INDUSTRIAL FIRE SAFETY SYSTEM

TEAM MEMBERS:

- JIYA SHAJIL
- THANMAYA RAJITH
- HELEN SANDRA BABU
- SAYANTH B S
- HARIF HANEEFA

INTRODUCTION

Imagine a system that never sleeps, constantly vigilant against the threat of fire, ready to alert you at the first sign of danger. This project brings that vision to life with a fire alarm system powered by Arduino, temperature sensors, and gas sensors. Combining precision technology with simplicity, the system detects unusual heat or flammable gases and instantly triggers an alarm, providing critical moments to prevent disaster. Designed to be efficient and affordable, this innovation aims to safeguard homes, workplaces, and communities. This report explores the design, functionality, and potential applications of this lifesaving system.

PROBLEM STATEMENT

Fire hazards pose serious risks, often resulting in severe damage if not detected early. Traditional fire alarm systems can be costly and inaccessible for smaller applications. This project addresses the need for an affordable and efficient solution by developing a fire alarm system using Arduino, integrating temperature and gas sensors for real-time fire detection and timely alerts.

SCOPE OF SOLUTION

- Provides a cost-effective and efficient solution for fire detection.
- Suitable for residential, commercial, and industrial applications.
- Monitors temperature and gas levels in real time for early hazard detection.
- Issues timely alerts to minimize damage and enhance safety.
- Modular and customizable for integration with other safety systems.
- Scalable design allows for adaptation to different environments and requirements.

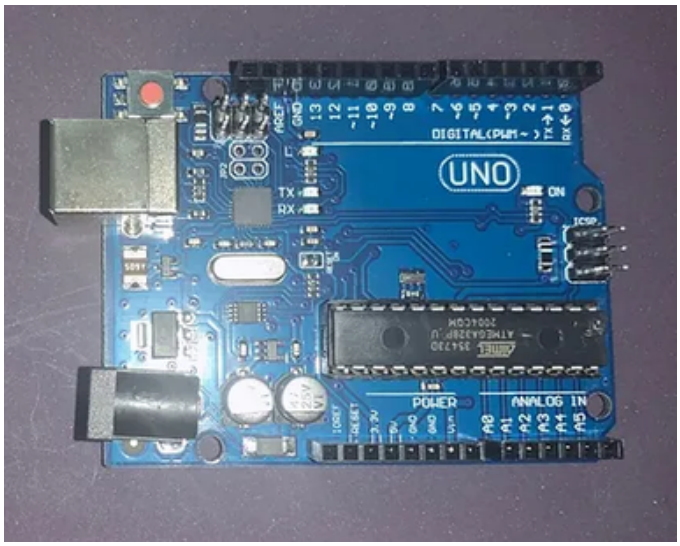
PRINCIPLE

The industrial fire safety system operates on the principle of real-time monitoring of environmental conditions, such as temperature and gas concentration. A significant rise in temperature or the presence of smoke/flammable gases indicates a potential fire. The system then promptly triggers an alert to ensure the safety of workers and facilitate evacuation.

REQUIRED COMPONENTS

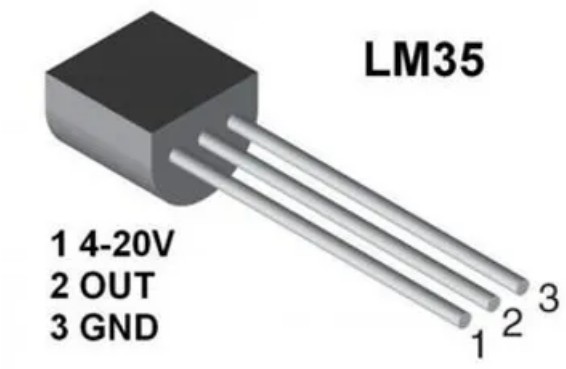
1. ARDUINO UNO R3

Arduino board is a microcontroller that is used to accept inputs from sensors connected and provide an output action on the desired device connected to it. The sensor inputs can be from light-detecting sensors, motion sensors (Ultrasonic or IR), temperature sensors, etc. The output from this device can be received through other output devices such as LED, Buzzer, Serial monitor, etc.



2. LM-35 TEMPERATURE SENSOR

Measures the temperature in the surroundings. A rapid rise in temperature indicates a fire.



