Fire Alarm Development System

Group ID: MCA-II **Date of Submission:** 06.11.2023

Group Details:

Group Members:

1. Ramya S J
2. Vennila S
3. Swetha M

1. Introduction:

Our project is based on “fire alarm system”. It is used to protect us from fire. Here we are using smoke sensor and temperature sensor to detect smoke and temperature in a particular area.it helps us to give an alert message before fire starts based on the smoke detection. we can use this project in some applications. some applications are firework factories, transportations, hospitals and some other industries.to use this project we can prevent people and other things from disaster. It is a disaster prevent application to reduce the problems related to fire. For example we can take firework factories – we saw many fire accidents in those factories because of this many people died and so many things are burned. Here the organization face great loss . To avoid this problem we can use our project . here our MQ-2 sensor detect some gases like Methane(CH_4) , Butane(C_4H_{10}), LPG,Propane(C_3H_8),Alcohol,Carbon Monoxide(CO),Smoke.

2. Components required:

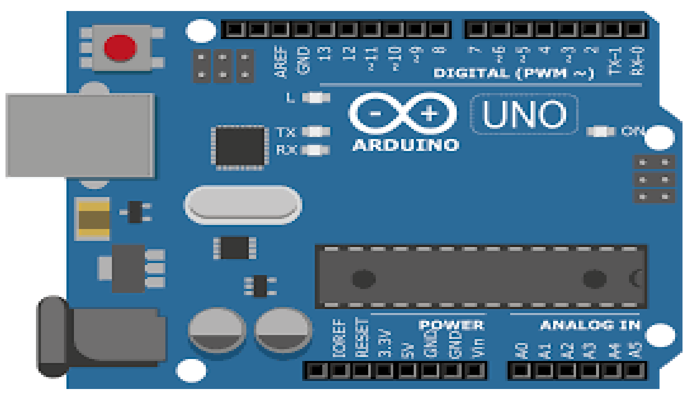
2.1. Table of components

Sensors	Actuators	Other Components
MQ-2 Gas Sensor	Piezo Buzzer	Arduino UNO Board, Bread Board.
DHT-11 Temperature and humidity sensor	Light Emitting Diode	Jumper wires, USB Cable, Resistors.

2.2. Components Explanation

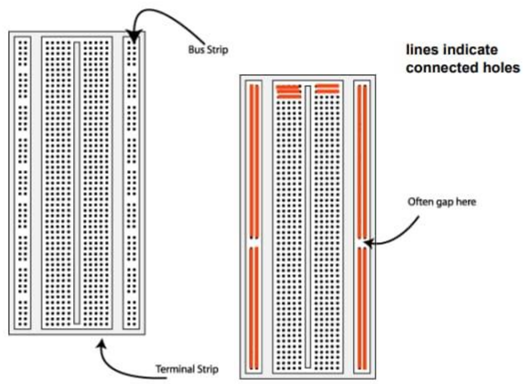
Arduino UNO Board:

Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online.



Bread board:

A breadboard (sometimes called a plugblock) is used for building temporary circuits. It is useful to designers because it allows component to be removed and replaced easily. It is useful to the person who wants to build a circuit to demonstrate its action, then to reuse the components in another circuit.

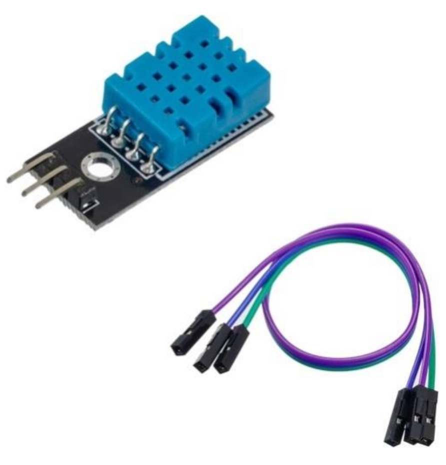


DHT-11 temperature and Humidity Sensor:

The DHT11 is a low-cost digital temperature and humidity sensor commonly used in various DIY electronics and Internet of Things (IoT) projects. It is popular among hobbyists and makers due to its affordability and simplicity. Here are some key features and information about the DHT11 temperature sensor.

Temperature: The DHT11 can measure temperatures in the range of 0°C to 50°C (32°F to 122°F).

Humidity: It can measure relative humidity in the range of 20% to 80%.

