GAS LEAKAGE DETECTION AND PREVENTION

Project Synopsis

This IoT-based LPG gas leakage detection and prevention system is designed to improve safety in residential and commercial settings. Using an MQ2 gas sensor integrated with an ESP Wroom 32D microcontroller, the device monitors gas levels continuously. When gas concentration surpasses a specified threshold, the system activates an LED indicator and buzzer, shuts off the gas supply using a relay mechanism, and turns on an exhaust fan to help ventilate the area. Additionally, it promptly sends an email notification to the user, alerting them to the hazard. In severe cases where fire is detected, a connected fire sensor triggers a sprinkler system to extinguish the fire. This solution offers a comprehensive and efficient approach to prevent gas leakage and fire incidents, protecting both property and lives.

Master of Computer Applications

SUBMITTED BY

ROHIT DAS (12312748)

ATHAR ALI (12316128)

FAIZAN KHAN (12311404)

VISHAL ROY (12312612)

ZISHAN RAZA (12320000)

LOVELY PROFESSIONAL UNIVERSITY,

JALANDHAR

Table of contents

1. Introduction

Overview of the Project

The IoT-based LPG gas detection and leakage prevention system is designed to provide a robust and real-time solution for monitoring gas levels in residential and commercial environments. The system uses an ESP Wroom 32D microcontroller paired with an MQ2 gas sensor to detect the presence of LPG gas. When the gas concentration exceeds a predefined safety threshold, the device triggers alerts through an LED indicator and buzzer, shuts off the gas supply via a relay-controlled mechanism, and sends an email notification to the user. This ensures timely awareness and action to mitigate potential hazards. The integration of IoT technology facilitates remote monitoring, adding convenience and safety for users.

Importance of Gas Leakage Detection

Gas leakage is a critical safety issue, as undetected leaks can lead to severe consequences, including fires, explosions, and exposure to toxic gases. Such incidents pose significant risks to human life, property, and the environment. Early detection of gas leaks is essential to prevent accidents and ensure the safety of occupants. The importance of an automated gas detection system lies in its ability to respond immediately, providing alerts and preventive actions without relying solely on human intervention. By automating the detection and response process, this system aims to minimize the risk of accidents, offering a proactive safety measure that enhances overall household and workplace safety.

2. Objectives

Purpose and Goals of the System

The primary purpose of the IoT-based LPG gas detection and leakage prevention system is to safeguard lives and property by providing early detection and automatic intervention during gas leaks. The system aims to offer an efficient, real-time monitoring solution that ensures immediate response in the event of gas leakage. The main goals are to:

- Detect gas leakage promptly and accurately.
- Alert users with visual and auditory signals.
- Automatically shut off the gas supply to prevent escalation.
- Notify users via email for remote awareness and timely action.

Key Features

- **Real-time Gas Monitoring:** Continuous monitoring of LPG levels to detect any sudden rise in gas concentration.
- **Visual and Auditory Alerts**: LED indicator and buzzer are triggered to notify occupants instantly.
- **Automatic Gas Shut-off:** A relay motor controls the gas regulator to stop the gas supply when leakage is detected.
- **Email Notification:** The system sends an alert to the user's email for remote awareness.
- **User-friendly Display:** A display screen shows gas concentration levels and system status.
- **Exhaust Fan Activation:** To prevent gas buildup, the system activates an exhaust fan via a relay switch, pumping out any accumulated gas and improving indoor air safety.
- **Fire Detection and Firefighting Mechanism:** In case of a fire, a flame sensor detects the presence of flames and triggers the sprinkler system to contain and extinguish the fire, adding an extra layer of safety.

3. System Components

ESP Wroom 32D Microcontroller

The ESP Wroom 32D is a powerful and versatile microcontroller unit (MCU) with built-in Wi-Fi capabilities, making it ideal for IoT applications. It serves as the brain of the system, processing data from the gas sensor and controlling other components such as the LED, buzzer, and relay.

MQ2 Gas Sensor

The MQ2 sensor is used to detect the presence of flammable gases, including LPG. It provides real-time readings of gas concentration in PPM (parts per million), which are processed by the ESP32 to determine if the levels are safe or dangerous.

IR Flame Sensor

Detects the presence of fire, enabling the system to trigger a water sprinkler system if flames are detected. This component increases the system's effectiveness in high-risk fire scenarios.

