



PALESTINE POLYTECHNIC UNIVERSITY
College of Information Technology and Computer Engineering
Computer Engineering Department

Graduation Project

**LoRa-Based Emergency Detection and
Response System for War Zones**

Submitted by

Ahmed Al-tell - 207801

Khaled Hroub - 201086

Adam Al-Amor - 201149

Supervisor

Dr. Liana Al-Tamimi

2024 – 2025

Certification and Anti-Plagiarism Declaration

This is to declare that the graduation project under the supervision of **Dr.Liana Al-Tamimi** having the title "**LoRa-Based Emergency Detection and Response System for War Zones**" was prepared by students listed below in partial fulfilment of the requirements for the degree of Bachelor in **Computer Systems Engineering** and no part hereof has been reproduced illegally (in particular: cut and paste) which can be considered as Plagiarism.

We certify that we will not commit any plagiarism, cheating, or any other academic integrity violations. We will be responsible and liable for any consequences if a violation of these declarations is proven.

Date: 12 , Jun 2025

Graduation Project Group's Students:

Ahmed Al-tell:

Khaled Hroub:

Adam Al-Amor:

الملخص

يقدم هذا المشروع نظاماً متكاملًا للكشف والاستجابة للطوارئ، يعتمد على مكونات مادية وبرمجية متقدمة، يستخدم النظام جهاز معالجة بيانات محلي، مع حساسات متخصصة للكشف الفوري عن الحرائق وتسرب الغاز والاهتزازات، كما يعتمد النظام على تقنية اتصال باستخدام الأمواج الراديوية منخفضة الطاقة وذات المدى البعيد والتي تتيح نقل البيانات بشكل موثوق في المناطق النائية أو التي تعاني من ضعف أو انعدام الاتصال، من الناحية البرمجية، تستخدم خوارزميات معالجة اللغة الطبيعية لاكتشاف الكلمات المفتاحية المرتبطة بالطوارئ وطلب النجدة من خلال الكلام، ويصدر النظام تحذير صوتي عند وقوع أي خطر، كما وترسل بيانات المستشعرات المختلفة والجمل الصوتية الخاصة بالاستغاثة بعد تحويلها إلى نص باستخدام خوارزمية معينة إلى مركز الطوارئ.

تعزز واجهة العرض الالكترونية من تفاعل مراكز الطوارئ واستجابتهم للحوادث، من خلال عرض التاريخ والوقت الفعلي للحدث، ونوعه، واسم المشترك، وموقعه على الخريطة، بالإضافة إلى تسهيل عملية التواصل بين مركز الطوارئ والأشخاص في مواقع الخطر.

بشكل عام يجمع هذا المشروع بين بنية تحتية قوية وتقنيات متعددة لمعالجة البيانات في الوقت الفعلي، مما يساهم في إنشاء نظام طوارئ ذكي يمثل طوق نجاة للأشخاص، ويهدف إلى تحسين وضمان سلامتهم في الظروف الخطرة خاصة في الحروب، يمثل هذا النظام حلاً فعالاً وقابلاً للتطوير يمكن تكيفه مع التطورات المستقبلية في مجالات السلامة والأنظمة الذكية في المباني.

ABSTRACT

This project presents an integrated emergency detection and response system powered by cutting-edge hardware and software components. The system utilizes a local processing unit along with dedicated sensors for real-time detection of fire, gas leaks, and vibrations.

The system also utilizes Long-Range Low-Power Radio Frequency (LoRa) communication technology, enabling the secure transmission of data in remote or sparsely connected areas.

On the software side, Natural Language Processing (NLP) algorithms detect emergency words and distress calls through voice. The system notifies through a sound when any risk occurs. Moreover, various sensor data and distress voice sentences, after text-conversion using a proprietary algorithm, are sent to the emergency center.

The electronic display screen enhances emergency centers' response and communication to incidents through the display in real time of the incident date and time, nature of incident, subscriber name, as well as his/her location on a map. It also avails mechanisms for communication between the emergency center and individuals in hazardous locations.

In summary, this project combines a robust infrastructure with a range of real-time data processing technologies for integration into a smart emergency system that is a lifeline to people, in an effort to improve and offer them protection in risky situations, especially during wars. The system is an effective and scalable solution that can be upgraded based on future technology in safety and smart building systems.

ACKNOWLEDGEMENT

In the name of "Allah", the most beneficent and merciful who gave us strength, knowledge and helped us to get through this project.

To the people that have inspired, supported, and molded us into the people that we are today, our families, friends and our supervisor.

We would've never been able to reach this achievement without your support, care and encouragement.

We would like to express our sincere appreciation to our graduation project supervisor Dr.Liana Al-Tamimi for his guidance, continuous encouragement, and support throughout the semester, for being there to answer our questions and provide us with valuable suggestions and tips which helped us in choosing some system components.

We would also like to thank our colleague, Eng. Mohammad Abu Amar, who introduced us to the idea of using LoRa technology and the module itself, based on his previous project. His support in helping us obtain the component was greatly appreciated.

Moreover, it is our duty to thank our families for their generous encouragement and continuous support throughout our life.

We are truly grateful for all your support throughout the whole education stage. At last, we would like to thank all the people who helped, supported, and encouraged us to successfully finish the graduation project.

Table of Contents

ACKNOWLEDGEMENT.....	5
Table of Contents.....	6
List of Figures:.....	8
List of Tables:.....	10
Abbreviations.....	11
Chapter 1: Introduction.....	12
1.1 Preface.....	12
1.2 Problem Statement.....	12
1.3 Project Aims and Objectives.....	13
1.4 Project Requirements.....	14
1.4.1 Functional Requirements:.....	14
1.4.2 Non Functional Requirements.....	15
1.5 System Description.....	15
1.6 Project Limitations/constraints:.....	16
1.7 Project Schedule.....	17
1.8 Report Outline.....	18
Chapter 2: Theoretical Background.....	19
2.1 Preface.....	19
2.2 Theoretical Background Concepts.....	19
2.3 Literature Review.....	21
2.3.1 Gas Leakage and Ventilation System.....	21
2.3.2 Smart House.....	22
2.4 Summary.....	24
Chapter 3: System Design.....	25
3.1 Preface.....	25
3.2 System components and design alternatives.....	25
3.2.1 Hardware components.....	25
3.2.2 Software Components.....	33
3.3 Conceptual System Description.....	36
3.4 Algorithms and Methodologies.....	38
3.4.1 Flow Chart.....	39
3.4.2 Sequence Diagram.....	40
3.4.3 use cases.....	41
3.5 Schematic Diagram.....	41
3.6 Summary.....	42
Chapter 4: Implementation and Testing.....	43

4.1 Overview.....	43
4.2 Implementation issues.....	43
4.3 Implementation challenges.....	51
4.4 Validation and testing.....	51
4.5 Summary.....	55
Chapter 5: Results and Discussion.....	56
5.1 Detailed analysis of the results/experiments.....	56
5.3 Justifications of the obtained results.....	62
5.4 Summary.....	63
Chapter 6: Conclusion and Future Work.....	64
6.1 Conclusion.....	64
6.2 Future work.....	65
References.....	66

List of Figures:

Figure 1.1 System description.....	15
Figure 2.1:LoRa communication [20].....	19
Figure 3.1: Raspberry Pi [6].....	25
Figure 3.2: loRa32 [8].....	26
Figure 3.3: MQ-5[10].....	28
Figure 3.4: IR Flame [12].....	29
Figure 3.5: SW-420 Vibration [14].....	30
Figure 3.6: SR-ULM10 USB Microphone [15].....	31
Figure 3.7: USB Mini Stereo Speaker [16].....	31
Figure 3.8: eSpeak software.....	32
Figure 3.9: NLP algorithm [19].....	33
Figure 3.10: Block diagram.....	36
Figure 3.11: System conceptual diagram.....	37
Figure 3.12: flow chart diagram.....	39
Figure 3.13: Sequence diagram.....	40
Figure 3.14: use cases diagram.....	41
Figure 3.15: Schematic Diagram.....	42
Figure 4.1: Top view of sensors placement layout on the system chassis.....	44
Figure 4.2: Placing LoRa and Raspberry Pi.....	45
Figure 4.3: Field testing of LoRa module across varying distances.....	45
Figure 4.4: speaker and microphone, and alarm bell setup.....	46
Figure 4.5: Power system.....	47
Figure 4.6: login page.....	48
Figure 4.7: main interface.....	49
Figure 4.8: subscriber log.....	50
Figure 4.9 alarm log.....	51

Figure 4.10: hardware testing.....	52
Figure 4.11: LORA testing.....	53
Figure 4.12: software testing.....	53
Figure 4.13: error software testing.....	54
Figure 4.14 verifying sensor readings.....	55
Figure 5.1 LORA Result.....	62

List of Tables:

Table 1: List of Abbreviations.....	10
Table 1.1: project GANNT(introduction).....	16
Table 1.2: project GANNT(Graduation Project).....	17
Table 2.1: List of Features and Limitations of Previous Projects.....	22
Table 3.1: Comparison Between Arduino and Raspberry Pi.....	25
Table 3.2: Comparison Between Cellular Networks LoRa.....	27
Table 3.3 Comparison Between MQ-5 Gas sensor and MQ-2 Gas sensor.....	28
Table 3.4: Comparison Between MQ-2 sensor and IR Flame sensor.....	29
Table 3.5: Comparison Between SW-420 and Piezoelectric Vibration Sensor.	30
Table 5.1: LoRa Transmission Test Results.....	57

Abbreviations

Abbreviation	Meaning
LoRa	Long Range
NLP	Natural Language Processing
IoT	Internet of Things
GPIO	General Purpose Input/output
VCC	Voltage, Common Collector
GND	Ground
PC	Personal Computer
USB	Universal Serial Bus
RPI	Raspberry Pi
AP	Access point
PHP	Hypertext Preprocessor
PY	Python
PCBs	Printed Circuit Boards
DB	Database
IDE	Integrated Development Environment
D0	Digital
AI	Artificial intelligence

Table 1: List of Abbreviations

Chapter 1: Introduction

1.1 Preface

The aim of this project is to develop an integrated emergency detection and response system, especially in war areas. The system combines a number of sensors to detect fires, toxic gas leaks and explosion vibrations. It also utilizes artificial intelligence to detect speech and has a communication channel with emergency centers via LoRa technology even if the traditional network collapses. Enhancing safety and speeding up emergency response, the project also includes a web portal at the emergency center by which homeowners are able to pre-register their household data for faster and more accurate emergency management.

1.2 Problem Statement

In the tragic circumstances of war zones, where families are at risk at all times, a good emergency system could prove to be a savior. This system would continually detect fires, toxic gases, and the telltale tremors of an impending explosion, which would immediately sound alarms for first responders, even if all other networks collapse. A computer-based voice recognition system powered by an AI would recognize desperate cries for help, transmitting crucial information—what, where, and when—to the capable hands that can offer assistance. Meanwhile, families would be guided by distinctive voice commands, ushering them to safety amidst calamity.

However, use of such a strong system is severely handicapped by the dire energy crisis and the constant threat of destruction prevalent in such areas. The absence of steady power sources, especially consistent electricity, seriously hampers performance when reliability is most required. Besides, the constant threat of missile attacks poses a monumental threat; a single hit would render the system inoperable at its moment of greatest necessity.

To bridge this crucial void between the ideal and unsightly reality, our system is directed to the optimal allocation of its components in a building's plan to offer their protection. Notably, it will utilize rechargeable batteries as a primary solution to the vast lack of an accessible source of electricity, to supply the power required for constant operation.

1.3 Project Aims and Objectives

Our system aims to create a comprehensive emergency response system to enhance household safety in high-risk areas. The objectives of this system include:

1. Real-Time Hazard Detection

Developing a system that identifies and responds to various hazards in real time. This includes:

- Immediate sensor alerts for fires, gas leaks and vibrations from explosions.
- Rapid transmission of detected events to emergency responders.

2. AI-Driven Speech Recognition

Implement AI-based speech recognition to:

- Detects emergency keywords and automatically generates distress signals.
- Convert spoken commands into text, transmitted through LoRa communication to relevant emergency centers.

3. Robust LoRa Communication: Ensure seamless transmission of emergency alerts to response centers without internet dependency.

4. Audio-Based Emergency Instructions

Enable the system to deliver audio instructions for occupants during emergencies by:

- Converting received text-based emergency instructions into audible guidance.
- provide clear and live, step-by-step information for safe evacuation or other safety measures.

5. Emergency Center Website

We will create a website for emergency centers to include the following:

- Homeowners register on a web portal and provide details like home description and owner name.
- This information, stored in advance, will match alerts to household locations without needing real-time GPS, allowing quick response times without dependence on internet-based dynamic location services.

1.4 Project Requirements

Functional requirements describe what the system should do, detailing the specific behaviors and functions that must be implemented.

1.4.1 Functional Requirements:

1. **Hazard Detection:** Detects fire, gas leaks and vibrations.
2. **Data Processing:** Analyze sensor data with microcontrollers and use AI-powered speech recognition to process emergency keywords.
3. **Communication:** Transmit alerts via LoRa, including type, timestamp, and pre-registered location, Transmit alerts via LoRa, including incident type, timestamp, and pre-registered location. Receive and convert emergency center responses into audio instructions.

