**1. Background**

Liquefied Petroleum Gas (LPG) is a highly flammable fuel commonly used in households, industries, and commercial settings for cooking, heating, and other energy applications. While LPG is efficient and convenient, leaks pose severe hazards, including fire outbreaks, explosions, and health risks due to inhalation of toxic gases. According to safety reports, gas leaks are a leading cause of domestic and industrial accidents, often resulting in loss of life and property.

**2. Problem Statement**

One of the major challenges associated with LPG usage is the difficulty in detecting leaks at an early stage, especially in confined spaces where gas accumulation can lead to catastrophic consequences. Since LPG is heavier than air, it tends to settle in low-lying areas, making detection by human senses unreliable. In many cases, leaks are only noticed when the gas concentration is high enough to cause health issues or an explosion risk. Traditional detection methods, such as manual checks and odor detection, are insufficient and prone to human error. Therefore, an automated gas leak detection system is necessary to enhance safety and minimize potential hazards.

**3. Project Objective**

This project aims to design and develop an LPG gas leak detection system that automatically detects gas leaks and triggers an alarm to alert users. The system will integrate a **gas sensor** (such as an MQ-2 or MQ-5 sensor) with a **microcontroller-based circuit** to continuously monitor the surrounding air for LPG concentrations. If the gas level exceeds a predefined threshold, the system will activate an **alarm** (buzzer, LED indicator, or a connected alert system) to warn users of the leak.

**4. Significance of the Project**

The implementation of this system will provide several benefits, including:

* **Enhanced Safety:** Early detection reduces the risk of explosions and fire accidents.
* **Automation and Reliability:** Unlike manual detection, the system works continuously and provides instant alerts.
* **Cost-Effective Solution:** The design is affordable and can be integrated into homes and industries.
* **Scalability:** The system can be upgraded with additional features, such as SMS notifications or automatic gas shut-off mechanisms, for better functionality.

By developing this gas leak detection system, this project aims to contribute to **household and industrial safety** while addressing a critical need for **early warning systems** in gas-powered environments.

### **Overview of Gas Leak Detection**

Gas leak detection is important for preventing accidents caused by flammable gases like LPG. Over time, different methods have been used to detect leaks, ranging from **manual methods** (such as smelling the gas) to **electronic sensors** that detect gas automatically. Newer technologies, like **wireless and IoT-based detection systems**, allow remote monitoring and alerts.

### **Choosing the Right Gas Sensor**

For LPG leak detection, **MQ-series sensors** are commonly used because they are affordable and effective. Some examples include:

* **MQ-2 Sensor:** Detects LPG, methane, and smoke.
* **MQ-5 Sensor:** Detects LPG and natural gas with higher accuracy.
* **MQ-6 Sensor:** Highly sensitive to LPG and provides faster response.

## **Hardware Components and Their Functions**

### **1. Gas Sensor (MQ-6 or MQ-2)**

#### ****Function:****

The gas sensor is the core component that detects **LPG (Liquefied Petroleum Gas) leaks** in the surrounding air. It continuously monitors gas concentration and sends a signal when the level exceeds a preset threshold.

#### ****Why This Sensor?****

* **High Sensitivity:** MQ-6 and MQ-2 sensors are highly sensitive to LPG.
* **Fast Response Time:** They quickly detect gas leaks, ensuring early warning.
* **Affordable and Widely Available:** Cost-effective and easy to integrate with microcontrollers.

### **2. Microcontroller (Arduino Uno)**

#### ****Function:****

The microcontroller acts as the **brain** of the system. It receives signals from the gas sensor, processes the data, and triggers the alarm if a gas leak is detected.

#### ****Why This Microcontroller?****

* **Arduino Uno:** Chosen for **ease of programming, reliability, and affordability**.

### **3. Buzzer (Alarm System)**

#### ****Function:****

The buzzer **produces a loud sound** when a gas leak is detected to alert users.

#### ****Why This Buzzer?****

* **Immediate Warning:** Provides an instant audio alert for safety.

### **4. LED Indicator**

#### ****Function:****

The LED provides a **visual alert** when the system detects a gas leak.

#### ****Why This LED?****

* **Visual Confirmation:** Alerts users in case they don’t hear the buzzer.
* **Different Colors Possible:** Green for safe, Red for danger.

### **5. Resistors and Connecting Wires**

#### ****Function:****

* **Resistors:** Protect components by controlling the flow of electrical current.
* **Connecting Wires:** Establish electrical connections between the components.

#### ****Why These Components?****

* **Essential for Circuit Stability:** Prevents damage to the microcontroller and sensors.
* **Ensures Proper Functioning:** Helps in making secure connections.

