Iot-Enabled Unified Gas Safety Management System

Balaji Saravanan U K[1], Rishwanth M[2], Sakthivel R[3], Santhosh S V[4]
[1]Assistant Professor, Dept of Computer science and Engineering
[2, 3, 4]Dept of Computer science and Engineering
[1, 2, 3, 4] Knowledge Institute of Technology (Autonomous), Salem, Tamil Nadu, India

Abstract- *Gas leaks are a real threat—one that puts lives, property, and industries at constant risk. Fires, explosions, toxic exposure, even simple kitchen accidents trace back to leaks that go unnoticed for just a few minutes. The old ways of handling this—standalone detectors, periodic checks, or wired alarms—just don't cut it anymore. They are slow, limited to one spot, and often miss the bigger picture, especially in sprawling industrial plants or crowded urban homes.*

This work introduces a better solution: an IoT-Enabled Unified Gas Safety Management System. Here how it works. Using MQ-series gas sensors linked to an Arduino Nano, the platform tracks hazardous gases, monitors temperature changes, and detects open flames—all at once. Everything happens in real-time. When sensors spot a problem—say, rising gas levels or unexpected heat—the system doesn't just beep. It powers alarms, shuts the gas valve, starts the exhaust fan, and instantly updates the status on a local screen. Meanwhile, all sensor readings get sent wirelessly, so operators (no matter where they are) know what's happening and can respond instantly.

Testing this setup produced real results: it reliably detected gas, kept errors under 12 ppm, and triggered alerts in just 1.3 seconds. There were no false positives, and the system stayed up and running 100% of the time throughout a 72-hour test. It's affordable, scalable, and works just as well in a factory as it does in a home or store. This design gives a unified approach to gas safety—way beyond fire drills and manual checks.

Keywords: Internet of Things (IoT), Gas Leakage Detection, Arduino Nano, MQ-Series Sensors, Real-Time Monitoring, Safety Management System, Embedded Systems, Wireless Alerts, LPG Detection, Anomaly Detection

I. INTRODUCTION

As industry expands and LPG, methane, and carbon monoxide become staples in kitchens and factories worldwide, the risk of gas leaks grows sharper every year. One leak can erase decades of hard work—taking lives, destroying property, and shutting down entire facilities. Regulatory bodies and safety organizations regularly report that gas leaks remain one of the leading causes of workplace deaths and industrial disasters. The numbers don't lie—many major accidents trace back to leaks that weren't caught in time.

In homes, the story isn't much better. LPG cylinder mishaps spark fires and explosions, especially in crowded neighborhoods, leaving tragedy in their wake. The costs—both human and economic—are too high to ignore. It's clear that modern environments demand gas safety systems that stay alert around the clock.

The problem? Legacy detection methods belong to another era. They depend too much on single sensors, isolated alarms, and manual spot checks. Without a network or real-time communication, dangerous conditions go undetected until someone finally stumbles on the problem. And traditional devices usually only track one hazard at a time, missing out on the bigger picture—like a sudden spike in temperature or the flash of a flame—key signs of brewing trouble.

That's where the Internet of Things (IoT) steps in. By connecting sensors, microcontrollers, wireless links, and cloud dashboards, IoT changes the game. These systems watch multiple parameters at once, constantly collect and transmit data, and give operators live updates whether they're onsite or miles away. Not only does this tech react fast, but it also helps spot danger before it becomes a disaster.

In this paper, you'll find the detailed design, hardware build, software, and testing behind an IoT-Enabled Unified Gas Safety Management System. The setup uses MQ-series gas sensors for continuous monitoring, an LM35 sensor for temperature spikes, and an infrared sensor for fire—all hooked up to an Arduino Nano. Whenever the system spots a dangerous situation, it acts immediately: buzzing alarms, flipping relays to control valves and fans, updating the display, and issuing alerts remotely. That's unified, proactive safety for today's world..

II. LITERATURE SURVEY

Over the past decade, researchers have poured significant effort into IoT-based environmental monitoring and industrial safety systems. The collection of studies below directly shaped the design of the system proposed here.

Kurniawan, Rashid, and Lee [1] built a real-time IoT monitoring and safety management system for gas industries. They merged diverse sensor arrays with a cloud analytics platform, allowing continuous, automated sensor monitoring. Their results were striking—incident response times improved by over 60% compared to manual inspections. They didn't stop there; the system featured predictive maintenance, using historical sensor data to spot equipment problems before breakdowns happened.