4. **Database Management:** Maintain an incident database at the emergency center, logging event details and response notifications.
5. **Audio Instructions:** Convert received text instructions into clear, step-by-step audio guidance for occupants.
6. **Emergency Center Website:** Allow homeowners to pre-register details (like owner name, address) on a web portal for quick location matching without relying on real-time GPS or internet.

1.4.2 Non Functional Requirements

outline the quality attributes and constraints of the system, focusing on how the system performs its functions, Non Functional Requirements:

1. **Reliability:** High system uptime and consistent communication success.
2. **Performance:** Fast data processing and accurate speech recognition.
3. **Scalability:** Support multiple sensors per type and handle communication for numerous homes.
4. **Maintainability:** Quick error detection and efficient update process.

1.5 System Description

The High-Risk War Zone Emergency Detection and Response System is designed to continuously scan homes or buildings in war zones for potential dangers such as fires, gas leaks, and explosions. In addition to its robust sensor system, the project includes a web-based central platform in the emergency center where information is supplied by the homeowners about their homes with a description, house number and other essential details. It eliminates the need for real-time GPS since the static location of each house is already pre-recorded. Upon a situation, the system identifies it and cross-compares the house number with the pre-registered database such that response teams respond appropriately and in time.

Voice emergency alerts are also facilitated by the AI in the system by listening for specific distress words.

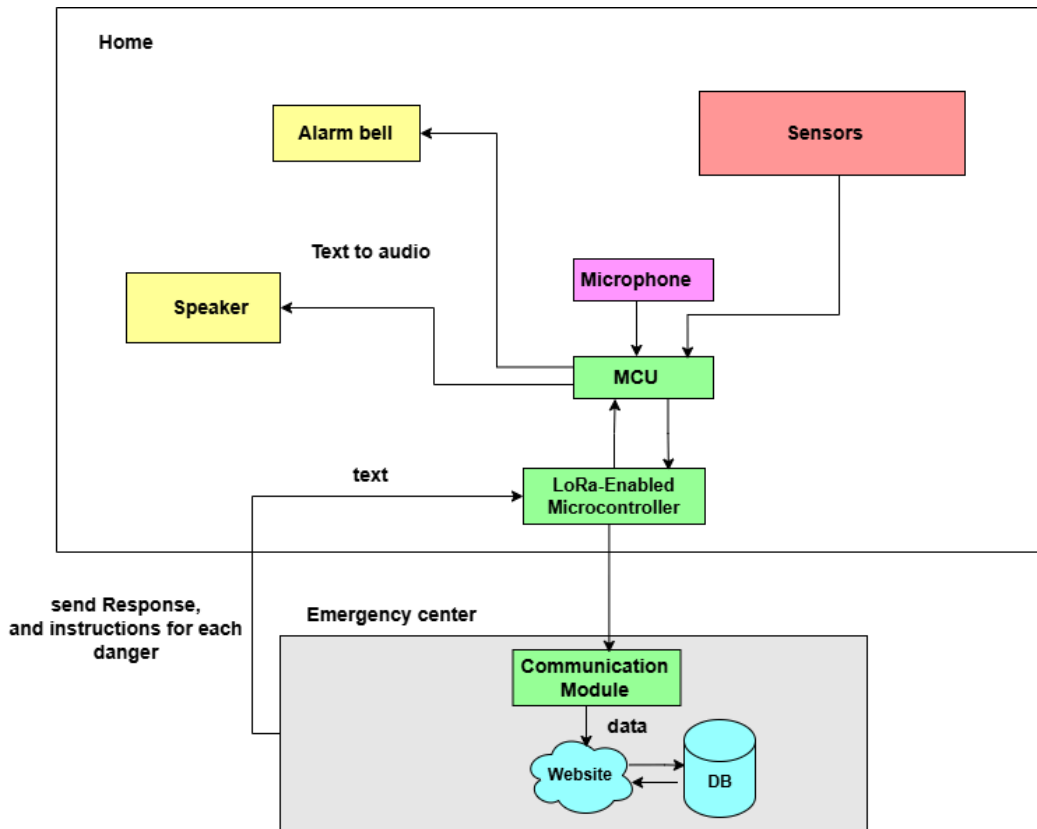


Figure 1.1 System description

1.6 Project Limitations/constraints:

Availability of LoRa Devices: One of the key constraints is the potential unavailability of LoRa devices due to likely import bans or restrictions on overseas purchases, which can delay or influence the implementation schedule of the project.

LoRa Technology Limitations: The LoRa technology is designed to be used in situations where the other systems do not function, e.g., remote or dangerous areas. Its operation may, however, be hindered by certain environmental factors like weather. Additionally, regions that suffer extensive damage during wars present a very real likelihood of system component destruction.

Infrastructure Constraints: The system is modeled for locations that have some infrastructure, such as a hardened room to protect equipment, access to solar or backup power, and the ability to install LoRa communication equipment. This requirement limits the system's deployment in locations that lack these facilities, especially in severely devastated or remote locations where these infrastructures are unavailable. Due to this, additional preparation or updating may be required before the system can be successfully implemented.

1.7 Project Schedule

	name	time	Start date	End date
1	Project Selection and Proposal Writing	15 days	5/10/2024	20/10/2024
2	Research and Detailed Planning	25 days	21/10/2024	15/11/2024
3	Procurement of Initial Hardware	32 days	16/11/2024	18/12/2024
4	Initial Software Development	27 days	19/12/2024	15/1/2025

Table 1.1: project GANNT(introduction)

	name	time	Start date	End date
1	Hardware Setup and Initial Testing	17 days	15/2/2025	4/3/2025
2	IoT Integration, Database Setup, and Software Development	20 days	5/3/2025	25/3/2025
3	AI Model Training and Testing	23 days	26/3/2025	18/4/2025
4	System Testing, Refinement, Documentation, and Final Presentation	27 days	19/4/2025	16/5/2025

Table 1.2: project GANNT(Graduation Project)

1.8 Report Outline

This report is organized as follows: Chapter 2 introduces some literature review including available technologies and related projects. It also goes briefly over the theoretical background of the project, hardware and software components. Chapter 3 discusses the conceptual design of the system, block diagrams, flowcharts and detailed hardware connections. Chapter 4 discusses the hardware and software implementations of the project. Chapter 5 shows the testing process which is made until we reach the final system design. Finally, Chapter 6 presents the final results of the project, future works and conclusion.

Chapter 2: Theoretical Background

2.1 Preface

In this chapter, the theoretical basis and certain research on the building of a system for danger-prone zones are mentioned. The chapter explains the technologies and techniques through which the system can work effectively, like using sensor networks in voice recognition and LoRa technology in emergency alarms. Additionally, the chapter reviews other similar projects and tries to suggest what is different in this one.

2.2 Theoretical Background Concepts

In this section, we will provide an overview and discuss the technologies and techniques that will be used in this project.

Natural Language Processing (NLP): In our project, Natural Language Processing (NLP) is in action by helping the system to detect the words of emergency being spoken and converting the words into processable words that can be delivered over the LoRa communication network. Upon hearing someone say words like "help," "fire," or "gas leak," the system hears carefully and speech-to-text technology is utilized to convert the sounds into words. This enables the system to signal emergency conditions even when it is not possible or safe to type. Upon identification of such keywords, the data are sent wirelessly to an emergency center. Conversely, the emergency center can also reply in text messages such as "we are coming" or "arriving in 5 minutes," and the system will then convert these messages from text to speech again using text-to-speech technology so that they are spoken aloud with a speaker. This two-way conversion of words and text facilitates emergency communication with the ability to deliver quick and clear exchange of information, which is vital when it comes to saving lives and making quick decisions at momentous times.[1]

LoRa (Long Range) communication:

LoRa is a very important technology for this project. It is useful where there isn't a mobile network or internet connection, like in remote or destroyed areas. In these places, LoRa runs at low frequencies like 868 MHz or 915 MHz and uses something called Chirp Spread Spectrum, enabling it to send data far away without consuming high power. That's why it's suitable for small battery-operating devices. In case of emergency, LoRa can send essential sensor information like fire, gas, vibration, or voice detection even without an internet connection. It can cover large spaces, hence it helps to connect the sensors to the core system or emergency center in a swift manner, thus responding timely.[2]

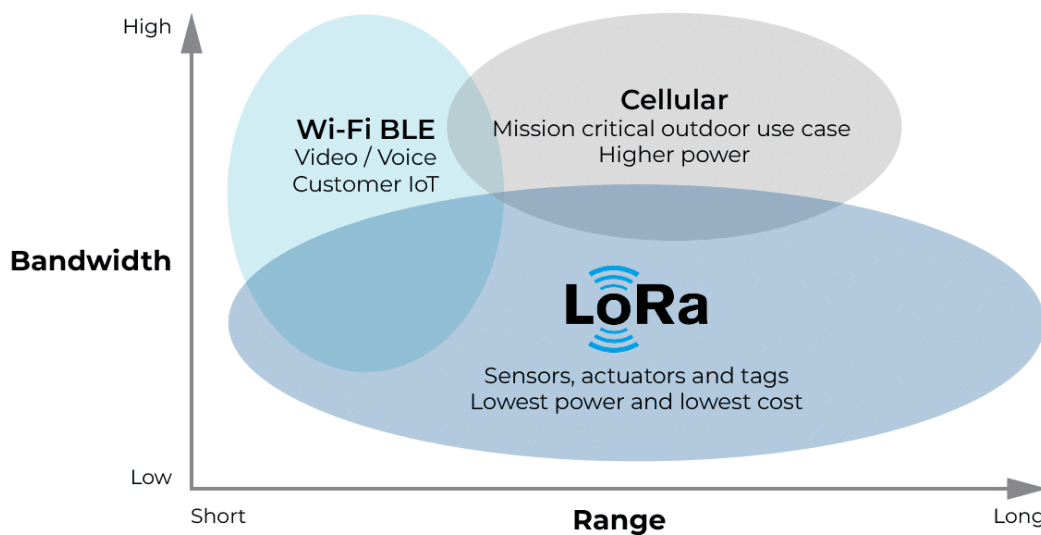


Figure 2.1:LoRa communication [20]

Database Management:

MySQL is used as a database to hold important information such as pre-registered homeowner data and incident reports in our system. Homeowners pre-register their data such as house addresses and descriptions in the web interface of the emergency center, hence facilitating quick matching of hazard warnings with individual homes without applying real-time GPS. The database offers accurate

location identification that allows for quick emergency response. MySQL also keeps incidents and sensor data and reporting and monitoring incidents is easy. Its advanced features such as complex queries, indexing support and transactions, ensure good management of large volumes of emergency data.

Speech-to-Text and Text-to-Speech Processing

Speech-to-text and text-to-speech technology are utilized in our project to enable communication in a more suitable and faster manner in the case of emergency situations. The system records the user's voice and converts it into letters by the help of a speech-to-text tool. It comes in handy when a person cannot type or type safely, i.e., in case of a fire accident or leakage of gas. For example, when a person utters a word like "fire" or "help," the system will recognize this and react. When the emergency center wants to react, they can send a text message stating something like "arriving soon" or "we are coming." The system translates the message to voice response through text-to-speech and has it read out through a speaker. This allows one to hear the response even when one cannot see a screen, allowing for convenient emergency communication.

2.3 Literature Review

Below, we will introduce some of the previously proposed projects that are similar to our project:

2.3.1 Gas Leakage and Ventilation System

Description

This IoT-based system is intended to detect toxic gas leakages and fire hazards in residential spaces. On detection, it automatically activates ventilation systems, fire suppression systems (e.g., water pumps), and sends SMS alerts to the

residents. It employs an Arduino Mega, fire and gas sensors, and a GSM module for communication and control [3].

- **Methods and Technologies Used in the Project:**

- A central processing unit is used to control sensors and manage system logic.
- A wireless communication module sends alert messages to designated users.
- Environmental sensors monitor gas concentrations and smoke presence.

2.3.2 Smart House

- Description

The Smart House project was developed to provide basic disaster detection and response in residential buildings. It includes fire detection, theft alarms, and automated safety responses such as alarms, power shutdown, and alerting the homeowner via pre-programmed communication channels[4].

- **Methods and Technologies Used in the Project:**

- A control unit processes input from sensors and activates corresponding safety actions.
- Fire detection components identify hazardous smoke levels.
- Safety mechanisms are activated to manage fires, including shutting off power.
- A user interface allows remote control and status monitoring of the system.

Project Name	Key Features	Limitation
Gas Leakage and Ventilation System	• Detecting Gas and Smoke. • Automatically starts ventilation and fire control systems. • Sends alerts messages to users Via SMS.	• Detects fire, gas leaks, and explosions. • Recognize voice commands like “help” in real time. • Sends alerts using LoRa when there is no internet. • Has a web portal for monitoring and control.
Smart House	• Detects fire and theft • Actives alarms and cuts of electricity. • Automatically dials and emergency phone number.	• Uses an old and less powerful microcontroller. • No support for advanced communication Methods. • Cannot detect explosions or vibrations.
LoRa-Based Emergency Detection and Response System for War Zones	• Detects fire, gas, and explosions. • Recognize voice commands like “help” in real time. • Sends alerts using LoRa. • Has a web portal for monitoring and control	• Relies on the availability of electricity. • Still in testing phase, especially in • dangerous environments.

