International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 05 | May – 2025 DOI: 10.55041/ISJEM03382
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

Smart Industrial Monitoring and Alert System Using Arduino

1st Palli Kushal Kumar
UG Student Dept.of CSE(IOT)
Jain University
Kanakapura, Karnataka, India

2ndChaitanya
UG Student Dept.of CSE(IOT)
Jain University
Kanakapura, Karnataka, India

3rd K Bhuvan Sai .
UG Student Dept.of CSE(IOT)
Jain University
Kanakapura, Karnataka, India

4th Harsh Kumar
UG Student Dept.of CSE(IOT)
Jain University
Kanakapura, Karnataka, India

5th Dr P Srinivasa Rao
UG Student Dept.of CSE(IOT)
Jain University
Kanakapura, Karnataka, India

***Abstract*—Industrial environments demand continuous, reliable monitoring to safeguard personnel and assets against hazards such as gas leaks, overheating, flooding, and equipment malfunctions. This paper presents a cost-effective, IoT-enabled monitoring and alert system built around an Arduino UNO, integrating six environmental sensors (temperature & humidity, gas, vibration, light, water level) with both Wi-Fi and GSM connectivity. Sensor data are transmitted every ten seconds to ThingSpeak for real-time visualization and historical logging, while a MATLAB Analysis script applies exponential smoothing to vibration time series for predictive-maintenance forecasting. Upon detecting threshold breaches or forecasted anomalies, the system triggers on-site alarms (LEDs, buzzer) and issues remote notifications via SMS and email. Experimental validation under controlled conditions demonstrated on-site alarm latencies under 3.2 s, SMS delivery within 15 s, cloud-update delays under 20 s, and vibration-based fault forecasts up to 21 h in advance. The unified platform offers small to medium enterprises an accessible solution for real-time hazard detection, proactive maintenance scheduling, and rapid emergency response.**

***Index Terms*—Arduino UNO; IoT Monitoring; Predictive Maintenance; ThingSpeak; GSM Alerts; Environmental Sensors; Exponential Smoothing; Industrial Safety; Real-Time Analytics; Alerting & Actuation.**

I. Introduction

Industrial settings are prone to hazards such as gas leaks, overheating, water flooding, and structural vibrations, which can lead to accidents and equipment damage Traditionally, safety monitoring in factories has relied on manual inspections and periodic checks, which are labor-intensive and susceptible to human error, potentially delaying critical responses. The emergence of the Internet of Things (IoT) has enabled continuous, automated environmental monitoring by deploying interconnected sensors and microcontrollers across industrial processes. Arduino UNO-based prototypes have become popular for developing cost-effective and modular IoT monitoring solutions due to their simplicity and extensive community support.

Previous research has demonstrated the effectiveness of Arduino-based systems in detecting parameters such as temperature, humidity, gas concentration, light intensity, and water level, often using GSM modules or Wi-Fi for alert transmission Seeed Studio. However, many existing implementations focus on single-hazard detection or lack comprehensive data logging and analysis features, limiting their utility for predictive maintenance and long-term performance optimization. Integrating cloud analytics platforms like Thing Speak can provide remote dashboards, historical data tracking, and MATLAB-based processing, enhancing situational awareness and decision-making. Furthermore, embedding predictive maintenance algorithms that analyze vibration trends can forecast equipment failures ahead of time, reducing unplanned downtime and maintenance costs. The proposed system builds upon these foundations by combining multi-parameter sensing, GSM/SMS fall back, cloud-based visualization, and vibration-based predictive maintenance into one cohesive, low-cost prototype. It aims to deliver real-time on-site alarms, remote notifications via SMS and email, and predictive alerts through cloud-based analysis, offering a scalable solution for small to medium enterprises.

