Chapter 1

INTRODUCTION

Fire accidents are one of the major causes of loss of life and property in residential, industrial, and commercial areas. Many fire incidents become severe due to late detection and lack of immediate communication to concerned authorities. Traditional fire alarm systems are often limited to local alerts such as sirens or buzzers, which may go unnoticed when no one is present at the location.

With the advancement of embedded systems and wireless communication technologies, smart fire alert systems have become more efficient and reliable. The Fire Alert System using GSM–Arduino module is designed to detect fire-related hazards such as flame, smoke, or gas leakage and immediately send alert notifications through SMS and call using a GSM module. This ensures real-time monitoring and quick response even when the user is far from the affected area.

The system uses Arduino Uno as the main controller, which continuously monitors data from sensors like flame and smoke sensors. When abnormal conditions are detected, the Arduino triggers a buzzer for local alert and activates the GSM module to send warning messages to predefined mobile numbers. This makes the system suitable for homes, offices, industries, warehouses, and remote locations.

2. Brief History of Fire Alert Systems

The development of fire alert systems has evolved significantly over time to improve safety and reduce damage:

Early Fire Detection Systems

In earlier days, fire detection relied mainly on human observation, watch towers, and manual alarms. These methods were slow and highly dependent on human presence and attention.

Conventional Fire Alarm Systems

Later, manual call points and conventional fire alarms were introduced, which used heat or smoke detectors connected to audible alarms. Although effective, these systems could only provide local alerts and required people to be nearby to take action.

Automated Electronic Fire Systems

With the introduction of electronics, automatic fire detection systems using sensors became common. These systems could detect fire conditions without human intervention but still lacked remote communication capabilities.

Microcontroller-Based Systems

The development of microcontrollers like Arduino brought flexibility, low cost, and easy programming to fire detection systems. Sensors could be interfaced easily, and system behavior could be customized based on requirements.

GSM-Based Fire Alert Systems

The integration of GSM technology marked a major advancement. GSM-based fire alert systems can send SMS and voice calls to users, fire stations, or authorities, enabling quick response. These systems are reliable, work over long distances, and do not depend on internet availability.

Chapter 2

Problem Statement

Fire accidents in residential buildings, industries, and public places cause severe loss of life and property due to delayed detection and lack of timely alerts. Traditional fire detection systems often depend on manual monitoring or local alarms, which may go unnoticed when no one is present at the location.

There is a need for an automated fire alert system that can continuously monitor environmental conditions, detect fire-related hazards such as flame and smoke, and immediately notify responsible persons remotely. This project aims to develop a low-cost, reliable, and automated fire alert system using Arduino and GSM technology to provide early warning and reduce damage.

2. Objectives of the Project

1. To design and develop an automatic fire detection system using sensors.
2. To detect fire, smoke, and gas leakage at an early stage.
3. To generate local alerts using buzzer and LED indications.
4. To send SMS and call alerts to the user through a GSM module.
5. To minimize human intervention in fire detection.
6. To provide a cost-effective and easy-to-install safety solution.
7. To improve response time during emergency situation

3. Proposed Solution

The proposed system uses an Arduino UNO as the main controller along with flame sensor, smoke/gas sensor, GSM module, buzzer, and LED.

The sensors continuously monitor the surrounding environment.When smoke, gas, or flame is detected beyond a predefined threshold, the Arduino processes the data.

The system activates a buzzer and LED to provide an immediate local warning.

Simultaneously, the GSM module sends an SMS alert and makes a call to a pre-registered mobile number.

This ensures that the user is informed even if they are far from the affected location.

The system operates automatically and continuously, ensuring real-time monitoring.

4. Advantages of the Proposed Solution

1. Early Fire Detection – Detects fire hazards at an initial stage.
2. Remote Alert System – Sends SMS and call alerts to users instantly.
3. Low Cost – Uses affordable and easily available components.
4. Real-Time Monitoring – Continuously monitors environmental conditions.
5. Easy Installation – Simple hardware setup and coding.
6. Reduced Human Dependency – Works automatically without manual supervision.
7. Scalable Design – Can be extended with more sensors or IoT features.

Chapter 3

3.1 Design Thinking Process

1. Empathize

Understanding the problem from the user's perspective.