### **Summary**

| **Component** | **Function** | **Reason for Selection** |
| --- | --- | --- |
| **MQ-2 Gas Sensor** | Detects LPG leaks | High sensitivity, fast response, affordable |
| **Microcontroller (Arduino)** | Processes sensor data, triggers alarm | Reliable, easy to program, supports automation |
| **Buzzer** | Alerts users with sound | Immediate warning, low power consumption |
| **LED Indicator** | Visual warning | Extra alert system for safety |
| **Resistors & Wires** | Protects components and connects them | Prevents circuit damage, ensures stability |

## **5. Software Design**

### **5.1 Programming Environment**

The gas leak detection system was programmed using **Arduino IDE** and simulated on **Wokwi**, an online microcontroller simulation platform.

### **5.2 Code Functionality**

The system continuously monitors the gas concentration using the **MQ gas sensor**. If the gas level exceeds the predefined safety threshold, the system triggers an alarm by activating a **buzzer** and **red LED**. A **green LED** remains on when gas levels are normal.

Additionally, a **manual reset button** is included to silence the alarm temporarily for **30 seconds**, after which the system automatically reactivates if gas levels remain unsafe.

### **5.3 Key Features**

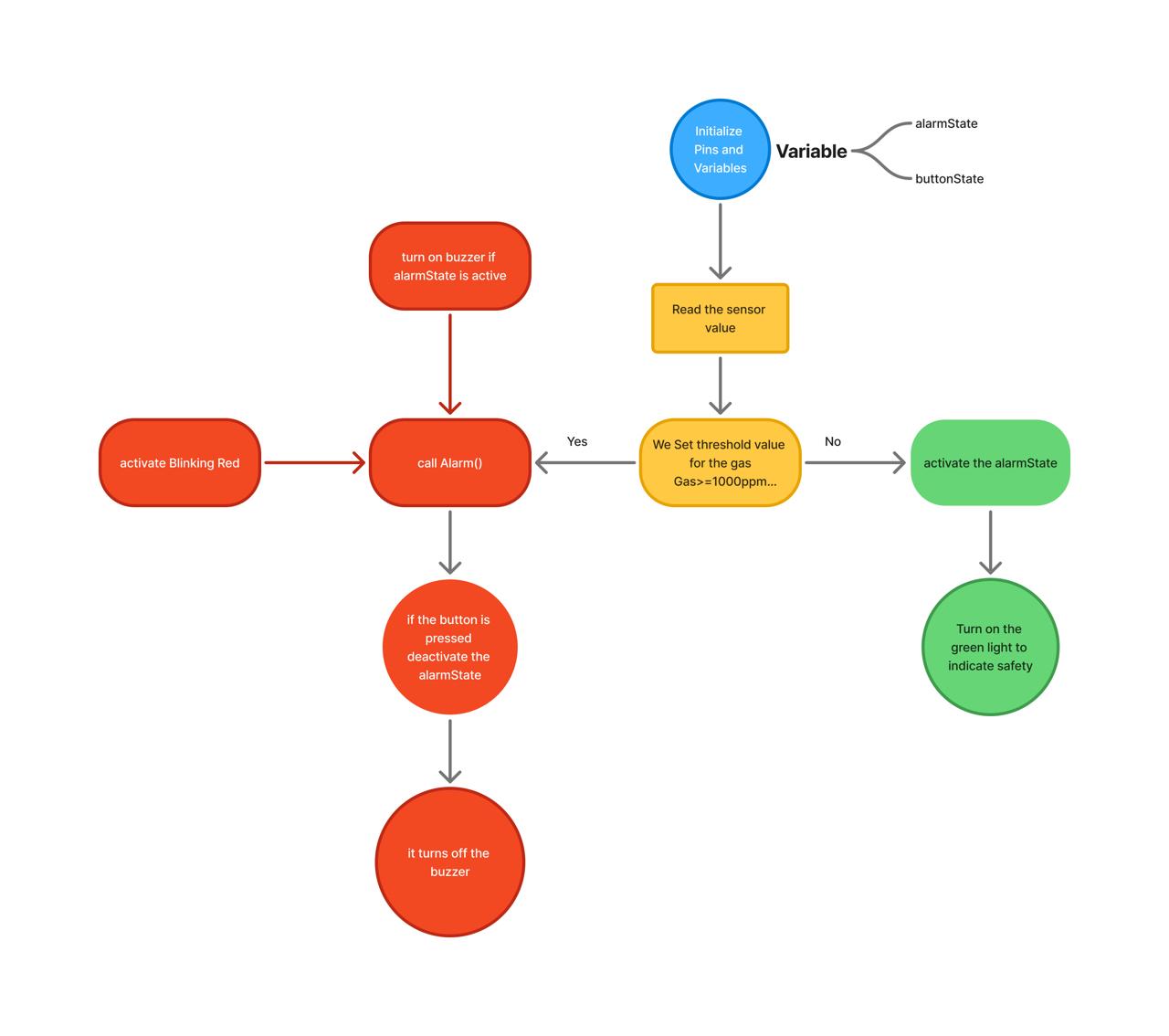
* **Real-time gas monitoring:** The microcontroller reads sensor values continuously.
* **Alarm system:** The buzzer and red LED are activated when gas levels exceed **940mV (1000ppm)**.
* **Manual alarm deactivation:** A button press silences the alarm for **30 seconds** before reactivation.
* **Visual safety indication:** A green LED remains on under safe conditions.