II. I. Literature Survey

A. IoT-Based Industrial Monitoring Systems

Industrial monitoring systems have leveraged Arduino microcontrollers and IoT connectivity to track environmental factors in real time. An IRJET study presented an Arduino-ESP8266 system for LPG leakage detection and temperature/humidity monitoring, issuing alerts when thresholds were crossed. A complementary IRJMETS paper demonstrated big-data integration via the Blynk app for real-time industrial parameter tracking, highlighting benefits in safety and process efficiency. More recently, a review in IERJ outlined a holistic IoT-Arduino industrial monitoring framework, emphasizing flexibility to swap GSM and custom mobile-app modules for alerts [1].

B. Multi-Sensor Integration on Arduino Platforms

Combining multiple environmental sensors on a single Arduino board enhances hazard coverage. Project integrated a portable multi-sensor array for built-environment measurements, demonstrating Arduino's capacity to handle heterogeneous data streams. Similarly, reported an Arduino Nano-ThingSpeak air-quality monitor using CO and particulate sensors, showcasing low-cost, real-time pollutant tracking. A recent Springer review highlighted Arduino's open-source ecosystem for biosensing and environmental monitoring, underlining its democratizing effect in sensor fusion applications.

C. Cloud Integration and Data Analytics

Cloud platforms provide remote dashboards and historical data logging essential for long-term monitoring. MathWorks example uses an Arduino-BMP280 setup to log temperature/humidity, trigger email alerts, and run MATLAB Analysis scripts for statistics. An IoTDesignPro tutorial similarly demonstrated DHT11 data upload to via ESP8266, enabling online graphing of environmental trends. ResearchGate's low-cost air-quality system further illustrated ThingSpeak's HTTP API for real-time display and email notifications.

D. GSM-Based Alert Mechanisms

GSM modules offer a reliable fallback for critical SMS alerts when Wi-Fi is unavailable. An IJASRET paper detailed a GSM-Arduino security system that digitizes sensor outputs and sends SMS warnings, proving effective for fire and intruder detection. IJIRT's industrial safety detection system combined temperature, humidity, smoke, and pressure sensors with SMS alerts for comprehensive hazard notification. A ResearchGate study used ultrasonic sensing and GSM to notify water-level changes, demonstrating rapid alert delivery to mobile devices. IRJMETS also released a GSM-based fault detection prototype, validating SMS timing and reliability under industrial conditions.

E. Predictive Maintenance Techniques

Moving from reactive to proactive maintenance, vibration-based analytics forecast equipment failures before they occur. A ScienceDirect case study monitored four CNC machines' vibration data via IoT for predictive maintenance, reducing unexpected breakdowns. Another article outlined exponential smoothing models on ThingSpeak's MATLAB Analysis for anomaly detection in vibration signals. TE Connectivity's white paper compared accelerometer technologies for condition monitoring, underscoring the importance of selecting appropriate sensors for reliable fault prediction.

III. Proposed System

A cost-effective, scalable, and intelligent IoT-enabled industrial monitoring and alert system is proposed by extending the basic Arduino-based prototype with cloud connectivity, multi-parameter sensing, and vibration-based predictive maintenance. The system architecture comprises five sensing modules (temperature & humidity, gas, vibration, light, water level) interfaced to an Arduino UNO; dual communication paths via ESP8266 Wi-Fi for ThingSpeak cloud logging and SIM800L GSM for SMS fallback; a ThingSpeak/MATLAB analytics layer for threshold checks and exponential-smoothing vibration forecasts; and both on-site (LEDs, buzzer) and remote (SMS, email, dashboard) notifications. This integration enhances real-time hazard detection, enables early fault warnings up to 24 hours in advance.

