Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 6, No.1, June : 2024

# SMART HOME GAS LEVEL MONITORING SYSTEM WITH ESP32 AND BLYNK INTEGRATION

**Dr. S. Mahendran Associate professor & HOD,** Er. Perumal Manimekalai College of Engineering, Hosur. drmahendrans@gmail.com
**Dr. N. Neeladevi Associate professor,** Er. Perumal Manimekalai College of Engineering, Hosur
**Keerthana. B, Pavithra. M, Megala. M,** Department of Electrical and Electronics Engineering**,** Er. Perumal Manimekalai College of Engineering, Hosur

## ABSTRACT
In this project, we are going to make an IOT gas detection system in which the MQ-2 gas sensor will sense if there is gas nearby. If gas is nearby, then the buzzer will start beeping and the Red LED will light up and a warning will be displayed on a web page which we will create using The ESP32 module. This web page will be accessible using any connected device like a mobile, Tablet or PC.
How Does the Gas Alarm Works:
The MQ-2 gas sensor will give the output in the form of analog voltage. We have set a condition in our code that if the output value of the sensor is greater than 400, then the buzzer will start to Beep and the red LED will light up and if the output value of the sensor is less than 600, then the Buzzer will remain quiet and the red LED will light up. The ESP used here will create a web pageAt an IP address and will send the data to this IP address and will print the data there. After uploading the code, this IP address can be seen in the serial monitor as shown below**.**
**Keywords**: IOT, Gas detecting, ESP32, MQ2 gas sensor, led, LCD display, Arduino software.

## 1. INTRODUCTION TO SMART HOME GAS LEVEL MONITORING SYSTEM:
The increasing prevalence of fire hazards in residential, commercial, and industrial environments necessitates advanced and reliable fire detection systems. Traditional gas detectors, while effective in alerting occupants to the presence of gas, often suffer from limitations such as delayed response times and the inability to notify users remotely. These limitations can lead to significant property damage and, in the worst cases, loss of life.
**Sensor Technology**: Gas monitoring systems employ various sensor technologies such as mq2 gas sensors to detect the occupancy of leakage detection.
**Connectivity:** The ESP32 connects to a local Wi-Fi network, enabling it to send sensor data to a cloud-based IOT platform. This real-time data transmission allows for constant monitoring of environmental conditions.
Upon detecting gas or unusual temperature changes, the ESP32 triggers an alarm and sends an alert to the cloud platform.
**Mobile Applications:** The cloud platform interfaces with a dedicated mobile application, providing users with instant notification about potential fire hazards. Users can receive alerts on their smart phones, even when they are away from the monitored premises, ensuring timely responses to emergencies.
The following components are used to create hardware module.

- ESP32
- LCD display
- MQ2 gas sensor
- Buzzer
- Breadboard
- Red LED
- 5V DC
- Flame sensor

## 2. Scope of the method:

- Gas Detection and Monitoring: Utilize gas sensors to detect gas levels in real-time and continuously monitor the environment.
- Real-time Alerts: Send immediate alerts to users through multiple channels (SMS, email, mobile app notifications) when gas level is detected.
- Remote Access and Control: Allow users to monitor and control the gas detector remotely via a mobile app or web interface.
- Data Logging and Analytics: Low gas levels and incidents to a cloud platform for historical analysis and predictive maintenance.
- Integration with Smart Home Systems: Integrate the gas level detector with existing smart home systems for automated responses (e.g., turning off HVAC systems, unlocking doors).
- User-Friendly Interface: Provide an intuitive interface for users to interact with the system, receive notifications, and view data logs.

## 3. The Flow of the System:

The flow of Smart home gas level monitoring system with ESP32 typically involves several stages, including monitoring gas levels, processing data, and communicating with users and operators. Here's a Step by step flow of the system

**1. Initialization**

Setup: Initialize the system by configuring pins, connecting to Wi-Fi, and setting up necessary variables.