Table 2.1: List of Features and Limitations of Previous Projects.

Key Differences:

Our project is different from the previous systems because it uses smart voice recognition and a long-range, low-energy communication method. This allows it to work even when there is no internet. Also, our system can detect explosions by sensing vibrations, and it can process data quickly and efficiently. These features make it more useful in dangerous areas like war zones, where normal systems might not work because of damage or weak infrastructure.

Key Similarities:

All the systems aim to detect dangers like gas leaks and fires as early as possible. They use sensors to check the environment and automatically react—by sending alerts, turning on alarms, or starting devices like fans or water pumps. Each system is built to respond quickly in real-time to help protect people and their homes.

2.4 Summary

This chapter has provided an overview of the core concepts and technologies supporting our system, including LoRa communication, AI-driven speech recognition and hazard detection sensors. Through a review of related projects, we highlighted the unique features of our system, particularly its focus on real-time emergency response in high-risk zones. This theoretical foundation sets the stage for the technical details and implementation strategies discussed in the following chapters.

Chapter 3: System Design

3.1 Preface

This chapter delves into the hardware and software components required for our system. It presents various design alternatives for each element, explaining the rationale behind the chosen options. A conceptual framework of the system's design and a block diagram visually representing the system's structure are also introduced. Additionally, the chapter outlines the system's algorithms and methodologies using clear and concise flowcharts, while schematic diagrams provide a detailed view of the connections and interactions between components.

3.2 System components and design alternatives

This section provides a detailed analysis of the hardware and software components utilized in the system, along with a comparison of alternative design choices to justify the selected configurations.

3.2.1 Hardware components

- Raspberry Pi

In our system, we need a Raspberry Pi microcontroller shown in Figure 3.1 for voice-to-text conversion, there are several options like Arduino as mentioned in table 6.3, but we choose Raspberry Pi due to its superior processing power, advanced audio handling capabilities and extensive software support, while the Arduino can capture basic audio signals or act as a supporting device to transmit data, it lacks the computational capacity and library support needed for complex speech recognition tasks, the Raspberry Pi simplifies implementation by directly connecting to microphones and running advanced speech-to-text libraries, making it the optimal choice for this application[5].

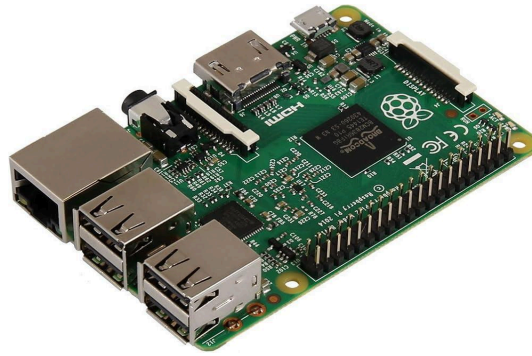


Figure 3.1: Raspberry Pi [6]

Feature	Arduino	Raspberry Pi
Type	Microcontroller board	Single-board computer
Processor	8-bit AVR (e.g., ATmega328P)	Powerful, capable of running a full operating system
GPU	None	Broadcom VideoCore VI
RAM	2KB to 8KB	2GB, 4GB, or 8GB LPDDR4
Storage	There is no built-in storage, Flash memory only	MicroSD USB
Number of Pins	14 digital, d analog	40 GPIO
I/O Ports	Simple embedded systems like blinking LEDs, sensors	Complex systems like IoT hubs, AI and multimedia apps
Power Supply	Limited to program memory (e.g., 32KB on Arduino Uno)	Expanded via SD card
Wireless Connectivity	Requires modules for Wi-Fi, Bluetooth, or Ethernet	Built-in connectivity options (in some models)

Table 3.1: Comparison Between Arduino and Raspberry Pi

- loRa32

The LoRa32 module, as illustrated in Figure 3.2, plays a vital role in our system by enabling reliable long-range communication. While there are alternative options, such as Cellular Networks (4G/5G), detailed in table 3.2, we opted for LoRa due to its advantages in low-power communication. This module facilitates efficient data exchange between sensors (e.g., fire, gas and vibration) and the emergency center, ensuring reliable performance even in areas where traditional networks are compromised.

LoRa supports two-way communication, allowing the emergency center to send audio instructions to households. Additionally, the ESP32 component integrated into the system transmits commands to activate the alarm bell at the emergency center upon receiving distress signals from specific locations. Its low power consumption makes it ideal for use with solar panels and rechargeable batteries, ensuring uninterrupted operation during power outages.

To enhance durability, the LoRa32 module should be housed in a shock-resistant and reinforced enclosure, capable of withstanding harsh environmental conditions and physical impacts[7].



Figure 3.2: loRa32 [8]

Feature	Cellular Networks (4G/5G)	LoRa
Range	Large (city-wide or nationwide)	Long (up to 15 km in rural areas)
Power Consumption	High	Low
Data Transfer	High (support real-time data)	Low (suitable for small packets)
Infrastructure	Requires cellular towers and SIM cards	Does not require cellular infrastructure
Cost	High operational	Low operational cost
Latency	Low	Higher
Deployment Complexity	Simple (if network exist)	Requires LoRa gateways and setup
Ideal use cases	Real-time apps, large data.	IoT sensors, remote monitoring emergency alerts.

Table 3.2: Comparison Between Cellular Networks LoRa

- Gas sensor

In our system we will use an MQ-5 sensor shown in Figure 3.3, for gas detection, there are several options as mentioned in table 3.3, but we choose MQ-5, We choose MQ-5 because it is specifically designed for detecting LPG (cooking gas) combustible gases such as methane and propane, offering higher sensitivity and accuracy for these gases compared to the more general-purpose MQ-2. This makes MQ-5 ideal for applications like detecting gas leaks from cooking appliances, where precise detection is critical for safety. It has a faster response time and optimized performance[9].



Figure 3.3: MQ-5[10]

Feature	MQ-5	MQ-2
Target Gases	LPG, methane, propane	LPG, methane, propane, smoke, CO
Sensitivity	High for LPG	General-purpose
Accuracy	More accurate for LPG	Less precise for LPG
Applications	LPG leak detection	General gas and smoke detection
Response Time	Faster	Moderate

Table 3.3 Comparison Between MQ-5 Gas sensor and MQ-2 Gas sensor

- Flame sensor

Our system will use an IR Flame Sensor shown in Figure 3.4, for fire detection. There are several options, as mentioned in Table 3.4, but we chose the IR Flame Sensor because it is specifically designed to detect infrared radiation emitted by flames, making it ideal for fire detection. compared to the MQ-2 sensor, which is mainly designed for detecting smoke and gases, the IR Flame Sensor offers higher accuracy in identifying actual flames and heat, ensuring faster and more reliable detection of fires. This makes the IR Flame Sensor a more specialized and effective choice for immediate fire response in environments where the presence of flames needs to be detected with precision. Its ability to detect

flames directly, as opposed to the MQ-2's smoke detection, ensures quicker activation of safety protocols in case of fire hazards[11].



Figure 3.4: IR Flame [12]

Sensor	Output Type	Best for	Power Consumption	Environmental Suitability
MQ-2	Analog with digital option	Gas leak detection, (smoke, LPG, CO, methane)	Moderate	Sensitive to other gases
IR Flame Sensor	Analog and Digital	Flame detection via infrared radiation	Low	Affected by ambient light

Table 3.4: Comparison Between MQ-2 sensor and IR Flame sensor

- Vibration Sensor Module

Our system will use the SW-420 vibration sensor shown in Figure 3.5, module for vibration detection. Table 3.5 mentions several options, but we chose SW-420 because it is a cost-effective, simple and reliable solution for detecting vibrations and shocks. Unlike piezoelectric sensors, designed for continuous vibration monitoring and requiring more complex signal processing, the SW-420 provides a straightforward digital output, making it easier to integrate with our Raspberry Pi-based system. Its design is ideal for detecting sudden shocks or vibrations, such as those caused by explosions, which are critical for our

emergency detection system. The SW-420’s compact size, low power consumption and ease of use make it a practical and efficient choice for our system requirements[13].

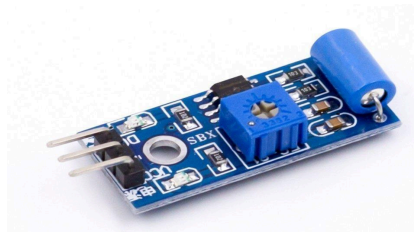


Figure 3.5: SW-420 Vibration [14]

Feature	SW-420 Vibration Sensor	Piezoelectric Vibration Sensor
Output Type	Digital (ON/OFF)	Analog (requires ADC)
Sensitivity	Detects sudden shocks	Detects subtle vibration
Ease of Use	Simple, direct GPIO connection	Requires signal processing
Cost	Low	Higher
Power Consumption	Low	Low to moderate
Integration	Easy with Raspberry Pi	Requires ADC or additional circuits

Table 3.5: Comparison Between SW-420 and Piezoelectric Vibration Sensor

- Microphone

The SR-ULM10 Microphone shown in Figure 3.6 was chosen for its high sensitivity, compact size, and compatibility with embedded systems like the Raspberry Pi. It plays a critical role in detecting audio cues during emergencies, such as distress calls or spoken keywords, which the AI-powered speech recognition module processes.



Figure 3.6: SR-ULM10 USB Microphone [15]

- Speaker

The speaker is an essential component of the emergency detection and response system, designed to provide real-time audio alerts and guidance to occupants during critical situations. It complements the other detection and communication modules by ensuring occupants are promptly informed and guided through audible instructions.



Figure 3.7: USB Mini Stereo Speaker [16]

3.2.2 Software Components

In this section, we will introduce the software components we're going to use.

- eSpeak software

eSpeak is an open-source text-to-speech synthesizer that runs on Raspberry Pi, converting written text into spoken words. It supports multiple languages and can be controlled via the command line or integrated into Python programs. eSpeak allows customization of speech parameters such as speed, pitch and volume, making it ideal for projects requiring voice feedback, like home automation or emergency response systems. It can easily be installed on a Raspberry Pi, adding voice functionality to various applications[17].

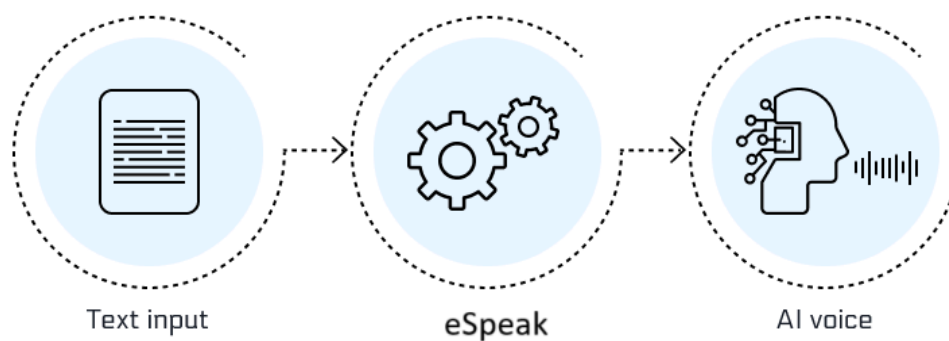


Figure 3.8: eSpeak software

- NLP Algorithm

NLP (Natural Language Processing) algorithms are designed to help computers understand, interpret and generate human language. These algorithms include tasks like tokenization (breaking text into words), part-of-speech tagging (identifying grammatical components), sentiment analysis (determining the emotional tone of text) and machine translation (translating languages). Advanced techniques like word embeddings (Word2Vec) and transformer models (GPT, BERT) enable a deeper understanding and generation of text, making NLP

essential for applications such as chatbots, language translation and content classification[18].

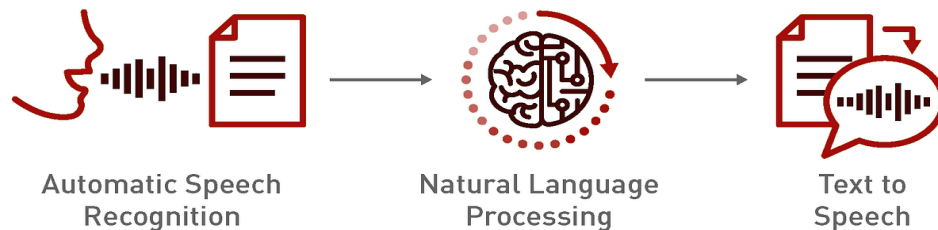


Figure 3.9: NLP algorithm [19]

- Vosk

Vosk is an open-source speech recognition library that converts audio to text with high accuracy, even without an internet connection. Vosk supports multiple languages, including Arabic, and is lightweight and fast, making it suitable for use on resource-constrained devices like the Raspberry Pi. Vosk relies on pre-trained voice models and can be easily used in artificial intelligence applications, voice assistants, and voice interaction with systems. The library provides APIs in multiple languages, such as Python and Java, making it easy to integrate into various types of projects.