Fire accidents can occur unexpectedly in homes, offices, industries, and laboratories.

Many fire incidents go unnoticed in early stages, especially when people are not present.

Manual fire monitoring systems are unreliable and slow in emergency situations.

Users need an automatic, fast, and remote alert system.

A reliable system that can detect fire early and immediately notify users even when they are away.

2.Problem Definition:

To design a low-cost, real-time fire alert system that can detect fire-related parameters and notify responsible persons through GSM-based communication.

Key Challenges Identified:Early detection of fire and smoke.

Reliable alert delivery without internet dependency.

Low power consumption and cost efficiency.

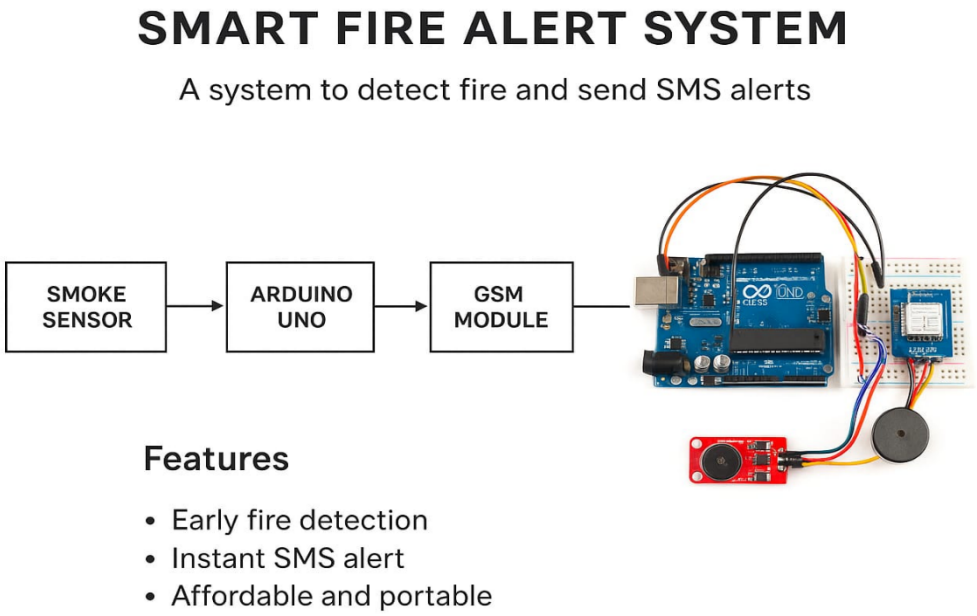
Easy installation and maintenance.

3. Ideate

Generating possible solutions to the defined problem.

3.2 Methodology

The fire alert system continuously monitors environmental conditions using smoke and flame sensors. When a fire-related condition is detected, the Arduino processes the data and activates local and remote alert mechanisms using a GSM module.



Working Procedure – Step by Step

1. Power Supply Initialization

The system is powered ON.Arduino UNO initializes all connected sensors and modules.

2. Sensor Monitoring

Smoke sensor continuously measures smoke/gas concentration.Flame sensor monitors infrared radiation from fire.

3. Data Processing

Sensor readings are sent to the Arduino.Arduino compares readings with predefined threshold values.

4. Normal Condition Check

If smoke and flame levels are below thresholds:System remains in monitoring mode.

buzzer remain OFF.No GSM alert is sent.