**2. Continuous Monitoring**

Loop: Enter the main loop of the program where the system continuously monitors the environment for gas.

**3. Gas level Detection**

Read Sensor: Read data from the gas sensor connected to the ESP32 analog input pin.
Threshold Check: Compare the sensor reading with a predefined threshold to determine if gas is present.

**4. Alert Mechanism**

Trigger Alarm: If gas is detected (sensor reading exceeds the threshold), trigger the alarm mechanisms.
Activate Buzzer: Turn on the buzzer to produce an audible alarm sound.
Illuminate LEDs: Light up LEDs for visual indication of the alarm.

**5. Notification**

Send Alert: Send alerts to notify users about the detected gas level.
Connect to Wi-Fi: Ensure the ESP32 is connected to the Wi-Fi network.
HTTP Request: Send an HTTP POST request to a server or cloud platform with information about the gas detection event.
Optional: Send notifications via SMS, email, or mobile app alerts.

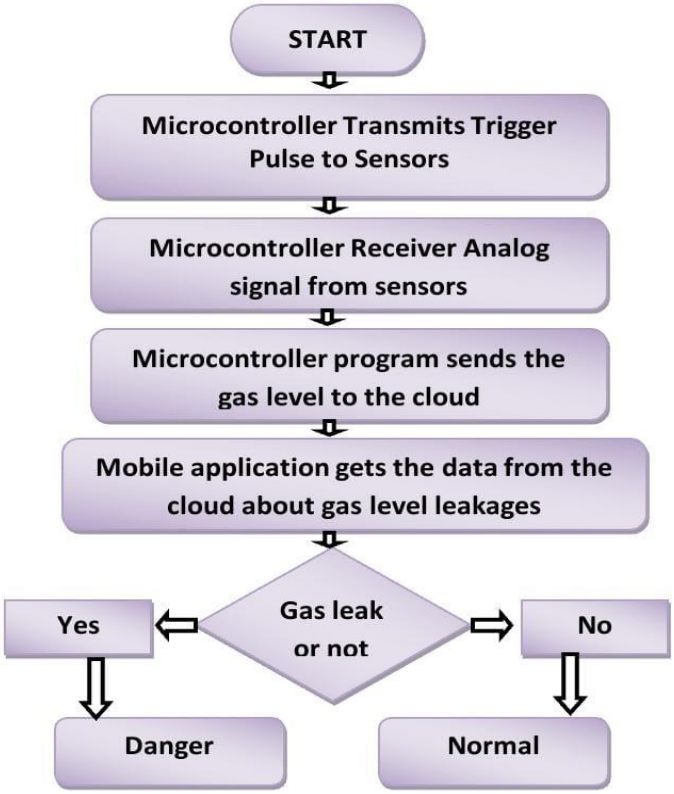
**6. Reset Mechanism**

Reset: Once the gas clears and the sensor reading falls below the threshold, reset the alarm mechanisms.
Deactivate Buzzer: Turn off the buzzer**.**
Turn Off LEDs: Switch off the LEDs.

