RESEARCH PROJECT REPORT

ON

"IOT Based Gas Detection System"

SUBMITTED TO

Fergusson College (Autonomous)

FOR THE DEGREE OF

M.Sc.

(ELECTRONIC SCIENCE)

BY

Killedar Prajakta (236903)

Estd. 1884

DEPARTMENT OF ELECTRONIC SCIENCE
FERGUSSON COLLEGE (Autonomous)
PUNE - 411004

April 2025

Contents

Introduction

An **IoT-based Gas Detection System** is a smart safety solution designed to detect the presence of harmful gases in the environment and send real-time alerts to users via the internet. This system typically uses **gas sensors** (like MQ-series sensors) to continuously monitor air quality and detect gases such as **LPG, methane, carbon monoxide**, and more.

The data from the sensors is processed by a **microcontroller** (like Arduino or ESP32), which is connected to the **Internet of Things (IoT) platform**. When dangerous gas levels are detected, the system can **trigger alarms**, **send notifications to smartphones**, and even **control ventilation systems automatically**.

This technology is especially useful in **industrial settings, kitchens, and gas storage areas**, where early detection of gas leaks can prevent accidents, save lives, and reduce property damage.

Objectives

- **To detect the presence of harmful or flammable gases** (like LPG, methane, or CO) in the environment using gas sensors.
- **To design a real-time monitoring system** that alerts users immediately upon detection of unsafe gas levels.
- **To implement IoT technology** (e.g., using ESP8266 or ESP32) for sending gas level data to a cloud platform or mobile device.
- **To improve safety and prevent accidents** by generating automatic alerts (buzzer, LED, mobile notifications) during gas leaks.
- **To create a portable, low-cost, and user-friendly system** that can be easily installed in homes, industries, or gas storage facilities.
- **To store and analyze gas level data remotely**, enabling users to track historical data for better decision-making and preventive maintenance.
- **To enhance awareness and promote early warning systems**, especially in high-risk environments, ensuring timely response and safety.

Scope of Project

- **Real-Time Monitoring:**

The system allows continuous real-time monitoring of harmful or flammable gases in residential, commercial, and industrial environments.

- **Remote Accessibility:**

Using IoT, users can access gas level data and receive alerts on their smartphones from anywhere in the world.

- **Scalability:**

The project can be easily expanded to monitor multiple gas types, additional sensors, or multiple locations using cloud integration.

- **Data Logging and Analysis:**

The system can store sensor readings on an IoT platform (e.g., ThingSpeak, Blynk), enabling users to analyze trends and make safety decisions.

- **Automation Possibility:**

It can be integrated with ventilation systems, gas valves, or exhaust fans to automatically respond to gas leaks.

- **Safety Enhancement:**

Useful for applications in kitchens, industries, chemical plants, and fuel stations where gas leaks can be dangerous or fatal.

- **Low-Cost and Energy Efficient:**

The system is affordable, consumes low power, and can run continuously with minimal maintenance.

Literature Review

The use of gas detection systems has become increasingly important in both domestic and industrial environments to prevent accidents caused by toxic or combustible gases. Traditional gas detectors are standalone systems that generate local alerts only, lacking remote monitoring or data analysis capabilities.

