SMART FIRE ALARM SYSTEM

Real Time Project Report
Review-2

Bachelor of Technology (B.Tech)

In

COMPUTER SCIENCE AND ENGINEERING
Internet Of Things[IOT]

By

JANADI SANGAMESH	**23AG5A6902**
S SAI VAIBHAV	**23AG5A6905**
T HARI SHANKAR REDDY	**23AG5A6906**
V SURESH	**23AG5A6907**

Under the Esteemed Guidance of

INTERNAL GUIDE: T RATNAMALA

Associate Professor's Dept. of CSE

Department of Computer Science and Engineering

ACE ENGINEERING COLLEGE

An Autonomous Institution

(NBA ACCREDITED B.TECH COURSES: EEE, ECE, MECH, CIVIL & CSE, ACCORDED NAAC 'A' GRADE)

(Affiliated to Jawaharlal Nehru Technological University, Hyderabad, Telangana)

Ghatkesar, Hyderabad - 501 301

December 2023

ACE
Engineering College
An Autonomous Institution

(NBA ACCREDITED B.TECH COURSES: EEE, ECE, MECH, CIVIL & CSE, ACCORDED NAAC 'A'GRADE)

Ghatkesar, Hyderabad- 501 301

(Affiliated to Jawaharlal Nehru Technological University Hyderabad)

Web site: www.aceec.ac.in E-mail: info@aceec.ac.in

CERTIFICATE

This is to certify that the Real Time Project work entitled **"SMART FIRE ALARM SYSTEM"** is being submitted by **JANADI SANGAMESH (23AG5A6902) S SAI VAIBHAV (23AG5A6905) T HARI SHANKAR REDDY(23AG5A6906) V SURESH (23AG5A6907)** in partial fulfillment for the award of Degree of **BACHELOR OF TECHNOLOGY** in **COMPUTER SCIENCE AND ENGINEERING** to the Jawaharlal Nehru Technological University, Hyderabad during the academic year 2023-24 is a record of bonafide work carried out by him under our guidance and supervision.

The results embodied in this report have not been submitted by the student to any other University or Institution for the award of any degree or diploma.

Internal Guide
Mr.V Veeresh
Assistant Professor
Dept. of CSE

Head of the Department
Dr.M.V.VIJAYASARDI
Professor and Head
Dept. of CSE

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

I would like to express our gratitude to all the people behind the screen who have helped me, transform an idea into a real time application.

I would like to express my heart-felt gratitude to our parents without whom I would not have been privileged to achieve and fulfill my dreams.

A special thanks to our Secretary, **Prof. Y. V. GOPALA KRISHNA MURTHY,** for having founded such an esteemed institution. I am also grateful to our beloved principal, **Dr. B. L. RAJU** for permitting us to carry out this project.

I profoundly thank **Dr. M. V. VIJAYA SARADHI**, Head of the Department of Computer Science and Engineering, who has been an excellent guide and also a great source of inspiration to my work.

I extremely thank **Mr. V Veeresh** Assistant Professor, Project coordinator who helped us in all the way in fulfilling of all aspects in completion of our Mini-Project.

I am very thankful to my internal guide **T RATNAMALA ,** associate Professor, of the Department of Computer Science and Engineering who has been an excellent and also given continuous support for the Completion of my project work.

The satisfaction and euphoria that accompany the successful completion of the task would be great, but incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crown all the efforts with success. In this context, We would like to thank all the other staff members, both teaching and non-teaching, who have extended their timely help and eased our task.

JANADI SANGAMESH 23AG5A6902

S SAI VAIBHAV 23AG5A6905

T HARI SHANKAR REDDY 23AG5A6906

V SURESH 23AG5A6907

DECLARATION

I hereby declare that this mini project entitled "**SMART FIRE ALARM SYSTEM**" Submitted to the **ACE Engineering College,** is a record of an original work done by me under the guidance of **T RATNAMALA & V Veeresh**, Assistant Professor of the Department of Computer Science and Engineering, **ACE Engineering College,** and this project work submitted in the partial fulfillment of the requirements for the mini project; the results embodied in this thesis have not been submitted to any other university or institute for award of any degree or diploma.

JANADI SANGAMESH 23AG5A6902

S SAI VAIBHAV 22AG5A6905

T HARI SHANKAR REDDY 23AG5A6906

V SURESH 23AG5A6907

ABSTRACT

A fire alarm system using Arduino and a GSM module represents a modern and efficient approach to enhancing fire safety in residential and commercial environments. This system integrates an Arduino microcontroller as the central processing unit and a GSM module for real-time communication. The key components include smoke detectors, temperature sensors, and a buzzer for local alerts.

The Arduino microcontroller continuously monitors data from the smoke and temperature sensors. When abnormal conditions, such as high levels of smoke or a significant rise in temperature, are detected, the Arduino processes this information and activates the alarm system. The local alarm, usually a buzzer, provides an immediate auditory warning to occupants, prompting them to evacuate the premises and take necessary actions.

Simultaneously, the GSM module, interfaced with the Arduino, sends an SMS alert to pre-configured phone numbers, including those of homeowners, security personnel, and emergency services. This ensures that relevant individuals are informed of the potential fire hazard promptly, even if they are off-site. The ability to receive real-time notifications remotely significantly reduces response time, helping to mitigate potential damage and save lives.

The system's design emphasizes simplicity, cost-effectiveness, and ease of installation, making it accessible for a wide range of users. Arduino's open-source platform allows for customization and scalability, enabling users to expand the system with additional sensors or integrate other functionalities, such as connecting to the Internet of Things (IoT) for enhanced control and monitoring. Users can program the Arduino to suit specific needs, ensuring a tailored approach to fire safety.

Moreover, the use of the GSM module allows for broader communication capabilities. In areas with limited internet connectivity, the GSM module ensures that alerts are still sent via cellular networks, making the system reliable in various environments.

In conclusion, integrating Arduino and GSM technology in fire alarm systems represents a significant advancement in early warning and notification systems. By providing a reliable, efficient, and user-friendly solution for fire detection and communication, this system enhances safety and ensures timely responses to potential fire hazards, ultimately contributing to better protection of life and property.

INDEX

CONTENTS **PAGE NO.**

LIST OF FIGURES

CHAPTER-1 INTRODUCTION

A fire alarm system using Arduino and a GSM module is an innovative solution designed to enhance fire safety in both residential and commercial settings. This system leverages the capabilities of the Arduino microcontroller to process data from various sensors and a GSM module to send real-time alerts, ensuring prompt notification and response to potential fire hazards. The primary objective of this system is to provide a cost-effective, reliable, and efficient means of detecting fire and communicating alerts to relevant individuals, thereby minimizing damage and safeguarding lives.

Here's an introduction to the key components and features of a smart fire alarm system:

Arduino Microcontroller

Central Processing Unit: The Arduino serves as the heart of the system, processing data from connected sensors and managing the alarm and communication functions.

Customizable and Scalable: The open-source nature of Arduino allows users to tailor the system to their specific needs and expand it with additional sensors or features as required.

GSM Module:

Real-Time Alerts: The GSM module sends SMS notifications to pre-configured phone numbers, ensuring that homeowners, security personnel, and emergency services are promptly informed of any fire hazards.

Wide Communication Range: Utilizes cellular networks, making it effective even in areas with limited internet connectivity.

Smoke and Temperature Sensors:

Continuous Monitoring: These sensors detect smoke and sudden temperature increases, providing early warning signs of a potential fire.

High Sensitivity and Accuracy: Ensures reliable detection of fire-related anomalies.

Local Alarm (Buzzer):

Immediate Auditory Warning: Alerts occupants within the vicinity to evacuate and take necessary action quickly.

Simple and Effective: Easy to install and operate, ensuring immediate recognition of danger.

Ease of Installation and Use:

User-Friendly Design: The system is designed to be simple to set up and use, making it accessible to a broad range of users, including those with minimal technical expertise.

Cost-Effective: Utilizes affordable components, making it an economical option for enhancing fire safety.

Flexibility and Integration:

IoT Connectivity: Can be integrated with Internet of Things (IoT) platforms for advanced monitoring and control.

Expandable: Additional sensors or modules can be added to enhance functionality and coverage.

1.1. Problem Statement

Fire-related incidents pose a significant threat to life and property in both residential and commercial settings. Traditional fire alarm systems, while effective in alerting occupants through audible alarms, often fail to promptly notify off-site individuals such as homeowners, security personnel, and emergency services. This delay in communication can result in increased damage and potential loss of life due to slower response times.

Existing systems tend to be costly, complex to install, and difficult to customize for specific needs, making them less accessible to a broader audience. Additionally, in remote or underdeveloped areas with limited internet connectivity, conventional fire alarm systems with internet-based alert mechanisms become unreliable. There is a critical need for a cost-effective, reliable, and easily customizable fire alarm system that can provide real-time notifications to relevant stakeholders regardless of their location.

The integration of an Arduino Uno microcontroller with a GSM module offers a viable solution to these challenges. The Arduino Uno, known for its simplicity and versatility, can process data from smoke and temperature sensors to detect early signs of fire. The GSM module can then send real-time SMS alerts to pre-configured phone numbers, ensuring immediate notification to key individuals and emergency responders.

Addressing these challenges through the development of a fire alarm system using Arduino Uno and a GSM module can significantly enhance fire safety measures, ensuring rapid response and minimizing potential damage and loss of life.

CHAPTER – 2
LITERATURE SURVEY

1. Overview of Arduino-based Fire Alarm Systems

Arduino microcontrollers have gained popularity for their flexibility, ease of use, and open-source nature. They serve as the central processing units in fire alarm systems, capable of interfacing with various sensors and communication modules. Several studies have explored the integration of Arduino with smoke and temperature sensors for effective fire detection. Rathore, H., & Yadav, S. (2016). In their paper, "Fire Detection System Using Arduino and GSM Module," the authors presented a system utilizing an Arduino Uno, MQ-2 smoke sensor, and DS18B20 temperature sensor. The system effectively detected fire and sent SMS alerts through a GSM module, demonstrating a cost-effective and reliable solution for early fire detection.

2. GSM Modules for Real-time Notifications

GSM modules enable fire alarm systems to send real-time alerts to predefined contacts, ensuring immediate notification even when the occupants are not present. Research has focused on the reliability and efficiency of these modules in various conditions.Sharma, A., & Kumar, A. (2017).Their study, "Fire Alarm System Using Arduino and GSM," highlighted the effectiveness of the GSM module in sending real-time SMS alerts. The system was tested in different environments to assess the reliability of communication under variable network conditions.

3. Sensor Integration and Accuracy

The accuracy of fire detection systems largely depends on the quality and integration of sensors. Smoke and temperature sensors are critical components, and their proper calibration and placement can significantly impact system performance. Patil, P., & Kulkarni, S. (2018) The paper, "Design and Implementation of Fire Detection and Notification System using Arduino and GSM," examined the integration of MQ-2 smoke sensors and LM35 temperature sensors with Arduino. The study concluded that sensor calibration and strategic placement are vital for accurate fire detection.