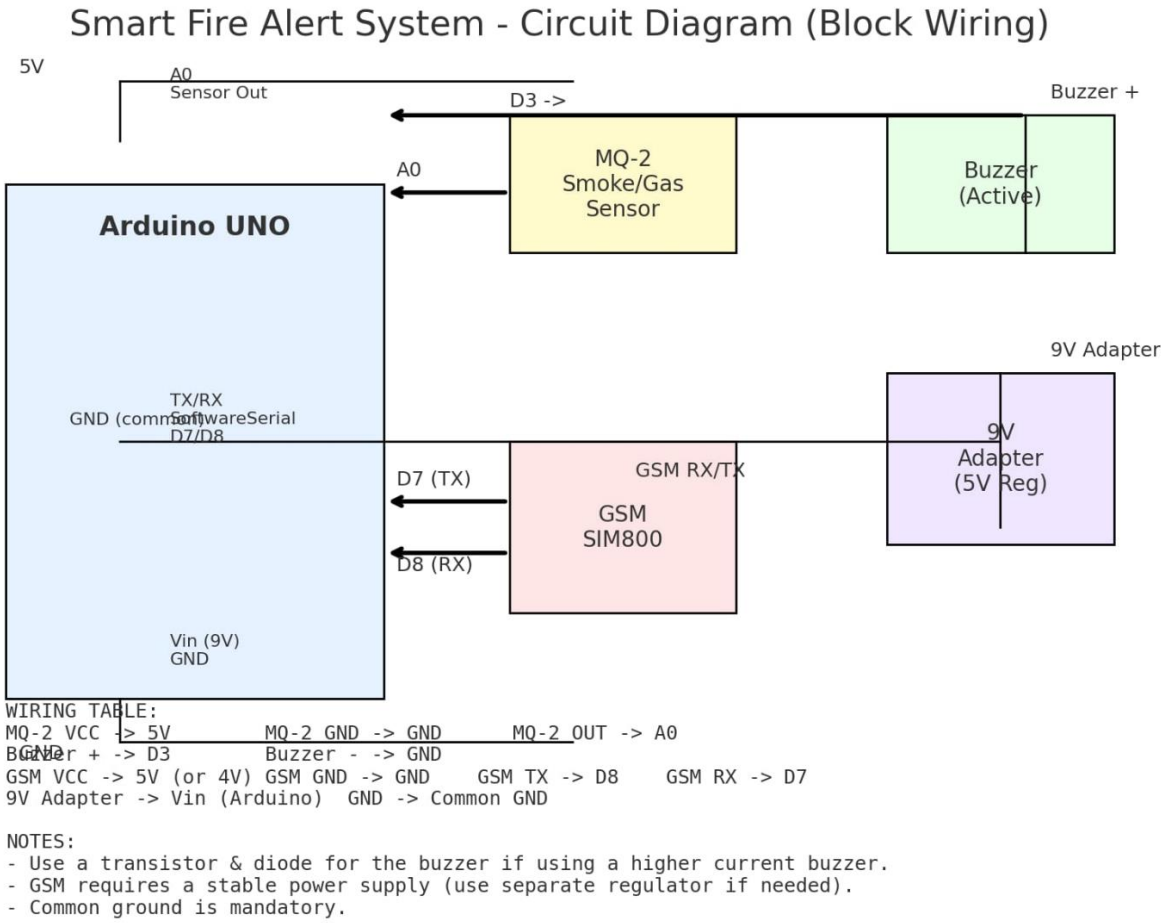
5. Fire Detection ConditionIf smoke or flame exceeds threshold:

Fire condition is confirmed.

6. Local Alert Activation

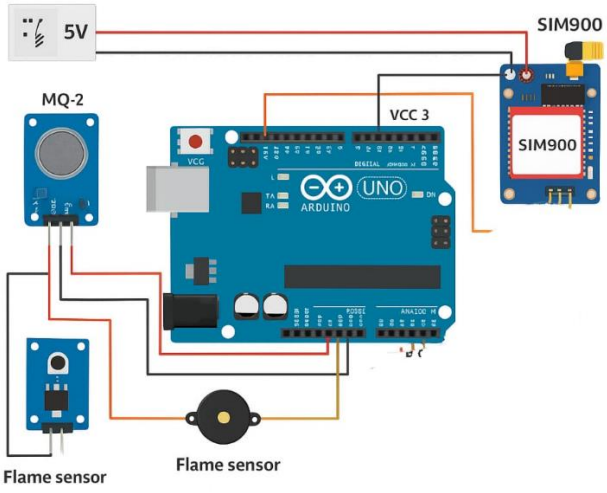
Buzzer is activated to alert nearby people. GSM Alert Transmission

GSM module sends an SMS alert to stored mobile numbers.GSM module makes an emergency call to the user