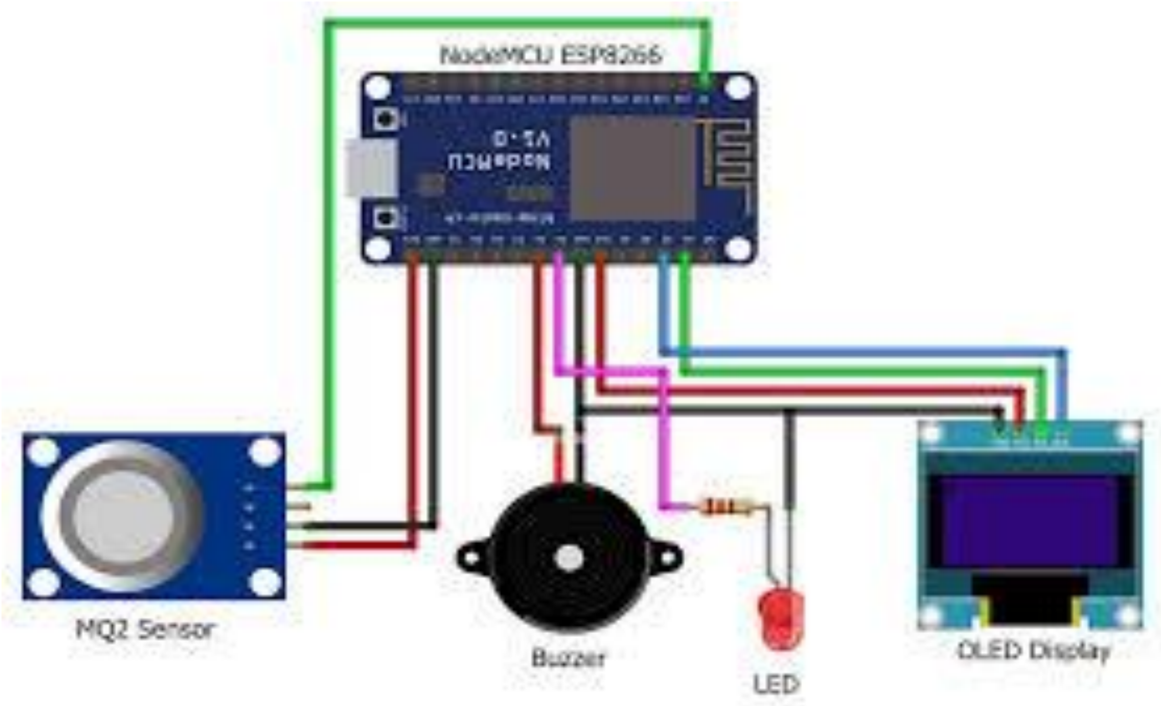
Recent advancements in **Internet of Things (IoT)** technology have enabled the development of **smart gas detection systems** that can send real-time data over the internet, enhancing both safety and user convenience. Researchers have explored various methods and platforms for integrating gas sensors with microcontrollers and IoT cloud services.

In a study by **Patil et al. (2020)**, a gas leakage detection and smart alerting system was implemented using an **MQ-2 gas sensor**, **Arduino Uno**, and **GSM module** to send SMS alerts during gas leakage. Although effective, it lacked real-time remote monitoring through the internet.

Kumar and Singh (2021) proposed an IoT-based air quality and gas monitoring system using **ESP8266** and **ThingSpeak**. This approach allowed real-time data logging and online visualization but was limited to specific gas types and required stable internet connectivity.

Sharma et al. (2022) developed a mobile app-based notification system integrated with a gas sensor and Wi-Fi module. Their work highlighted the importance of user-friendly interfaces and emphasized the need for data analytics in improving response time and predictive maintenance.

Block Diagram



➢ **Gas Sensing:**

- The system uses a gas sensor (like **MQ-2**, **MQ-135**) to detect the concentration of gases such as LPG, methane, smoke, or carbon monoxide in the air.
- The sensor gives an **analog voltage output** based on the amount of gas detected.

➢ **Signal Processing:**

- A **microcontroller** (like **ESP8266** or **Arduino**) reads this analog signal.
- The microcontroller compares the sensor value to a **predefined threshold**. If the gas concentration crosses this threshold, it means a **leak or dangerous level** is detected.

➢ **Local Alert System:**

- When gas is detected, the system **activates a buzzer or LED** to give a local alert to nearby people.
- This helps in taking immediate action to prevent accidents.

➢ **IoT Communication:**

- The microcontroller is connected to the internet via Wi-Fi.
- It **sends real-time data** (gas levels) to an **IoT platform** like **ThingSpeak**, **Blynk**, or **Firebase**.

- **Remote Alerts:**

- When a dangerous gas level is detected, the system can **send notifications to the user's smartphone** via the IoT platform.
- This ensures that the user is alerted even if they are not physically present near the device.

- **Data Logging (Optional):**

- The gas level data is stored on the cloud.
- Users can view the data in real-time or analyze historical trends for maintenance or safety analysis.