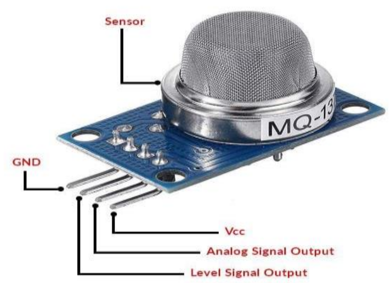


***MQ-2 Gas sensor*:**

The MQ-2 is a popular gas sensor module used in various electronic projects and applications to detect the presence of different gases. It is often used in safety systems, environmental monitoring, and IoT projects where gas detection is important. Here are some key features and information about the MQ-2 gas sensor

Gas Detection: The MQ-2 sensor is capable of detecting a variety of gases, including:

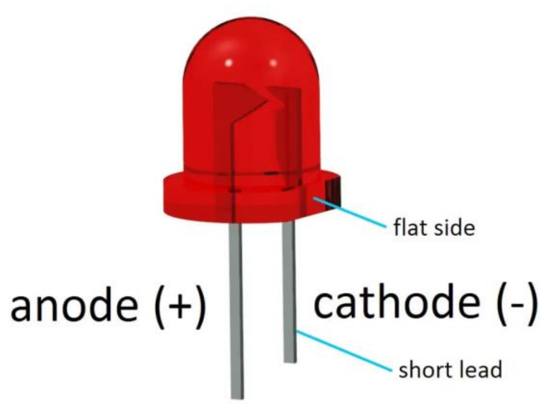
- ✓ LPG



- ✓ Propane
- ✓ Methane
- ✓ Butane
- ✓ Alcohol
- ✓ Hydrogen
- ✓ Smoke

LED'S:

The major uses of LED (Light Emitting Diodes) are to illuminate objects and even places. Its application is everywhere due to its compact size, low consumption of energy, extended lifetime, and flexibility in terms of use in various applications.



Jumpers:

A **jumper** is a short length of conductor used to close, open or bypass part of an electronic_circuit. They are typically used to set up or configure printed circuit_boards, such as the mother boards of computers. The process of setting a jumper is often called **strapping.**

***USB Cable*:**

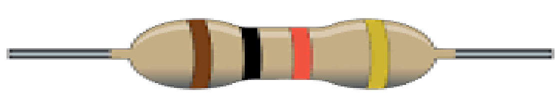
Many older Arduino boards, like the Arduino Uno, Arduino Mega, and Arduino Leonardo, use a USB Type-B connector on the board. The USB Type-A to USB Type-B cable is a standard USB cable that you can find in most computer accessories stores or online retailers. It's often referred to as a "USB A to B cable" for arduino.

Piezo Buzzer:

Piezo buzzers operate based on the piezoelectric effect. When an electrical voltage is applied to the piezoelectric material inside the buzzer, it deforms and produces sound waves. Conversely, when mechanical pressure or vibration is applied to the piezo element, it generates an electrical signal.

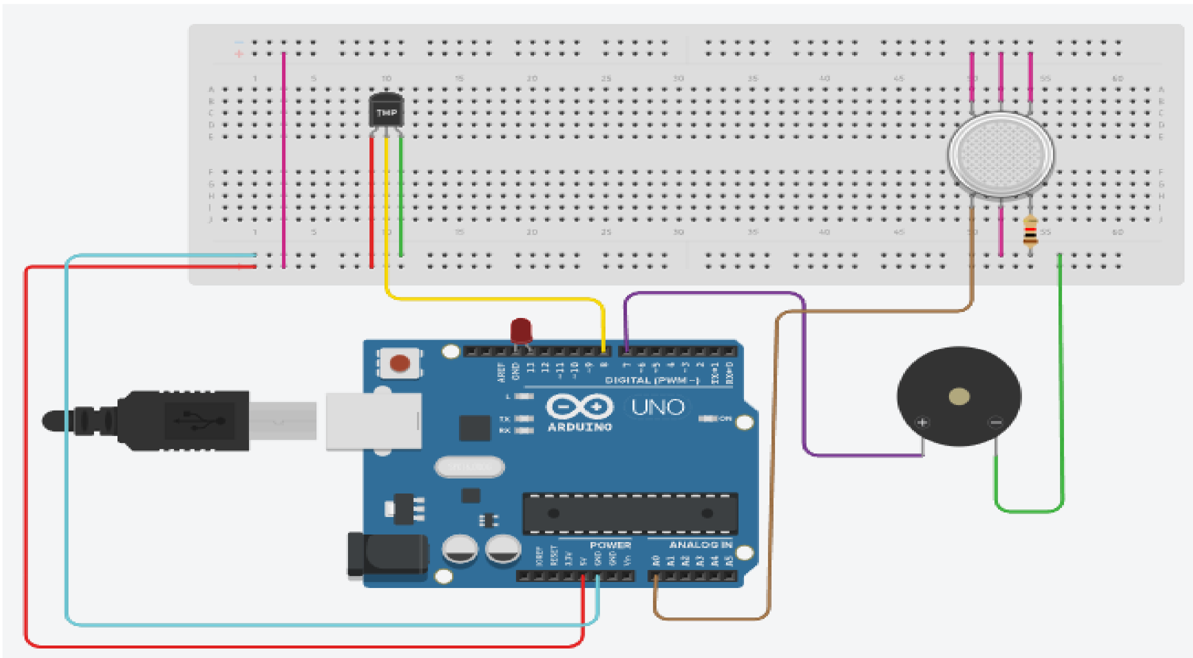
Resistor:

