FIRE ALARM SYSTEM USING ARDUINO AND GSM MODULE

REVIEW-2 BY

J.SANGAMESH

SAI VAIBHAV

HARI SHANKAR

SURESH

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.

LITERATURE SURVEY:

- M. S. Bin [23]Propose that Fire alarm is a real-time monitoring device which detects fire smoke and records photos via a camera in a room if a fire arises. Raspberry Pi and Arduino Uno are the embedded systems for developing the fire alarm system. The system's primary feature is the ability to transmit an alarm remotely if a fire is detected. The technology will display a picture of the room condition on a website when the presence of smoke is detected. To report the occurrence to the Firefighter via Short Message service, the system needs user confirmation (SMS). This approach has the advantage of reducing the risk of the Firefighter receiving erroneous alerts. The camera merely captures a picture so that this system uses a little power and storage.

LITERATURE SURVEY 2:

- K. Muheden,[24] Propose that For the safety of these regions it is of essential relevance to monitor domestic and industrial sites through sensors and to avoid issue through prediction. This article illustrates how the Wireless Sensors (WSN) approaches may be increased via the development of new design strategies and the improvement of low-cost systems for industrial and residential security. In this study, To assure and provide optimal solutions for the system, not only temperatures and moisture detectors, but also flames and gas detectors was used. Because of the construction of a basic electrical circuit, anybody may operate this wirelessly house protection systems. A notification was used as a method of informing people about the system. Signals from detectors like gas, flames, warmth, and humidity are received by the implant Arduino device, which were constructed using Android Studio. It will enable access to the WIFI networks and deliver a warning alert to smartphone customers if it finds that the obtained data has control levels over a specified threshold, in order to pre-monitor the potential of a fire.