Components

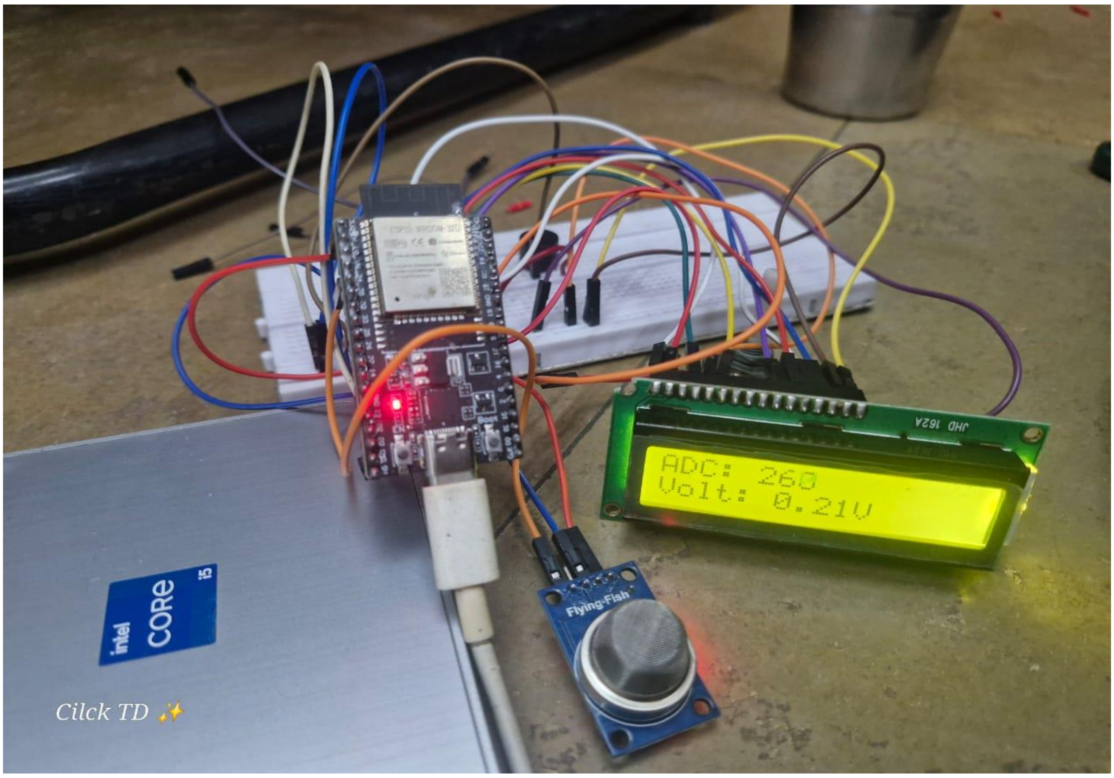
- **MQ-2 Gas Sensor** – For detecting gases like LPG, methane, and smoke.
- **ESP8266 NodeMCU / ESP32 / Arduino Uno** – Microcontroller with Wi-Fi capability for processing and sending data.
- **Buzzer** – To give a sound alert when gas is detected.
- **LED** – For visual indication of gas presence.
- **16x2 LCD / OLED Display (optional)** – To show real-time gas concentration levels.
- **Wi-Fi Module (ESP-01)** – Only needed if you're using Arduino Uno (not required with NodeMCU or ESP32).
- **Power Supply (5V Adapter / USB / Battery)** – To power the circuit.
- **Resistors (e.g., 220Ω for LED)** – For current limiting and safe operation.
- **Breadboard** – For prototyping the circuit without soldering.
- **Connecting Wires / Jumper Wires** – For making connections between components.
- **IoT Platform (e.g., Blynk / ThingSpeak / Firebase)** – For real-time data monitoring and alert notifications.