- XAMPP

It is the program used to create a virtual local server. The name is an abbreviation, where each letter represents one of the five main components. The software package contains the Apache web server, the MySQL (or MariaDB) database management system and the Perl and PHP programming languages. The

initial X indicates the operating systems it works with Linux, Windows and Mac OS.

- Arduino IDE

The Arduino Integrated Development Environment (IDE) is a software platform for writing, compiling, and uploading code to Arduino boards. It supports C and C++ programming languages and provides a simple, user-friendly interface for beginners and experts alike. The IDE allows users to interact easily with hardware components like sensors, motors and lights, making it ideal for building electronics projects. It includes a built-in library of functions to simplify coding and debugging, making it an essential tool for developing Arduino-based applications.

- Fritzing Software

Fritzing is an open-source electronics design software that allows users to create circuit diagrams, design Printed Circuit Boards (PCBs) and prototype projects. It provides an intuitive interface for beginners and experts to visualize circuits, connect components and document their designs, making it ideal for electronics and IOT projects.

- Website Technologies

1. **PHP:** A server-side scripting language used to create dynamic websites. It processes data, interacts with databases and sends responses to users.
2. **JavaScript:** A client-side programming language used to add interactivity to websites, like handling user actions (clicks, form submissions) and updating the content without reloading the page.
3. **MySQL:** A relational database management system used to store and retrieve data, such as user information or emergency records, in tables.

4. **CSS:** A styling language used to design the layout and appearance of web pages, controlling colors, fonts, spacing and responsiveness across different screen sizes.
5. **HTML:** A markup language used to structure the content of web pages, like headings, paragraphs, images and links.

3.3 Conceptual System Description

The following figure 3.10 shows the block diagram of LoRa-Based Emergency Detection and Response System for War Zones. The diagram represents the project's main components and how they are interconnected. The system begins with a sensor box equipped with flame, gas and vibration sensors connected to a Raspberry Pi microcontroller which acts as the central processor. Once the data is processed it is passed to the IoRa32 module which uses LoRa communication technology to transmit information over long distances even in areas with limited or no traditional networks.

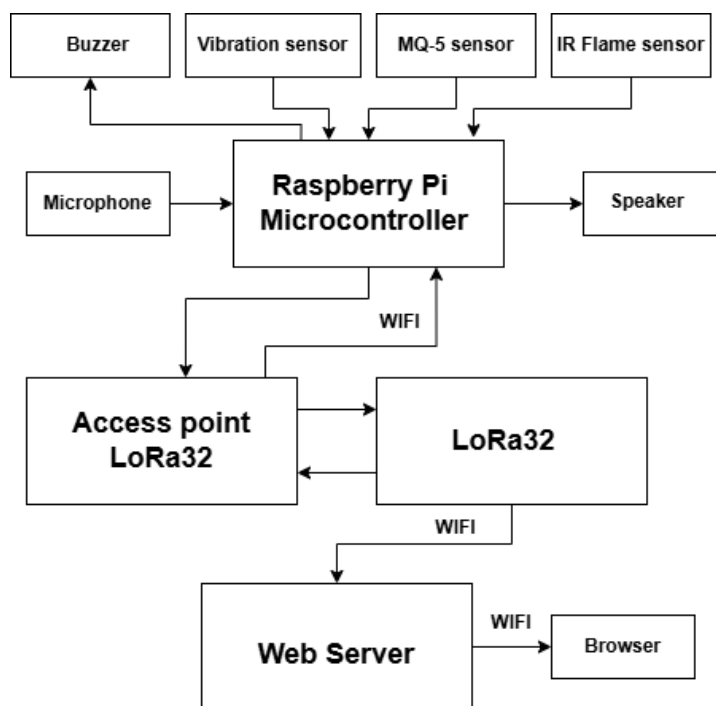


Figure 3.10: Block diagram

The system also includes outputs such as an alarm bell and a loudspeaker that generate immediate audible alerts to warn people of detected hazards. The data is then sent to an emergency website, where respondents can access it in real time to take action. Figure 3.11 illustrates the system's working mechanism in more detail, showing how the various components work together. The sensors and microphone send the data to the Raspberry Pi, which processes it and decides on an appropriate response. Alerts are then sent via the IoRa32 module to both the local outputs and the emergency website. This ensures that people in the area are warned and emergency crews are notified as quickly as possible. The integration of these components ensures the system's reliability and effectiveness, especially in high-risk areas such as conflict zones or remote areas. Using long-range communications, low-power devices, and real-time data processing, the system can respond quickly to emergencies and help save lives.

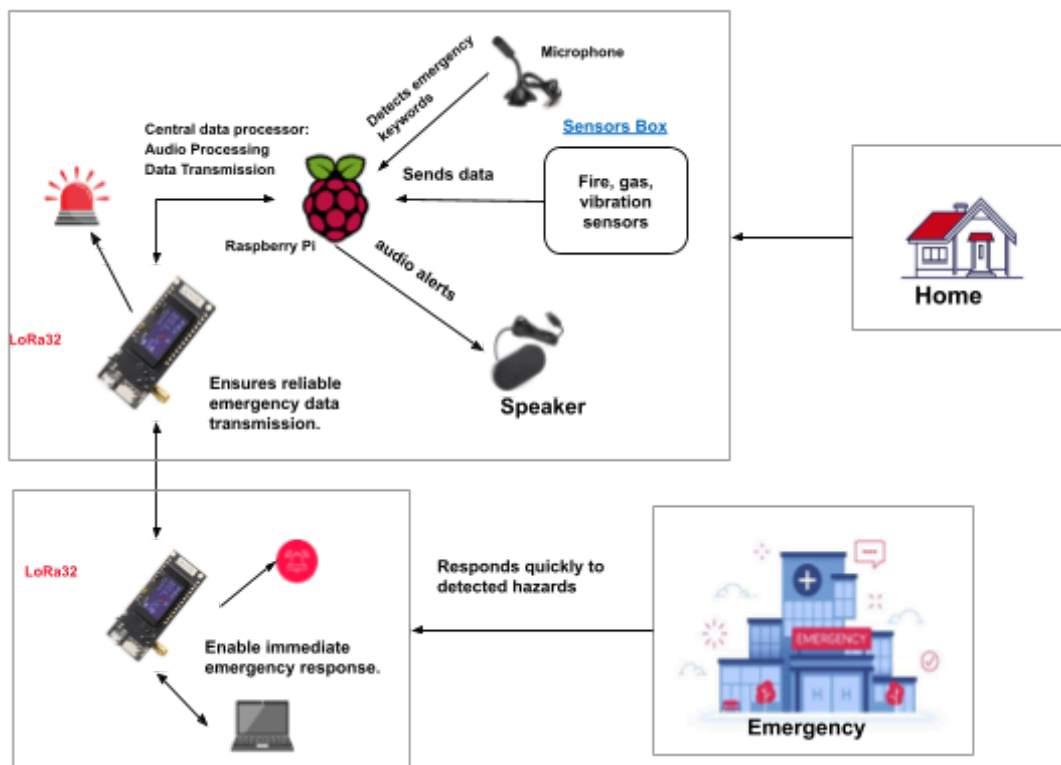


Figure 3.11: System conceptual diagram

3.4 Algorithms and Methodologies

Algorithm 1: NLP-Based Emergency Keyword Detection

```
1: Initialize microphone and audio system.  
2: while (Microphone Active) do  
3:   Capture and preprocess audio.  
4:   Convert audio to text and tokenize.  
5:   if (Emergency Keyword Detected) then  
6:     Trigger hazard alert and send data to LoRa.  
7:   end if  
8: end while
```

Algorithm 2: Text-to-Speech Conversion using eSpeak

```
1: Initialize the eSpeak library for text-to-speech functionality.  
2: while (System is Active) do  
3:   Check for incoming text commands or notifications.  
4:   if (New Text Command Received) then  
5:     Preprocess the text.  
6:     Pass the processed text to eSpeak for synthesis.  
7:     Generate speech output:  
8:     - Select desired language and voice settings.  
9:     - Adjust speed, tone and volume parameters.  
10:    Play the synthesized speech through the speaker.  
11:   end if  
12: end while
```

3.4.1 Flow Chart

The flowchart illustrates the operational process of our system, outlining steps from hazard detection and voice command analysis to sending alerts via LoRa and logging events in a local database as shown in figure 3.12.

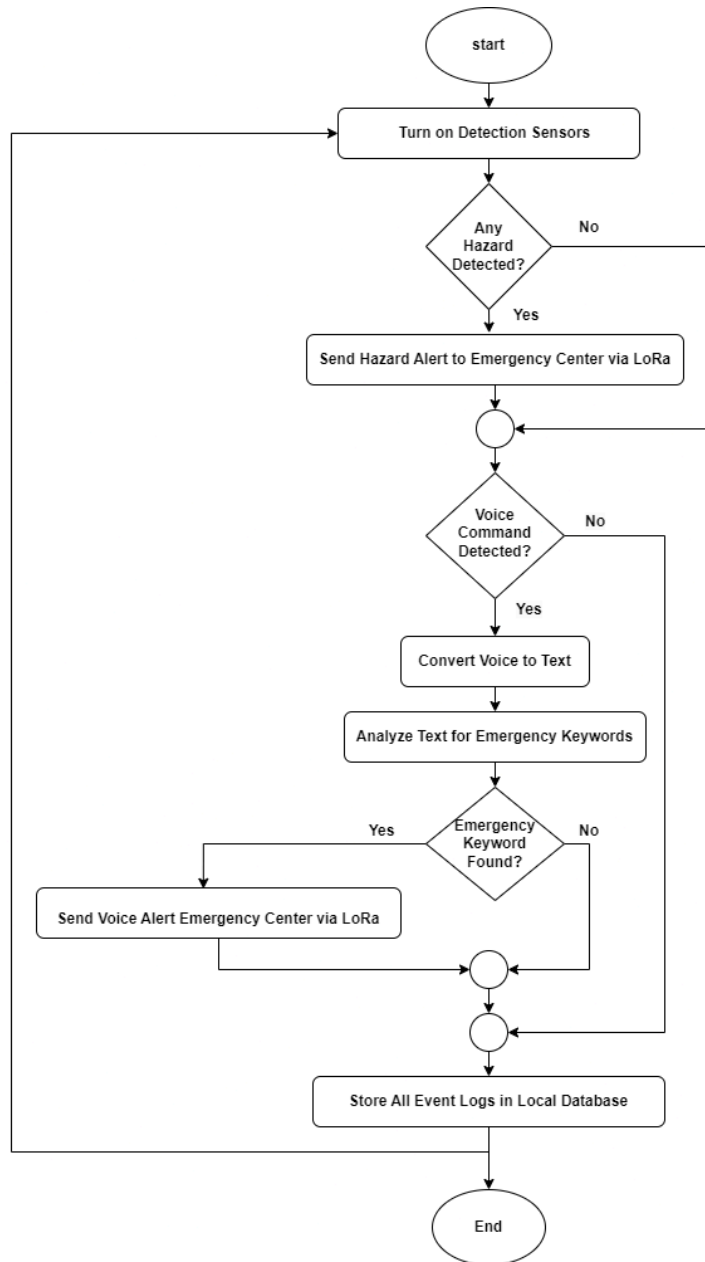


Figure 3.12: flow chart diagram

3.4.2 Sequence Diagram

This section delves into the sequence diagram, providing a detailed visual representation of the system's interactions. The sequence diagram is illustrated in Figure 3.13:

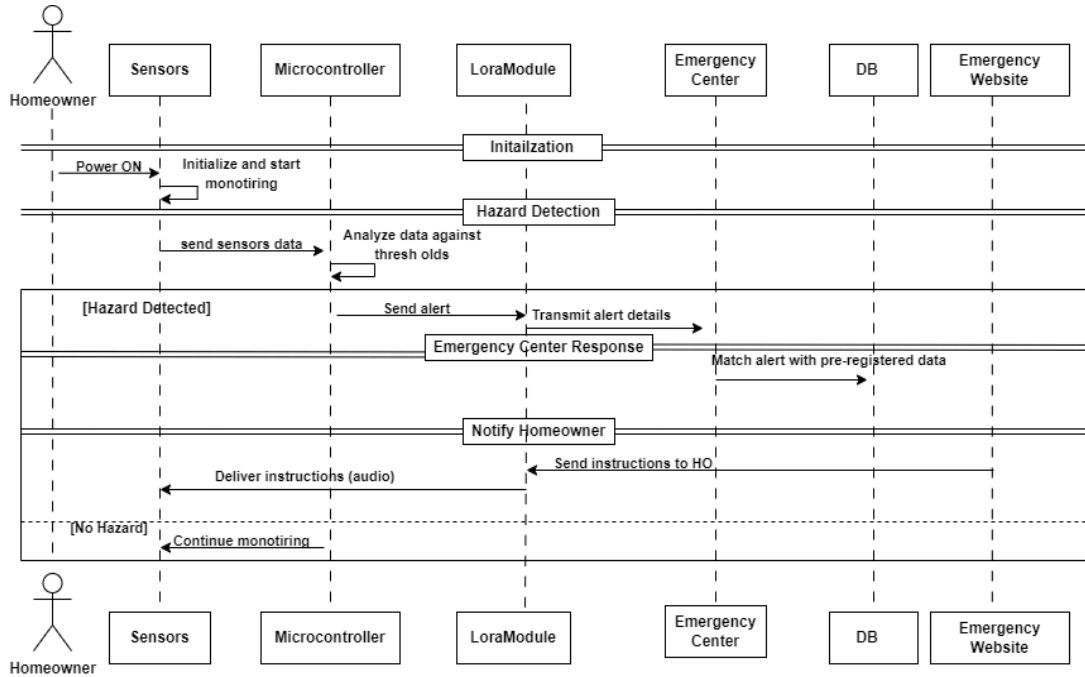


Figure 3.13: Sequence diagram

This sequence diagram illustrates our system design. Upon powering on, the system initializes and begins monitoring sensor data. If a hazard is detected by analyzing data against set thresholds, an alert is sent via the LoRa module to the Emergency Center. The center matches the alert with pre-registered information, notifies the homeowner and delivers audio instructions while also updating the homeowner's website interface. After addressing the alert, the system resumes monitoring to ensure continuous safety.