A resistor is a fundamental electronic component that is used to restrict or limit the flow of electric current in an electrical circuit. It is a passive component, which means it does not generate or amplify signals but rather opposes the flow of current.

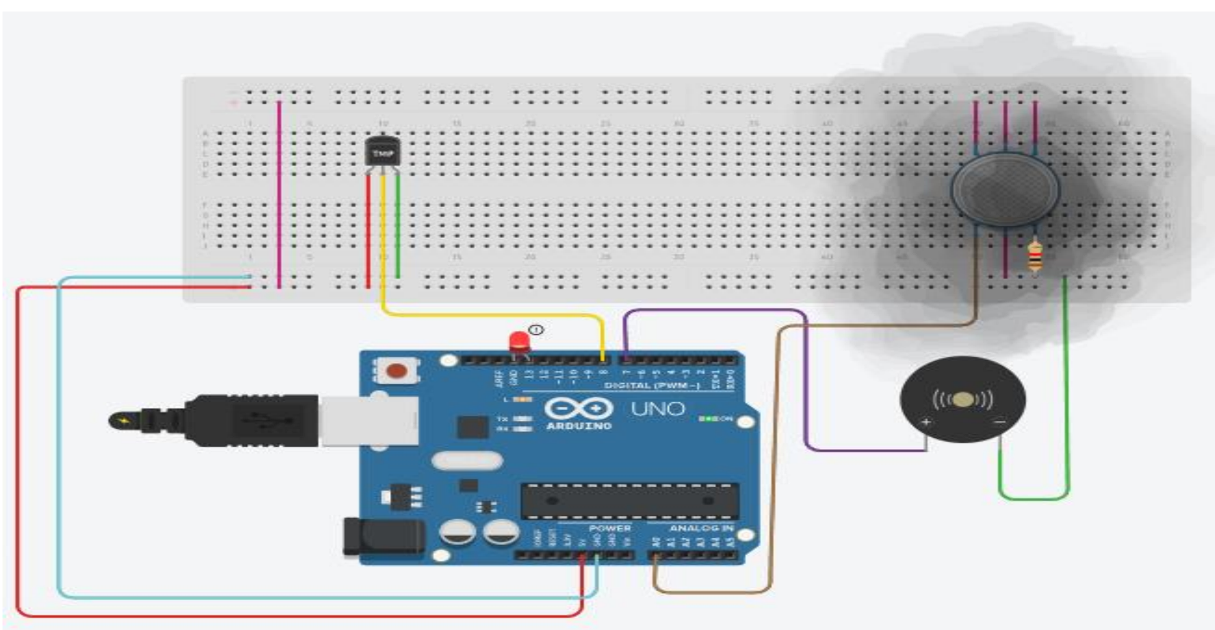


3. Exprementаl Approach:

3.1. Schematic Diagram



Before Simulation



After Simulation

```
Temperature in Degree C=  100.05        GasSensor=  903
Temperature in Degree C=  100.05        GasSensor=  149
Temperature in Degree C=  100.05        GasSensor=  216
Temperature in Degree C=  100.05        GasSensor=  909
Temperature in Degree C=  100.05        GasSensor=  928
```