Code

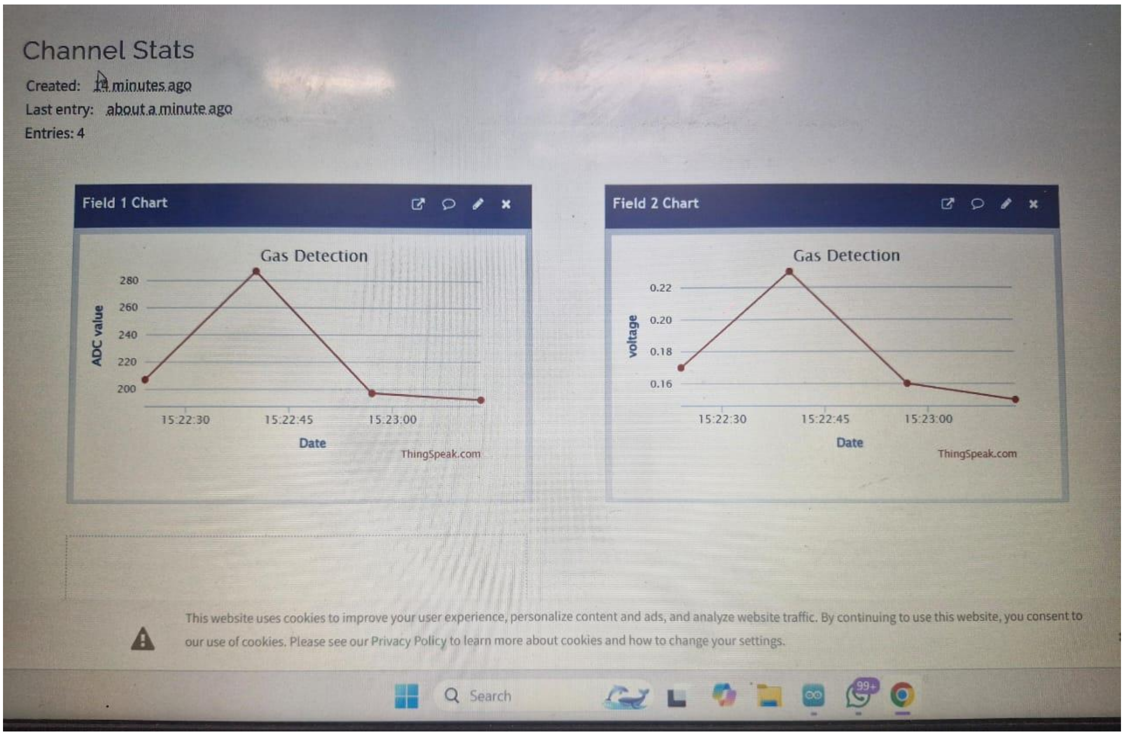
```
#include <WiFi.h>
#include <HTTPClient.h>
#include <LiquidCrystal.h>

#define WIFI_SSID "Tantrick OP"  // Replace with your Wi-Fi SSID
#define WIFI_PASS "12345678765"  // Replace with your Wi-Fi Password
#define THINGSPEAK_CHANNEL_ID "2904005"  // Replace with your ThingSpeak
Channel ID
#define THINGSPEAK_API_KEY "STA1CFBVHMZ1T5G5"  // Replace with your
ThingSpeak API Key
#define THINGSPEAK_URL "http://api.thingspeak.com/update"

// Hardware Configurations
#define MQ2_PIN 34    // Analog pin for MQ2 sensor
#define BUZZER  25    // Buzzer pin

// ESP32 ADC Configuration
#define ADC_MAX  4095   // 12-bit ADC range (0-4095)
#define VOLTAGE_REF 3.3 // ESP32 ADC reference voltage

// LCD pin configuration (RS, EN, D4, D5, D6, D7)
LiquidCrystal lcd(4, 5, 18, 19, 21, 22);

void setup() {
  Serial.begin(115200);
  pinMode(BUZZER, OUTPUT);

  // Initialize LCD
  lcd.begin(16, 2);
  lcd.print("Gas Detector");
  delay(2000);
  lcd.clear();

  // Connect to Wi-Fi
  WiFi.begin(WIFI_SSID, WIFI_PASS);
  lcd.print("Connecting...");
  Serial.print("Connecting to Wi-Fi");

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  Serial.println("\nWiFi Connected!");
  lcd.clear();
  lcd.print("WiFi Connected");
  delay(1000);
}

void loop() {
  int sensorValue = analogRead(MQ2_PIN);
  float voltage = (sensorValue / (float)ADC_MAX) * VOLTAGE_REF;  // Convert ADC
to voltage
```

```
  // Display on Serial Monitor
  Serial.print("Gas Sensor Value: ");
  Serial.print(sensorValue);
  Serial.print("  |  Voltage: ");
  Serial.print(voltage, 3); // Display voltage with 3 decimal places
  Serial.println(" V");

  // Display ADC value on LCD
  lcd.setCursor(0, 0);
  lcd.print("ADC: ");
  lcd.print(sensorValue);
  lcd.print("    ");  // Clear extra characters if value length changes

  // Display Voltage on LCD
  lcd.setCursor(0, 1);
  lcd.print("Volt: ");
  lcd.print(voltage, 2); // Show voltage with 2 decimal places
  lcd.print("V  ");

  // Gas detection alert
  if (sensorValue > 160) { // Adjust threshold based on calibration
    digitalWrite(BUZZER, HIGH);
    Serial.println("🚨 Gas Detected! Alert!");
  } else {
    digitalWrite(BUZZER, LOW);
  }

  // Send data to ThingSpeak
  if (WiFi.status() == WL_CONNECTED) {
    HTTPClient http;

    // Construct URL dynamically with Channel ID
    String url = "http://api.thingspeak.com/update";
    url += "?api_key=" + String(THINGSPEAK_API_KEY);
    url += "&field1=" + String(sensorValue);
    url += "&field2=" + String(voltage, 2);

    http.begin(url);
    int httpResponseCode = http.GET();

    Serial.print("HTTP Response Code: ");
    Serial.println(httpResponseCode);
    http.end();
  } else {
    Serial.println("Wi-Fi not connected, skipping data upload.");
  }

  delay(15000); // Send data every 15 seconds (ThingSpeak limit)
}
```