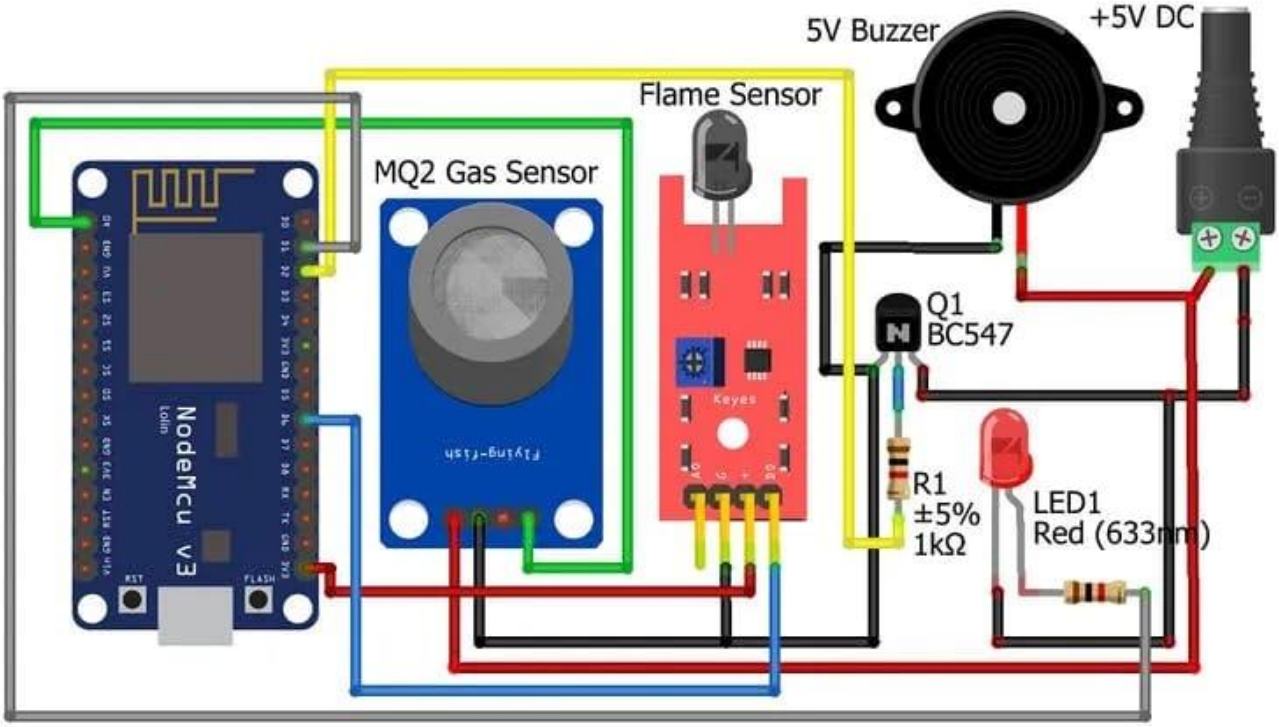
**7. Continuous Operation**

Repeat: Continue monitoring the environment for gas by looping back to the monitoring stage has grown.

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 6, No.1, June : 2024




## 4. The Block Diagram of the System




**1.** Initialization: When powered on, the ESP32 initializes its components and connects to the Wi-Fi network**.**

**2.** Monitoring: The ESP32 continuously reads the analog value from the gas sensor.

**3.** Detection: If the sensor value exceeds a predefined threshold, indicating the presence of gas.

- The ESP32 activates the buzzer and LED.

- The ESP32 sends a notification via Wi-Fi to designated server or device.

**4.** Alerting: The buzzer sounds an alarm, and the LED provides a visual indication to alert gas is nearby.

**5.** Notification: A remote alert is sent, providing information about the gas level and potentially the location of the detector.

## 5. CONCLUSION

Creating a gas level detector alarm system using the ESP32 microcontroller offers several advantages and demonstrates the integration of IOT technology into everyday safety devices. The system can be customized to meet specific needs, such as adjusting the gas sensitivity threshold or adding additional features like temperature monitoring. Implementing a gas level detector alarm using the ESP32 microcontroller is an impactful project that combines hardware and software to create a smart safety device. This project not only enhances safety through real-time alerts and monitoring but also provides valuable learning opportunities in the field of IOT and embedded systems. By leveraging the capabilities of the ESP32, users can create a highly effective and adaptable gas detection system tailored to their specific needs.

## 6. SUGGESTIONS

Regularly calibrate the gas sensor to ensure accurate detection. Implement a mechanism to adjust the sensitivity threshold, either through software (e.g., a web interface) or hardware. By incorporating these suggestions, you can enhance the functionality, reliability, and user experience of your gas detector alarm system using the ESP32, making it a more robust and versatile safety device.