16x2 I2C LCD Display

Displays real-time gas levels and system status updates. It shows gas concentration levels in PPM and indicates when gas leaks or fires are detected.

Servo Motor

Connected to the gas regulator, the servo motor closes the valve when a gas leak is detected, preventing further gas flow and mitigating potential danger.

LED Indicator

The LED acts as a visual alert system that turns on when a gas leak is detected, signalling the occupants of potential danger.

Buzzer

The buzzer provides an auditory alert, sounding an alarm when gas levels exceed the safety threshold. This ensures immediate awareness even if the LED signal is unnoticed.

Relay Module

The relay module controls the gas by starting on the exhaust fan and also in case of worst-case scenario like fire, it will turn on the water sprinkler system.

Power Supply

The power supply provides the necessary electrical power to run the microcontroller, sensor, and other connected components. The system typically uses a 5V power source compatible with the ESP32 and peripheral devices.

Relay Water System

In case of fire, the flame sensor triggers the relay-controlled sprinkler system, which releases water to contain the fire and prevent it from spreading.

Exhaust Fan

The exhaust fan in this project is used to ventilate the area when a gas leak is detected. The fan is controlled by a relay, which is triggered by the ESP32 microcontroller, ensuring an automated response to mitigate potential hazards effectively.

Breadboard

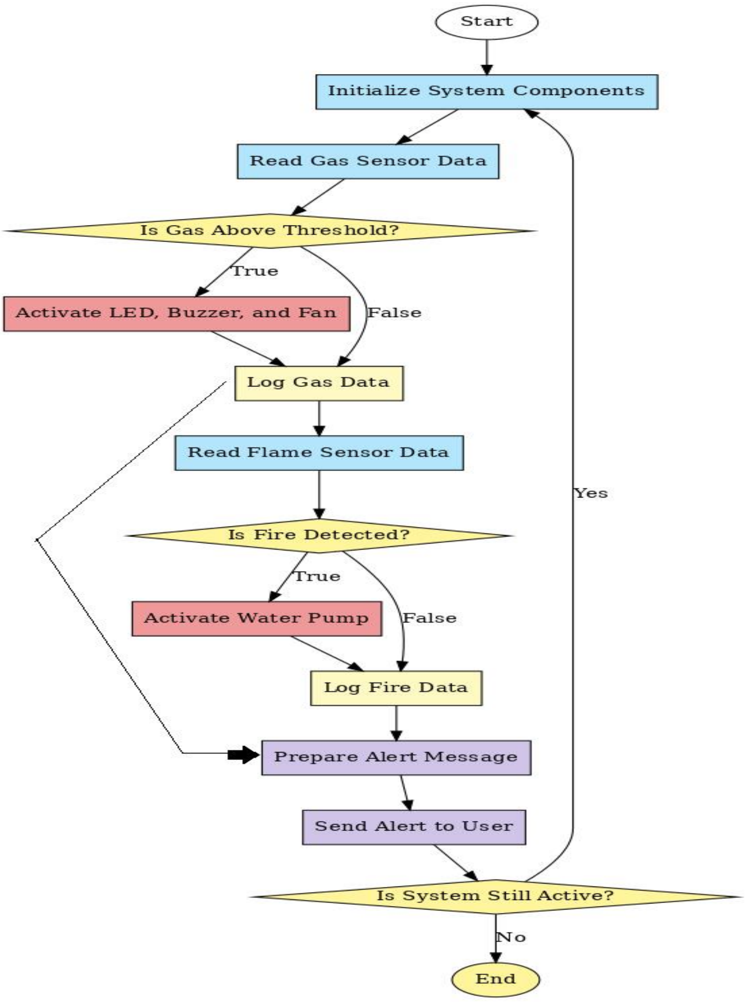
A breadboard is used in this project to facilitate easy and temporary connections between various components without the need for soldering. It allows the components such as the MQ2 gas sensor, flame sensor, ESP32, and other electronic parts to be placed and connected securely. The breadboard ensures that the components are organized and can be easily modified or tested during the development phase.

Jumper Wires

Jumper wires are used to establish connections between the components on the breadboard and the microcontroller (ESP32). These wires are flexible, easy to use, and come in various lengths, allowing for convenient and quick prototyping. They help to connect sensors, actuators, and other modules to the breadboard, ensuring a functional circuit setup for testing and deployment.