Implementation



Results on ThingSpeaks



Future Scope

- **Mobile App Integration:**

  A custom mobile application can be developed to provide real-time alerts, historical data visualization, and remote control of the system.

- **Multiple Gas Detection:**

  The system can be expanded to detect multiple gases simultaneously using additional sensors (e.g., MQ-7 for CO, MQ-135 for air quality).

- **Automatic Control Systems:**

  Integration with exhaust fans, ventilation systems, or automatic gas shut-off valves can be added for automated response to gas leaks.

- **Battery Backup and Solar Power:**

  Adding battery backup or solar power support can make the system more reliable during power failures or in remote areas.

- **AI and Predictive Maintenance:**
  Machine learning algorithms can be used to predict potential gas leaks or

- **GPS and Location Tracking:**

  Useful for mobile applications like gas delivery trucks or vehicles carrying hazardous materials, to locate the source of the leak.

- **Industrial-Grade Deployment:**

  The system can be ruggedized for use in harsh industrial environments and support multiple network protocols like LoRa or Zigbee.

- **Voice Alert System:**

  A speaker or voice module can be added for audible warnings, especially useful for visually impaired individuals.

Conclusion

The **IoT-Based Gas Detection System** successfully demonstrates how modern technology can be utilized to improve safety in environments where hazardous gases may be present. By integrating gas sensors with a microcontroller and IoT platform, the system is capable of **detecting harmful gases in real-time** and immediately alerting users both **locally** (via buzzer/LED) and **remotely** (via smartphone notifications).

This project proves to be a **cost-effective, portable, and scalable** solution suitable for homes, industries, and gas storage facilities. It not only enhances safety but also enables **remote monitoring and data logging**, which can be useful for preventive maintenance and analysis.

Overall, this system helps in **preventing accidents**, **protecting lives and property**, and promoting **smart automation** in the field of safety and environmental monitoring.

.

References

IoT based Smart Gas Monitoring System

- *Authors:* A. S. Pradeep et al.
- *Published in:* International Journal of Engineering Research & Technology (IJERT)
- *Link:* https://www.ijert.org/iot-based-smart-gas-monitoring-system
- *Abstract:* Focuses on detecting gas leaks using sensors like MQ-2 and alerts via IoT platforms.

IoT Based Gas Leakage Detection System with Blynk Application

- *Authors:* K. Rajeswari et al.
- *Published in:* International Research Journal of Engineering and Technology (IRJET)
- *Link:* https://www.irjet.net/archives/V6/i4/IRJET-V6I4252.pdf
- *Abstract:* Describes an IoT setup that uses NodeMCU, MQ sensors, and Blynk app for gas leak alerts.

Gas Leakage Detection and Smart Alerting and Prediction Using IoT

- *Authors:* Dr. P. Jayalakshmi et al.
- *Published in:* International Journal of Recent Technology and Engineering (IJRTE)
- *Link:* https://www.ijrte.org/download/volume-8-issue-2s2/
- *Focus:* Uses predictive analysis alongside sensor data for safety applications.