**Project Report**

**on**

IoT-Based Gas Leakage Detection System  
 USING UBIDOTS FOR ALERT

in partial fulfilment for the award of the degree of

**BACHELOR OF ENGINEERING**

IN

**CSE (AI&ML)**

**Submitted by:**

**24BAI70069**

**24BAI70180**

**24BAI70111**

**24BAI70094**

**Under the Guidance of**

**Mr. KHUSHAL THAKUR SIR**

**ACADEMIC UNIT-I**

****

**Chandigarh University**

**17-April -2025**

**Index**

|  |  |  |
| --- | --- | --- |
| **S. No.** | **Content** | **Page No.** |
| 1 | Project Overview | 01 |
| 2 | Objective and Problem Statement | 02 |
| 3 | Proposed Solution & Methodology | 03 |
| 4 | Key Findings / Results | 04 |
| 5 | Conclusion & Learnings | 05 |
| 6 | References | 06 |
| 7 | Appendix (if required) | - |

# 1. Project Overview:

# This project presents the design and implementation of a Gas Leakage Detection System that leverages the ESP32 microcontroller, MQ-6 gas sensor, and cloud integration through the Ubidots platform. The primary aim is to enhance safety in residential, commercial, and industrial spaces by detecting harmful gas concentrations at an early stage and alerting users before dangerous levels are reached. Gas leaks can be extremely hazardous, potentially leading to fire outbreaks, explosions, and health hazards. Manual detection is not always reliable, especially during sleeping hours or when occupants are away from the premises.

# By combining real-time gas monitoring with Internet of Things (IoT) capabilities, this system not only activates local alerts through a buzzer and LED when a threshold is crossed but also uploads gas levels to the Ubidots cloud. This enables users to monitor gas concentration remotely through their smartphone or computer via Ubidots dashboards. Additionally, the system displays real-time values on a 16x2 LCD to keep local users informed. Overall, this project demonstrates how a low-cost microcontroller and a few electronic components can be used to build an intelligent and reliable gas leakage detection solution suitable for smart homes and industrial IoT safety applications.

# 2. Objective and Problem Statement:

The objective of this project is to design and develop an IoT-based gas leakage detection system that can detect and alert users about dangerous levels of gas in the air. Gas leaks are a major safety concern in households and industries due to the risk of explosions, fire, or poisoning caused by gases like LPG, methane, and propane. The key issue addressed here is the delay or absence of alerts in traditional setups that depend solely on human sensing or periodic checks.

Our aim is to automate the gas monitoring process using sensors, process the data using a microcontroller (ESP32), and provide both local (buzzer and LED) and remote (cloud-based) alerts using the Ubidots IoT platform. This ensures 24/7 monitoring and instant notification in case of a leak, even when users are away. We also aim to implement a real-time display using a 16x2 LCD for local monitoring. By addressing this problem with a smart system, we contribute to increasing safety, preventing loss, and building awareness around using IoT technologies in everyday safety-critical applications. Furthermore, the system is scalable and can be integrated with mobile notifications or automated emergency services in the future.

# 3. Proposed Solution & Methodology

The proposed solution involves the use of an ESP32 microcontroller as the brain of the system, which collects data from an MQ-6 gas sensor. This sensor can detect flammable gases such as LPG, methane, and propane. The analog output of the sensor is connected to one of the ADC pins of the ESP32. A 16x2 I2C LCD is used to continuously display the gas concentration level in raw ADC values, which can be calibrated into ppm (parts per million) in advanced setups. A buzzer and an LED are triggered when the gas level exceeds a preset threshold, alerting the user locally about a potential leak.

The ESP32 also connects to Wi-Fi and sends the gas level data to the Ubidots cloud platform using MQTT protocol. The Ubidots dashboard displays real-time and historical graphs for gas concentration, and users can set alerts or automation rules (e.g., send an SMS or email when gas is detected). In addition, if the gas level goes significantly beyond the safe limit, automated alerts such as SMS messages, emails, or even phone calls can be sent to the user to ensure rapid awareness and response. These alerts can be configured using Ubidots’ event-based trigger system or

The system is powered by a USB supply or battery and is coded using Arduino IDE with libraries such as WiFi.h, UbidotsESPMQTT.h, and LiquidCrystal\_I2C.h. The methodology ensures real-time monitoring, dual-alerting mechanisms (local and remote), and cloud connectivity for intelligent data visualization and analysis.

Components Used:  
- ESP32  
- MQ-6 Gas Sensor  
- 16x2 I2C LCD  
- Buzzer  
- LED  
- Ubidots  
  
Methodology:  
1. Interface MQ-6 with ESP32 for analog gas readings.  
2. Use LCD to display gas values.  
3. Use buzzer and LED to alert when gas exceeds threshold.  
4. Connect ESP32 to Wi-Fi and send real-time data to Ubidots cloud.  
5. Visualize and monitor using Ubidots dashboard.

# 4. Key Findings / Results:

After extensive testing and calibration, the Gas Leakage Detection System proved to be highly responsive and accurate in detecting flammable gases. When the gas level detected by the MQ-6 sensor exceeded the defined safety threshold, the buzzer and LED were immediately activated. Simultaneously, the gas concentration values were successfully transmitted to the Ubidots cloud platform where they were displayed in real time. The integration with Ubidots enabled remote monitoring of the system from any internet-enabled device through a custom dashboard. Users could view the live gas levels as well as historical trends in the form of line graphs and set custom alerts.

The 16x2 LCD displayed the gas value continuously, making it easy to assess the air quality locally. Tuning the gas threshold and ensuring proper pre-heating of the MQ-6 sensor were essential steps to ensure consistent performance. The system remained stable under different conditions and was sensitive to real gas leaks simulated with lighter gas during testing. This indicates that the device is effective and practical for real-life use in kitchens, labs, or gas-powered equipment zones.

# 5. Conclusion & Learnings

In conclusion, this project successfully demonstrates a low-cost, efficient, and real-time gas leakage detection system using ESP32 and the MQ-6 sensor with cloud support from Ubidots. The solution offers immediate local alerts and remote monitoring capabilities, addressing the crucial issue of gas leak detection in homes and industries. By combining microcontroller programming, sensor integration, and cloud services, we created a system that is both reliable and easy to scale.

The major learning outcomes from this project include understanding sensor calibration, real-time data processing, integration with cloud platforms like Ubidots, and designing user-friendly interfaces through LCDs and buzzer/LED indicators. We also learned how to use the MQTT protocol to publish data to IoT dashboards and the importance of choosing correct thresholds. Future improvements may include adding GSM modules for SMS alerts, using a mobile app for instant notifications, or integrating machine learning to predict hazardous trends. Overall, the project emphasizes how embedded systems and IoT can be combined to create impactful safety solutions for real-world problems.

Future Scope:  
- Add SMS/Call alert via GSM  
- Enable mobile app notifications  
- Add temperature sensing

# 6. References

1. https://www.electronicwings.com - For MQ-6 sensor interfacing and circuit diagrams
2. https://docs.ubidots.com - For Ubidots platform documentation and MQTT setup
3. MQ-6 Sensor Datasheet - Technical specifications and operating conditions
4. Arduino Reference - Syntax and library documentation for ESP32
5. ESP32 Pinout Reference - GPIO pin capabilities and ADC configurations
6. https://randomnerdtutorials.com - Tutorials on ESP32 and cloud integration
7. https://components101.com - Sensor breakdown and characteristics