Serial Monitor Readings

3.2. Connection Process

Step 1: Gather the Components

Before you start, make sure you have the following components:

-Arduino board (e.g., Arduino Uno)

-DHT11 temperature and humidity sensor

-Gas sensor (e.g., MQ-2)

-Breadboard and jumper wires

-USB cable for connecting your Arduino to a computer

Step 2: Connect the DHT11 Sensor

- The DHT11 sensor has four pins: VCC, GND, DATA, and NC (No Connection).
- Connect the VCC (power) pin of the DHT11 to the 5V output on your Arduino.
- Connect the GND (ground) pin of the DHT11 to a GND (ground) pin on your Arduino.
- Connect the DATA pin of the DHT11 to pin 8 on your Arduino.

Step 3: Connect the Gas Sensor

- The MQ-2 gas sensor typically has four pins: VCC, GND, AOUT (analog output), and DOUT (digital output). For this example, we'll use the analog output (AOUT) to measure the gas sensor value.
- Connect the VCC (power) pin of the gas sensor to the 5V output on your Arduino.
- Connect the GND (ground) pin of the gas sensor to a GND (ground) pin on your Arduino.
- Connect the AOUT (analog output) pin of the gas sensor to analog pin A1 on your Arduino.

Step 4: Connect LEDs (Optional)

- If you want to use LEDs to indicate when the gas level is above a threshold, you can connect them to pins 7 and 13 on your Arduino.
- Connect a current-limiting resistor (around 220-470 ohms) to each LED's anode (longer lead).
- Connect the other end of the resistors to pins 7 and 13 on your Arduino.
- Connect the cathodes (shorter leads) of the LEDs to GND.

Step 5: Connect the Arduino to Your Computer

- Connect your Arduino board to your computer using the USB cable.

Step 6: Upload the Code

- Now, you can upload the code you provided in your initial question to your Arduino board using the Arduino IDE.

Step 7: Monitor the Serial Output

- Open the Arduino IDE's Serial Monitor (Tools > Serial Monitor) to view the temperature, humidity, and gas sensor readings. If the gas sensor value is above 1000, the LEDs on pins 7 and 13 should turn on.

- Please ensure that you have the necessary libraries installed for the DHT sensor and follow the appropriate pin connections for your gas sensor model. Double-check your connections to avoid any issues with your circuit.

3.3. PIN Connections

MQ-2 Gas Sensor:

- ✓ VCC to A1 in Arduino
- ✓ GND to (–ve) in BreadBoard
- ✓ D0 to (–ve) in Bread Board with Resistor
- ✓ A0 to (+ve) in Bread Board

DHT-11 Temperature and Humidity Sensor:

- ✓ (-ve) to (-ve) in Bread Board
- ✓ Output pin to pin 8 in Arduino Board
- ✓ (+ve) to (+ve) in Bread Board

Piezo Speaker:

- ✓ (+ve) red wire to pin 7 in Arduino Board
- ✓ (-ve) black wire to (-ve) in Bread Board