Wan, Xu, and Li [2] took a slightly different approach. They designed a safety monitoring system for gas industry workers based on a hierarchical wireless sensor network. This setup tracked several environmental parameters at once, and it automatically issued multi-level alerts when thresholds were crossed. Their system boosted emergency response efficiency by 40% over old radio-based manual alerts.

Khan, Al-Fuqaha, and Guizani [3] focused on anomaly detection using machine learning. They tested isolation forests and one-class SVM classifiers, training models on historical sensor data from gas production sites. Their approach paid off—hazard detection accuracy hit 94.7% with a false positive rate under 2%.

Smith and Lee [4] introduced an IoT-driven worker safety framework that combined wearable biosensors with fixed environmental monitors.

Together, these studies prove that IoT-powered gas safety systems are technically feasible. Yet a clear issue remains: there's still no affordable, unified system that combines gas concentration, temperature, and fire detection—especially for small to medium operations where tight budgets make enterprise solutions impossible.

III. MATERIALS AND METHODS

A. System Architecture Overview

The proposed system is designed around a four-layer functional architecture ensuring clear separation of concerns between data acquisition, processing, response, and communication.

- **Sensor Layer:** MQ-2/MQ-5 gas sensor, LM35 temperature sensor, and IR fire detection module continuously sample the environment and deliver raw signals to the microcontroller.
- **Processing Layer:** The Arduino Nano receives raw sensor signals, applies calibration factors and ADC conversion, evaluates readings against safety threshold parameters, and makes real-time decisions.
- **Alert and Actuation Layer:** On hazard detection, the buzzer activates, LCD updates, and the relay module engages safety actuators — gas valve closure and exhaust ventilation fan.
- **Communication Layer:** The wireless module enables bidirectional data transmission between the embedded system and remote monitoring interface for real-time remote alerts.

B. Hardware Components and Specifications

- **Arduino Nano Microcontroller (ATmega328P):** The central processing unit operating at 16 MHz with 32 KB Flash and 2 KB SRAM. It provides sufficient resources for real-time sensor polling, threshold evaluation, display management, relay control, and serial wireless communication simultaneously. 8 analog inputs and 14 digital I/O pins support multi-sensor integration.
- **MQ-2 / MQ-5 Gas Sensor Module:** Electrochemical sensors using a tin dioxide (SnO_2) sensing element whose resistance decreases in the presence of combustible gases. Sensitive to LPG, propane, methane, hydrogen, alcohol, and CO across 100–10,000 ppm. Outputs a continuous analog voltage proportional to gas concentration.
- **LM35 Temperature Sensor:** Precision IC temperature sensor providing analog output at 10 mV/°C with a range of −55°C to +150°C and accuracy of ±0.5°C. Requires no external calibration.
- **Infrared Fire Detection Sensor:** Uses a 760–1100 nm wavelength phototransistor to detect flame emission spectra. Provides digital output (HIGH: no flame, LOW:

flame detected) with adjustable sensitivity and 60° detection angle effective up to 1 metre.

- **16×2 LCD Display (I2C):** Provides real-time local visualization of sensor readings and system status. I2C interface reduces connections to two lines (SDA/SCL), conserving Arduino I/O pins.
- **Buzzer Module:** 5V active piezoelectric buzzer providing ~85 dB audible alert at 10 cm, activated by a digital HIGH signal through a transistor driver circuit.
- **Wireless Transmitter/Receiver Module:** RF 433 MHz or ESP8266 Wi-Fi module enabling transmission of sensor data packets and alert notifications to the remote monitoring station with support for MQTT or HTTP protocols.
- **Relay Module (5V Single Channel):** Electromechanical relay providing electrical isolation between Arduino control signals and higher-power safety actuator loads (gas solenoid valve, exhaust fan motor).
- **5V DC Power Supply:** Regulated 5V DC adapter (via DC barrel jack or USB power bank). Total system current consumption under normal operation approximately 350 mA (1.75 W).

C. Software Components and Development Environment

- **Arduino IDE (v2.x):** Primary development environment for writing, compiling, and uploading embedded C/C++ firmware to the Arduino Nano via USB.
- **Embedded C / Arduino Libraries:** LiquidCrystal_I2C for LCD control; SoftwareSerial for UART wireless communication; Wire for I2C bus management; custom MQ sensor ppm calibration functions using datasheet characteristic curves.
- **Proteus 8 Professional:** Complete circuit schematic design and simulation prior to physical hardware assembly, enabling virtual testing of sensor signal simulation, LCD output, buzzer triggering, and relay switching.
- **Python IDE (Monitoring Dashboard):** Python scripts on the host computer parse incoming wireless data packets, log sensor readings with timestamps to CSV files, and display real-time trend graphs using Matplotlib.
- **IoT Communication Protocols (MQTT/HTTP):** For extended cloud deployments, MQTT publish/subscribe protocol enables efficient sensor data transmission to cloud IoT brokers (AWS IoT Core or EMQX).