3.3.1 Materials Used

3.3.2 System Diagram (circuit connections)



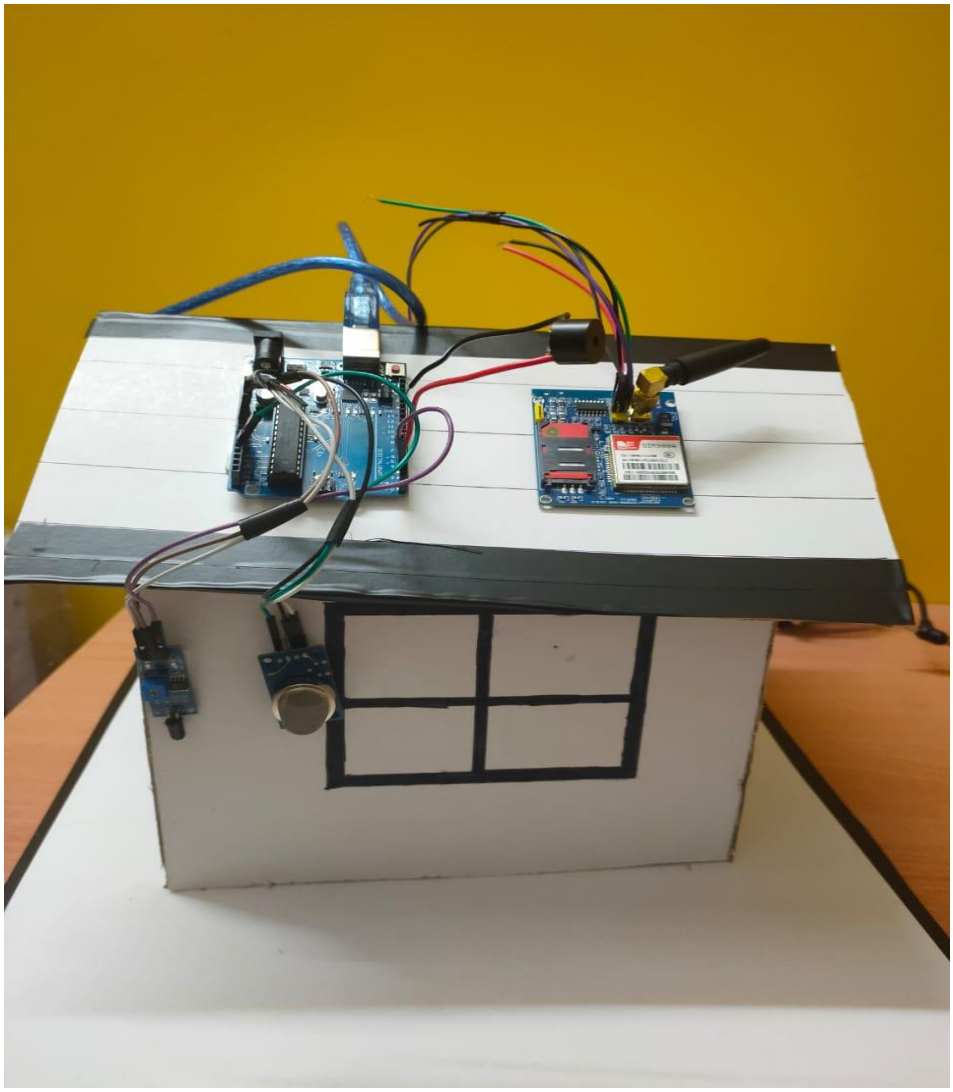
Chapter 4

Implementation

```
#include <SoftwareSerial.h>

SoftwareSerial sim900(7, 8);  // RX, TX

const int flamePin = 2;

const int mqPin = A0;

const int buzzerPin = 9;

int mqThreshold = 350;

void setup() {

  pinMode(flamePin, INPUT);

  pinMode(mqPin, INPUT);

  pinMode(buzzerPin, OUTPUT);

  digitalWrite(buzzerPin, LOW)

  Serial.begin(9600);

  sim900.begin(9600);

  delay(1000);

 // Initialize SIM900

  sim900.println("AT");

  delay(500);

  sim900.println("AT+CMGF=1");  // SMS text mode

  delay(500);

}

void loop() {
```

```
  bool flame = (digitalRead(flamePin) == LOW); // LOW = flame detected
  int smokeValue = analogRead(mqPin);
  bool smoke = smokeValue > mqThreshold;
  Serial.print("Flame = "); Serial.print(flame);
  Serial.print(" | Smoke = "); Serial.println(smokeValue);

  if (flame || smoke) {
    digitalWrite(buzzerPin, HIGH);
    String message = "FIRE ALERT! ";
    if (flame) message += "Flame detected. ";
    if (smoke) message += "Smoke level high: " + String(smokeValue);
sendSMS("+91XXXXXXXXXX", message);
   makeCall("+91XXXXXXXXXX");
 delay(60000); // cooldown to prevent repeated calls/SMS every second
  }
  else {
    digitalWrite(buzzerPin, LOW);
  }
  delay(500);
}
void sendSMS(const char* number, String text) {
  sim900.println("AT+CMGF=1");
  delay(300);
  sim900.print("AT+CMGS=\"");
```

```
  sim900.print(number);
  sim900.println("\"");
  delay(300);
  sim900.print(text);
  delay(200);
 sim900.write(26); // CTRL+Z
  delay(3000);
}
void makeCall(const char* number) {
  sim900.print("ATD");
  sim900.print(number);s
  sim900.println(";");
  delay(15000);   // let it ring for 15 seconds
  sim900.println("ATH"); // hang up
  delay(1000);
}
```



PICTURE OF REAL PROTOTYPE

Chapter 5

Results and Analysis

User Testing & Feedbacks

1. User Testing

Testing Scenarios:

Smoke Generation Test: Smoke was introduced near the sensor to test smoke detection.

Flame Detection Test: A small flame source was used to test flame sensor response.

Alert Verification Test: SMS and call alerts were checked on registered mobile numbers.

False Alarm Test: System behavior was observed under normal lighting and environmental conditions.

Testing Procedure:

1. The system was powered ON and allowed to stabilize.

2. Users simulated fire conditions using smoke and flame sources.

3. Observed buzzer, LED, SMS, and call alerts.

4. Feedback was collected regarding response time and alert clarity.

2. User Feedback

