Project Report On

Fire Alarm and Smoke Detection

Student Name: Vandita Bhardwaj **UID: 24MCI10037**
Branch: MCA(AIML) **Section/Group: 24MAM-1/B**
Semester: 03 **Date of Performance: 07.11.25**
Subject Name: Internet Of Things **Subject Code: 24 CAH-723**

1. **Aim:** To design and develop an IoT-based fire alarm and smoke detection system that can sense fire or smoke in the surroundings and immediately send alerts to the user's mobile device through the internet.

2. **Objectives:**

- To detect the presence of fire or smoke using sensors.
- To alert the user immediately through IoT-based notifications.
- To display real-time fire and smoke status on a monitoring dashboard or mobile app.
- To enhance safety by providing an early warning system.

3. **Components Required:**

Sno	Name of Component	Qty.
1.	NodeMCU (ESP8266)	1

2.	Gas Sensor	1
3.	Bluetooth	1
4.	LED	3
5.	Resistor (230 Ohm)	3
6.	Breadboard	1

4. Details of Components:

a. ESP32 Microcontroller: The ESP32 microcontroller is the main control unit of the project. It features built-in Wi-Fi and Bluetooth, making it ideal for IoT-based applications. In this system, it reads data from the MQ-2 gas sensor and the flame sensor, processes the information, displays the results on the LCD, and sends real-time updates to the Blynk IoT platform. The ESP32 operates at 3.3V logic level and provides multiple GPIO and ADC pins for connecting sensors and output devices.

b. MQ-2 Gas and Smoke Sensor:

The MQ-2 sensor is used to detect the presence of smoke and flammable gases such as

LPG, methane, and propane. It provides an analog voltage that varies with the gas concentration. In this project, the analog output (AO) of the MQ-2 is connected to GPIO 35 of the ESP32. When smoke or gas levels increase beyond a predefined threshold, the system triggers an alarm and sends alerts through the Blynk application.

c. Flame Sensor:

The flame sensor is used to detect fire or flame using infrared (IR) radiation. It produces a digital output signal—LOW when a flame is detected and HIGH when no flame is present. The sensor is connected to GPIO 16 of the ESP32. When it detects fire, the ESP32 activates the buzzer, displays the warning on the LCD, and notifies the user through the IoT platform.

d. Buzzer:

The buzzer acts as an audible alarm to alert nearby people whenever smoke or fire is detected. It is connected to GPIO 18 of the ESP32 and turns ON automatically during any danger condition. This component ensures immediate local notification of hazardous situations.

e. Breadboard:

The breadboard is used for building and testing the circuit connections without soldering. It allows easy placement of components and helps in troubleshooting and modifying the connections during the prototype stage.

f. Connecting Wires:

Jumper wires (male-to-male and male-to-female) are used to establish electrical connections between the ESP32, sensors, LCD, and buzzer. They make the setup neat, flexible, and easy to rearrange during testing.

g. USB Cable:

The USB cable is used to upload the program to the ESP32 microcontroller from the Arduino IDE. It also supplies 5V DC power to the board during programming and testing stages.

h. Power Supply (5V DC):

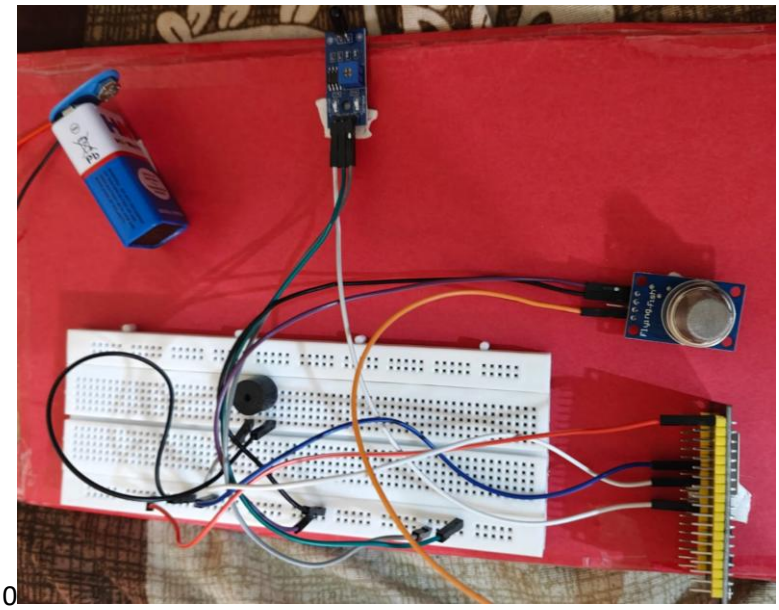
A 5V DC power supply provides the necessary voltage for operating the ESP32, LCD, and sensors when the system is running independently. It can be provided using a mobile adapter, USB power source, or regulated power module.