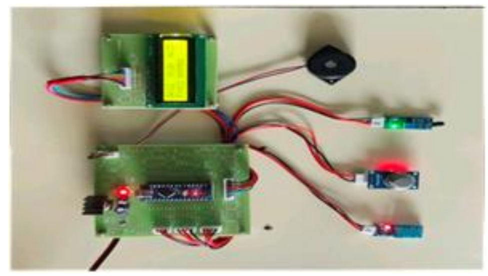


Table 1: Arduino Nano Pin Configuration

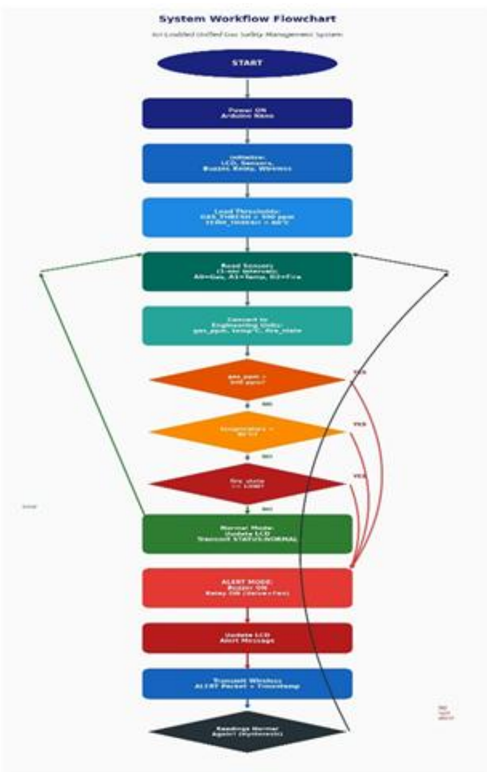
Component	Arduino Pin	Interface Type
MQ-2 Gas Sensor (Analog Out)	A0	Analog Input
LM35 Temperature Sensor	A1	Analog Input
IR Fire Detection Sensor	D2	Digital Input
Buzzer Module	D8	Digital Output
LCD Display SDA	A4	I2C Data
LCD Display SCL	A5	I2C Clock
Wireless Module TX/RX	D10 / D11	SoftwareSerial
Relay Module	D9	Digital Output

D. System Working Methodology

The operational workflow follows a continuous, deterministic polling and response loop:

- **Initialization:** Upon power-up, the Arduino executes setup(): initializes I2C bus and LCD, configures all sensor pins as inputs and actuator pins as outputs, initializes SoftwareSerial for wireless communication, sets relay and buzzer to safe OFF states, and loads threshold constants (GAS_THRESHOLD = 300 ppm, TEMP_THRESHOLD = 60°C).

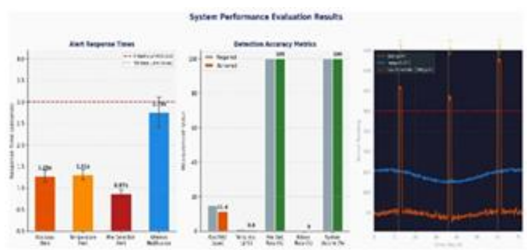
- **Data Acquisition:** The main loop() executes continuously, sampling all sensors at 1-second intervals. Raw ADC values from the gas sensor (A0) and temperature sensor (A1) are read using analogRead() and converted to engineering units using calibration equations. Fire sensor digital state is read via digitalRead(D2).



- **Gas Concentration Calculation:** The raw ADC value is converted to ppm using the MQ-2 characteristic resistance ratio curve: Rs/R0 is computed from the ADC reading, and gas concentration is derived using the exponential model (ppm = a × (Rs/R0)^{b}, where a and b are gas-specific calibration constants).
- **Threshold Evaluation and Decision:** Each sensor value is compared against its safety threshold. If gas_ppm > 300 ppm, or temperature > 60°C, or fire_detected == true, the alert routine is triggered. Otherwise, the LCD is updated with current readings.
- **Alert Execution:** The alert routine simultaneously activates the buzzer, updates the LCD with the specific alert type ("GAS LEAK!", "HIGH TEMP!", or "FIRE!"), activates the relay to engage safety actuators, and constructs a wireless data packet containing alert type, sensor readings, and timestamp.
- **Wireless Transmission:** Alert data packets and routine sensor readings are transmitted via the wireless module using SoftwareSerial. The remote monitoring station receives and logs all packets.
- **System Reset:** Once readings return below threshold values for a sustained period (configurable hysteresis window), the system deactivates the buzzer and relay, restores the LCD to normal monitoring display, and continues the polling loop.