IMPLEMENTATION CODE:

```cpp
#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;

int thresh = 200;

int status = true;

String alertMsg;

String mob1 = "+918789801436";

String mob2 = "+917491981734";


void setup() {

 pinMode(red, 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() {
  Serial.println("Gas Val: " + String(analogRead(smoke)) + ", Flame state: " + String(!digitalRead(flame)));
  if (digitalRead(flame) == LOW || analogRead(smoke) > thresh) {
    digitalWrite(red, HIGH);
    siren(buzzer);
    digitalWrite(green, LOW);

    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"));
    }


    Serial.println(alertMsg);
    if(status) {
      status = false;
      String msg = "Alert Type: " + alertMsg;
      SendMessage(msg, mob1);
      delay(8000);
      SendMessage(msg, mob2);
    }
```
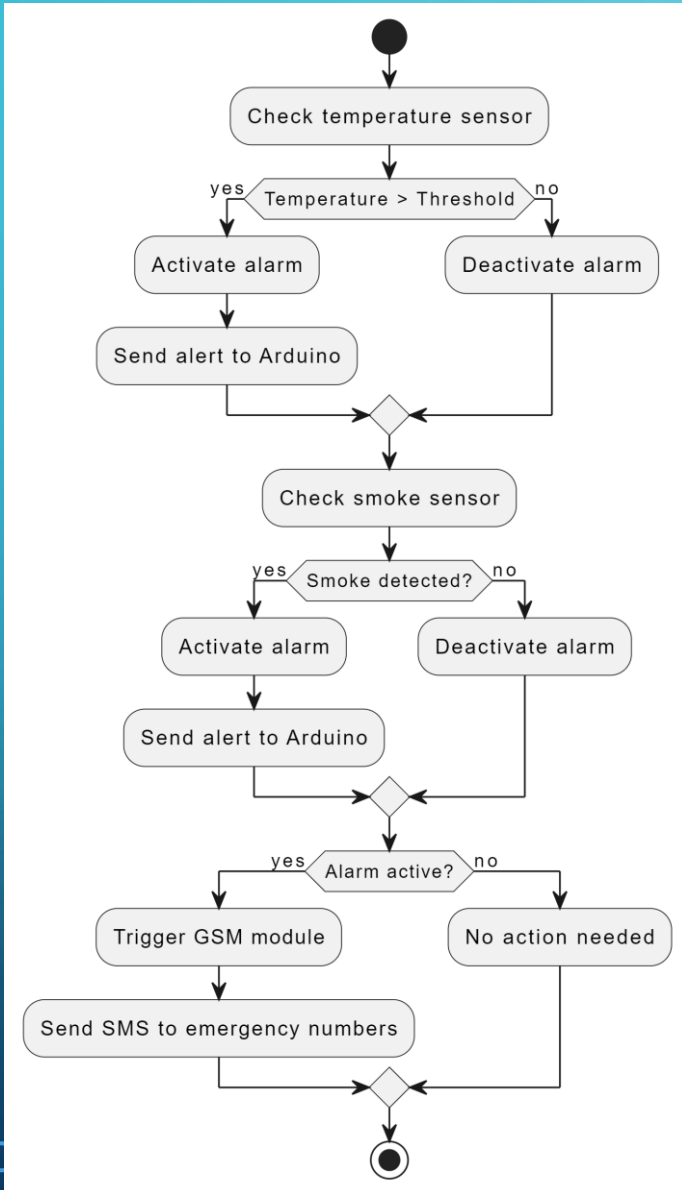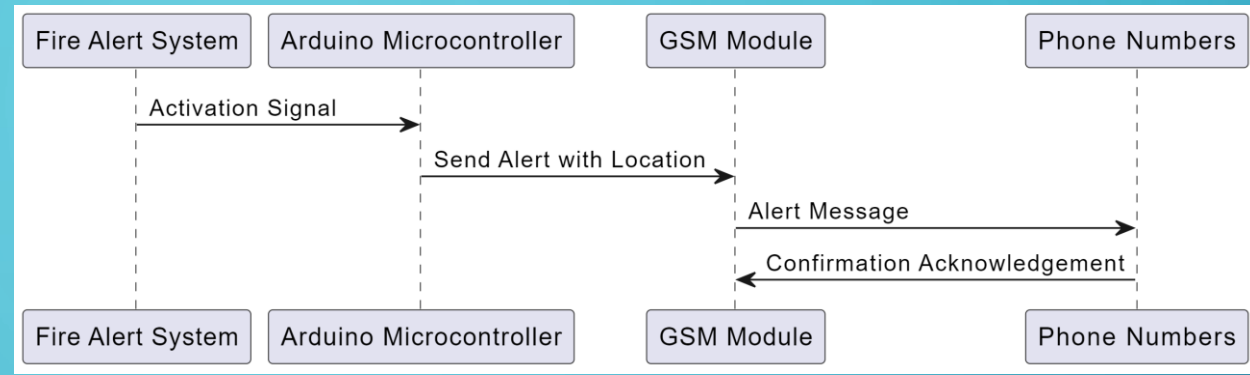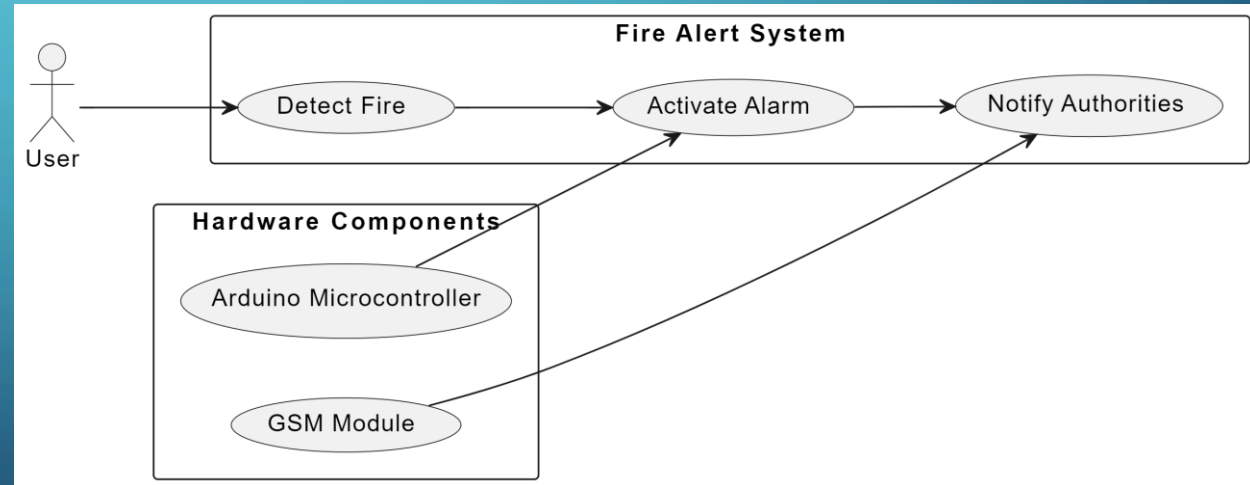
```
  } else {

    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(red, LOW);

    noTone(buzzer);

    digitalWrite(green, HIGH);

  }

  delay(500);

  lcd.clear();

}


void SendMessage(String msg, String mob) {

  Serial.println(msg);


  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);

}
```