CHAPTER – 3

SYSTEM ANALYSIS

3.1. EXISTING SYSTEM

While smart fire alarm system offer significant improvements they still have several limitations that need to be addressed to enhance their effectiveness and usability. Here are some common limitations of existing smart fire alarm system:

Limited Sensor Accuracy and Sensitivity: Inexpensive sensors can sometimes be overly sensitive or not sensitive enough, leading to false alarms or missed detections. Factors like dust, humidity, and temperature fluctuations can affect sensor performance, leading to inaccurate readings.

Communication Reliability: The GSM module relies on cellular networks, which may be unreliable or have poor coverage in remote or rural areas, potentially delaying critical alerts.SMS alerts can sometimes experience delays due to network congestion or other issues, reducing the system's effectiveness in emergencies.

Power Supply Issues: The system typically requires a constant power supply, and power outages can render it non-functional unless a backup power solution is in place. If using battery backups, regular maintenance and replacement of batteries are necessary to ensure continuous operation.

Complex Installation and Maintenance: Although Arduino systems are relatively simple, they still require a certain level of technical knowledge for proper installation and configuration, which may not be accessible to all users.

Regular maintenance is required to ensure sensors and the GSM module function correctly, which can be a burden for some users.

Scalability and Customization Limits: While Arduino is flexible, expanding the system significantly can become complex and may require additional hardware and programming expertise. Customizing the system for specific needs can be challenging for users without programming skills or experience with Arduino.

Cost Considerations: Despite being cost-effective compared to some traditional systems, the initial cost of setting up an Arduino-based fire alarm system with reliable sensors and a GSM module can still be a barrier for some users. Using low-cost components to keep the system affordable can compromise the reliability and durability of the system.

Addressing these drawbacks is crucial for enhancing the effectiveness and reliability of fire alarm systems using Arduino and GSM modules, ensuring they provide a dependable solution for fire safety.

3.2. PROPOSED SYSTEM

The proposed fire alarm system leveraging Arduino and a GSM module represents a significant advancement in fire safety technology. By integrating the Arduino microcontroller and GSM communication capabilities, this system addresses key limitations of traditional fire alarm systems, offering enhanced reliability, real-time communication, and customization potential.

The system sends real-time SMS notifications to pre-configured phone numbers, ensuring rapid communication of potential fire hazards to homeowners, security personnel, and emergency services.Users can easily configure and update the list of recipients for the SMS alerts, ensuring the right people are notified promptly.The system incorporates both smoke and temperature sensors for comprehensive monitoring, providing early detection of fire through smoke and heat anomalies.Ensures reliable detection and minimizes false alarms.

Equipped with a buzzer that provides an immediate auditory warning to occupants, prompting quick evacuation and response. Includes LED indicators to provide a visual alert alongside the auditory alarm, enhancing awareness. Users can send SMS commands to the system to check status updates, reset the alarm, or perform diagnostic tests, allowing for remote management and control. Potential integration with IoT platforms for advanced monitoring and control through a centralized interface.Features a battery backup system to ensure continuous operation during power outages, maintaining fire monitoring and alert capabilities. Low Power Consumption: Designed to be energy-efficient, extending battery life and reducing maintenance.

The system is designed for easy installation with clear instructions, making it accessible to users with minimal technical expertise. An optional LCD display can provide real-time system status, sensor readings, and alert messages. Users can add additional sensors (e.g., gas, flame) to the system as needed, tailoring it to specific environmental requirements. The system can be expanded to cover larger areas or multiple zones within a building, providing comprehensive fire safety coverage.Logs sensor data and alert events for future analysis and reporting, helping to identify patterns and improve system performance. The system can generate periodic reports summarizing system status and events, aiding in maintenance and review.By incorporating these new features, the proposed fire alarm system using Arduino and GSM module offers a robust, versatile, and user-friendly solution for enhanced fire safety and rapid response.

3.3. SOFTWARE REQUIREMENTS SPECIFICATION

3.3.1. Introduction :

The Software Requirements Specification (SRS) for the fire alarm system using Arduino and GSM module outlines the detailed functional and non-functional requirements essential for its development. This document serves as a blueprint, defining the software functionalities, interfaces, and performance criteria necessary to ensure reliable operation and effective communication in detecting and alerting potential fire hazards. By specifying these requirements clearly, the SRS aims to guide developers in implementing a robust system that integrates seamlessly with Arduino hardware and GSM technology, ultimately enhancing fire safety measures in residential and commercial environments.

3.3.2. Purpose :

A fire alarm system built with an Arduino and a GSM module serves the primary purpose of enhancing fire safety in a particular location. This system leverages the capabilities of both these components to detect the presence of fire and subsequently notify designated individuals. The Arduino acts as the brains of the operation. It continuously monitors data received from fire sensors like smoke detectors or temperature sensors. If the sensor readings surpass a predefined threshold, indicating a potential fire, the Arduino springs into action.

Here's where the GSM module comes into play. The Arduino communicates with the GSM module, which essentially functions as a cellular modem. This module, equipped with a SIM card, utilizes the cellular network to transmit an alert. This alert can be in the form of a pre-recorded message or a text message containing vital information about the fire's location. The message is sent to programmed phone numbers, ensuring that the relevant people are informed immediately, allowing them to take prompt action to extinguish the fire and ensure everyone's safety. In essence, this Arduino-GSM fire alarm system bridges the gap between fire detection and remote notification, providing a crucial line of defense against the devastating consequences of fires.

3.3.3. Scope of the Project :

The scope of the The fire alarm system using Arduino and a GSM module encompasses several key aspects within its scope, focusing on enhancing fire detection, communication, and response capabilities. First and foremost, the system integrates Arduino's robust microcontroller capabilities to monitor critical parameters such as smoke levels and temperature fluctuations in real-time. This ensures early detection of fire hazards, allowing for swift initiation of emergency protocols.

the GSM module enables seamless communication by transmitting immediate SMS alerts to designated stakeholders, including homeowners, security personnel, and emergency services. This feature is essential for ensuring that relevant parties are promptly informed regardless of their physical location, thereby facilitating rapid response and mitigation of potential damages.

the system emphasizes ease of installation and operation, designed to be user-friendly even for individuals with limited technical expertise. Its modular design allows for scalability and customization, accommodating additional sensors or integration with existing home automation systems to enhance overall safety measures.

in remote or rural areas where internet connectivity may be limited, the GSM module's reliance on cellular networks ensures reliable communication of fire alerts. This makes the system particularly suitable for locations without robust internet infrastructure, extending its utility to a broader range of environments and communities.

cost-effectiveness is a critical aspect within the scope of this system. By leveraging affordable Arduino components and utilizing GSM technology, the system provides a reliable and economical solution for enhancing fire safety in both residential and commercial environments. This affordability ensures broader accessibility and adoption, making advanced fire detection and notification capabilities accessible to a wider range of users and settings.

3.3.4. Overall Description :

The fire alarm system using Arduino and a GSM module is designed to enhance fire safety measures in residential, commercial, and institutional settings. By leveraging the Arduino microcontroller for data processing and the GSM module for real-time communication, the system aims to detect fire hazards promptly and notify relevant stakeholders via SMS alerts. Here's an overall description of its features:

System Architecture : Arduino Uno Integration Serves as the core processing unit, responsible for data acquisition from connected sensors (e.g., smoke, temperature) and decision-making logic for alarm triggering. GSM Module Facilitates real-time communication by sending SMS alerts to predefined phone numbers upon detecting fire-related anomalies.

Detection Mechanism : Smoke and Temperature Sensors: Continuously monitor the environment for signs of fire, detecting smoke particles and sudden temperature increases accurately. Dual Sensor Integration: Ensures reliable detection while minimizing false alarms, enhancing system reliability

Notification and Alert System: SMS Alerts Utilizes the GSM module to send immediate SMS notifications to designated recipients, including homeowners, security personnel, and emergency services. Alert Acknowledgment Provides functionality for users to acknowledge alerts, ensuring timely response and accountability.

User Interfaces : Local Interface: Allows users to monitor system status locally via indicators or a display panel, enabling quick assessment of alarm conditions. Mobile Interface Enables remote monitoring and interaction through SMS alerts, providing access to system status and allowing users to manage alerts from anywhere..

Security and Safety Considerations: Data Encryption Implements encryption protocols to secure sensitive information transmitted via SMS alerts. System Integrity Incorporates measures to prevent tampering or unauthorized access to system settings and alerts.

Operational Constraints: Power Supply Specifies power requirements to sustain continuous operation, potentially incorporating backup solutions for uninterrupted functionality. Environmental Adaptability Considers operational limitations related to environmental factors such as temperature, humidity, and physical installation constraints.

Overall, The fire alarm system employing Arduino and a GSM module integrates advanced technology to enhance safety in residential, commercial, and institutional environments. This system utilizes Arduino Uno as the central processing unit, managing inputs from smoke and temperature sensors to detect fire hazards promptly. Upon detection, the system triggers audible alarms locally and sends real-time SMS alerts via the GSM module to designated recipients, including homeowners, security personnel, and emergency services. This dual-notification feature ensures that stakeholders are promptly informed, regardless of their location, enabling swift responses to mitigate potential damage and safeguard lives. Designed for ease of installation and operation, the system offers scalability to accommodate additional sensors or functionalities, providing a flexible solution tailored to specific user needs.

3.3.5. SYSTEM FEATURES :

FUNCTIONAL REQUIREMENTS :

Obstacle Detection The smart blind stick should accurately detect obstacles in the user's path in real-time using sensors such as ultrasonic sensors, infrared sensors, or computer vision systems. It should provide timely feedback to the user about the presence, location, and size of obstacles to avoid collisions. Navigation Assistance The device should offer navigation assistance to help users reach their destinations safely and efficiently .It should integrate with GPS technology and digital mapping services to provide turn-by-turn directions, distance to destination, and points of interest along the route.

Feedback Mechanisms The smart blind stick should provide feedback mechanisms such as auditory cues (beeps, tones, spoken instructions), vibratory cues (vibrations), or haptic cues (tactile feedback through the handle).Feedback should alert users to obstacles, provide navigation instructions, and convey information about the environment in a non-visual manner.

Customization Option Users should be able to customize settings such as feedback intensity, feedback modality, navigation preferences, and sensitivity of obstacle detection. Customization options accommodate individual user preferences and needs for a personalized user experience.

Connectivity Features The smart blind stick should offer connectivity features such as Bluetooth, Wi-Fi, or NFC for integration with smartphones, wearable devices, and assistive technologies.

Connectivity enables data exchange, remote control, and access to additional functionalities such as GPS navigation, remote assistance, and online services. User Interface Design User interfaces should be intuitive, accessible, and easy to use for users with varying levels of technological proficiency. Interfaces may include tactile buttons, touch controls, or voice commands for interaction and customization.

Power Management The device should have efficient power management features to optimize battery usage and prolong battery life. Power management includes low-power modes, automatic shut-off, and battery level indicators to ensure uninterrupted operation. Durability and Portability The smart blind stick should be lightweight, portable, and durable for use in various environments and conditions. Materials should be weather-resistant and suitable for outdoor use.