## 7. FUTURE SCOPE

This appliance is a basic step for attaining an active result in regular interest.

**We can develop this project in many ways by:**

1) By providing a central management system that confirms only valid instruction is directed to the client that is handling the Security issues.
2) Also, inquiries are often done using the previous gas detecting data by which users can get instructions or ideas on gas leakage levels and their opportunities in upcoming days.
3) According to this analysis can be used detecting a gas levels by a user.
4) We can create a mobile application that will be useful for peoples to know the gas levels and leakages for particular industries and homes.

## 8. REFERENCES

1. Li,W.;Zhang,W.sensor selection for improving accuracy of target localization in wireless visual sensor networks.IET Wireless.Sens.Syst.2012,2,293-301.
2. Pagano,S, Peirani,S, Valle,M indoor ranging and localisation algorithm based on received signal strength indicator using statistical parameters for wireless sensor networks' Wirel.Sens.Syst.2015,5,243-249.
3. Rathore,M.M. Ahmad,A. Paul,A. Wan,J. Zhang,D.Real,D. Real-time response system: Exploiting IOT and big data for public health.J.Med.Syst.2016,40,283.
4. Rathore,M.M. Paul,A. Ahmad, Anisetti,M. Jeon,G.Hadoop-based intelligent care system. Analytical approach for big data in iot.ACM trans.inf.technol.2017, 18,8.
5. Blake,D.Aircraft cargo compartment gas detector Alarm incidents on us-registered aircraft,19741999;aviation administration: Washington,DC,USA,2000.
6. Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2015). Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. IEEE Communications surveys & Tutorials, 17(4), 2347-2376. doi: 10.1109/comst.2015.2444095.
7. Atzori, L., Iera, A., & Morabito, G. (2010). The Internet of Things: A Survey. Computer

Networks,54(15), 2787-2805. doi: 10.1016/j.comnet.2010.05.010.
8. Bandyopadhyay, D., & Sen, J. (2011). Internet of Things: Applications and Challenges in Technology and Standardization. Wireless Personal Communications, 58(1), 49-69. doi: 10.1007/s11277-011-0288-5.
9. Han, Y., Yang, Y., Zhang, H., & Peng, Z. (2018). Design and Implementation of Industrial IoT Gateway Based on Edge Computing. 2018 IEEE International Conference on Industrial Technology (ICIT), 444-448. doi: 10.1109/icit.2018.8352106.
10. Hasib, M. A., Islam, M. M., & Amin, M. R. (2021). An Efficient IoT-Based Industrial Load Monitoring and Controlling System. Journal of Electrical and Computer Engineering Innovations, 9(1), 17-25. doi: 10.25236/jecei.2021.010103.
11. Kalaivanan, K., Kumar, K. S., & Kumar, M. P. (2018). Internet of Things (IoT) and Its Applications in Electrical Power Sector. 2018 IEEE International Conference on Communication and Signal Processing (ICCSP), 1743-1747. doi: 10.1109/iccsp.2018.8524303.
12. Li, S., Xu, L. D., & Zhao, S. (2015). The Internet of Things: A Survey of Topics and Trends. Information Systems Frontiers, 17(2), 261-274. doi: 10.1007/s10796-014- 9492-7.
13. Wang, J., Chen, M., Haas, Z. J., & Hua, K. (2017). A Survey of Industrial Internet of Things: APerspective from Energy, Environment and Engineering. IEEE Access, 5, 22418-22434. doi: 10.1109/access.2017.2770876.
14. Wu, F., Tan, J., Liu, R., & Shi, W. (2014). Mobile Internet of Things: A Survey. Journal of Internet of Things, 1(1), 1-13. doi: 10.11648/j.iot.s.2014020601.11.
15. Zhang, Y., Li, X., Li, Y., Zhang, W., & Wang, B. (2019). Industrial Internet of Things: A Comprehensive Survey. IEEE Access, 7, 82717-82743. doi: 10.1109/access.2019.2924726