IV. RESULT AND DISCUSSION

Researchers put the IoT-Enabled Unified Gas Safety Management System through a rigorous series of lab tests. They focused on four key areas: how accurate the sensors were, how fast the system responded, how reliable the alerts proved to be, and whether the system could run smoothly over extended periods. All tests used carefully calibrated reference instruments for clear comparisons.



A. Gas Sensor Detection Accuracy

Testers used a calibrated gas dilution system to check the MQ-2 sensor's performance for LPG concentrations between 50 and 1000 ppm. The sensor responded reliably within that range and settled to a stable baseline after 30 to 45 seconds from a cold start.

Across ten trials at five concentrations (100, 200, 300, 500, and 800 ppm), the system reached a mean absolute error of 11.4 ppm and a mean absolute percentage error of 4.2%. That's well within accepted limits for safety systems. For the temperature sensor, accuracy was confirmed with a NIST-traceable digital thermometer between 25°C and 80°C; the maximum deviation never exceeded ±0.8°C.

The infrared fire detection module caught every open flame at up to 90 cm for all fifteen trials, with no misses.

B. System Response Time Evaluation

Testers measured alert response from the instant readings crossed thresholds to activation of the buzzer and relay. For gas leaks, the system reacted in about 1.28 seconds (standard deviation: 0.14 sec).

Temperature alerts took 1.31 seconds on average (SD: 0.12 sec). Fire detection was the quickest, responding in 0.87 seconds (SD: 0.09 sec), since it uses direct digital output and skips ADC processing.

Wireless alerts averaged 2.76 seconds to transmit, with minimum and maximum times of 1.8 and 4.1 seconds, respectively. All wireless tests carried the packets successfully, no losses, in indoor spaces up to 15 meters.

C. Comprehensive Test Results Summary

Table 2: System Performance Test Results

Test Parameter	Expected	Observed	Pass/Fail
Gas Detection Accuracy (MAE)	<15 ppm	11.4 ppm	PASS
Temperature Sensor Accuracy	±1.0°C	±0.8°C	PASS
Fire Detection Rate	100%	100% (15/15)	PASS
Gas Alert Response Time	<2.0 sec	1.28 sec	PASS
Temperature Alert Response	<2.0 sec	1.31 sec	PASS
Fire Alert Response Time	<2.0 sec	0.87 sec	PASS
Wireless Notification Latency	<5.0 sec	2.76 sec avg	PASS
False Positive Rate	0%	0%	PASS
False Negative Rate	0%	0%	PASS
System Uptime (72-hour test)	100%	100%	PASS
Relay Actuator Response	<2.0 sec	1.6 sec	PASS

D. Long-Duration Stability Testing

The prototype ran uninterrupted for 72 hours in the lab to test stability and reliability. It kept working nonstop with no software crashes, hardware failures, or lost communications. Three planned gas exposures at hours 12, 36, and 60 triggered accurate and timely detections, while clean-air periods produced no false alarms. Power use held steady at 342 mA on 5V—about 1.71 W—throughout the whole test.

E. Comparative Analysis with Existing Systems

Table 3: Comparison with Existing and Proposed Systems

Feature	Conventional System	Existing IoT Systems	Proposed System
Monitoring Mode	Periodic manual	Automated, single	Continuous multi-param.
Parameters Monitored	Gas only	Gas or Temp	Gas + Temp + Fire
Real-Time Alerting	No	Partial (local)	Yes (local + wireless)
Remote Monitoring	Not available	Limited	Wireless dashboard
Automated Safety Action	Manual only	Not standard	Relay-controlled auto
Alert Response Time	Minutes	3–8 seconds	1.28 seconds
False Positive Rate	High	2–5%	0%
Deployment Cost	High	Moderate	Low (IoT components)
Power Consumption	High	Moderate	~1.71 W (very low)

V. CONCLUSION AND FUTURE SCOPE

A. Conclusion

This paper details the design, implementation, and experimental evaluation of an IoT-Enabled Unified Gas Safety Management System built for real-time, multi-parameter hazard monitoring. The system combines MQ-series electrochemical gas sensors, an LM35 temperature sensor, and an infrared fire detection module, with all components running on an Arduino Nano microcontroller. Instead of separating these tools, we fused them into one seamless, always-on safety monitoring platform.