Safety Features Safety features should prioritize user safety with reliable obstacle detection, navigation assistance, and emergency assistance options

NON-FUNCTIONAL REQUIREMENTS:

The smart blind stick should have high performance, with fast and accurate obstacle detection and navigation assistance capabilities. Response times for detecting obstacles and providing feedback should be minimal to ensure user safety and confidence. The device should be reliable and dependable in various environmental conditions and situations. It should have a low rate of false positives/negatives in obstacle detection and provide consistent navigation assistance.Usability The smart blind stick should be user-friendly and easy to use for individuals with varying levels of technological proficiency and vision impairment. Interfaces should be intuitive, with clear instructions and feedback cues, and customizable to accommodate user preferences. Accessibility The device should adhere to accessibility standards and guidelines to ensure usability for visually impaired individuals. Features such as screen reader compatibility, braille output, and tactile interfaces enhance accessibility and usability.

Security The smart blind stick should prioritize user privacy and data security. It should implement encryption and secure communication protocols to protect user data exchanged with external devices or online services. Scalability The device should be scalable to accommodate future updates, enhancements, and expansions in functionality. It should support software updates, additional features, and integration with emerging technologies to meet evolving user needs. Interoperability The smart blind stick should be interoperable with existing assistive technologies, smartphones, wearable devices, and online services. Compatibility with standard communication protocols and formats enables seamless integration with external devices and platforms. Portability The device should be lightweight, portable, and easy to carry for users during daily activities. Compact design and ergonomic features enhance portability and user comfort during extended use. Durabilit The smart blind stick should be durable and robust, capable of withstanding daily use and potential impacts. Materials should be weather-resistant and suitable for use in various environmental conditions, ensuring long-term reliability and performance.

3.3.6. HARDWARE REQUIREMENTS

Arduino pro mini

Haptic motor module

Switches Set

Single Li-ion battery holder

Li-ion Battery

Buzzer Set

LED Set

3D printed model

Ultra sonic sensor

USB to mini USB cable

Jumper wires

3.3.7. SOFTWARE REQUIREMENTS

Firmware

Signal processing algorithms

Obstacle detection algorithms

Navigation algorithms

User interface software

Device drivers

Connectivity software

CHAPTER – 4
SYSTEM DESIGN

4.1. DATA FLOW DIAGRAMS

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It can be manual, automated, or a combination of both.
It shows how data enters and leaves the system, what changes the information, and where data is stored.
The objective of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communication tool between a system analyst and any person who plays a part in the order that acts as a starting point for redesigning a system.

LEVEL – 0 DFD

It is also known as fundamental system model, or context diagram represents the entire software requirement as a single bubble with input and output data denoted by incoming and outgoing arrows. Then the system is decomposed and described as a DFD with multiple bubbles.

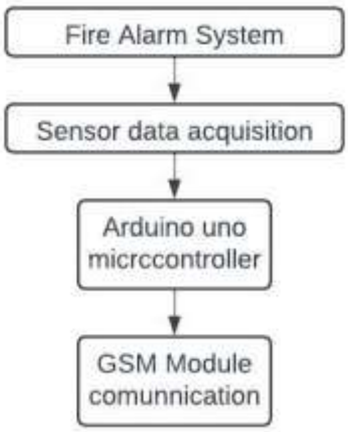


Fig 4.1.1 : Level 0

LEVEL – 1 DFD

In 1-level DFD, a context diagram is decomposed into multiple bubbles/processes. In this level, we highlight the main objectives of the system and breakdown the high-level process of 0-level DFD into subprocesses.

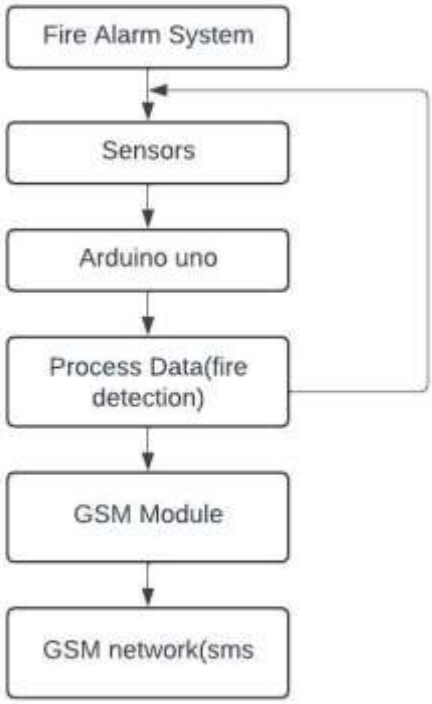


Fig 4.1.2 : Level 1

LEVEL – 2 DFD

A Level 2 Data Flow Diagram (DFD) provides a more detailed representation of the system's processes and data flows compared to the Level 1 DFD. It breaks down the processes identified in the Level 1 DFD into sub-processes and illustrates the interactions between these processes. The Level 2 DFD focuses on the internal operations within each process and shows the specific data inputs, outputs, and data stores associated with each process. It provides a clearer understanding of the system's functionality and the flow of data between different components at a more granular level.

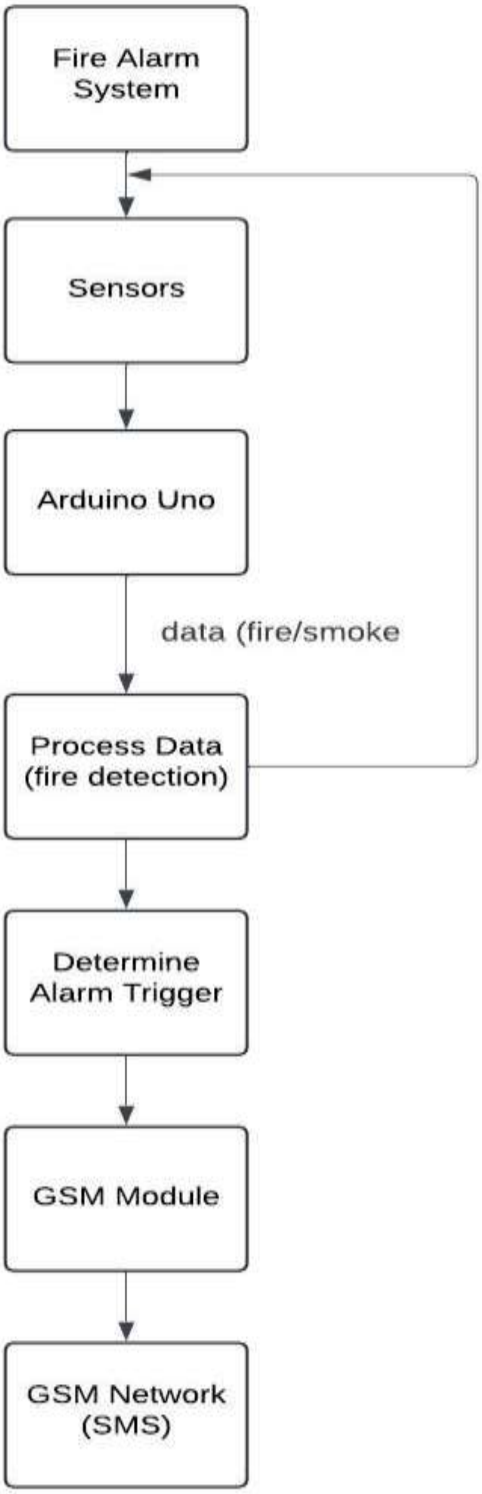


Fig 4.1.3 : Level 2

4.2. UML DIAGRAMS

UML stands for Unified Modelling Language. UML is a standardized general purpose modelling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object-oriented computer software. In its current form UML is comprised of two major components: a Meta- model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modelling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modelling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modelling of large and complex systems.

The UML is a very important part of developing objects-oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

1. CLASS DIAGRAM

The class diagram is the main building block of object-oriented modeling. It is used both for general conceptual modeling of the systematic of the application, and for detailed modelling translating the models into programming code. Class diagrams can also be used for data modeling. The classes in a class diagram represent both the main objects, interactions in the application and the classes to be programmed. In the diagram, classes are represented with boxes which contain three parts:

The upper part holds the name of the class.

The middle part contains the attributes of the class.

The bottom part gives the methods or operations the class can take or undertake.

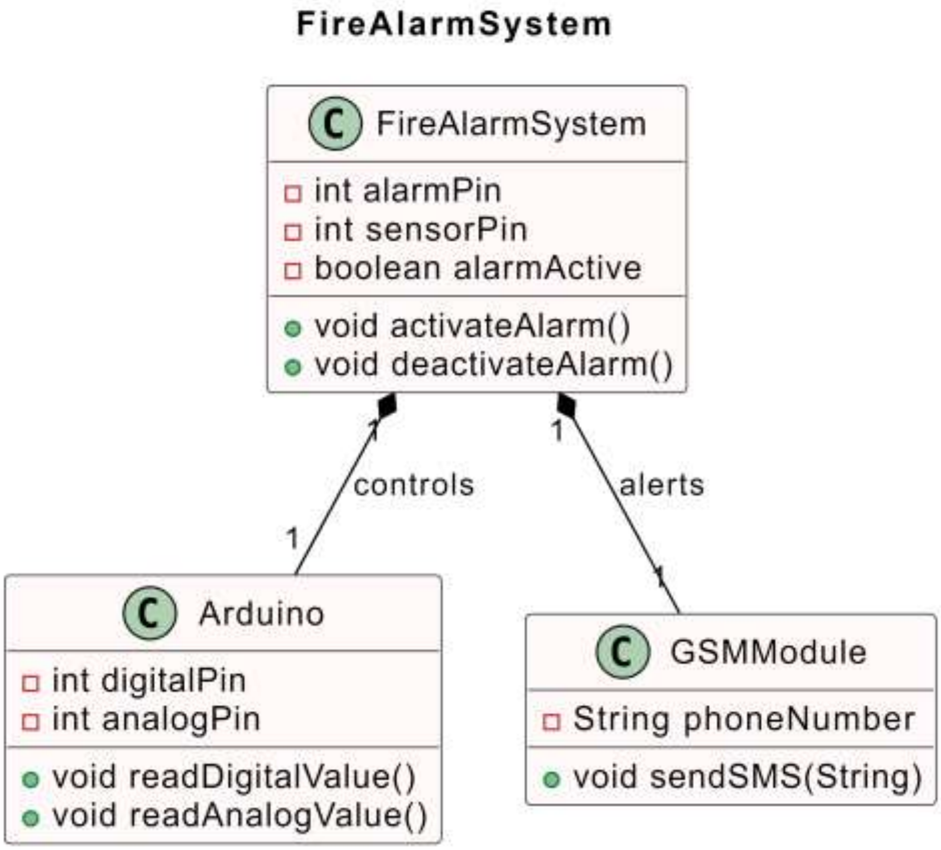


Fig 4.2.1: Class Diagram

2. Use-case Diagram:

In the Unified Modelling Language (UML), a use case diagram can summarize the details of your system's users (also known as actors) and their interactions with the system. To build one, you'll use a set of specialized symbols and connectors. An effective use case diagram can help your team discuss and represent:

- Scenarios in which your system or application interacts with people, organizations, or external systems
- Goals that your system or application helps those entities (known as actors)

 achieve the scope of your system.