Feedback was gathered from users after testing the system.

Positive Feedback:

The system was easy to understand and operate.

Immediate buzzer and LED alerts were helpful for local warning.

SMS and call alerts were received clearly and on time.

Users appreciated the internet-independent GSM communication.

3. Feedback Analysis

Parameter User Response

Ease of Use Very Good

Alert Speed Fast

Alert Clarity Clear

Reliability Good

4. Improvements Implemented Based on Feedback

Sensor threshold values were fine-tuned.

Alert messages were made clearer.

Buzzer activation time was optimized.

GSM signal stability was improved using an external antenna.

Chapter 6

Conclusion & Future Work

The Fire Alert System using GSM–Arduino Module was successfully designed and implemented to provide early fire detection and instant alert notification. The system effectively monitors fire-related parameters using smoke and flame sensors and processes the data through an Arduino UNO. Upon detecting a fire condition, the system activates local alerts using a buzzer and LED and sends remote notifications through SMS and phone calls via the GSM module.

The system operates independently of the internet, making it reliable in remote and low-connectivity areas. Testing and user feedback confirmed that the system is cost-effective, easy to use, and capable of providing timely alerts, thereby reducing potential damage to life and property. Overall, the project meets its objectives of safety, reliability, and affordability.

Future Work

The current system can be further enhanced with the following improvements:

1. Battery Backup Integration

Ensures system operation during power failures.

2. Temperature Sensor Addition

Improves accuracy by monitoring heat levels along with smoke and flame.

3. Multiple Contact Support

Enables alert notifications to be sent to multiple users simultaneously.

4. IoT Integration

Allows real-time monitoring through a mobile application or web dashboard.

5. GPS Module Integration

Provides exact location information during emergency alerts.

References

1. Arduino Official Documentation

Arduino, “Arduino UNO – Technical Specifications and Datasheet.”

2. SIM900 GSM Module Datasheet

SIMCom Wireless Solutions, “SIM900 GSM/GPRS Module Hardware Design.”

3. MQ-2 Smoke Sensor Datasheet

Hanwei Electronics, “MQ-2 Semiconductor Gas Sensor Technical Data.”

4. Flame Sensor Module Datasheet

“Infrared Flame Sensor Module – Technical Specifications.”

5. Arduino IDE Documentation

Arduino, “Getting Started with Arduino Programming.”

https://youtu.be/wufOuwQgl40?si=fmgwAJ3weo8aNsDn