4. System Design and Architecture

Flow Diagram



Components Connections

ESP32 Microcontroller:
The main controller for all sensors and outputs, handling Wi-Fi for email alerts, and controlling various actuators.

MQ2 Gas Sensor:

- Analog Pin: Connect the analog output (A0) of the MQ2 to GPIO 34 on the ESP32 to read gas levels.

- Power: Connect VCC to the 5V pin and GND to ground.

Flame Sensor:

- Analog Pin: Connect the analog output to GPIO 35 to monitor fire levels.
- Power: Connect VCC to 3.3V or 5V, and GND to ground.

Servo Motor (Gas Valve):

- Control Pin: Connect to GPIO 14 to open/close the gas valve based on leakage detection.
- Power: Connect VCC to an external 5V supply, GND to ground.

Relay Module (Exhaust Fan and Sprinkler):

- Control Pin: Connect IN1 and IN2 (for fan and sprinkler) to GPIO 12 and any other available GPIO.
- Power: VCC to 5V, GND to ground.

LED Indicator:

- Connect the positive lead to GPIO 13 (with a resistor), and the ground lead to GND.

16x2 LCD Display:

- I2C Interface: Connect SDA and SCL to the respective pins on the ESP32 (usually GPIO 21 for SDA and GPIO 22 for SCL).

Exhaust Fan:

Controlled by one of the relay outputs, powered separately.

System Workflow

1. The ESP32 reads gas levels through the MQ2 sensor and checks for flames via the flame sensor.
2. If gas levels exceed the threshold:
    - It triggers an LED, closes the gas valve using the servo, and activates the exhaust fan to remove gas.
    - Sends an email alert to the user using SMTP.
3. If fire is detected, it triggers the sprinkler via the relay module.

5. Working Principle

Gas Detection Mechanism

The system utilizes an MQ2 gas sensor to continuously monitor the concentration of LPG and other flammable gases in the environment. The sensor outputs an analog signal, which the ESP Wroom 32D microcontroller reads and processes. If the detected gas level exceeds a predefined safety threshold, the system activates preventive measures. If levels are safe, the system remains in standby mode, displaying the current gas concentration on an LCD screen.

Alert and Notification Process

When the gas concentration surpasses the safety threshold, the system triggers an immediate response to alert occupants and prevent hazards:

- **Visual and Auditory Alerts**: An LED indicator lights up, and a buzzer sounds to warn occupants of the leak.
- **Email Notification:** The ESP32 microcontroller connects to the internet through Wi-Fi to send an email notification to the user, ensuring they are promptly informed, even if they are away from the premises.
- **Exhaust Fan Activation:** The relay module turns on an exhaust fan to vent the gas, quickly lowering the gas concentration in the affected area.

Gas Regulator Shutdown

In addition to alerting, the system uses a servo-controlled gas valve to close the gas regulator, immediately halting gas flow. This action reduces the potential for gas accumulation and minimizes the risk of ignition.

Fire Detection and Sprinkler Activation

To address worst-case scenarios, the system includes a flame sensor to detect fire. Upon detecting flames, the ESP32 activates a relay that turns on a sprinkler system to extinguish the fire. This layered approach of detection, notification, and active response provides an integrated safety solution to protect lives and property from gas-related hazards.