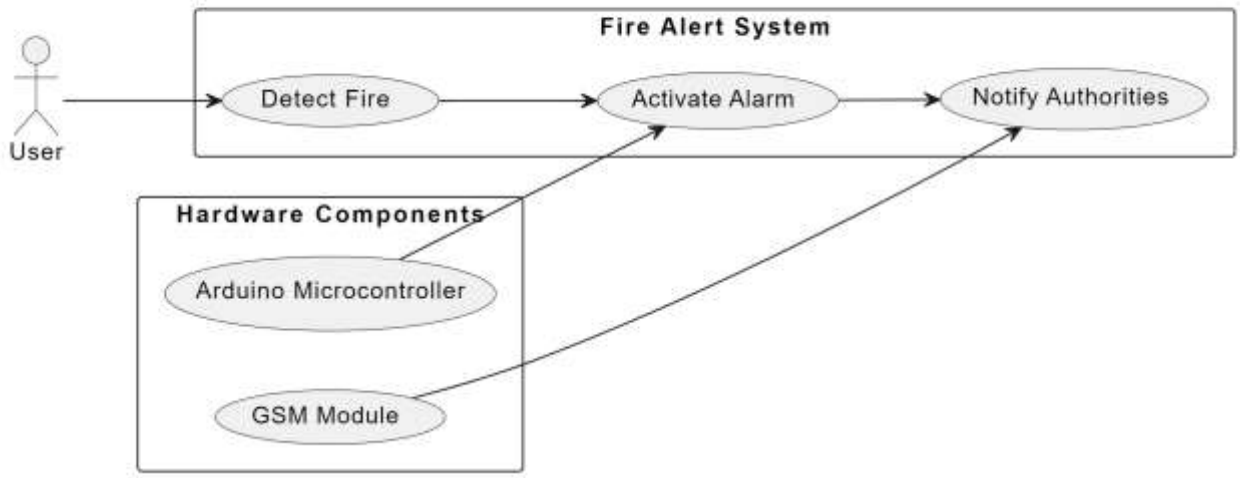


Fig 4.2.2: Use Case Diagram

3. Activity Diagram:

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc

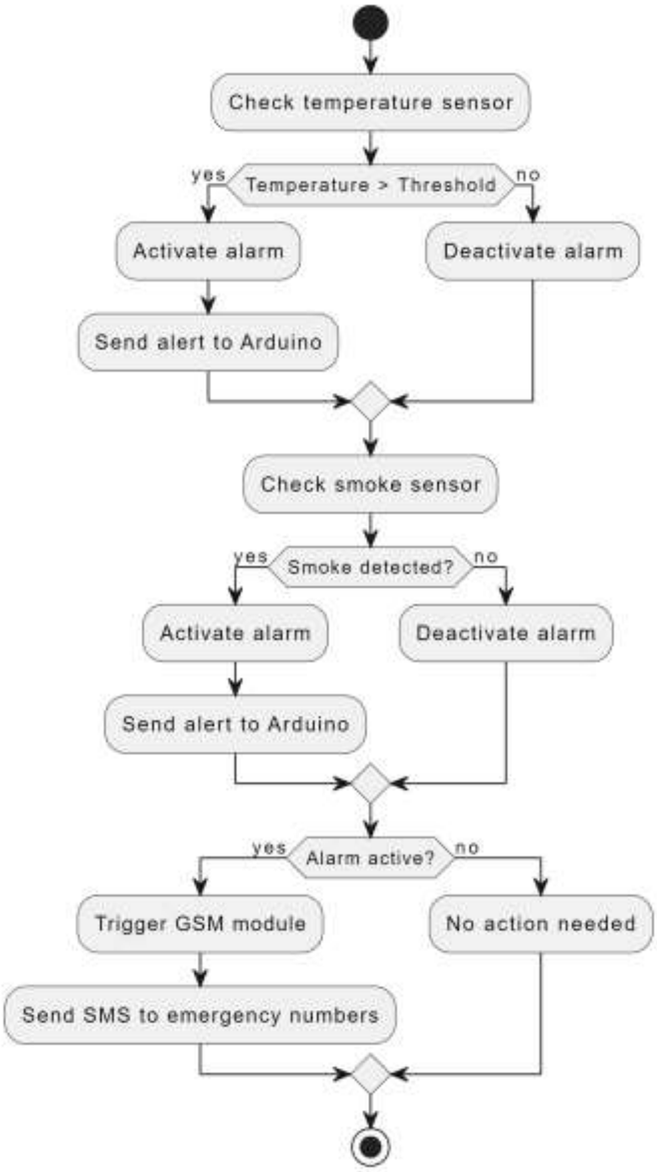


Fig 4.2.3: Activity Diagram

4. Sequence Diagram:

A sequence diagram or system sequence diagram (SSD) shows process interactions arranged in time sequence in the field of software engineering. It depicts the processes involved and the sequence of messages exchanged between the processes needed to carry out the functionality. Sequence diagrams are typically associated with use case realizations in the 4+1 architectural view model of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios. For a particular scenario of a use case, the diagrams show the events that external actors generate, their order, and possible inter-system events. All systems are treated as a black box; the diagram places emphasis on events that cross the system boundary from actors to systems. A system sequence diagram

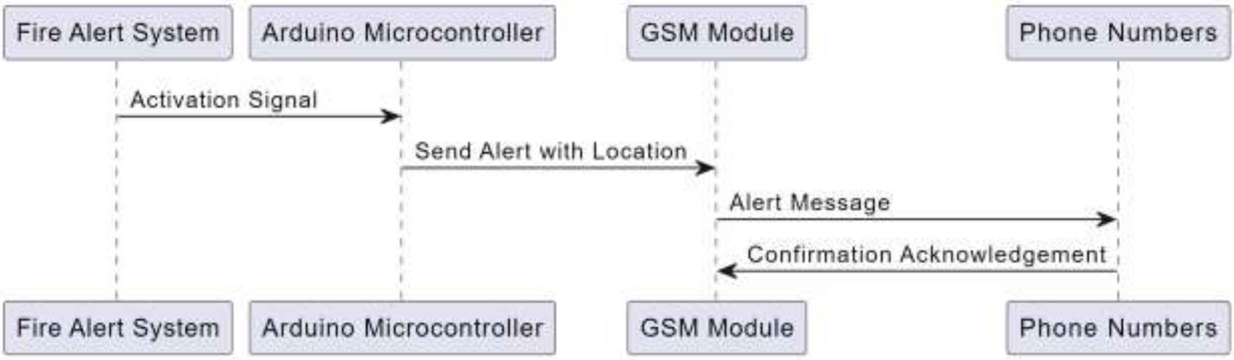


Fig 4.2.4: Sequence Diagram

5. ER DIAGRAM:

An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how "entities" such as people, objects or concepts relate to each other within a system. ER Diagrams are most often used to design or debug relational databases in the fields of software engineering, business information systems, education and research. Also known as ERDs or ER Models, they use a defined set of symbols such as rectangles, diamonds, ovals and connecting lines to depict the interconnectedness of entities, relationships and their attributes. They mirror grammatical structure, with entities as nouns and relationships as verbs. ER diagrams are related to data structure diagrams (DSDs), which focus on the relationships of elements within entities instead of relationships between entities themselves. ER diagrams also are often used in conjunction with data flow diagrams (DFDs), which map out the flow of information for processes or systems.

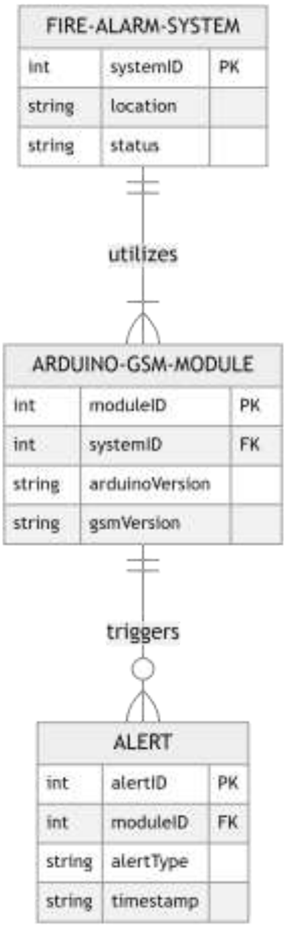


Fig 4.2.5 ER DIAGRAM

4.3. ARCHITECTURAL DIAGRAM

Architecture diagramming is the process of creating visual representations of software system components. In a software system, the term architecture refers to various functions, their implementations, and their interactions with each other.

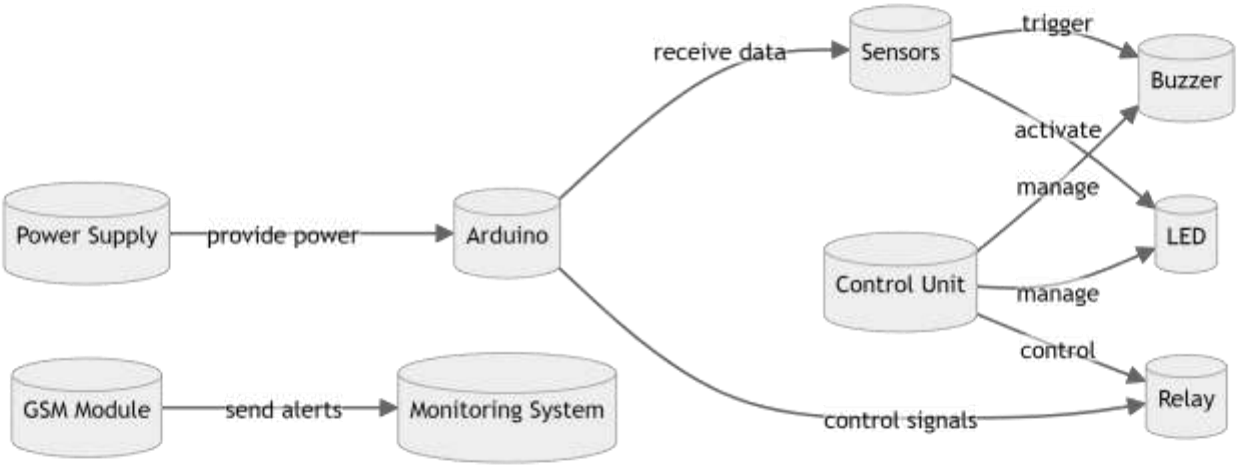


Fig 4.3: Architectural Diagram

s

CHAPTER – 5
MODULES

To implement this project, we have designed following modules.

fire Detection module

A fire detection module in a fire alarm system using Arduino and GSM modules represents a pivotal advancement in safety technology, combining robust sensor capabilities with real-time communication functionalities. At its core, this system integrates various sensors—such as smoke detectors, heat sensors, and flame detectors—each playing a crucial role in early fire detection. Smoke detectors, utilizing technologies like ionization or photoelectric sensors, monitor air quality for particles indicative of smoke, a primary early sign of fire. Heat sensors complement this by monitoring temperature changes, activating alarms upon sudden increases, while flame detectors identify specific light frequencies emitted by flames, ensuring comprehensive fire detection across different scenarios.

Practical implementation of this system involves meticulous sensor calibration and strategic placement to optimize detection accuracy and minimize false alarms. Calibration ensures sensors are sensitive enough to detect potential fire hazards while resilient against non-fire-related disturbances like dust or humidity. Strategic sensor placement considers airflow dynamics and potential fire sources, ensuring comprehensive coverage throughout the monitored area. Power considerations are equally critical, with the system typically equipped with reliable power sources and backup options, such as batteries, to ensure continuous operation during power outages or emergencies.