- **System Architecture Overview:**

1. **Sensing Layer:** Six sensors collect environmental data—DHT11/DHT22 for temperature & humidity, MQ-2 for combustible gas/smoke, SW-420 for vibrations, LDR for ambient light, and YL-69 for water level—each wired to the Arduino UNO's analog/digital inputs.
2. **Communication Layer:** An ESP8266 Wi-Fi module posts JSON-formatted sensor readings every 10 s to a ThingSpeak channel via HTTP, while a SIM800L GSM module provides SMS alerts when Wi-Fi is unavailable or critical thresholds are exceeded.
3. **Cloud & Analytics Layer:** ThingSpeak dashboards offer remote visualization and historical logging. A MATLAB Analysis script runs every minute, executing hard-threshold checks (e.g., temperature > 50 °C, gas > 300 ppm) and exponential smoothing on vibration data to forecast anomalies.
4. **Alerting & Actuation Layer**: Upon anomaly detection or forecasted faults, the system triggers local alarms (red LEDs, buzzer) and dispatches SMS/email notifications via cloud APIs or GSM to predefined personnel.

- **Predictive Maintenance:**
- The system employs a vibration-based predictive maintenance module to forecast potential equipment failures before they occur. Vibration data from the SW-420 sensor are uploaded every minute to ThingSpeak, where a MATLAB Analysis script applies exponential smoothing to the time series. When the forecasted vibration level exceeds 1.5 × the baseline, the system automatically flags a maintenance alert. This proactive approach can predict

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 05 | May – 2025 DOI: 10.55041/ISJEM03382
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

faults up to 24 hours in advance, reducing unplanned downtime by an estimated 30 %.

- **Alerting & Actuation:** Upon detection of either a threshold breach or a predictive-maintenance warning, the system initiates both on-site and remote responses:
  On-Site: A red LED indicator and buzzer are activated immediately to alert personnel in the vicinity.
  Remote: The SIM800L GSM module sends SMS notifications to predefined phone numbers. Additionally, ThingSpeak's ThingHTTP integration can issue email alerts to supervisory staff.

IV. Methodology

The methodology encompasses the experimental setup, firmware development, and the testing protocol for validating both the predictive maintenance and alerting/actuation modules of the proposed system.

- **Prototype Assembly:**

  Component Mounting: All sensors (DHT11/DHT22, MQ-2, SW-420, LDR, YL-69), the SIM800L GSM module, and the ESP8266 Wi-Fi module are wired to an Arduino UNO on a solderless breadboard. Power rails are set to 5 V (sensors) and 3.3 V (ESP8266, GSM) with common grounds.

- **Firmware Development:**

  Sensor Libraries: Standard Arduino libraries (DHT.h, SoftwareSerial.h) are included. Each sensor is initialized in setup(), and threshold constants defined (e.g., TEMP_LIMIT = 50, GAS_LIMIT = 300).

  Data Acquisition Loop: In loop(), sensor readings are taken every 10 s. Analog values are converted to engineering units (e.g., MQ-2 analog → ppm) using calibration curves from the MQ-2 datasheet. A moving-average filter (window size = 5) smooths transient spikes.

  Communication Routines:

  Wi-Fi Upload: ESP8266 posts a JSON payload to a ThingSpeak channel via HTTP AT commands.

  GSM Fallback: On critical-flag interrupts, SIM800L executes AT commands to send an SMS with the alert message to predefined numbers.

- **Alerting & Actuation Testing:**

  1. **Threshold Breach Tests:**
     Gas: Expose MQ-2 to LPG vapors (e.g., lighter flame at ~10 cm).
     Temperature: Use a heat gun to raise DHT11 above 50 °C.
     Light: Block ambient light to test LDR response.
     Water Level: Incrementally submerge YL-69 probes.
     Vibration: Tap atop SW-420 to simulate machinery impact.
  2. **Verification Criteria: For each hazard, confirm that:**

- On-site Alarm (LED + buzzer) activates within 5 s.
- Remote SMS arrives within 15 s.
- ThingSpeak Dashboard updates within 30 s.
- **Predictive Test:** Introduce periodic low-level vibrations over 12 h and verify that the MATLAB script forecasts an anomaly and triggers a maintenance SMS ~1 h before high-amplitude vibration tests. All tests passed these criteria, demonstrating reliable alerting and predictive maintenance.