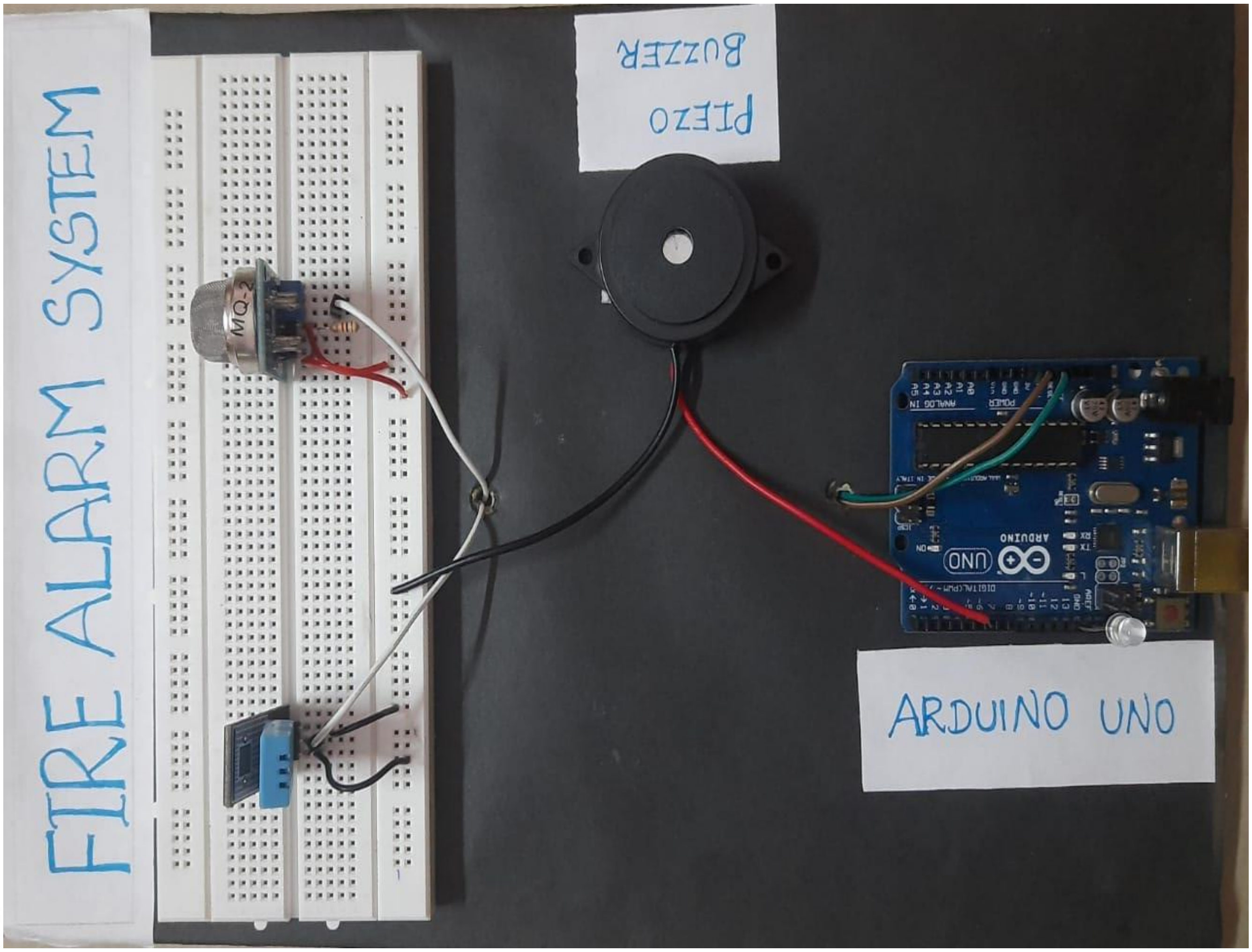
LED:

- ✓ (+ve) to pin 13 in Arduino Board
- ✓ (-ve) to GND in Arduino Board

Source Positive and Negative:

- ✓ Source (+ve) to GND in Arduino Board
- ✓ Source(-ve) to 5v in Arduino Board

Real Picture:



4. Code

4.1. Source Code

```
#include <DHT.h>
#define DHTPIN 8
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
void setup() {
```

```
  pinMode(A0, INPUT);
  pinMode(A1, INPUT);
  pinMode(7,OUTPUT);
  pinMode(13,OUTPUT);
  // Initialize the DHT sensor
  dht.begin();
  Serial.begin(9600);
}
void loop() {
  // Read temperature and humidity from the DHT sensor
  float humidity = dht.readHumidity();
  float temperature = dht.readTemperature();
  float gasSensor = analogRead(A1);
  if(gasSensor>=1000){
    digitalWrite(7,HIGH);
    digitalWrite(13,HIGH);  }
  else{
    digitalWrite(7,LOW);
    digitalWrite(13,LOW);
  }
  // Print temperature and humidity
  Serial.print("Temperature in Degree C = ");
  Serial.print(temperature);
```

```
  Serial.print("\t");

  Serial.print("Humidity = ");

  Serial.print(humidity);

  Serial.print("\t");

  Serial.print("Gas Sensor = ");

  Serial.print(gasSensor);

  Serial.println();

 delay(1000);

}
```

4.2. Code Explanation

Void setup(){

}

- In this part of the code, we include the necessary DHT library to work with DHT sensors.
- We specify that we are using a DHT11 sensor connected to digital pin 8. We create a DHT object to interface with the sensor.
- Additionally, we set up the pins for various components, including an analog input (A0), another analog input (A1),
- a piezo buzzer (pin 7), and an LED (pin 13). Serial communication is initialized for debugging purposes at a baud rate of 9600.