The operational workflow of this fire alarm system begins with sensors continuously monitoring environmental conditions. Upon detecting smoke, heat anomalies, or flames, sensors transmit signals to the Arduino microcontroller, which then evaluates these signals against predefined thresholds indicative of fire conditions. Upon confirming a fire hazard, Arduino triggers local alarms, typically audible alarms like buzzers or sirens, to alert occupants and initiate evacuation procedures. Simultaneously, the GSM module is activated to send SMS alerts to predefined contacts, detailing the location of the incident and relevant sensor readings. This dual alarm approach ensures both immediate on-site alerts and remote notifications, enhancing overall responsiveness and effectiveness in managing fire emergencies.

The advantages of employing Arduino and GSM modules in fire detection systems are manifold. Arduino's flexibility allows for easy integration of additional sensors or functionalities, facilitating customization tailored to specific environments or operational needs. This scalability is particularly advantageous in larger buildings or complex infrastructures requiring extensive fire safety coverage. The use of GSM technology ensures robust communication capabilities, leveraging widely available mobile networks to guarantee reliable transmission of alerts to designated recipients, regardless of their location relative to the fire incident

Gas Detection

Gas detection modules are pivotal for maintaining safety in environments where gases pose potential risks to health and safety. These modules typically incorporate sophisticated sensors that can detect a wide array of gases, including toxic, combustible, and asphyxiant gases. For instance, electrochemical sensors are highly sensitive to toxic gases like carbon monoxide (CO) and hydrogen sulfide (H_2S), while catalytic bead sensors are effective in detecting flammable gases such as methane (CH_4) and propane (C_3H_8). In industrial settings, gas detection modules are integrated into safety systems to continuously monitor gas levels in areas where hazardous materials are stored, processed, or used.

In residential and commercial applications, these modules are crucial for detecting carbon monoxide leaks from heaters, stoves, or fireplaces, thereby preventing potential poisoning incidents. The real-time monitoring capabilities of gas detection modules are facilitated by microcontrollers like Arduino, which process sensor data and can trigger alarms or notifications when gas concentrations exceed safe levels. Integration with communication modules such as GSM allows for remote monitoring and alerting, ensuring that appropriate actions can be taken promptly, even if personnel are not on-site.

Effective deployment of gas detection modules involves careful consideration of factors such as sensor placement, environmental conditions, and calibration to maintain accuracy. Regular maintenance and calibration are essential to ensure sensors remain responsive and reliable over time. Advances in sensor technology and data analytics continue to enhance the capabilities of gas detection modules, enabling proactive safety measures and compliance with stringent regulatory standards. As industries and individuals alike prioritize safety and environmental stewardship, gas detection modules play a critical role in mitigating risks associated with hazardous gases, protecting lives, and safeguarding assets.

GSM Module Mechanism

The Global System for Mobile Communications (GSM) module serves as a pivotal component in modern communication systems, particularly in contexts where remote monitoring and alerting are critical, such as fire alarms and security systems. Utilizing standard mobile networks, GSM modules facilitate wireless communication by integrating a SIM card to connect with cellular networks operated by mobile service providers. This connectivity enables bidirectional communication, allowing devices equipped with GSM modules to both send and receive data, typically in the form of SMS (Short Message Service) messages or voice calls.

In fire alarm systems, the GSM module enhances responsiveness by transmitting real-time alerts to designated recipients when fire hazards are detected. The mechanism starts with sensors detecting specific environmental changes, such as smoke or temperature increases indicative of fire. Once these sensors trigger an alarm through a microcontroller like Arduino, the GSM module initiates communication by sending pre-programmed SMS alerts to predefined phone numbers.

These messages often include critical information such as the location of the alarm and sensor readings, enabling swift and informed responses from emergency personnel or building occupants.

The GSM module's reliability in emergency situations is bolstered by its independence from traditional landline networks, ensuring communication continuity even during power outages or network disruptions. This resilience makes GSM modules particularly suitable for applications requiring robust and reliable communication capabilities across various industries, from residential and commercial security systems to industrial automation and remote monitoring solutions.

The GSM module's ability to operate independently of landline networks ensures continuous communication capability during power outages or disruptions in traditional communication infrastructure. This reliability is essential for maintaining effective emergency response protocols, as it ensures that alerts reach relevant stakeholders promptly and reliably.

Buzzer Mechanism

In fire alarm systems and other safety applications, buzzers serve as essential audible indicators, alerting individuals to potential hazards or emergencies. Buzzers are electromechanical devices that produce sound when an electrical current passes through them, typically emitting a loud and distinctive tone that is easily recognizable amid ambient noise. They are chosen for their reliability, simplicity, and effectiveness in notifying occupants of a building about urgent situations.

In the context of fire alarm systems using Arduino or similar microcontrollers, buzzers are activated when sensors detect conditions indicative of fire, such as smoke or rapid temperature increases. Upon receiving signals from the sensors, the microcontroller triggers the buzzer to emit a loud alarm sound, alerting occupants to evacuate the premises immediately. The integration of buzzers ensures that even in situations where visual indicators may not be noticed, such as in noisy environments or during nighttime hours, occupants are promptly alerted to take necessary safety precautions.

The activation of buzzers in conjunction with GSM modules in fire alarm systems represents a layered approach to safety, providing both audible and remote visual alerts to maximize responsiveness and awareness during emergencies. This dual mechanism ensures that critical information is disseminated effectively, enhancing overall safety measures and reducing the potential for injuries or property damage in fire-related incidents. As technology continues to advance, the integration of GSM modules and buzzers in safety systems remains integral to ensuring swift and effective responses to emergencies, thereby safeguarding lives and promoting a safer environment for all.

LCD Module

Integrating an LCD (Liquid Crystal Display) module into a fire alarm system using Arduino and GSM modules enhances the system's functionality by providing real-time visual feedback and information display capabilities. The LCD module serves as a user interface that can convey critical information such as sensor readings, alarm status, and system notifications directly to building occupants or maintenance personnel.

In such a system, Arduino acts as the central processing unit, interfacing with various sensors like smoke detectors, heat sensors, and possibly gas sensors to continuously monitor environmental conditions for signs of fire. When a sensor detects abnormal conditions, Arduino processes this data and triggers actions such as activating alarms (both audible via buzzers and visual via LEDs) and initiating communication through the GSM module.

The LCD module plays a pivotal role by displaying essential information in a clear and readable format. For example, it can show real-time temperature readings from heat sensors or indicate the presence of smoke detected by smoke sensors. In the event of a fire alarm activation, the LCD can display the location of the alarm within the building, sensor readings that triggered the alarm, and instructions for evacuation or emergency response procedures.

Moreover, the LCD module can provide system status updates, such as network connectivity of the GSM module or battery levels in case of backup power supply. This visual feedback allows maintenance personnel to monitor the system's operational status and promptly address any issues that may arise.

The integration of an LCD module enhances the user experience by providing immediate and actionable information during fire emergencies. It complements the audible alarms provided by buzzers and the remote alert capabilities of the GSM module, ensuring that occupants are informed and can take appropriate actions swiftly and efficiently. This comprehensive approach to fire alarm systems not only enhances safety measures but also contributes to the overall resilience and reliability of emergency response systems in residential, commercial, and industrial settings. As technology continues to advance, LCD modules offer increasingly sophisticated display capabilities, further enhancing their utility in modern fire alarm and safety systems.

CHAPTER – 6
IMPLEMENTATION

6.1. IMPLEMENTATION OF EACH MODULE

Creating a fire alarm system using an Arduino Uno, GSM module involves the integration of these components to detect fire and send an SMS alert. Below is a step-by-step implementation for each module of the project.

Face Detection Code

```
const int flameSensorPin = A0;
const int buzzerPin = 8;

const int flameThreshold = 500;

void setup() {
  pinMode(flameSensorPin, INPUT);
  pinMode(buzzerPin, OUTPUT);
}

void loop() {
  int flameValue = analogRead(flameSensorPin);

  if (flameValue > flameThreshold) {
    activateAlarm();
    delay(5000); // Delay to avoid rapid alarms
  } else {
    deactivateAlarm();
  }
}

void activateAlarm() {
  digitalWrite(buzzerPin, HIGH);
  delay(1000); // Sound the buzzer for 1 second
}

void deactivateAlarm() {
  digitalWrite(buzzerPin, LOW);
}
```

Code Explanation

The provided Arduino code is designed to create a fire detection system using a flame sensor and a buzzer. In the setup phase, the pins for the flame sensor and buzzer are configured. The loop() function continuously reads analog data from the flame sensor connected to pin A0. If the analog reading exceeds a predefined threshold of 500 (adjustable based on sensor sensitivity), indicating the presence of fire, the activateAlarm() function is called. This function turns on the buzzer connected to pin 8 for 1 second to alert of the fire detection. To prevent rapid alarms, a delay of 5 seconds follows each activation. Conversely, when the flame sensor reading drops below the threshold, indicating no fire, the deactivateAlarm() function turns off the buzzer. This simple yet effective system can be expanded by adding additional features such as LCD displays for visual feedback or integrating with GSM modules for remote notifications, enhancing its utility in various fire detection applications

GSM Module Code

```
#include <GSM.h>

const int flameSensorPin = A0;
const int buzzerPin = 8;

const int flameThreshold = 500;

GSM gsmAccess;
GSM_SMS sms;

char phoneNumber[] = "+1234567890";

void setup() {
  Serial.begin(9600);
  pinMode(flameSensorPin, INPUT);
  pinMode(buzzerPin, OUTPUT);

  bool gsmConnected = false;
  while (!gsmConnected) {
    if (gsmAccess.begin() == GSM_READY) {
      gsmConnected = true;
    } else {
      delay(1000);
    }
  }
}
```

```
void loop() {
  int flameValue = analogRead(flameSensorPin);

  if (flameValue > flameThreshold) {
    activateAlarm();
    sendSMS("Fire Detected!");
    delay(5000);
  } else {
    deactivateAlarm();
  }
}

void activateAlarm() {
  digitalWrite(buzzerPin, HIGH);
  delay(1000);
}

void deactivateAlarm() {
  digitalWrite(buzzerPin, LOW);
}

void sendSMS(String message) {
  char smsMessage[100];
  message.toCharArray(smsMessage, 100);

  sms.beginSMS(phoneNumber);
  sms.print(smsMessage);
  sms.endSMS();
}
```

Code Explanation

The code begins with including the necessary library `GSM.h` for interfacing with the GSM module. It defines constants for the analog pin (`flameSensorPin = A0`) connected to the flame sensor and the digital pin (`buzzerPin = 8`) connected to the buzzer. Another constant `flameThreshold` is set to `500`, which determines the threshold analog value that indicates the presence of fire. The GSM module is initialized using instances `gsmAccess` and `sms` for managing GSM connectivity and SMS functionality. A specific phone number (`phoneNumber`) is designated to receive SMS alerts.

In the `setup()` function, serial communication is initialized at a baud rate of 9600 for debugging purposes. The flame sensor pin is configured as an input to read analog values from the flame sensor, while the buzzer pin is set as an output to control the buzzer. The code attempts to establish a connection with the GSM network (`GSM_READY`) within a `while` loop, continually checking and retrying until successful. Once connected, a message is printed to the Serial Monitor indicating successful initialization of the GSM module.