ACTIVTY DIAGRAM:

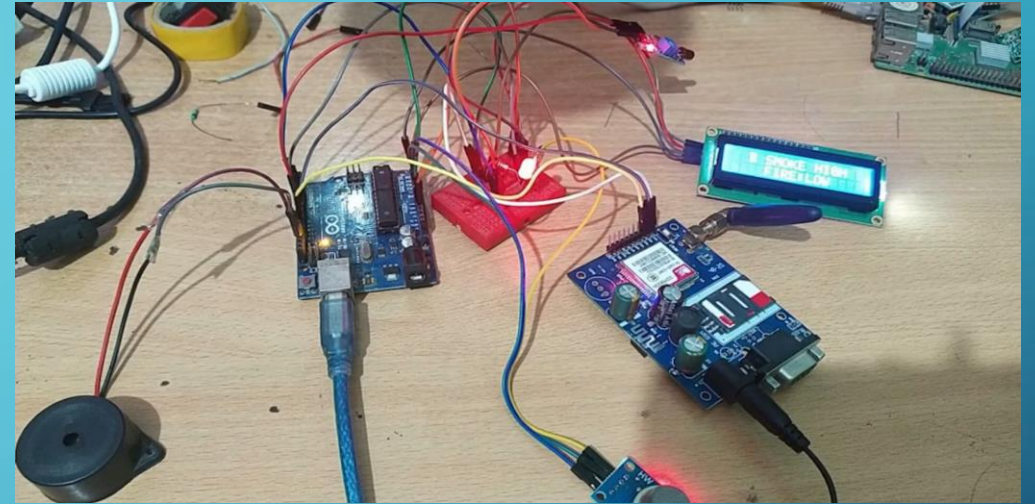
SEQUENCE DIAGRAM:

USE-CASE DIAGRAM:
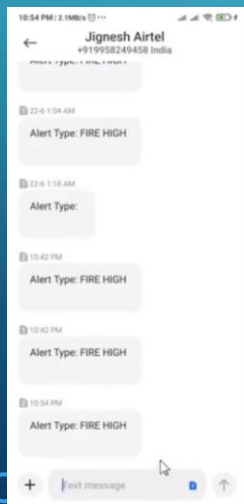
CIRCUIT DIAGRAM:
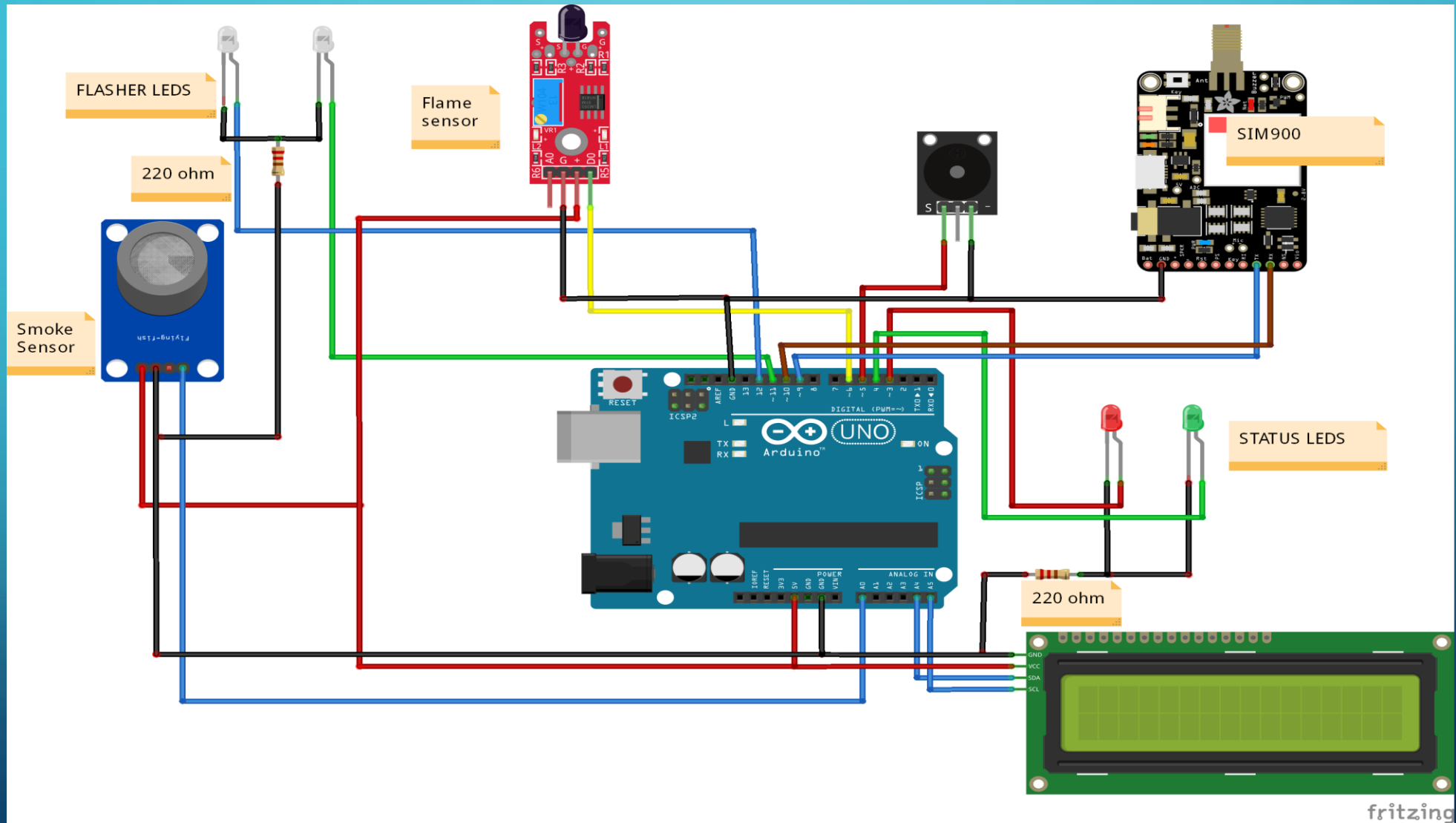
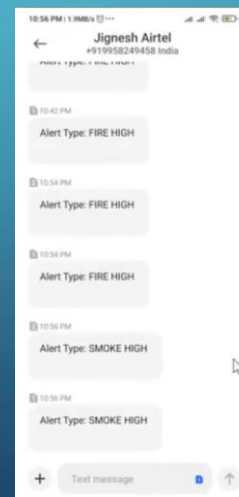# OUTPUT SCREENS:

•FIRE ALERT ON LCD SCREEN
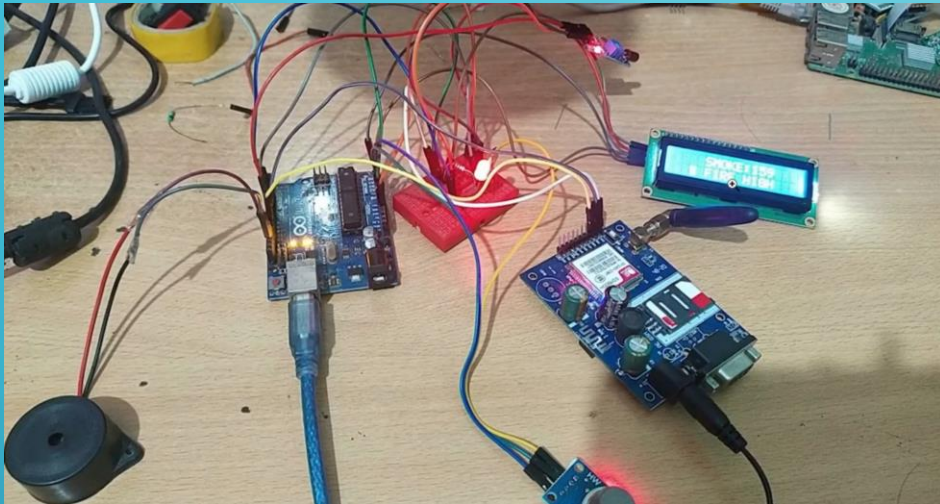
SMOKE ALAERT ON LCD SCREEN

FIRE ALART MESSAGE TO USER

•SMOKE ALERT MESSAGE ON MOBILE

THANK YOU!!!