5. Working of Designed Model:

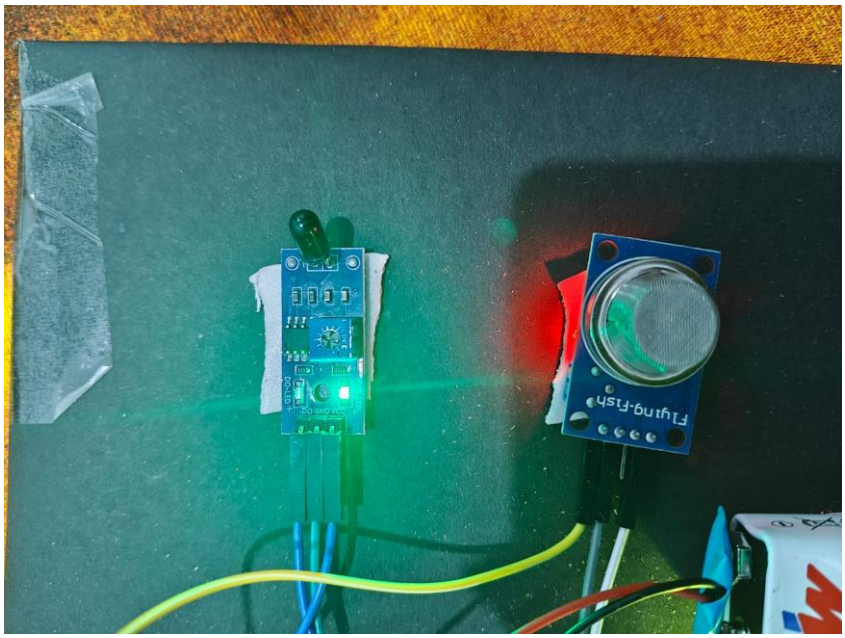
- The system is powered using a 5V DC supply, and the ESP32 microcontroller initializes all the connected components such as the MQ-2 gas sensor, flame sensor, buzzer, and I2C LCD display.
- The LCD display first shows a startup message ("System Ready" or "Monitoring...") and then begins displaying the real-time gas readings and flame detection status.
- The MQ-2 sensor continuously senses the concentration of smoke and flammable gases (like LPG, methane, propane) present in the environment and gives an analog output proportional to the gas level.
- The flame sensor detects the presence of fire or flame using infrared radiation. It provides a digital output — LOW when flame is detected and HIGH when there is no flame.
- The ESP32 reads data from both sensors at regular intervals and compares the readings with predefined threshold values set in the program.

- If the gas level is below the safe threshold, the system displays “SAFE” on the LCD and keeps the buzzer turned OFF.
- When the gas level increases beyond the warning or danger threshold, the LCD shows “HIGH” or “DANGER,” and the buzzer automatically turns ON to alert nearby people.
- If the flame sensor detects a fire (LOW signal), the system immediately activates the buzzer, displays “FIRE DETECTED” on the LCD, and sends an alert message to the user via the Blynk IoT platform.
- Through Blynk, all sensor readings (gas value, flame status, and system alerts) are transmitted in real time to a smartphone or web dashboard, allowing the user to monitor the system remotely.
- The Blynk app also provides instant notifications (log events) such as “ 🔥 FIRE DETECTED” or “⚠ HIGH GAS LEVEL,” ensuring quick user response even when not present near the system.
- The system therefore provides both local alerts (buzzer, LCD) and remote alerts (IoT notifications), ensuring maximum safety and quick detection of fire or smoke hazards.

6. Output of Deigned Model/Prototype:



0



Learning outcomes (What I have learnt):

1. Gained practical knowledge of IoT-based system design using sensors, microcontrollers, and cloud platforms.
2. Learned how to interface MQ-2 gas sensor, flame sensor, LCD display, and buzzer with the ESP32 microcontroller.
3. Understood how to collect, process, and transmit real-time sensor data using the Blynk IoT platform.
4. Acquired skills in Arduino IDE programming, including sensor calibration, threshold setup, and IoT integration.
5. Developed the ability to design a system that provides both local and remote alerts for safety applications.