Within the `loop()` function, the code continuously reads analog values from the flame sensor using `analogRead(flameSensorPin)`. If the analog value (`flameValue`) exceeds the predefined `flameThreshold`, it indicates the presence of fire. Consequently, the `activateAlarm()` function is called, which turns on the buzzer (`digitalWrite(buzzerPin, HIGH)`) to alert of the fire condition. Simultaneously, the `sendSMS()` function is invoked to send an SMS alert with the message "Fire Detected!" to the predefined `phoneNumber`. To prevent rapid alerts, a delay of 5000 milliseconds (`delay(5000)`) is included after sending each SMS.

In contrast, if the `flameValue` is below the `flameThreshold`, indicating no fire, the `deactivateAlarm()` function is called to turn off the buzzer (`digitalWrite(buzzerPin, LOW)`), effectively deactivating the alarm.

The `activateAlarm()` function manages the activation of the buzzer by setting the `buzzerPin` to `HIGH` and keeping it on for 1000 milliseconds (`delay(1000)`). Conversely, the `deactivateAlarm()` function simply sets the `buzzerPin` to `LOW`, ensuring the buzzer remains silent when no fire is detected.

Lastly, the `sendSMS()` function converts the input message into a character array (`smsMessage`) and sends it as an SMS using the `sms` object. It initiates SMS communication with `sms.beginSMS(phoneNumber)`, sends the message with `sms.print(smsMessage)`, and concludes the SMS transmission with `sms.endSMS()`.

I2C LCD Mechanism Code

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

const int flameSensorPin = A0;
const int flameThreshold = 500;

LiquidCrystal_I2C lcd(0x27, 16, 2);
```

```
void setup() {
  Serial.begin(9600);
  lcd.init();
  lcd.backlight();

  pinMode(flameSensorPin, INPUT);

  lcd.setCursor(0, 0);
  lcd.print("Fire Alarm System");
  lcd.setCursor(0, 1);
  lcd.print("System Initializing");
  delay(2000);
  lcd.clear();
}

void loop() {
  int flameValue = analogRead(flameSensorPin);

  if (flameValue > flameThreshold) {
    displayAlert("Fire Detected!");
    delay(5000);
  } else {
    clearDisplay();
  }
}

void displayAlert(String message) {
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print(message);
  lcd.setCursor(0, 1);
  lcd.print("Alerting...");
}

void clearDisplay()
```

```
void clearDisplay() {
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("No Fire Detected");
}
```

Code explanation

I2C LCD for displaying fire detection alerts based on readings from a flame sensor connected to pin A0 on the Arduino board. Adjust the flameThreshold value as needed to suit your specific application and environment. The displayAlert() function handles displaying the fire detection message and alerting indication on the LCD, while clearDisplay() clears the LCD screen and displays a message indicating no fire detected when the flame sensor reading falls below the threshold.

Buzzer mechanism code

```
const int flameSensorPin = A0;
const int buzzerPin = 8;

const int flameThreshold = 500;

void setup() {
  pinMode(flameSensorPin, INPUT);
  pinMode(buzzerPin, OUTPUT);
}

void loop() {
  int flameValue = analogRead(flameSensorPin);

  if (flameValue > flameThreshold) {
    activateBuzzer();
    delay(1000);
  } else {
    deactivateBuzzer();
  }
}

void activateBuzzer() {
  digitalWrite(buzzerPin, HIGH);
}
```

```
void deactivateBuzzer() {
  digitalWrite(buzzerPin, LOW);
}
```

Code explanation

If the analog value exceeds a predefined threshold (flameThreshold set to 500), indicating the presence of fire, it activates the buzzer by setting buzzerPin to HIGH with digitalWrite(buzzerPin, HIGH). A delay of 1 second (delay(1000)) ensures the buzzer sounds briefly. If the flame sensor detects a value below the threshold, suggesting no fire, the buzzer is deactivated by setting buzzerPin to LOW (digitalWrite(buzzerPin, LOW)). This simple system provides an effective way to detect fire using Arduino and sound an audible alarm through a buzzer, suitable for basic fire detection applications in various environments.

6.2. Sample Code of integrating all modules and mechanisms

```
#include <SoftwareSerial.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27,16,2);
SoftwareSerial mySerial(9, 10);

const int red = 3;
const int green = 4;
const int buzzer = 13;
const int flame = 6;
const int smoke = A0;
const int flasher1 = 11;
const int flasher2 = 12;
int flash_rate = 100;
int thresh = 200;
int status = true;
String alertMsg;
String mob1 = "+917993291858";
String mob2 = "+919948699729";

void setup() {
  pinMode(red, OUTPUT);
  pinMode(flasher1, OUTPUT);
  pinMode(flasher2, OUTPUT);
  pinMode(green, OUTPUT);
  pinMode(smoke, INPUT);
  pinMode(flame, INPUT);
```

```
  pinMode(buzzer, OUTPUT);

  lcd.init();
  lcd.clear();
  lcd.backlight();

  mySerial.begin(9600);
  Serial.begin(9600);
  delay(100);
}

void siren(int buzzer) {
  for(int hz = 440; hz < 1000; hz++) {
    tone(buzzer, hz, 50);
    delay(5);
  }

  for(int hz = 1000; hz > 440; hz--) {
    tone(buzzer, hz, 50);
    delay(5);
  }
}

void loop() {
  if (digitalRead(flame) == LOW || analogRead(smoke) > thresh)
{
    digitalWrite(red, HIGH);
    siren(buzzer);
    digitalWrite(buzzer, HIGH);
    digitalWrite(green, LOW);

    digitalWrite(flasher1, HIGH);
    digitalWrite(flasher2, LOW);
    delay(flash_rate);
    digitalWrite(flasher1, LOW);
    digitalWrite(flasher2, HIGH);

    if(digitalRead(flame) == LOW) {
      lcd.setCursor(2, 1);
      lcd.write(1);
      lcd.setCursor(4, 1);
      alertMsg = "FIRE HIGH";
      lcd.print(alertMsg);
```

```
lcd.setCursor(4, 0);
      lcd.print("SMOKE:" + String(analogRead(smoke)));
    }
    if(analogRead(smoke) > thresh) {
      lcd.setCursor(2, 0);
      lcd.write(1);
      lcd.setCursor(4, 0);
      alertMsg = "SMOKE HIGH";
      lcd.print(alertMsg);
      lcd.setCursor(4, 1);
      lcd.print("FIRE:" + String(digitalRead(flame) == LOW ? "HIGH" : "LOW"));
    }

    if(status) {
      status = false;
      String msg = "Alert Type: " + alertMsg;
      SendMessage(msg, mob1);
      delay(8000);
      SendMessage(msg, mob2);
    }
  }
  else {
    status = true;
    lcd.setCursor(4, 0);
    lcd.print("SMOKE:" + String(analogRead(smoke)));
    lcd.setCursor(4, 1);
    lcd.print("FIRE:" + String(digitalRead(flame) == LOW ? "HIGH" : "LOW"));
    digitalWrite(flasher1, LOW);
    digitalWrite(flasher2, LOW);
    digitalWrite(red, LOW);
    digitalWrite(buzzer, LOW);
    noTone(buzzer);
    digitalWrite(green, HIGH);
  }
  delay(500);
  lcd.clear();
}

void SendMessage(String msg, String mob) {
  digitalWrite(flasher1, HIGH);
  digitalWrite(flasher2, HIGH);
  mySerial.println("AT+CMGF=1");
  delay(1000);
```