3.4.3 use cases

The use case diagram (Figure 3.14) illustrates how the system monitors the environment, processes data, sends alerts via LoRa, triggers alarms, provides instructions and notifies the homeowner.

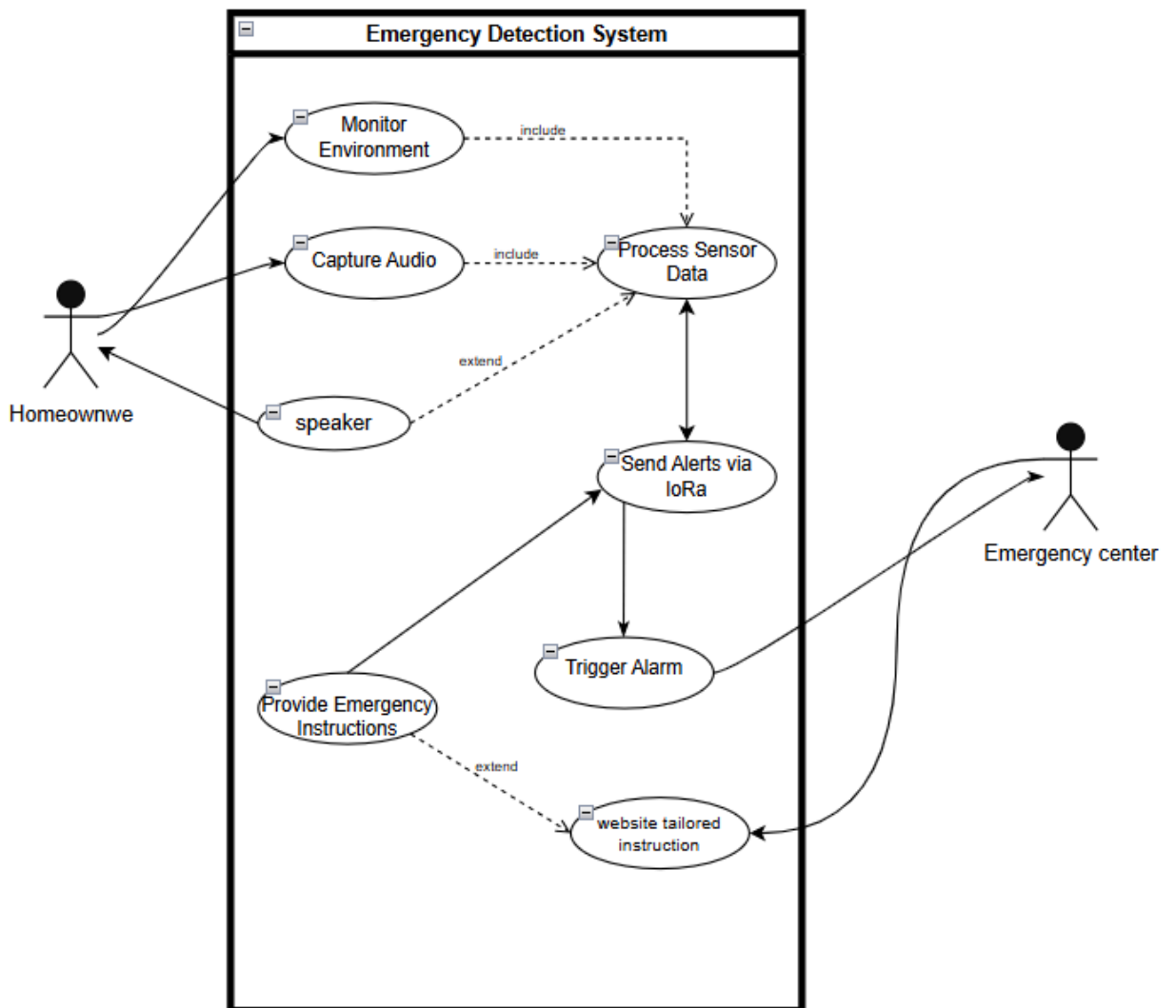


Figure 3.14: use cases diagram

3.5 Schematic Diagram

Figure 3.15 shows the hardware setup connecting the Raspberry Pi to sensors, a microphone, a speaker, a buzzer and LoRa modules for the emergency detection system.

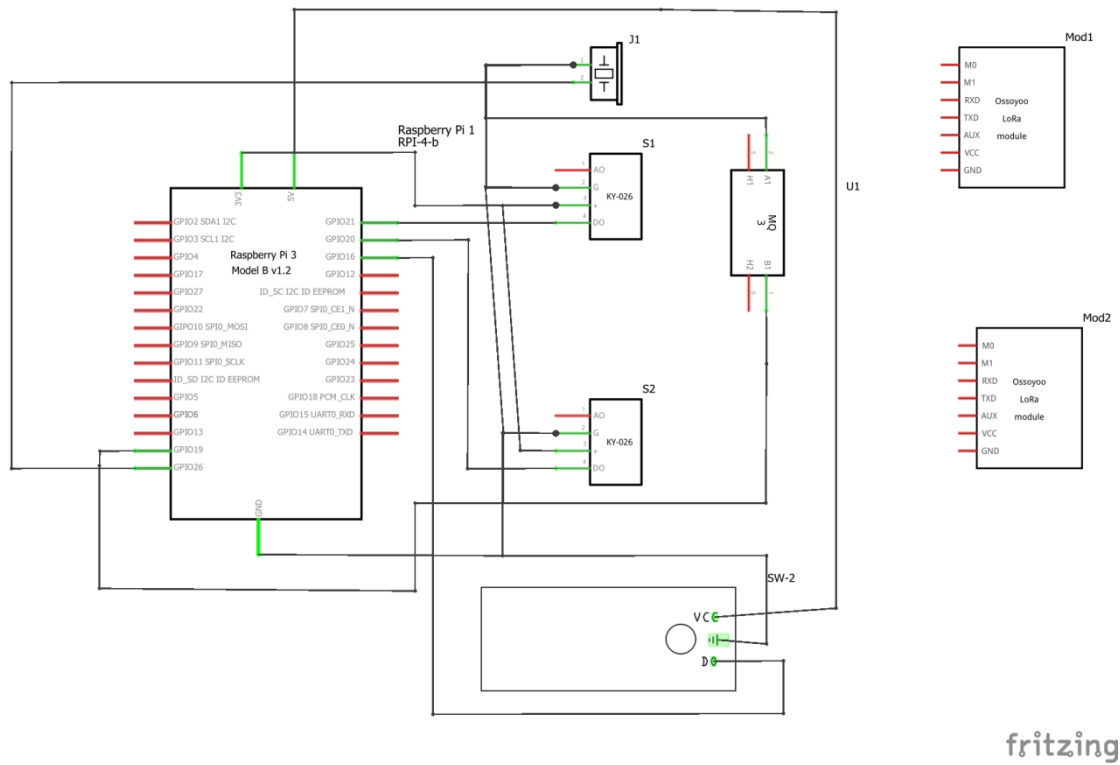


Figure 3.15: Schematic Diagram

3.6 Summary

In this chapter, the hardware and software components of the system are examined, along with alternative options for each. A conceptual overview of the system is provided, outlining its general flow and complemented by essential diagrams for clarity.

Chapter 4: Implementation and Testing

4.1 Overview

This chapter provides a comprehensive overview of the implementation phase of our Emergency Detection and Response System. It outlines both hardware and software development stages, including assembly procedures, technical obstacles, system integration, and performance validation. We also present a series of images that document critical steps and configurations throughout the process.

4.2 Implementation issues

This section summarizes the most significant implementation activities and development problems found. We are interested in the hardware component assembly, software module integration, and deployment strategies the most.

4.2.1 Hardware Implementation

The hardware development of the Emergency Detection and Response System was designed to have high accuracy for emergency detection, prompt response, and reliable wireless communication even under hostile or risky conditions.

Sensor Deployment

The setup included three primary sensors: flame, gas (MQ-5), and vibration (SW-420). The sensors are installed in two separate rooms rather than a single open area. This strategic positioning gives each room to be monitored separately for fire, gas leakages, or shock events. Such a setup enables faster and more specific reactions to emergencies as each sensor can trigger an alarm according to its local conditions, The figure 4.1 below shows the distribution of the different

sensors in the model, The image below shows the distribution of the different sensors in the model.

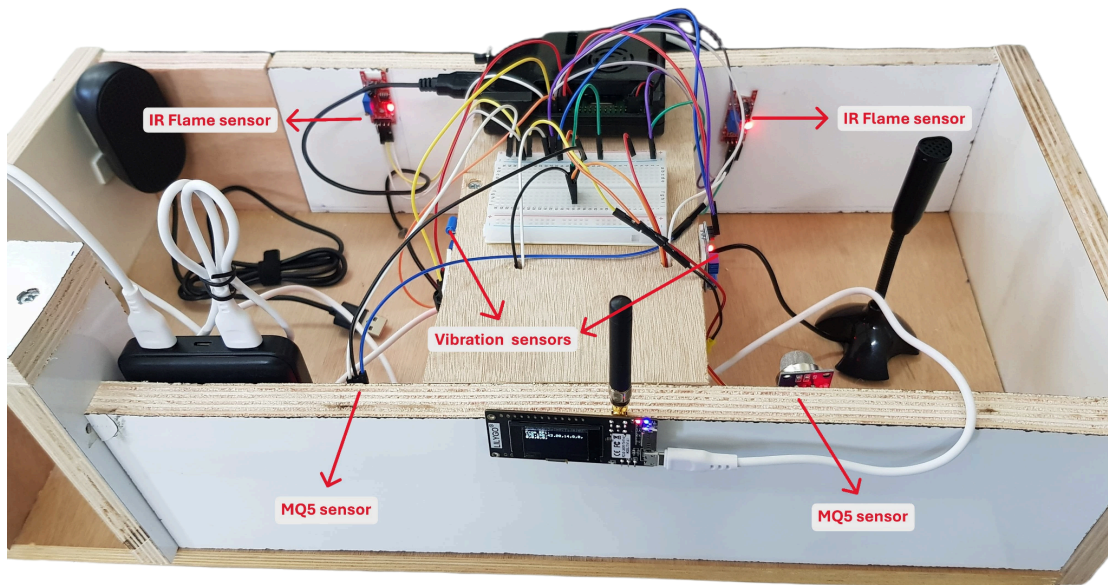


Figure 4.1: Top view of sensors placement layout on the system chassis

LoRa and Raspberry Pi Setup

The Raspberry Pi was placed in the center with the board to reduce wire management and electric noise. The LoRa module was placed in the middle next to it as shown in figure 4.2, and various tests were conducted over different distances: 30 meters, 100 meters, and 200 meters (line-of-sight). All the tests confirmed standard and robust data transmission, making LoRa an ideal solution for low-power long-distance communication in warzone environments, in the figure 4.3 below there is a message sent by emergency containing “we are coming” appearing on the LoRa32 OLED.