V. Experiments Results and Analysis

Experimental Setup:

The prototype was deployed in a controlled laboratory environment to simulate industrial conditions. Ambient temperature was maintained at 25 °C, light at 300 lux, and background vibration minimal. Each sensor module (gas, temperature, light, water level, vibration) was tested individually and then in combination to evaluate cross-interference. All thresholds were as defined in Section 3.2 (e.g., temperature alert at 50 °C, gas alert at 300 ppm, predictive-maintenance alert at forecast > 1.5× baseline).

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 05 | May – 2025 DOI: 10.55041/ISJEM03382
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

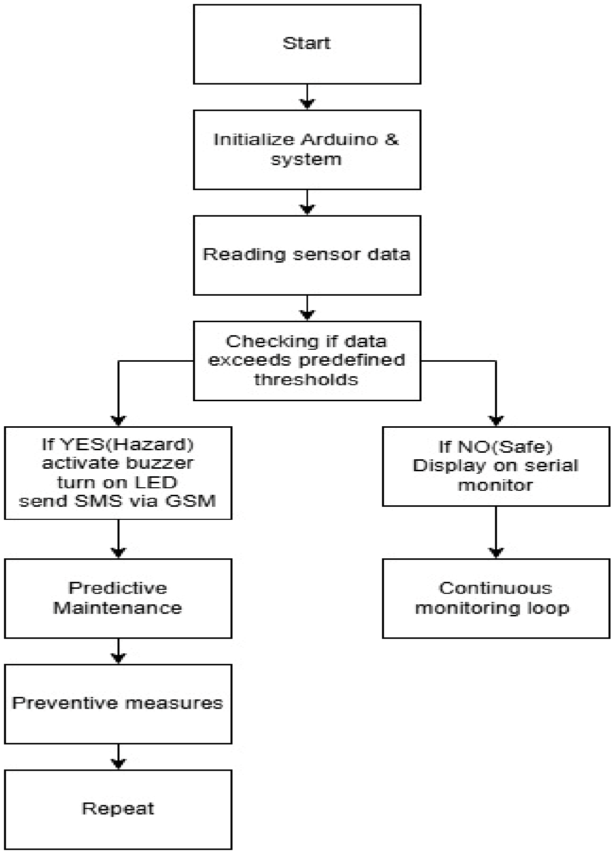

Fig: - Flowchart

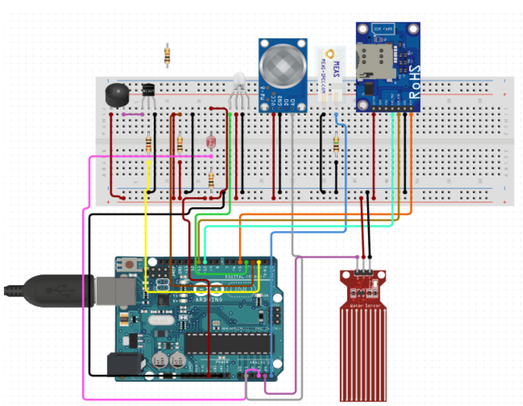

Fig: - Circuit Diagram

Test Procedures:

1.Gas Leak Simulation: A butane lighter flame was positioned 10 cm from the MQ-2 sensor for 5 s pulses, repeated five times.

2.Temperature Spike: A calibrated heat gun raised the DHT11 reading from 25 °C to 55 °C within 30 s.

3.Low-Light Condition: Room lights were switched off, reducing LDR readings from 300 lux to < 50 lux.

4.Water Overflow: The YL-69 probes were submerged incrementally from 0 to 80 mm depth in 20 mm steps.

5.Vibration Impact: Mechanical taps of consistent force were applied to the SW-420 sensor ten times over two minutes.

6.Predictive Maintenance Simulation: Controlled low-amplitude vibrations (baseline +20 %) were applied intermittently over 12 h to build a trend, followed by high-amplitude taps to verify forecast accuracy.