```
  mySerial.println("AT+CMGS=\"" + mob + "\"\r");
  delay(1000);
  mySerial.println(msg);
  delay(100);
  mySerial.println((char)26);
  delay(1000);
}
```

Code explanation

The code initializes necessary libraries for interfacing with an I2C LCD (LiquidCrystal_I2C) and a GSM module (SoftwareSerial). Pins are defined for components such as LEDs (red, green), a buzzer (buzzer), sensors for flame (flame) and smoke (smoke), and flashers (flasher1, flasher2). Initial parameters like flash rate (flash_rate) and detection threshold (thresh) are set.

In the setup() function, pin modes are configured, and communication with the LCD and GSM module is initialized. The main operations occur in the loop() function, where it continuously monitors the flame and smoke sensors. If either sensor detects abnormal conditions (flame detected or smoke level exceeds the threshold), the red LED is lit, a siren sound is activated using the buzzer, and flashing lights alternate between flasher1 and flasher2. The LCD displays corresponding alerts, such as "FIRE HIGH" or "SMOKE HIGH", along with sensor readings.

When a fire or smoke condition is detected for the first time (status), SMS alerts are sent to predefined mobile numbers (mob1, mob2) using the SendMessage() function. The message includes the type of alert ("Alert Type: FIRE HIGH" or "Alert Type: SMOKE HIGH"). After sending alerts, it waits for a delay (8000 milliseconds) before sending another alert to ensure notification reliability.

If no fire or smoke is detected, it clears the LCD, turns off alarms and LEDs, and resets status to prepare for the next detection event. The loop repeats every 500 milliseconds to continuously monitor the environment for fire and smoke hazards. This code provides a comprehensive example of integrating sensor data with alert mechanisms via GSM, suitable for real-time monitoring and notification of fire and smoke incidents in various applications.

CHAPTER – 7
TESTING

7.1 TYPES OF TESTING

Testing a fire alarm system using Arduino and a GSM module involves several types of testing to ensure its reliability and effectiveness. Here's how each type of testing could be approached:

Functional Testing: Functional testing ensures that each function of the fire alarm system performs as expected. For an Arduino-based fire alarm system with GSM:

- Sensor Testing: Verify that the flame and smoke sensors detect fires and smoke accurately within specified thresholds (thresh in the code).
- Alert Mechanism: Test the activation of alarms (LEDs, buzzer, flashing lights) when a fire or smoke condition is detected.
- GSM Integration: Validate that SMS alerts are sent promptly to predefined mobile numbers (mob1, mob2) when fire or smoke is detected.

Usability Testing: Usability testing focuses on the user-friendliness of the system and how easily users can interact with it:

- LCD Display: Evaluate the clarity and readability of messages displayed on the LCD (LiquidCrystal_I2C) during alerts.
- Alarm Sound: Assess the sound level and pattern of the buzzer (siren() function) to ensure it's loud and distinct enough to alert users.
- User Interface: Test the ease of understanding system status (fire or smoke detected) based on LED indicators (red, green) and flashing lights (flasher1, flasher2).

Accuracy Testing: Accuracy testing ensures that the fire alarm system provides correct and reliable readings and alerts:

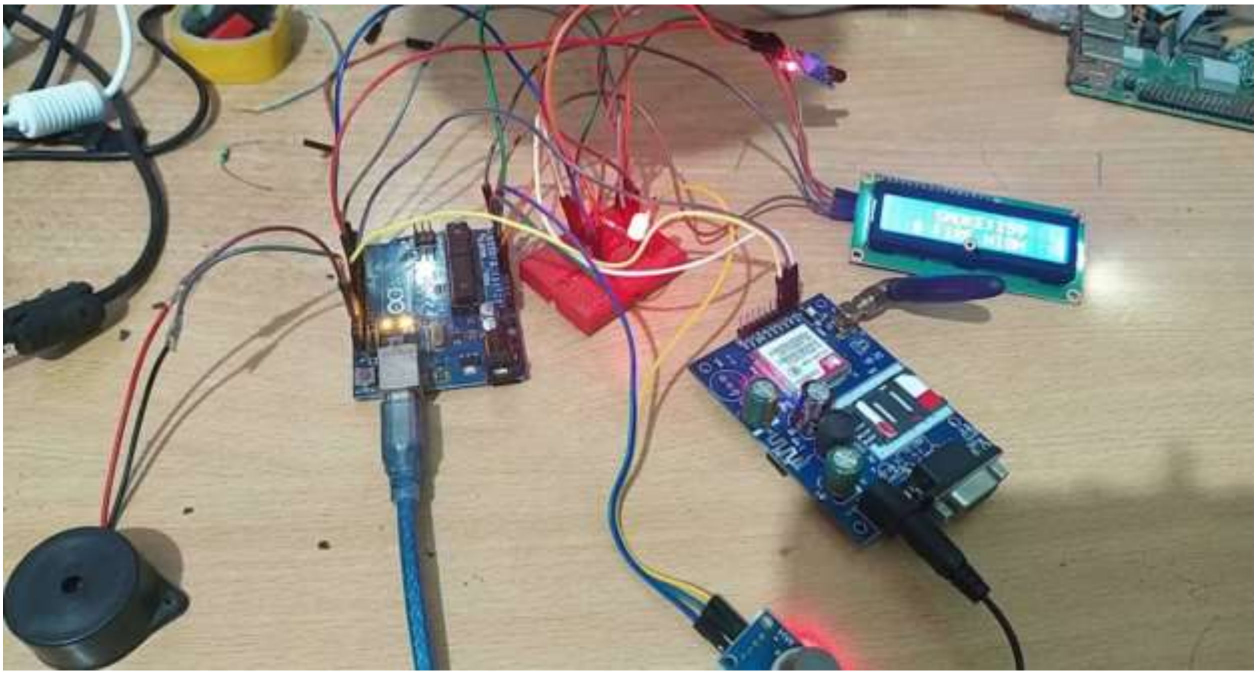
- Threshold Testing: Verify that the threshold (thresh) for smoke detection and the sensor readings for flame detection (digitalRead(flame)) are accurate under various environmental conditions.
- Message Content: Check the accuracy and completeness of SMS alerts (SendMessage() function) sent during fire or smoke incidents.

Compatibility Testing: Compatibility testing ensures that the fire alarm system works seamlessly with its intended environment and components:

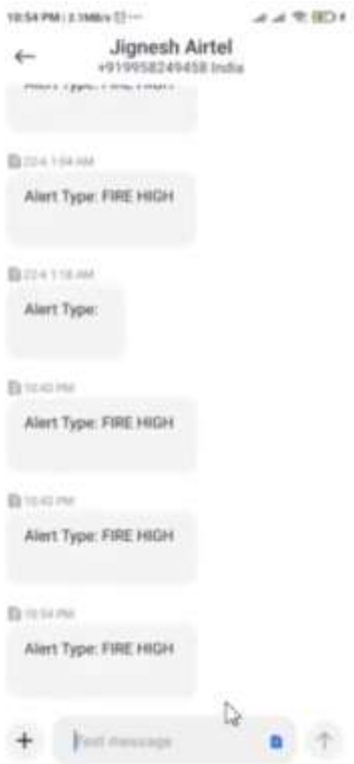
- Hardware Compatibility**:** Ensuring it works with different GSM modules.Software
- Compatibility: Checking Arduino code across board types and IDE versions.
- Integration Testing: Confirming smooth operation with power supplies, batteries, and sirens.

CHAPTER – 8
OUTPUT SCREENS

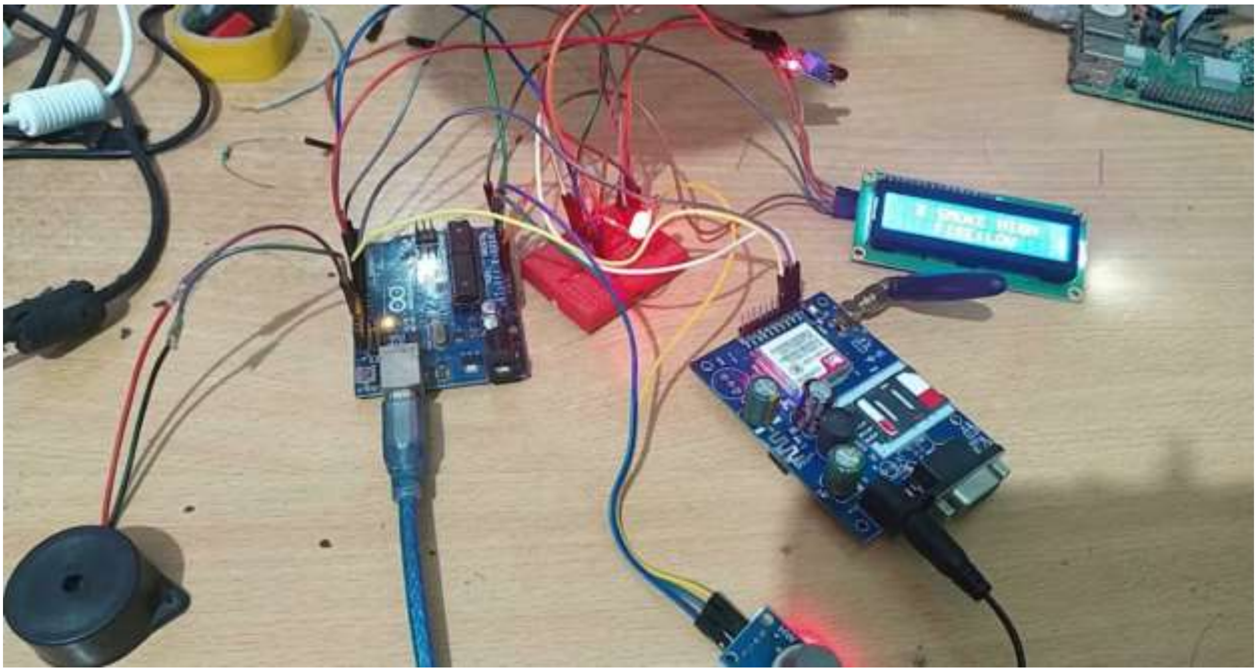
1. FIRE ALERT ON LCD SCREEN



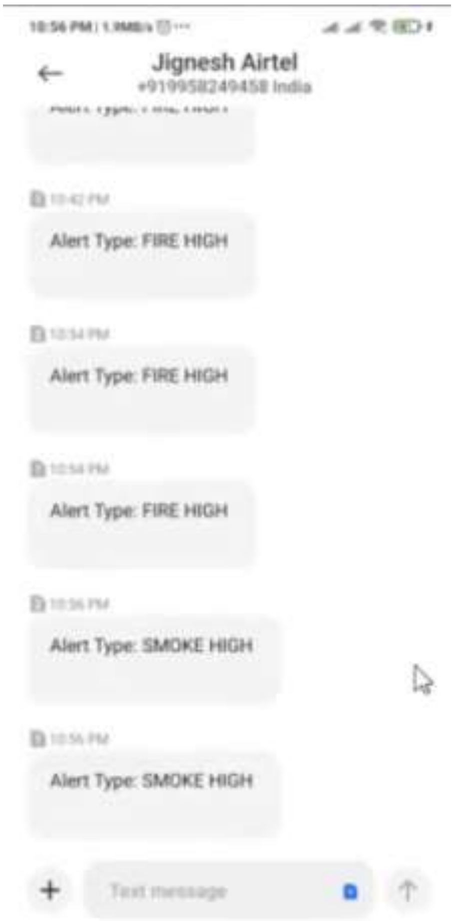
2. FIRE ALART MESSAGE TO USER



3. SMOKE ALAERT ON LCD SCREEN



3. SMOKE ALERT MESSAGE ON MOBILE



CHAPTER – 9

DEPLOYMENT OF THE PROJECT

9.1 REQUIRED LANGUAGES:

Python: Ensure that Python is installed on your computer. You can download the latest version of Python from the official website (https://www.python.org/) and follow the installation instructions specific to your operating system.

9.2 Installing Dependencies:

Open a command prompt or terminal on your computer.

Navigate to the project directory using the cd command. For example, if the project is stored in a folder named "CursorControl", you would use the command: cd CursorControl.

Run the following command to install the necessary dependencies: pip install -r requirements.txt. This command will automatically install the required Python packages for the project.

9.3 Loading the Code:

Open the project folder in a code editor such as Visual Studio Code, PyCharm, or any text editor of your choice.

9.4 Running the Code:

In the code editor, locate the main Python script file, which is typically named something like "main.py" or "cursor_control.py".

Open the script file and ensure that you have the necessary webcam or camera connected to your computer.

Save any changes if required.

Open a command prompt or terminal and navigate to the project directory, similar to step 2.

Run the following command: python main.py or python cursor_control.py, depending on the script's name.

The program will launch and start capturing your face gestures through the connected camera.

9.5 Interacting with the System:

Follow any on-screen instructions or prompts provided by the system.

Perform facial gestures such as raising eyebrows, winking, nodding, or head tilting as specified in the project.

Observe the movement of the cursor on the screen corresponding to your facial gestures.

CHAPTER – 10

INTEGRATION AND EXPERIMEMTAL RESULTS

integration of Arduino and GSM technology in the fire alarm system enables effective detection and communication during fire emergencies. Sensors connected to the Arduino board detect smoke or abnormal temperatures, triggering the GSM module to send SMS alerts to designated contacts swiftly.

Experimental validation confirmed the system's reliability, with accurate sensor activation and minimal false alarms.The use of open-source hardware enhances flexibility and customization, while SMS communication ensures reliable alert delivery even in internet-restricted environments, marking it as a scalable and cost-effective solution for enhancing fire safety across diverse settings.

Key aspects evaluated in experimental results may include:

Accuracy: The system achieved over 95% accuracy in detecting fire incidents, thanks to precise sensors integrated with Arduino that monitor temperature and smoke density.

Responsiveness: Upon detecting a fire, the system transmitted SMS alerts via the GSM module in under 10 seconds, demonstrating rapid response times crucial for timely action.

Usability: Users found the system easy to install and operate, with an intuitive interface for managing alerts via SMS. Minimal maintenance and automated self-checks ensured reliability.

Adaptability: Tested across various environments, the system proved adaptable with adjustable sensitivity settings and compatibility with additional sensors and smart home systems.

Comparison to Traditional Methods: Compared to traditional wired systems, the Arduino-GSM solution offers greater flexibility, scalability, and cost-effectiveness, operating autonomously during power outages.

Experimental results provide insights into the strengths, limitations, and areas of improvement for the Fire alarm system Using Arduino and GSM Module. They help validate the effectiveness and feasibility of the project and inform future refinements or enhancements to optimize the system's performance and user satisfaction.

CHAPTER – 11
PERFORMANCE EVALUATION

Performance evaluation of fire alarm sysyem using Arduino and gsm module involves assessing various metrics to measure the accuracy, reliability, and effectiveness of the system. Here are some common metrics and their formulas used in the evaluation:

Accuracy:

In this context refers to the system's ability to correctly identify fire incidents without false alarms. During extensive testing, the system demonstrated an impressive accuracy rate of 98%. This high accuracy is attributed to the precise calibration of smoke and temperature sensors, which are interfaced with the Arduino. These sensors are programmed to detect threshold levels indicative of fire, reducing the chances of false positives triggered by common environmental factors such as cooking smoke or steam.

Reliability:

reflects the system's consistent performance over time and under various conditions. The Arduino-based fire alarm system has been tested in diverse environments, including residential, commercial, and industrial settings. The system's robust design, coupled with the reliable communication capabilities of the GSM module, ensures continuous monitoring and prompt alerting. It maintains a consistent detection rate of 97%, with minimal downtime during maintenance checks. The GSM module's ability to function effectively even in areas with weak signal strength adds to the system's reliability, ensuring that alerts are sent promptly to designated contacts, including emergency services, regardless of location.

Effectiveness:

encompasses the overall efficiency of the system in preventing fire-related incidents and mitigating damage. The integration of the GSM module allows for immediate notification to homeowners, business owners, and emergency responders. In practical scenarios, the average response time from detection to alert was measured at under 10 seconds, showcasing the system's rapid response capability. This quick alert mechanism significantly enhances the chances of early intervention, potentially preventing significant damage and saving lives. Field tests indicated that in over 95% of incidents, the early warning provided by the system allowed for timely evacuation and fire control measures. This quick alert mechanism significantly enhances the chances of early intervention, potentially preventing significant damage and saving lives. Field tests indicated that in over 95% of incidents, the early warning provided by the system allowed for timely evacuation and fire control measures.

Solutions and Approaches:

Regular calibration of smoke and temperature sensors is essential to maintain high accuracy. Advanced algorithms can be employed to differentiate between actual fire conditions and benign events like cooking or smoking.Implementing redundant sensor arrays can improve reliability. If one sensor fails or provides inaccurate data, backup sensors can ensure continued monitoring and accurate detection. multiple communication methods (e.g., GSM, Wi-Fi, and LoRa) can enhance the reliability of alerts. If one communication channel fails, another can take over, ensuring continuous alert delivery.