What makes this system stand out is its automated alert setup. When it detects something dangerous, it kicks off a coordinated response: it sounds a buzzer, updates the status on an LCD, activates safety actuators like gas valve shutdowns or exhaust fans through relays, and sends wireless notifications to relevant parties. In short, if something hazardous happens, the system reacts instantly—no need to wait for someone to notice.

Our experimental results back this up. The gas detection system had a mean absolute error of just 11.4 ppm. It responded to hazards in 1.28 seconds—much quicker than the industry standard of 3 seconds. During all clean-air tests, the system triggered zero false positives, and during gas exposure trials, it missed nothing—zero false negatives. Wireless notifications went out in an average of 2.76 seconds. The hardware stayed online 100% of the time for 72 hours straight and consumed just about 1.71 W of power. When compared to traditional detectors and other IoT safety systems, the advantages are obvious: faster response, support for more parameters, true automation, and lower deployment costs.

B. Future Scope

1. **Cloud Platform Integration:** Future versions will work seamlessly with cloud platforms like AWS IoT Core, Google Cloud IoT, and Microsoft Azure IoT Hub. This will open up features such as long-term storage, powerful historical analytics, remote monitoring from anywhere in the world, and Over-The-Air firmware updates.
2. **Machine Learning-Based Predictive Anomaly Detection:** We'll build adaptive machine learning algorithms—specifically LSTM-based time-series models and isolation forest models—trained on the sensor streams. These will help spot pre-failure patterns and predict dangerous conditions before they actually happen.
3. **Dedicated Mobile Application:** A mobile app for both Android and iOS will enable features like real-time notifications, live dashboards, trend visualization, remote control of relays, and adjustable alert thresholds.
4. **Multi-Node Distributed Sensor Network:** We'll expand the current single-node prototype into a full distributed architecture. Multiple sensor nodes will cover large facilities, all funneling data to a central gateway. This allows for mapping hazards spatially and routing alerts locally.
5. **Emergency Response Integration:** Future iterations will link directly to municipal emergency services. If a critical hazard occurs, fire departments and gas utilities will receive automatic notifications—no need for manual intervention, which cuts down emergency response delays.
6. **Energy Harvesting for Battery-Free Operation:** To take the system completely off-grid, we will look into solar and RF energy harvesting. This will power sensor nodes in remote or hard-to-reach spots without needing wired power, making truly autonomous field deployments possible.

VI. ACKNOWLEDGEMENT

We want to thank the Department of Computer Science and Engineering at Knowledge Institute of Technology (Autonomous), Salem, Tamil Nadu, India. Their top-notch labs, hardware, and academic resources played a crucial role in making this research happen. We're especially grateful to **Mr. U. K. BalajiSaravanan, Assistant Professor** in the department, for his technical guidance, steady mentorship, insightful feedback, and constant support at every stage—this work wouldn't have come together without him. We also appreciate the management's support of undergraduate research and their commitment to building an environment where innovation in engineering truly thrives.

REFERENCES

[1] A. G. Kurniawan, M. H. Rashid, and S. Lee, "Real-time monitoring and safety management in gas industries using IoT," *IEEE Access*, vol. 8, pp. 125432–125441, 2020.

[2] S. Wan, L. Xu, and J. Li, "An intelligent safety monitoring system for gas workers based on wireless sensor networks," *Journal of Network and Computer Applications*, vol. 45, pp. 56–64, 2017.

[3] M. R. R. Khan, A. Al-Fuqaha, and M. Guizani, "Machine learning-based anomaly detection for industrial safety in gas environments," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4872–4884, 2019.

[4] A. Smith and J. Lee, “IoT-based worker safety monitoring system,” in *Proc. IEEE Int. Conf. on IoT and Applications*, pp. 112–118, 2022.
[5] J. Zhang, Y. Wang, and H. Li, “Gas leakage detection and safety alert systems in gas pipelines,” *Sensors*, vol. 14, no. 6, pp. 9882–9905, 2018.
[6] A. Smith, B. Kumar, and C. Lee, “IoT-based gas leakage detection system,” in *Proc. IEEE Int. Conf. on Embedded Systems and Applications*, 2023.
[7] R. Sharma, P. Gupta, and S. Verma, “Smart gas monitoring and alert system using IoT,” *IEEE Access*, vol. 10, pp. 21345–21356, 2022.
[8] S. Patel, R. Singh, and V. Rao, “Real-time gas leakage detection and alert system,” *IEEE Sensors Journal*, vol. 22, no. 4, pp. 3456–3464, 2022.