6. Code

```
#include <LiquidCrystal_I2C.h>
#include <ESP32Servo.h>
#include <WiFi.h>
#include <ESP_Mail_Client.h>

// Define pin numbers
const int gasSensorPin = 34;
const int fireSensorPin = 35;
const int ledPin = 13;
const int servoPin = 14;
const int sprinklerPin = 12;
const int gasThreshold = 400;

// Wi-Fi and Email settings
#define WIFI_SSID "vivo 1917"
#define WIFI_PASSWORD "aliansri1@"
#define SMTP_server "smtp.gmail.com"
#define SMTP_Port 587
#define sender_email "esp32alertmessage@gmail.com"
#define sender_password "lelshxhwwrdvpcgo"
#define Recipient_email "aaravaliansari@gmail.com"
#define Recipient_name "ESP32"

LiquidCrystal_I2C lcd(0x27, 16, 2);
Servo gasValve;
SMTPSession smtp;

int gasValue = 0;
int fireValue = 0;
bool emailSent = false;  // Flag to ensure emails aren't sent repeatedly
```

```
void setup() {
  // Initialize LCD
  lcd.init();
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print("Gas Leakage");
  lcd.setCursor(0, 1);
  lcd.print("Detection & Prev");
  delay(2000);
  lcd.clear();

  // Set pin modes
  pinMode(gasSensorPin, INPUT);
  pinMode(fireSensorPin, INPUT);
  pinMode(ledPin, OUTPUT);
  pinMode(sprinklerPin, OUTPUT);

  // Initialize servo
  gasValve.attach(servoPin);
  gasValve.write(0);

  // Turn off LED and sprinkler at the start
  digitalWrite(ledPin, LOW);
  digitalWrite(sprinklerPin, LOW);

  // Wi-Fi setup
  Serial.begin(9600);
  Serial.print("Connecting to Wi-Fi...");
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(200);
```

```
  }
  Serial.println("\nWi-Fi connected.");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());

  smtp.debug(1);
}

void sendEmail(String subject, String body) {
  // SMTP Session and message setup
  ESP_Mail_Session session;
  session.server.host_name = SMTP_server;
  session.server.port = SMTP_Port;
  session.login.email = sender_email;
  session.login.password = sender_password;
  session.login.user_domain = "";

  SMTP_Message message;
  message.sender.name = "ESP32";
  message.sender.email = sender_email;
  message.subject = subject.c_str();
  message.addRecipient(Recipient_name, Recipient_email);
  message.text.content = body.c_str();
  message.text.charSet = "us-ascii";
  message.text.transfer_encoding = Content_Transfer_Encoding::enc_7bit;

  // Connect and send email
  if (!smtp.connect(&session)) {
    Serial.println("Error connecting to SMTP server");
    return;
  }
```

```
  if (!MailClient.sendMail(&smtp, &message)) {
    Serial.println("Error sending Email: " + smtp.errorReason());
  } else {
    Serial.println("Email sent successfully.");
  }
}

void loop() {
  gasValue = analogRead(gasSensorPin);
  fireValue = analogRead(fireSensorPin);

  // Display gas value on LCD
  lcd.setCursor(0, 0);
  lcd.print("Gas Value: ");
  lcd.print(gasValue);

  if (gasValue > gasThreshold) {
    // Gas detected: turn on LED, close valve (rotate servo), display warning
    digitalWrite(ledPin, HIGH);
    gasValve.write(90);
    lcd.setCursor(0, 1);
    lcd.print("Gas Detected");

    if (!emailSent) {
      sendEmail("Gas Detected", "Warning: Gas leakage detected!");
      emailSent = true;
    }
  } else {
    // No gas: turn off LED, keep valve open (0 degrees)
    digitalWrite(ledPin, LOW);
    gasValve.write(0);
    lcd.setCursor(0, 1);
```

```
    lcd.print("No Gas          ");
  }

  // Check for fire
  if (fireValue > 500) {
    // Fire detected: turn on water sprinkler and display warning
    digitalWrite(sprinklerPin, HIGH);
    lcd.setCursor(0, 1);
    lcd.print("Fire Detected");

    if (!emailSent) {
      sendEmail("Fire Detected", "Warning: Fire detected!");
      emailSent = true;
    }
  } else {
    // No fire: turn off water sprinkler
    digitalWrite(sprinklerPin, LOW);
  }

  // Reset the email flag if neither gas nor fire is detected
  if (gasValue <= gasThreshold && fireValue <= 500) {
    emailSent = false;
  }
  delay(1000);
}
```

7. Testing and Calibration

Sensor Calibration Process

Calibrating the MQ2 gas sensor is essential for ensuring accurate detection of gas levels. The calibration process involves placing the sensor in a controlled environment with known concentrations of LPG and adjusting the baseline to determine a reliable threshold for gas detection. This step helps to account for environmental factors like humidity and temperature, which may influence sensor readings. Calibration typically includes preheating the sensor for a specific period to stabilize its response and measuring its output to set appropriate thresholds for safe and hazardous gas concentrations.