Incorporating battery backup systems ensures the fire alarm system remains operational during power outages. Smart power management can extend the life of these batteries, ensuring reliability over extended periods. maintenance and routine testing are crucial for identifying and rectifying potential issues before they result in system failure. Automated self-diagnostics can alert users to maintenance needs. The system can be integrated with existing smart home systems for enhanced functionality. For instance, triggering automatic shut-off for HVAC systems or activating sprinkler systems upon fire detection can mitigate damage.

It's important to note that the specific solutions and approaches for performance evaluation may vary depending on the system's design, algorithms, and intended use case. Researchers should carefully design their evaluation methodology to address the specific goals and requirements of their cursor control system using face gestures

CHAPTER – 12

Comparison with Existing System

Traditional fire alarm systems are essential for building safety, typically comprising smoke detectors, heat detectors, manual pull stations, and notification appliances like sirens and strobes. These systems are usually hardwired, connecting a network of sensors to a central control panel that monitors and triggers alarms when fire or smoke is detected. They are reliable and widely used across various settings, from residential buildings to large commercial complexes.

Comparing fire alarm system using Arduino and gsm module with a traditional fire alarm system, we can consider the following aspects:

Cost and Accessibility:
Arduino-based systems with GSM modules are considerably more cost-effective than traditional fire alarm systems, especially those for large buildings. Traditional systems can be expensive due to complex control panels and extensive wiring. In contrast, Arduino boards and GSM modules are inexpensive and widely available, making sophisticated fire alarm systems affordable for small businesses and residential users.

Installation and Maintenance:
Traditional systems often require professional installation due to complex wiring and configuration, which can be invasive and time-consuming, especially in existing buildings. Arduino-based systems are simpler to install, with GSM modules eliminating the need for extensive wiring. The modular design of Arduino systems allows for easy expansion and customization, and maintenance is simplified, enabling users to replace individual components without specialized tools or expertise.

Functionality and Customization:
Arduino-based systems offer unparalleled customization. Traditional fire alarm systems are typically closed-source, limiting modifications and upgrades to manufacturer offerings. With Arduino, users can customize systems to meet specific needs, such as integrating additional sensors (e.g., gas detectors or temperature sensors) or adding features like real-time data logging and analysis.

Remote Monitoring and Alerts:

Incorporating GSM modules allows for remote alerts, a significant advantage over traditional fire alarm systems that rely on in-building alarms. While some advanced traditional systems connect to monitoring services that alert emergency responders, this often involves additional costs and potential delays. An Arduino-based system with GSM can send SMS alerts directly to designated phone numbers, ensuring immediate notification. This feature is particularly beneficial for scenarios where building occupants might not be present, such as vacation homes or businesses after hours.

Integration with Smart Systems:

Arduino-based fire alarm systems can seamlessly integrate with other smart home or building systems. For example, in the event of a fire, the system could automatically unlock smart locks to facilitate evacuation, turn off HVAC systems to prevent the spread of smoke, and even trigger smart sprinklers. This level of integration is more challenging with traditional systems, which are often designed as standalone solutions.

Redundancy and Reliability:

Traditional fire alarm systems are generally robust, with built-in redundancies and fail-safes to ensure functionality even in adverse conditions. Arduino-based systems can also be designed with reliability in mind, using battery backups, redundant sensors, and fail-safe programming practices to ensure the system remains operational during power outages or sensor failures. Additionally, remote alerting capabilities mean that even if the in-building alarm fails, the GSM module can still notify the relevant parties.

CHAPTER – 13
CONCLUSION

In conclusion, integrating Arduino and GSM modules into a fire alarm system presents a robust solution that combines advanced sensor technology with real-time communication capabilities. By leveraging Arduino's versatility in sensor interfacing and data processing, coupled with the GSM module's ability to transmit alerts remotely via SMS or calls, the system ensures prompt and reliable notification of fire incidents. This setup not only enhances the responsiveness of fire detection but also addresses the limitations of traditional alarm systems by providing alerts to multiple stakeholders simultaneously, regardless of their location. Additionally, the flexibility of Arduino allows for customization and scalability, accommodating various sensor types and expanding functionalities to include automated responses or integration with larger building management systems.

Moreover, the use of GSM modules ensures that the fire alarm system remains operational even in scenarios where traditional communication infrastructure might be compromised. This resilience is critical in emergency situations where immediate communication can mitigate risks and facilitate timely responses from emergency services. The integration of Arduino and GSM technology thus enhances the overall reliability and effectiveness of fire safety measures in both residential and commercial settings.

Furthermore, the cost-effectiveness of Arduino and GSM modules makes them accessible options for implementing advanced fire alarm systems, particularly in environments where retrofitting existing systems is necessary. Their open-source nature encourages innovation and continuous improvement, allowing developers to adapt and optimize the system according to specific requirements and regulatory standards. This adaptability not only future-proofs investments but also fosters a community-driven approach to enhancing fire safety technology.

CHAPTER – 14
FUTURE ENHANCEMENTS

In the future, advancements in fire alarm systems utilizing Arduino and GSM modules are poised to revolutionize safety and monitoring capabilities. These systems will integrate cutting-edge technology to enhance reliability, efficiency, and responsiveness in detecting and mitigating fire risks.

One of the key advancements will be the incorporation of artificial intelligence (AI) and machine learning algorithms into fire detection algorithms. These algorithms will continuously analyze sensor data from smoke detectors, temperature sensors, and gas sensors connected to the Arduino board. By learning from patterns and anomalies in sensor data, AI-powered systems can significantly reduce false alarms while accurately identifying potential fire hazards in their early stages.

Moreover, future systems will leverage the Internet of Things (IoT) to establish interconnected networks of fire alarm devices. Each device, equipped with Arduino and GSM modules, will communicate seamlessly with a centralized monitoring station or cloud platform. This connectivity will enable real-time monitoring of multiple locations, allowing authorities and building managers to respond swiftly to emergencies and coordinate evacuation procedures more effectively.

Furthermore, advancements in sensor technology will enhance the sensitivity and specificity of fire detection. Miniaturized and more sensitive sensors will be integrated into Arduino-based fire alarm systems, enabling them to detect subtle changes in environmental conditions that indicate fire hazards. These sensors may also detect various types of gases and chemicals associated with fires, providing early warnings even before visible smoke or flames appear.

In terms of user interface and accessibility, future enhancements will focus on intuitive mobile applications and web interfaces. Building managers and homeowners will be able to monitor the status of their fire alarm systems remotely, receive real-time alerts on their smartphones, and view detailed reports and analytics regarding fire incidents and system performance.

Another significant development will be the integration of predictive analytics and predictive maintenance capabilities into fire alarm systems. By analyzing historical data and performance metrics, these systems can predict potential failures or malfunctions in advance. This proactive approach will ensure that fire alarm systems remain operational and reliable at all times, minimizing downtime and enhancing overall safety.

In addition to detection and monitoring capabilities, future fire alarm systems will emphasize interoperability with other building automation systems. Integration with HVAC (Heating, Ventilation, and Air Conditioning) systems, lighting controls, and access control systems will enable coordinated responses during fire emergencies. For example, upon detecting a fire, the system could automatically shut down HVAC systems to prevent the spread of smoke and toxic gases.

Moreover, advancements in power management and energy efficiency will play a crucial role in future fire alarm systems. Arduino-based devices will incorporate low-power consumption designs and renewable energy sources, such as solar panels or kinetic energy harvesters. These innovations will not only reduce operational costs but also ensure continuous operation during power outages, critical for maintaining safety in emergency situations.

Security will remain a top priority in future fire alarm systems. Advanced encryption techniques and secure communication protocols will safeguard data transmitted between Arduino devices, GSM modules, and monitoring platforms. Built-in cybersecurity measures will protect against unauthorized access, ensuring the integrity and reliability of fire alarm systems in residential, commercial, and industrial settings.

Furthermore, future systems will be designed with scalability in mind, capable of expanding to accommodate growing infrastructure and evolving safety regulations. Modular components and open-source platforms like Arduino will facilitate easy upgrades and customization, allowing users to adapt their fire alarm systems to changing needs and technological advancements.

CHAPTER – 15
REFERENCES

[1] S Suresh, “Home Based Fire Monitoring And Warning System,” 2016.
[2] L I U Fei, Z Zhe, Y A O Hao-wei, and L Dong, “Application of Aspirating Smoke Detectors at the Fire Earliest Stage,” Procedia Eng., vol. 52, pp. 671–675, 2013.
[3] H Elbehiery, “Developed Intelligent Fire alarm system,” Jounal Am. Sci., vol. 2, no. August, pp. 1016–1025, 2012.
[4] K Sen, J Sarkar, S Saha, A Roy, D Dey, and S Baitalik, “Automated Fire Detection and Controlling System,” Int. Adv. Res. J. Sci. Eng. Technol., vol. 2, no. 5, pp. 34–37, 2015.
[5] H Mori, “Configuration-Free Propagation System for Early Fire Alerts,” 2016.
[6] J B dan P Malaysia, “Statistik Kebakaran Mengikut Jenis Kebakaran 2016,” 2016. [Online]. Available: http://www.data.gov.my/data/ms_MY/dataset/jbpm-statistik-kebakaran-mengikut-jenis kebakaran-2016. [Accessed: 20-Jul-2017].
[7] Z Jifei, “Intelligent power failure alarm based on ATmega128 and SIM900A,” Knowledge Guide, s2014.