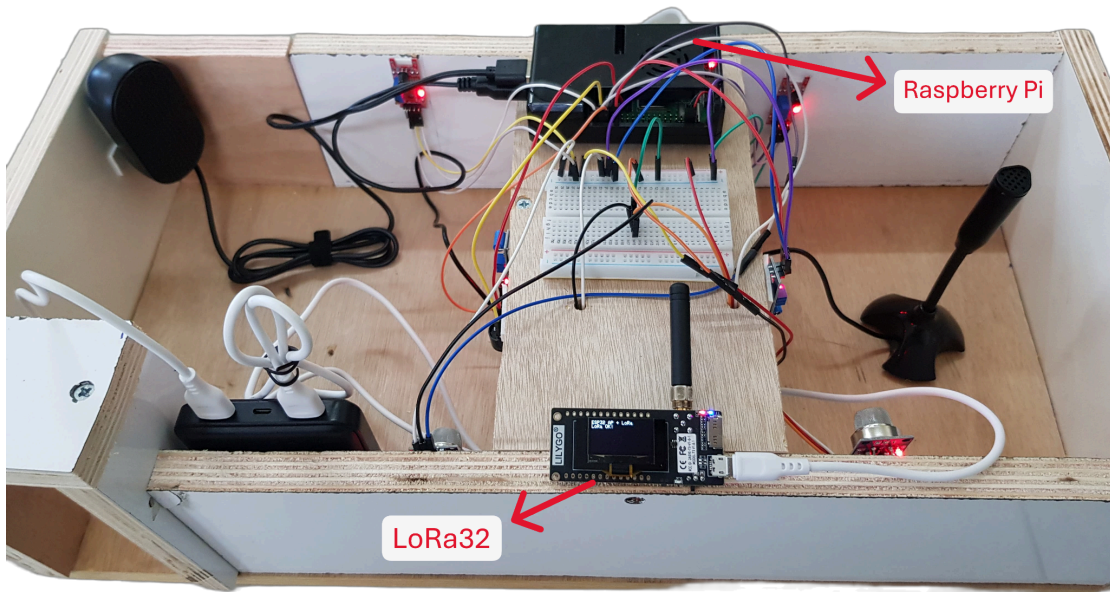


Figure 4.2: Placing LoRa and Raspberry Pi

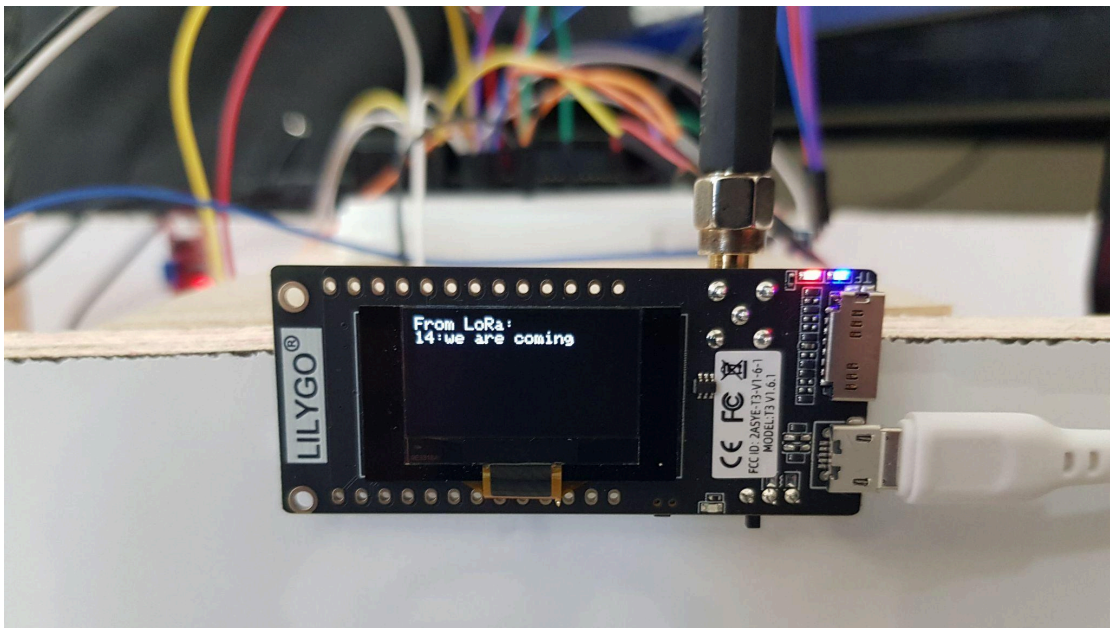


Figure 4.3: Field testing of LoRa module across varying distances

Microphone Integration

The microphone was positioned where it provides optimal acoustic coverage of the building as shown in figure 4.3. The microphone was set to pick up human voice continuously to sense emergency words such as "fire," "help," or "gas leak" faster.

Speaker and Buzzer Actions

The speaker was installed in the system but in a manner that allowed its sound to be audible throughout the building as shown in figure 4.3. It announces loud and repetitive emergency alerts like:

"Please evacuate immediately."

"Help is on the way."

An alarm bell has also been installed, which operates as soon as any sensor is activated in the event of a danger.

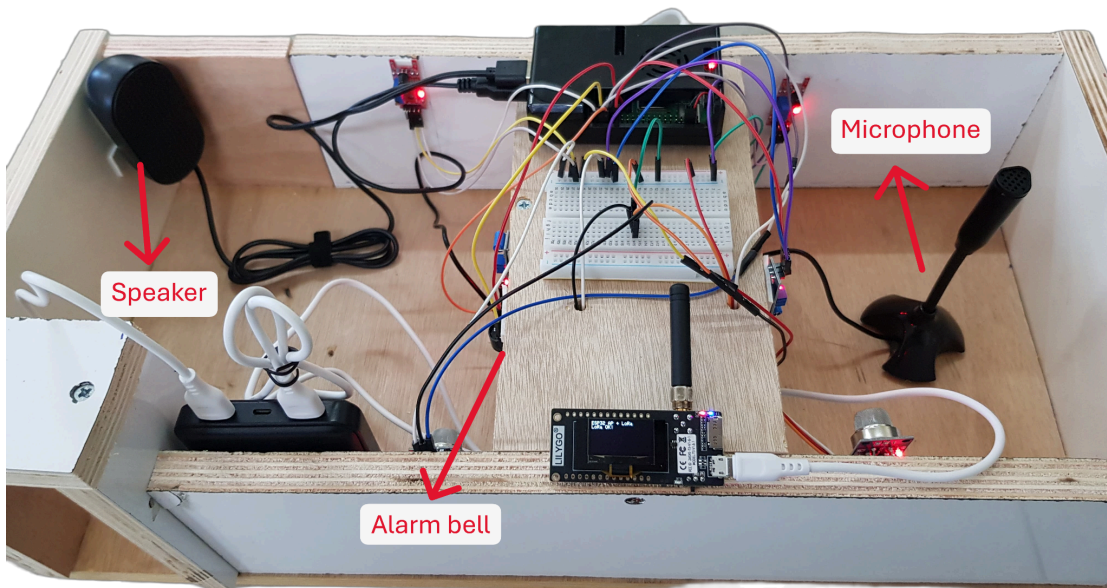


Figure 4.4: speaker and microphone, and alarm bell setup

Power system

All units are powered by a set of rechargeable lithium batteries, ensuring safe operation in the event of a power outage as shown in figure 4.5.

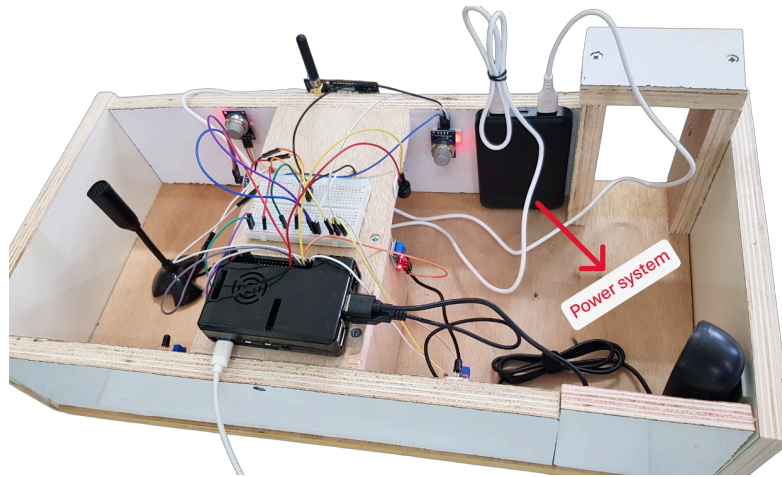


Figure 4.5: Power system

4.2.2 Software Implementation

This section describes the implementation details of software components of the system. It also explains the features and functions of each software component in the system.

Development Tools

Visual Studio Code served as the primary development environment to create the front-end and back-end modules of the system. XAMPP offered an in-built solution for executing the Apache server and MySQL database on the local machine for development and test environments. We created the website using HTML, CSS, and JavaScript for the front-end, and server-side programming and database queries were created using PHP. To represent and engage with real-time geospatial data, we employed the Google Maps JavaScript API.

Libraries and APIs

The application relies on a number of third-party libraries to enhance functionality and ease the development. Bootstrap was utilized to assist responsive design along with for common-sense UI elements. The Google Maps API was also utilized to

enable dynamic location tracking and graphical representation of incident locations on the website itself.

Website Features and Functionality

The login form allows the respective user to login to the system with a secure username and password. A registration form for new users is also present that requests basic information such as full name, phone number, and password. Input data is validated and stored securely in the database with respective encryption done on sensitive information.



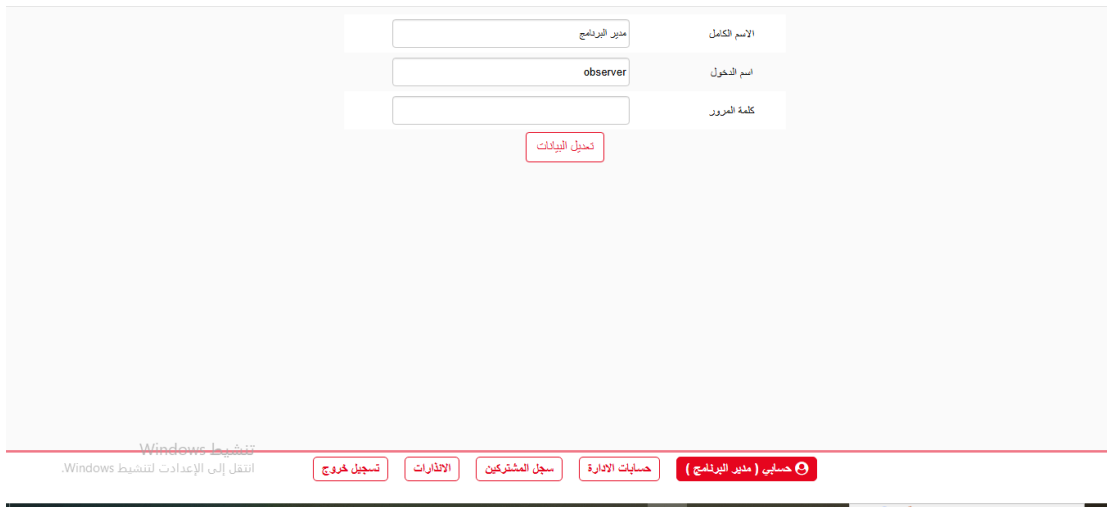
Figure 4.6: login page

Dashboard Interface

Upon logging in, the users are directed to the dashboard — the control center of the site. The interface provides access to the various modules of the system and displays user account information. Administrators can manage user accounts from here, view system notices in real time, display subscriber information, and track emergency logs via a simple navigation panel.

Administrative Account Management

Administrative accounts module allows the system administrator to add new high-privilege users. The interface displays a form where the administrator inputs the full name, username, and password for each new account. Inputted information is validated and inserted into the system, and existing accounts can be updated or deleted by navigating through the management panel.



The screenshot displays a web interface for administrative account management. It features a form with three input fields: 'الاسم الكامل' (Full Name) with the placeholder 'مدير البرنامج' (Program Manager), 'اسم الدخول' (Username) with the placeholder 'observer', and 'كلمة المرور' (Password). Below these fields is a red button labeled 'تحديث البيانات' (Update Data). At the bottom of the interface, there is a navigation bar with several buttons: 'تسجيل خروج' (Logout), 'الانذارات' (Alerts), 'سجل المشتركين' (Subscriber Log), 'حسابات الادارة' (Admin Accounts), and a red button labeled 'حسابي (مدير البرنامج)' (My Account (Program Manager)). On the far left of the navigation bar, there is a link for 'تنشيط Windows' (Activate Windows) with the text 'انتقل إلى الإعدادات لتنشيط Windows' (Go to settings to activate Windows).

Figure 4.7: main interface

Subscriber Log

The second part is a comprehensive list of all the system's subscribers in tabular form. Each record is filled with the subscriber's name, contact information, residential address, and exact geographical location of the device mounted on their property. Each device is given a unique identity by using a Device ID so that management and tracking can be exact. The subscriber record also includes record editing and deletion in real time to ensure up-to-date and pertinent information.