3. GAS SENSOR

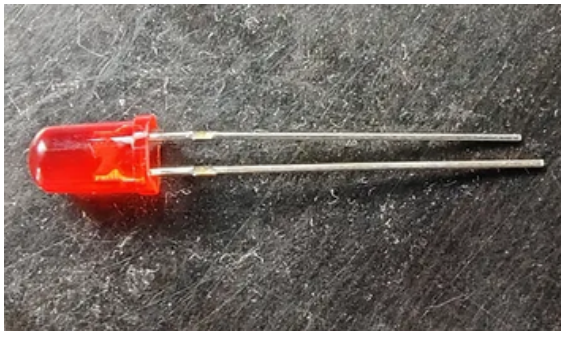
Detects smoke or flammable gases that may be present during a fire incident.

4. PIEZO BUZZER

Emits a sound to alert workers of a fire emergency.

5. LED

Provides a visual alert indicating the fire emergency.



REQUIRED SOFTWARES

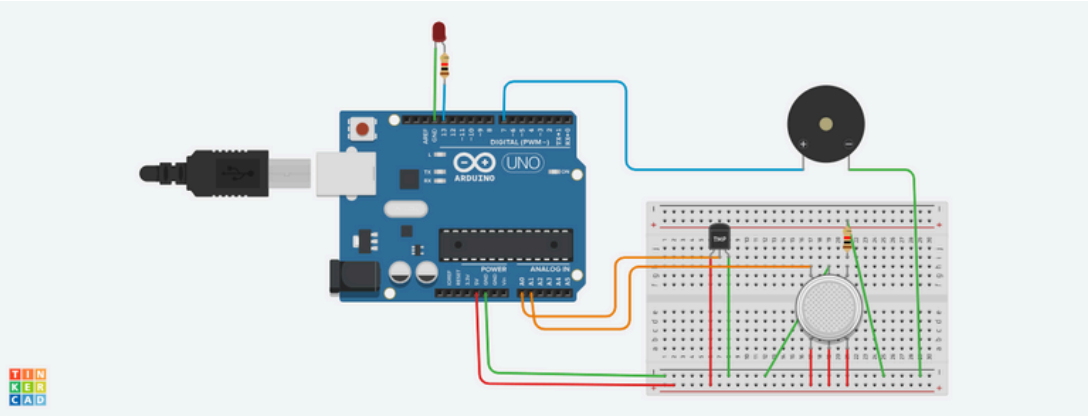
1. TINKERCAD

It is an online simulation software used for circuit design. It has all the electrical components required to built circuits and runs them.