Test Results and Observations

During testing, the system was exposed to varying concentrations of LPG to verify the accuracy of the sensor's readings and the system's response time. The results showed that the sensor effectively detected gas concentrations above the safety threshold and triggered the LED, buzzer, and email notification as intended. The relay motor successfully shut off the gas supply, confirming the reliability of the shutdown mechanism. Observations indicated that the system responded promptly, with minimal false alarms after the threshold was fine-tuned. The email notification was consistently sent within seconds of detection, demonstrating effective integration with the SMTP client.

8. Challenges and Solutions

Potential Issues

- **Sensor Calibration**: Ensuring accurate readings from the MQ2 gas sensor can be challenging, as environmental factors such as humidity and temperature can affect sensor accuracy.
- **Wi-Fi Connectivity**: Maintaining a stable internet connection to send notifications may be difficult in areas with poor connectivity.
- **Power Supply Consistency**: Power fluctuations or outages could disrupt the system's operation, compromising its reliability.
- **False Positives**: High sensitivity of the sensor may trigger alarms for non-hazardous levels of gas, causing unnecessary alerts.

Solutions Implemented

- **Calibration Process**: The sensor was calibrated in various conditions to establish a reliable threshold that accounts for typical environmental variances.
- **Fallback Mechanisms**: The system includes local alerts (LED and buzzer) as a backup in case email notifications cannot be sent due to connectivity issues.
- **Power Backup**: An uninterruptible power supply (UPS) was integrated to maintain system operation during power interruptions.
- **Adjustable Thresholds**: The threshold for gas detection was fine-tuned to reduce false alarms without compromising safety.

9. Conclusion

Summary of Project

This IoT-based LPG Gas Detection and Leakage Prevention System is designed to provide comprehensive safety measures for gas leak prevention and fire mitigation in residential and commercial settings. The system utilizes an MQ2 gas sensor to monitor the concentration of LPG and other flammable gases, connected to an ESP Wroom 32D microcontroller for real-time data processing. When gas levels exceed a predefined threshold, the system promptly alerts users through a visual LED indicator, an auditory buzzer, and an email notification, ensuring users are informed even if they are offsite.

In addition to notifications, the system activates a relay-controlled exhaust fan to vent out the gas and prevent further gas accumulation. The servo-controlled gas valve automatically shuts off the gas regulator, minimizing the risk of ignition. For extreme cases involving fire, a flame sensor triggers the sprinkler system to contain and extinguish the fire. By combining gas detection, alerting, and active response measures, this system offers an efficient, multi-layered approach to preventing gas-related accidents, ensuring user safety and property protection.

Future Enhancements

- **Mobile Application:** Developing a mobile app for more interactive notifications and control.

- **Battery-Powered Backup:** Implementing a rechargeable battery to keep the system functional during prolonged power outages.

- **Integration with Smart Home Systems:** Expanding the system to interact with other IoT devices, such as smart thermostats or ventilation systems.

- **Multi-Gas Detection:** Upgrading the sensor array to detect other hazardous gases, such as carbon monoxide.

10. References

Source Materials and Libraries Used

- **ESP32 Documentation:** Technical resources from Espressif for the ESP Wroom 32D.

- **MQ2 Sensor Datasheet:** Calibration and usage details for the MQ2 gas sensor.

- **Arduino IDE Libraries:**
    - WiFi.h for Wi-Fi connectivity.
    - SMTPClient.h for email notification.
    - Adafruit_SSD1306.h for display integration.

- **Online Tutorials and Articles:** Guides on setting up IoT projects and configuring SMTP for ESP32.

- **YouTube**

- **Wikipedia**
  https://en.wikipedia.org/wiki/Flame_detector
  https://en.wikipedia.org/wiki/Gas_detector
- Mahalingam, A.; Naayagi, R.T.; Mastorakis, N.E. Design and implementation of an economic gas leakage detector. In Proceedings of 6th International Conference on Circuits, Systems and Signals, Athens, Greece, 7–9 March 2012; pp. 20–24.
- Attia, H.A.; Halah, Y.A. Electronic Design of Liquefied Petroleum Gas Leakage Monitoring, Alarm, and Protection System Based on Discrete Components. Int. J. Appl. Eng. Res. 2016, 11, 9721–9726.
- Apeh, S.T.; Erameh, K.B.; Iruansi, U. Design and Development of Kitchen Gas Leakage Detection and Automatic Gas Shut off System. J. Emerg. Trends Eng. Appl. Sci. 2014, 5, 222–228.