Void loop(){

}

- In the loop() function, we read temperature and humidity from the DHT11 sensor using the dht.readHumidity() and dht.readTemperature() functions, respectively. We also
- read the gas sensor data from analog pin A1 using analogRead(A1).The code then checks the gas sensor value.
- If it's greater than or equal to 1000, it turns on both the piezo buzzer (connected to pin 7)
- and the LED (connected to pin 13). If the gas sensor value is less than 1000, it turns off both the piezo buzzer and the LED.
- The code prints temperature, humidity, and gas sensor values to the serial monitor for monitoring.A delay of 1000 milliseconds (1 second) is added to control the update rate of the data.
- This code allows for the integration of a DHT11 sensor for temperature and humidity readings and the control of an LED and a piezo buzzer based on the gas sensor's readings.
- Make sure your connections are correct, and the sensors are functioning properly for accurate data readings.

5. Experimental result and Discussion:

5.1 Experimental Results:

In this section, you can present the actual data and observations obtained during the execution of your code. The experimental results for your IoT project might look like the following:

Temperature and Humidity Data:

- During the experiment, the temperature readings ranged from [minimum temperature]°C to [maximum temperature]°C.
- The humidity levels varied between [minimum humidity]% and [maximum humidity]%.
- You can include tables or graphs to visualize this data.

Gas Sensor Data:

- Gas sensor readings were monitored continuously during the experiment.
- The gas sensor readings fluctuated between values, with values ranging from [minimum value] to [maximum value].
- You can also use a graph or chart to display the gas sensor data over time.

Digital Output Pins (7 and 13):

- The digital output pins (7 and 13) were controlled by the gas sensor reading.
- When the gas sensor reading exceeded the threshold of 1000, both pins were set to HIGH, turning on the corresponding LEDs.
- When the gas sensor reading dropped below 1000, both pins were set to LOW, turning off the LEDs.
- You can include images or videos to demonstrate the LED states.

5.2 Discussion:

In this section, you can analyze and discuss the results obtained from your IoT project. Here are some points to consider:

Temperature and Humidity Control:

- Discuss the importance of monitoring temperature and humidity in a specific application, such as a smart home system.
- Explain how the DHT sensor's data can be used to make decisions, like controlling a thermostat or ventilation system.

Gas Sensor and Safety:

Explain how the gas sensor can trigger alerts or actions in case of gas leakage or poor air quality.

Control Logic:

- Discuss the control logic implemented in your code, where the digital output pins respond to the gas sensor reading.
- Explain how this logic can be adapted for various applications and provide examples.

Challenges and Limitations:

- Mention any challenges faced during the experiment, such as sensor calibration, data accuracy, or hardware issues.
- Address any limitations of the project, including the range of sensors, response time, and the threshold value.

Future Improvements:

Suggest possible improvements or enhancements for your IoT system, like integrating more sensors, adding remote monitoring, or enhancing the user interface.

6. Conclusion:

- In conclusion, the development of our Fire Alarm System represents a critical step forward in enhancing safety and security in residential and commercial environments. The implementation of this system serves as a proactive measure to detect and respond to fire hazards promptly, minimizing the potential for loss of life and property damage.
- While our Fire Alarm System represents a significant advancement in fire safety technology, it's important to recognize that there is ongoing potential for refinement and expansion.

Throughout the course of this project, we have achieved the following key outcomes:

- ✓ Early fire detection
- ✓ Audible and visual alarms
- ✓ Integration with emergency services
- ✓ User-friendly interface

In summary, our Fire Alarm System is a critical addition to the field of safety and security. As fire safety remains a top priority, this system stands as a testament to the capabilities of technology in safeguarding lives and property. We are dedicated to the ongoing improvement and refinement of this system, with the goal of making it an indispensable tool for fire prevention and response in both residential and commercial settings.

7. References:

- ✓ ***https://youtu.be/vjfDk49-Ezs?si=7wR2A-lSPmriNBAn***
- ✓ ***https://youtu.be/MCT3nA6cc-s?si=ri6EKMKDM9b8vz2K***
- ✓ ***https://www.projectsof8051.com/iot-based-fire-alerting-system-project/***
- ✓ ***https://www.tinkercad.com/things/2a6w8bEeL7c***
- ✓ ***https://chat.openai.com/***