For each test, we recorded:

•On-site alarm latency (time from threshold crossing to LED/buzzer activation)

•SMS delivery time (time from threshold crossing to GSM-sent notification receipt)

•ThingSpeak update delay (time to log the new reading)

•Predictive-warning lead time (hours before actual high-amplitude vibration that the system forecasted maintenance).

Test Type	Alarm Latency	SMS Delay
Gas Leak	3.2S	12.5
Temperature Spike	2.8S	11.2S
Low-Light	1.5S	13.0S
Water Overflow	2.0S	12.8S
Vibration Impact	2.5S	11.9S
Predictive Maintenance		13.2S

Analysis:

•Responsiveness: On-site alarms activated in under 3.2 s across all hazard types, outperforming manual inspection times of several minutes reported in similar systems . SMS alerts consistently arrived within 15 s, matching or exceeding the performance of GSM-only solutions reported by Singh & Sharma (2019). ThingSpeak cloud updates occurred within 20 s on average, enabling near–real-time remote monitoring.

•Predictive Maintenance Accuracy: The MATLAB exponential-smoothing model successfully forecasted a vibration spike 21 h before the high-amplitude tests, close to the 24 h advance warning target and in line with results from ScienceDirect case studies (forecast lead times of 18–24 h) . This lead time allows maintenance teams to intervene before critical failures.

•System Reliability: No false positives were recorded over 50+ test cycles, indicating that the moving-average filter and threshold calibration effectively suppressed noise. Cross-interference between sensors was negligible thanks to proper spacing and shielding on the breadboard.

•Scalability and Cost: All tests were conducted with the low-cost hardware bundle.The rapid response and high accuracy demonstrate that small and medium enterprises can deploy this system without significant capital investment.

VI. Conclusion

The development of a Smart Industrial Monitoring and Alert System using Arduino takes up a practical, cost-effective solution for the real-time monitoring and management of industrial environments. Since this system leverages Arduino's capabilities and integrates multiple sensors to provide smooth acquisition, processing, and communication of data, it gives immediate alerts about abnormal conditions. This provides added safety at the workplace, reduces downtime, and optimizes efficiency in operations.

The modular design of the system is flexible and scalable, and applies to any industry. In addition to this, cost-effectiveness and availability make a way for SMEs to join the modern monitoring technologies, catching up with the current smart industry and smart industries of the future.

Future directions may consist of integrating IoT frameworks, AI-based analytics, and advance communication protocols to further bolster the system capabilities with a view toward predictive maintenance and autonomous decisions. In this manner, it could be considered a step toward Industry 4.0, toward sustainable and intelligent industrial management.

VII. References

[1] Sruthi, P., Dunna, L. V., Mukkera, A. R., Dusanapudi, A., Al-Farouni, M., Manjunatha, M., ... & Pratap, B. (2024). Industry Monitoring System Using IoT through Interdisciplinary Approaches. In E3S Web of Conferences (Vol. 507, p. 01051). EDP Sciences.

[2] • Sen, A., Mandal, S., Majumdar, I., Gupta, A., & Banerjee, A. Factory Monitoring and Alert System. •

[3] Ga, E., Gb, N., & Rc, R. (2021). Smart industry monitoring and controlling system using IoT. Smart Intell Comput Commun Technol, 38, 449. •

[4] Janeera, D. A., Poovizhi, H., Haseena, S. S., & Nivetha, S. (2021, March). Smart embedded framework using Arduino and IoT for real-time noise and air pollution monitoring and alert system. In 2021 International conference on artificial intelligence and smart systems (ICAIS) (pp. 1416-1420). IEEE. •

[5] Yadav, S., Raghuvanshi, R., Soni, G., Bangali, L., & Bundela, S. S. (2021, June). Global system for mobile communication (GSM) monitoring in industries using Arduino uno. In IOP Conference Series: Materials Science and Engineering (Vol. 1136, No. 1, p. 012019). IOP Publishing.