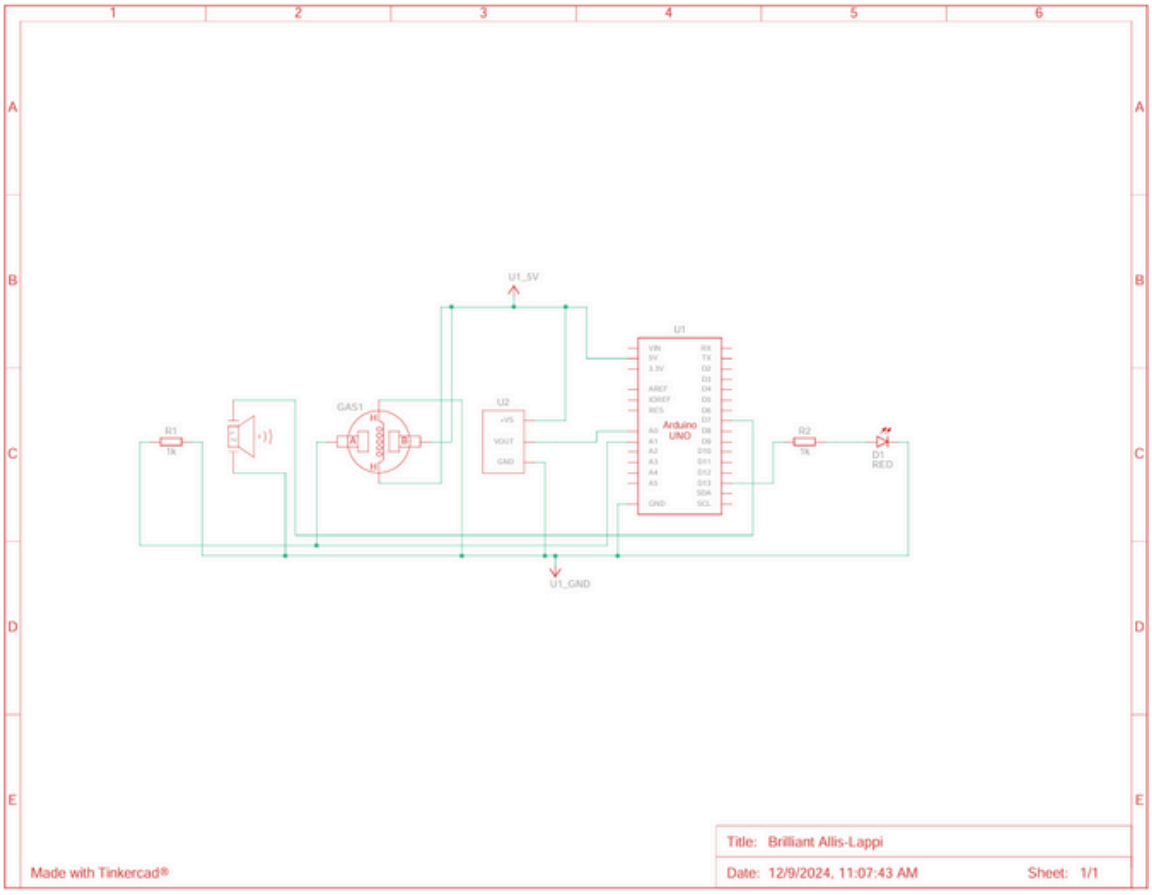
2. FRITZING

Fritzing is an open-source initiative to develop amateur or hobby CAD Software for the design of electronic hardware intended to allow designers and artists to build more permanent circuits from prototypes.

SIMULATED CIRCUIT



SCHEMATIC



WORKING

Power Supply:
The Arduino Uno and sensors are powered by a 5V source. The power rails on the breadboard distribute power to all components.
Data Collection:
The TMP36 sensor outputs an analog signal proportional to the temperature. This signal is fed to one of the analog pins (e.g., A0) of the Arduino Uno.
The gas sensor outputs an analog signal based on the concentration of gas or smoke detected. This signal is also connected to an analog pin (e.g., A1) on the Arduino.
Data Processing:
The Arduino reads the analog values from the TMP36 and the gas sensor. These values are converted to their respective temperature and gas concentration.
Threshold values for temperature and gas concentration are predefined in the code. If the measured values exceed these thresholds, it indicates a fire or gas hazard.
Alerts:
When a fire is detected (either high temperature or gas levels), the Arduino activates the buzzer and the LED.
The buzzer sounds continuously, and the LED glows as long as the hazardous conditions persist.
Output Actions:
The system can be extended to send notifications or activate a sprinkler system using additional modules.
Code Integration:
The Arduino code includes logic for reading sensor values, comparing them to threshold limits, and triggering the alert system.

C O D E

```
float temp;
float vout;
int LED = 13;
int gasSensor;
int piezo = 7;

void setup() {
  pinMode(A0, INPUT); // TMP36 Temperature sensor
  pinMode(A1, INPUT); // Gas sensor
  pinMode(LED, OUTPUT);
  pinMode(piezo, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  // Read TMP36 sensor
  vout = analogRead(A0);
  temp = (vout * 5.0 / 1023.0 - 0.5) * 100;

  // Read gas sensor
  gasSensor = analogRead(A1);

  // LED and buzzer for high temperature
  if (temp >= 80) {
    digitalWrite(LED, HIGH);
    tone(piezo, 2000); // High-pitched tone for high temperature
    delay(500);
    digitalWrite(LED, LOW);
    noTone(piezo);
    delay(500); // Blink every 1 second
  }
  // LED and buzzer for gas detection
  else if (gasSensor >= 100) {
    digitalWrite(LED, HIGH);
    tone(piezo, 1500); // Tone for gas detection
    delay(500);
    digitalWrite(LED, LOW);
    noTone(piezo);
    delay(500); // Blink every 1 second
  }
  else {
    digitalWrite(LED, LOW); // Turn off LED if no alerts
    noTone(piezo);          // Stop buzzer
  }
```

```
 // Debug output
  Serial.print("Temperature (C): ");
  Serial.print(temp);
  Serial.print("\t");
  Serial.print("Gas Sensor: ");
  Serial.print(gasSensor);
  Serial.println();

  delay(100); // Short delay for smooth loop execution
}
```

VIDEOS

videos

GERBER FILES

Files

GITHUB LINK

github

THANK YOU