### **5.4 Code Implementation**

The program follows these logical steps:

1. **Initialize components:** Gas sensor, LEDs, buzzer, and button are set up.
2. **Read gas levels:** The system continuously checks the sensor’s output.
3. **Compare with threshold:** If gas exceeds **940mV**, the alarm is triggered.
4. **Manual reset option:** If the button is pressed, the alarm turns off for **30 seconds**.
5. **Restore monitoring:** After **30 seconds**, if gas levels remain high, the alarm reactivates.

### **5.5 Flowchart Representation**



### **5.6 Code Explanation**

* **Gas Sensor Reading:**
* int LPG = analogRead(MQ);

Reads the sensor’s analog value.

* **Threshold Check & Alarm Activation:**
* if (LPG >= 940) {
* digitalWrite(GLED, LOW);
* Alarm();
* }

If the gas level exceeds the threshold, the system turns off the green LED and triggers the alarm function.

* **Manual Deactivation with Timer:**
* if (buttonstate == 0) {
* alarmState = 0;
* startTime = millis();
* }
* if (alarmState == 0 && (millis() - startTime >= deactivationTime)){
* alarmState = 1;
* }

If the button is pressed, the alarm is disabled for **30 seconds**, after which it reactivates.

* **Alarm Function:**
* void Alarm() {
* digitalWrite(RLED, HIGH);
* if (alarmState == 1) {
* tone(BUZZER, 400);
* }
* delay(300);
* digitalWrite(RLED, LOW);
* noTone(BUZZER);
* delay(300);
* }

The alarm function blinks the red LED and activates a **400 Hz buzzer sound** to alert users.

### **5.7 Libraries and Tools Used**

* **Arduino IDE** – for writing and uploading the program.
* **Wokwi Simulator** – for testing the microcontroller-based system.

## **6. Working Principle**

The gas leak detection system is designed to continuously monitor air quality for **LPG (Liquefied Petroleum Gas) leaks**. It operates based on the following principles:

### **6.1 Gas Detection Process**

1. The **MQ gas sensor** measures the concentration of gas in the surrounding environment and outputs an **analog voltage signal** corresponding to the gas level.
2. The **Arduino microcontroller** reads this signal and compares it with a predefined **threshold value (940mV, equivalent to 1000ppm)**.
3. If the gas concentration remains **below** this threshold, the system considers the environment **safe**, and a **green LED** stays ON.

### **6.2 Alarm Activation**

1. If the **gas level exceeds the threshold**, the system enters **alert mode**:
   * The **green LED turns OFF**.
   * The **red LED starts blinking** to indicate danger.
   * A **buzzer sounds an alarm at 400Hz** to warn of a leak.

### **6.3 Manual Alarm Deactivation**

1. A **manual reset button** is included to allow users to temporarily silence the alarm:
   * If the button is pressed, the **alarm is deactivated for 30 seconds**.
   * After **30 seconds**, if gas levels are still high, the alarm will automatically reactivate.

### **6.4 System Recovery & Safety Indication**

1. If gas levels return to normal (**below the threshold**), the system **resets**:
   * The **red LED turns OFF**.
   * The **buzzer stops sounding**.
   * The **green LED turns ON** to indicate safety.

## **7. Challenges and Solutions**

During the development of the gas leak detection system, several challenges were encountered. Below are the key issues and their solutions:

### **7.1 False Alarms Due to Noise or Sensitivity**

* **Challenge:** The system sometimes triggered false alarms due to minor fluctuations in sensor readings.
* **Solution:** A small delay and averaging method were added in the code to filter out minor fluctuations.

### **7.2 Alarm Reset Implementation**

* **Challenge:** The manual alarm reset button initially caused unexpected behavior, sometimes failing to reset the alarm properly.
* **Solution:** A **30-second timer** was introduced, ensuring that the alarm would not restart immediately after deactivation.

## **8. Conclusion and Future Enhancements**

### **8.1 Conclusion**

The gas leak detection system successfully detects **LPG leaks** and triggers an **alarm system** to warn users. The project was implemented using an **MQ gas sensor, Arduino microcontroller, buzzer, and LED indicators**. Testing confirmed that the system effectively responds to **dangerous gas levels**, ensuring safety in homes, industries, and commercial spaces.

### **8.2 Future Enhancements**

To improve the system, the following enhancements can be considered:

1. **GSM Module for Remote Alerts** – Adding a GSM module to send **SMS alerts** to users when a gas leak is detected.
2. **Wi-Fi/IoT Integration** – Connecting the system to the internet for **real-time monitoring** via a mobile app or web dashboard.
3. **Automatic Gas Shut-off** – Integrating an **electronic gas valve** to automatically shut off the gas supply when a leak is detected.
4. **Battery Backup** – Ensuring the system remains functional during **power outages** by adding a rechargeable battery.
5. **Improved Calibration** – Implementing machine learning techniques to further reduce **false alarms** and increase sensor accuracy.