#	الاسم	الغنوان	الهاتف	الموقع	Device ID	تعديل	حذف
13	fadi			الموقع على الخريطة		تعديل	حذف
9	mohammad			الموقع على الخريطة		تعديل	حذف
14	mohammad		0563989898	الموقع على الخريطة	1234	تعديل	حذف
7	احمد بونس			الموقع على الخريطة		تعديل	حذف
8	حسن			الموقع على الخريطة		تعديل	حذف
11	خالد ابراهيم		0560000004	الموقع على الخريطة		تعديل	حذف
10	راحي محمود			الموقع على الخريطة		تعديل	حذف
12	لؤي اكرم		0567345678	الموقع على الخريطة		تعديل	حذف

Figure 4.8: subscriber log

Alarm Log

The alarm log module is where all the emergency events are logged. Each record contains comprehensive details of the alarm, such as the time of the incident, the name of the subscriber involved, the nature of the alarm (such as fire or gas leak), and the exact site of the incident, which is graphically indicated on an interactive map. The record also indicates the response status — whether an emergency crew has been sent, not yet sent, or the incident has already been attended to. The module also provides the facility for administrators to raise alerts or provide commands to the emergency responders directly from the interface.

رقم السجل	المشارك	تاريخ وقت الإقذار	نوع الإقذار	الموقع على الخريطة	الرسائل	الحالة
1	راحي محمود	10:28:17 2025-02-18	fire	الموقع على الخريطة	رسائل	working
2	mohammad 0563989898	10:28:17 2025-02-18	fire	الموقع على الخريطة	رسائل	working
3	راحي محمود	10:28:17 2025-02-18	fire	الموقع على الخريطة	رسائل	working
4	mohammad 0563989898	10:28:17 2025-02-18	fire	الموقع على الخريطة	رسائل	done
5	راحي محمود	10:28:17 2025-02-18	fire	الموقع على الخريطة	رسائل	done
6	mohammad 0563989898	10:28:17 2025-02-18	fire	الموقع على الخريطة	رسائل	new
7	راحي محمود	10:28:17 2025-02-18	fire	الموقع على الخريطة	رسائل	new
8	mohammad 0563989898	10:28:17 2025-02-18	fire	الموقع على الخريطة	رسائل	new
9	راحي محمود	10:28:17 2025-02-18	fire	الموقع على الخريطة	رسائل	new
10	mohammad 0563989898	10:28:17 2025-02-18	fire	الموقع على الخريطة	رسائل	new
11	راحي محمود	10:28:17 2025-02-18	fire	الموقع على الخريطة	رسائل	new

Figure 4.9: alarm log

Database Integration

The back-end database is run in MySQL, in some number of related tables that specify users, subscribers, alarms, and system logs. The implementation is designed to enable data normalization and referential integrity. PHP scripts access the database securely, with prepared statements that prevent SQL injection and defend system security. Updates are done in real-time with AJAX communication, enabling seamless interaction between the client-side interface and server-side logic.

4.3 Implementation challenges

We faced many problems during the implementation of the LoRa-Based Emergency Detection and Response System for War Zones due to its multiple and integrated components. The most prominent of these challenges was the communication between the three controllers, the data transfer mechanism, the connection with the artificial intelligence libraries, and ensuring the accuracy of the readings of the various sensors (fire, gas, vibrations). In addition, the process of searching for appropriate libraries for audio analysis, where several libraries were worked on to find the best method for analysis. One of the major challenges was transferring the data to the electronic interface using LoRa technology, as it required adjusting the signals and taking into account the extent of coverage and interference.

4.4 Validation and testing

During project implementation, system components are tested and verified to ensure their functionality. These include:

1. Functional testing: To verify that the system operates according to requirements, and the primary verification of the device's ability to implement the emergency detection and response system.



Figure 4.11: LORA testing

The following figure 4.12 shows the process of testing sensors by displaying the readings of each sensor to ensure the accuracy of the work.

 The image shows a Thonny IDE window. The main editor displays a Python script named 'all_project.py'. The script includes a function 'transcribe_audio_vosk' and a 'while True' loop that reads sensor data (vibration, flame1, flame2, mq2) and triggers a buzzer. The terminal window at the bottom shows the execution output, including 'voice record=' and 'Sending: 192.168.214.48,14,0,0,0,0,0,0' messages, and 'Received from ESP32: from lora from web: 14:we will went' messages.


```

all_project.py
59 return np.max(np.abs(audio_data)) > threshold
60
61 def transcribe_audio_vosk(audio_data):
62     """ما احسن من هذا صوتا لوجت Vosk"""
63     q = queue.Queue()
64
65     def callback(indata, frames, time, status):
66         if status:
67             print(status)
68             q.put(bytes(indata))
69
70     audio_bytes = audio_data.tobytes()
71     recognizer.AcceptWaveform(audio_bytes)
72     result = json.loads(recognizer.Result())
73     return result.get("text", "").strip()
74
75
76
77 while True:
78     try:
79         # ما احسن من هذا صوتا لوجت
80         vibration_value = int(GPIO.input(VIBRATION_PIN)) #True:1 False:0
81         flame1_value = int(GPIO.input(FLAME1_PIN))
82         flame2_value = int(GPIO.input(FLAME2_PIN))
83         mq2_value = int(not GPIO.input(MQ2_PIN))
84
85         if vibration_value==1 or flame1_value==1 or flame2_value==1 or mq2_value==1 :
86             GPIO.output(BUZZER_PIN,True)
87         else:
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602

```

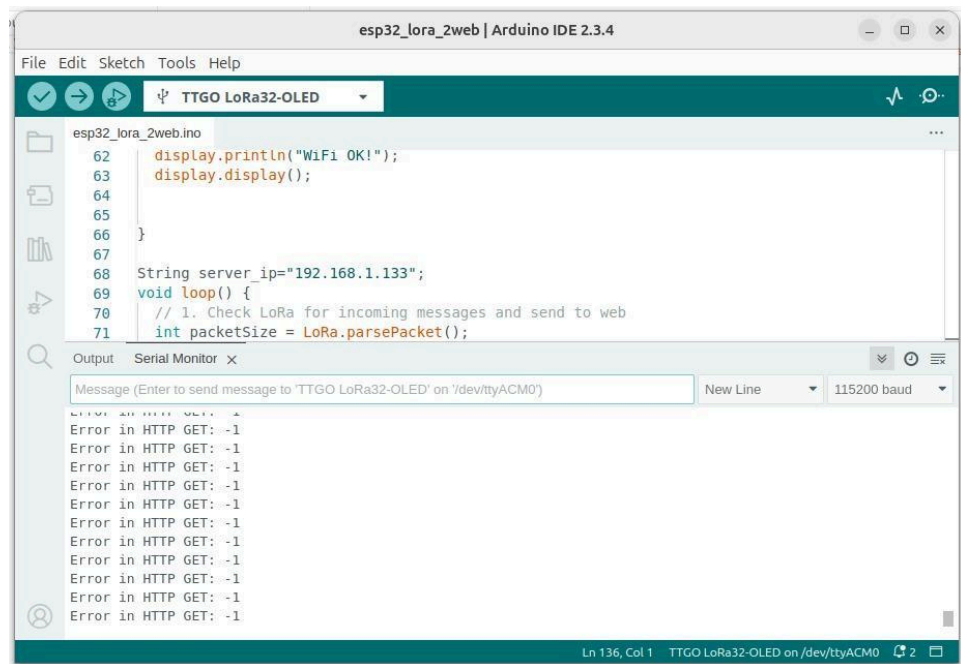


Figure 4.13: error software testing

The validation process included ensuring the accuracy and reliability of system components, conducting sensor calibration and accuracy tests to ensure accurate readings, testing system integration to ensure smooth operation of sensor communication modules, verifying the reliability and security of real-time data transmission, evaluating the website's user interface for ease of use, improving energy efficiency to prevent battery drain, and verifying location coordinates directly and automatically. The following figure 4.14 shows the process of verifying sensor readings.

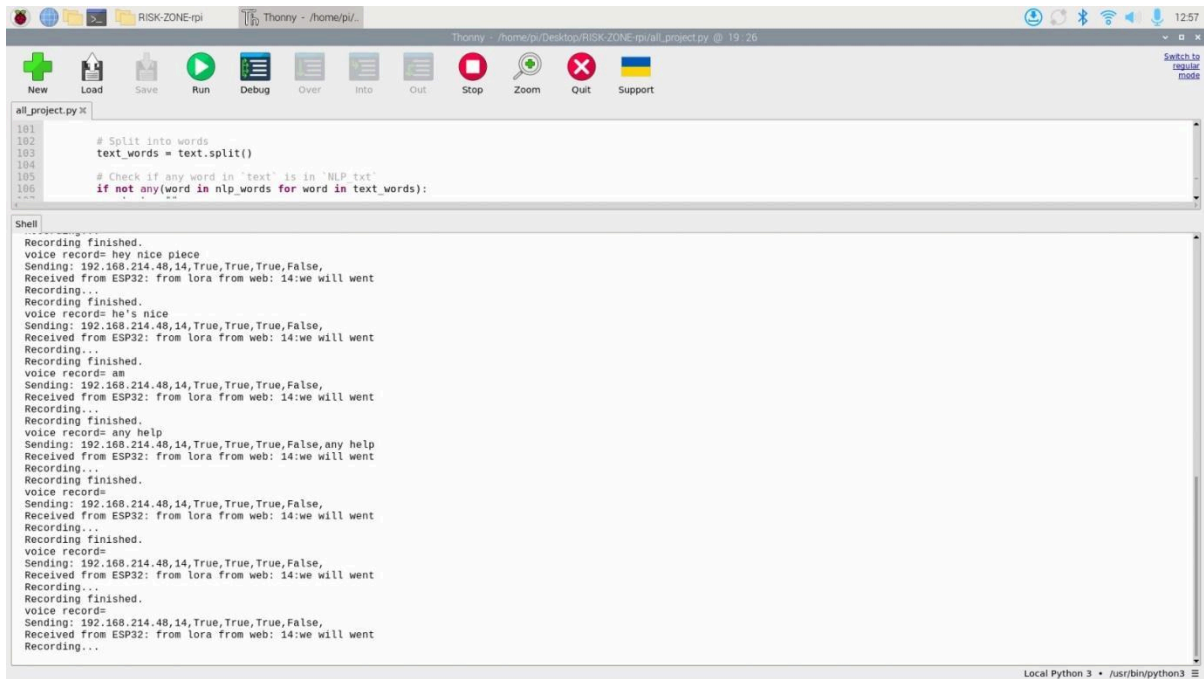


Figure 4.14: verifying sensor readings

4.5 Summary

In this chapter, we discussed implementation issues by identifying the problems we faced during work. We also discussed how to implement the software and hardware components, in addition to presenting the challenges we faced when working on implementing the project.

Chapter 5: Results and Discussion

5.1 Detailed analysis of the results/experiments

A series of experiments were conducted to test the system's effectiveness in detecting various emergencies and successfully transmitting data to the emergency center these experiments included the following:

1. **LoRa data transmission test:** Data transmission via LoRa was tested in the university environment, in various locations, and at various distances. Several experiments were conducted, including changing the location of the site's server, while the building's model was installed in one location, the Projects Lab. The first experiment was conducted from the fourth floor of Building C, while the second experiment was conducted from Airport Street on the university campus, at various distances, up to 200 meters away. Data transmission was also tested outside the university environment and over a distance of 500 meters in Yatta.
2. **Detect emergency words:** The response to emergency keywords, such as help, fire, gas, and vibration, was tested and successfully captured, transmitted to the site, and displayed on the LoRa screen in the emergency center.
3. **Response time:** In most test cases, the response time was very good, with a maximum of 10 seconds.
4. **Response effectiveness:** The system responds in most cases, and notifications are sent to the site based on the values generated by the various sensors. The notifications are displayed as text messages on the site. For example, if there is a fire, a "flame" is displayed, along with the time of the incident, the home address, and other details.

Test No.	Distance	Data sent and received	Response time	Comments
1	30 - 40 m	Sensors	1s	Excellent low-latency performance across all devices.
		Microphone	2s	
		Speaker	5s	
2	100 - 120 m	Sensors	1s	Sensors maintain minimal latency; slight increase in latency for Microphone and Speaker compared to Test 1.
		Microphone	3s	
		Speaker	6s	
3	200m	Sensors	2s	Moderate latency increase for Sensors and Speaker; Microphone latency remains consistent with Test 2.
		Microphone	3s	
		Speaker	7s	
4	500m	Sensors	8s	Response time increases slightly across all devices, indicating a slight performance drop, which is both normal and acceptable.
		Microphone	10s	
		Speaker	18s	

Table 5.1: LoRa Transmission Test Results

5.2 Error / success rate calculations

➤ Target Response Time for Test 1, Test 2, and Test 3:

1. Sensors ≤ 5 seconds
2. Microphone ≤ 5 seconds
3. Speaker ≤ 10 seconds

Test number 1 (30-40 m):

Results of the test:

- Sensors response time: 1s
- Microphone response time: 2s
- Speaker response time: 4s

Calculations:

1. Sensors:

Actual response time: 1s

Target response time: 2s

Comparison: $1s \leq 5s \Rightarrow \text{Success}$

2. Microphone:

Actual response time: 2s

Target response time: 2s

Comparison: $2s \leq 5s \Rightarrow \text{Success}$

3. Speaker:

Actual response time: 5s

Target response time: 6s

Comparison: $5s \leq 10s \Rightarrow \text{Success}$

Summary for Test #1:

- Number of Successes: 3
- Number of Errors: 0
- Attempts: 3
- Success Rate = $\frac{\text{Total of successes}}{\text{Total Attempts}} \times 100\% = \frac{3}{3} \times 100\% = 100\%$
- Errors Rate = $\frac{\text{Number of Errors}}{\text{Total Attempts}} \times 100\% = \frac{0}{3} \times 100\% = 0\%$

Test number 2 (100-120 m):

Results of the test:

- Sensors response time: 1s
- Microphone response time: 3s
- Speaker response time: 5s

Calculations:

1. Sensors:

Actual response time: 1s

Target response time: 5s

Comparison: $1s \leq 5s \Rightarrow \text{Success}$

2. Microphone:

Actual response time: 3s

Target response time: 5s

Comparison: $2s \leq 5s \Rightarrow \text{Success}$

3. Speaker:

Actual response time: 5s

Target response time: 10s

Comparison: $5s \leq 10s \Rightarrow \text{Success}$

Summary for Test #2:

- Number of Successes: 3
- Number of Errors: 0
- Attempts: 3
- Success Rate = $\frac{\text{Total of successes}}{\text{Total Attempts}} \times 100\% = \frac{3}{3} \times 100\% = 100\%$
- Errors Rate = $\frac{\text{Number of Errors}}{\text{Total Attempts}} \times 100\% = \frac{0}{3} \times 100\% = 0\%$

Test number 3 (200 m):

Results of the test:

- Sensors response time: 2s
- Microphone response time: 3s
- Speaker response time: 7s

Calculations:

1. Sensors:

Actual response time: 2s

Target response time: 8s

Comparison: $2s \leq 8s \Rightarrow \text{Success}$

2. Microphone:

Actual response time: 3s

Target response time: 10s

Comparison: $3s \leq 10s \Rightarrow \text{Success}$

3. Speaker

Actual response time: 5s

Target response time: 10s

Comparison: $5s \leq 10s \Rightarrow \text{Success}$

Summary for Test #3:

- Number of Successes: 3
- Number of Errors: 0
- Attempts: 3
- Success Rate = $\frac{\text{Total of successes}}{\text{Total Attempts}} \times 100\% = \frac{3}{3} \times 100\% = 100\%$
- Errors Rate = $\frac{\text{Number of Errors}}{\text{Total Attempts}} \times 100\% = \frac{0}{3} \times 100\% = 0\%$

➤ Target Response Time for Test 4 (500 m):

1. Sensors ≤ 10 seconds
2. Microphone ≤ 10 seconds
3. Speaker ≤ 15 seconds

Results of the test:

- Sensors response time: 8s
- Microphone response time: 10s
- Speaker response time: 18s

Calculations:

4. Sensors:

Actual response time: 8s

Target response time: 10s

Comparison: $8s \leq 10s \Rightarrow \text{Success}$

5. Microphone:

Actual response time: 10s

Target response time: 10s

Comparison: $10s \leq 10s \Rightarrow \text{Success}$

6. Speaker:

Actual response time: 18s

Target response time: 15s

Comparison: $18s \geq 15s \Rightarrow \text{Failed}$

Summary for Test #4:

- Number of Successes: 2
- Number of Errors: 1
- Attempts: 3
- Success Rate = $\frac{\text{Total of successes}}{\text{Total Attempts}} \times 100\% = \frac{2}{3} \times 100\% \approx 66.67\%$
- Errors Rate = $\frac{\text{Number of Errors}}{\text{Total Attempts}} \times 100\% = \frac{1}{3} \times 100\% \approx 33.33\%$

5. Experiments were conducted to detect emergency situations and ensure that data reaches the website.
6. Sending messages from the emergency center, i.e., through the website, to people at risk, informing them of the intervention process to resolve the problem, the following figure 5.1 illustrates the transmission process between controllers, showing the transmission results on LORA, including sensor values and sent messages.



Figure 5.1: LORA Result

5.3 Justifications of the obtained results

The results obtained through the experiments are consistent with our goals as the system detects and responds to emergencies the rationale for each observed result is presented below:

1. The accuracy of the sensor readings was achieved through proper connection and proper sensor settings with the microcontroller. It was confirmed that the gas, fire, and vibration readings were within the expected range.
2. Successful LoRa communication between the transmitter and receiver module was achieved through properly programming the components and identifying the sender and receiver, making it ideal for sending emergency alerts in areas with limited connectivity.
3. The effectiveness of the voice-to-text and text-to-text conversion was demonstrated by the use of well-trained AI libraries, as the system was able to recognize key emergency phrases and respond to them vocally.
4. Successful communication between the three microcontrollers was achieved by carefully assigning roles to each microcontroller .
5. Emergency detection and sending alerts to the website were successful, justified by the good integration between the sensors and microcontrollers.
6. Sending messages from the emergency center to people in danger was justified by the successful implementation of two-way communication.

5.4 Summary

This chapter presents the project's results, demonstrating that the system is capable of accurately detecting various emergency situations under ideal conditions, with some challenges in noisy environments or environments with poor communication coverage. The integration of sensors, artificial intelligence, and the LoRa interface has been shown to be effective in building a reliable and scalable smart response system.

Chapter 6: Conclusion and Future Work

6.1 Conclusion

In conclusion a smart integrated system was designed and developed for emergency detection and response focusing on high-risk environments such as war zones or other high-risk locations. The system combines sensors to detect fires and gas leaks and vibrations resulting from explosions. Additionally, it utilizes artificial intelligence (AI) voice recognition technologies, which analyze and recognize a person's voice to determine whether they are in a critical situation. Data is transmitted via LoRa technology to ensure communication even in the event of a traditional network failure, ensuring system continuity.

The project results demonstrated the system's effectiveness in early detection of hazards and rapid transmission of data to the emergency center. The center communicates with people and sends text messages, which are converted into voice messages to inform the person that the Civil Defense has intervened to resolve the emergency.

In addition, a Civil Defense website was designed to receive emergency calls and provide immediate intervention. Anyone can create an account by entering their data and ID number to identify the risks they are exposed to in emergency situations.

Overall, this project represents an important step toward using modern technology and AI to protect lives in the most difficult and dangerous circumstances, with broad potential for future development.

6.2 Future work

Future work on the Emergency Detection and Response System in High-Risk War Zones focuses on several key areas to develop the project and increase its efficiency and effectiveness, including:

1. Adding new sensors to enhance the system's capabilities to ensure emergency detection from all directions.
2. Working on using new artificial intelligence algorithms to analyze data from various sensors.
3. Using long-life batteries to ensure the system continues to operate for longer periods.
4. Protecting data stored and transmitted between components.
5. Directly linking the system with local emergency systems or rescue teams to speed up response and enhance field coordination.

نموذج إقرار وتوثيق استخدام أدوات الذكاء الاصطناعي في مشروع التخرج

Student Name(s): Khaled Hroub, Adam Al-Amour, Ahmad Al-Tall	Project Title: LoRa-Based Emergency Detection and Response System for War Zones
University ID Number(s): 207801 201086 201149	Academic Supervisor Name: Dr. Liana Al-Tamimi
Year/Semester: 2025	Faculty / Department: Faculty of Information Technology and Computer Engineering

Purpose of Use	Tool Name	Locations in Report (Chapters/Pages)	Main Prompt Used	Was the Output Modified ?	Estimated Usage Percentage	comments
Translation clarity verification	ChatGPT	Chapter 1 and 3 – While writing the abstract in English	Is this abstract clear to a non-technical reader	Yes	4%	Review of English presentation style
Generating technical comparisons	ChatGPT	Chapter 3 – Comparative tables	Compare MQ-2 and MQ-5 in terms of usage and accuracy	Yes	4%	Used to assist in phrasing the comparison table
Reviewing and improving research questions	ChatGPT	Chapter 2 – Literature review	we improve the comparison question between our project and others?	Yes	5%	Enhanced comparison formulation in the table

Data analysis	Gemini	Chapter 5 Evaluation and results	Assistance in reviewing test data patterns, interpreting them, and justifying results	Yes	6%	Used solely to analyze the meaning of test results and improve their formulation
Proofreading	Grammarly	All report	_____	Yes	15%	_____
writing software codes	GitHub Copilot	Chapter 4 Programmi ng LoRa data transmissi on	Programming data sending and receiving between two ESP32 units via LoRa periodically	Yes	6%	The generated code was modified to match the project structure

إقرار فريق مشروع التخرج:

نقر بأن استخدام أدوات الذكاء الاصطناعي تم بشكل مسؤول وبما يتوافق مع السياسات الجامعية، وقد راجعنا
وحررنا كافة المخرجات بما يعكس فهمنا الشخصي.

توقيع الطالب / الطلبة:

التاريخ:

References

[1] IEEE Public Safety Technology, "Breaking language barriers: NLP-powered emergency communication systems for multilingual communities," Accessed: Nov. 9, 2024. [Online]. Available:

<https://publicsafety.ieee.org/topics/breaking-language-barriers-nlp-powered-emergency-communication-systems-for-multilingual-communities>

[2] D. R. Sharma, R. R. Raghuwanshi, T. Chandak, and D. Ramdasi, "LoRa-based IoT system for emergency assistance and safety in mountaineering," **Int. J. Safety and Security Eng.**, vol. 13, no. 3, pp. 491-500, Jun. 2023. Accessed: Nov. 10, 2024. [Online]. Available:

<https://doi.org/10.18280/ijssse.130311>

[3] D. Najjar, J. Shabaneh, and R. Zaro, "Design and development of safety monitoring system for gas leakage and ventilation," PPU, 27 Dec. 2022. Accessed: Nov. 11, 2024. [Online]. Available: <https://scholar.ppu.edu/handle/123456789/8964>

[4] I. Badawi, R. J. Abd-Aalla, and R. A. Abu-Sharar, "Smart House," PPU, Jun. 2009. Accessed: Nov. 8, 2024 [Online]. Available:

<https://scholar.ppu.edu/handle/123456789/7491>

[5] Raspberry Pi Foundation, "Raspberry Pi 4 Model B – Technical specifications," Accessed: Nov. 13, 2024. [Online]. Available:

<https://www.raspberrypi.com/documentation/>

[6] "Raspberry Pi 4," 12Geeks, Accessed: Nov. 14, 2024. [Online]. Available:

<https://12geeks.com/shop/raspberry-pi-3-model-b-972.html>

- [7] Heltec Automation, "LoRa32 V2 – LoRa + OLED Display ESP32 Development Board," Accessed: Nov. 14, 2024. [Online]. Available: <https://www.heltec.org/project/lorawan-esp32/>
- [8] "TTGO LoRa32 SX1276 OLED with Arduino IDE," Random Nerd Tutorials, Accessed: Nov. 14, 2024. [Online]. Available: <https://randomnerdtutorials.com/ttgo-lora32-sx1276-arduino-ide>
- [9] Seeed Studio, "Grove - Gas Sensor (MQ-5) Documentation," Accessed: Nov. 15, 2024. Available: https://files.seeedstudio.com/wiki/Grove-Gas_Sensor-MQ5/res/MQ-5.pdf
- [10] "Grove - Gas Sensor (MQ5)," Seeed Studio Wiki, Accessed: Nov. 17, 2024. [Online]. Available: https://wiki.seeedstudio.com/Grove-Gas_Sensor-MQ5
- [11] Elprocus, "Flame Sensor: Working and Its Applications," Accessed: Nov. 17, 2024 Available: <https://www.elprocus.com/flame-sensor-working-and-its-applications/>
- [12] "Flame Sensor Working and Its Applications," Elprocus, Accessed: Nov. 17, 2024. [Online]. Available: <https://www.elprocus.com/flame-sensor-working-and-its-applications>
- [13] Electroschematics, "SW-420 Vibration Sensor: Working and Applications," Available: [Accessed: Nov. 20, 2024]. <https://www.electroschematics.com/sw-420/>
- [14] "SW-420 Vibration Sensor Module," Components101, Accessed: Nov. 20, 2024. [Online]. Available: <https://components101.com/sensors/sw-420-vibration-sensor-module>

- [15] "SR-ULM10 Ultracompact Clip-On Lavalier Microphone with USB-A Connector," Saramonic, Accessed: Nov. 20, 2025. [Online]. Available: <https://saramonicusa.com/sr-ulm10-ultracompact-clip-on-lavalier-microphone-with-usb-a-connector-for-mac-windows-computers-with-a-built-in-6-56-foot-2m-cable>
- [16] "USB Mini Stereo Speaker," LDLC, Accessed: Nov. 25, 2024. [Online]. Available: <https://www.ldlc.com/en/product/PB00254891.html>
- [17] "eSpeak: Open-Source Text-to-Speech Synthesizer," eSpeak Project, Accessed: Nov. 27, 2024. [Online]. Available: <https://espeak.sourceforge.net/>
- [18] "Hugging Face NLP Course," Hugging Face, Accessed: Nov. 30, 2024. [Online]. Available: <https://huggingface.co/learn/nlp-course>
- [19] "Speech Recognition in Python," Medium, Accessed: Dec. 1, 2024. [Online]. Available: <https://medium.com/naukri-engineering/speech-recognition-in-python-fcda027a97a1>