International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 13 Issue: 03 | Mar 2026 www.irjet.net p-ISSN: 2395-0072

IoT-Based Weather and Environmental Monitoring System with Real-Time Cloud Visualization and GSM Alerts Using ESP32

MD. Iqbal Basha[1], Nurpasha Dilsha Begum[2], Loya Jaya Chandra Manikanta[3], Dr. Ch. Rambabu[4]

[1,2,3]Final Year B.Tech Students, [4]Associate Professor - Department of Electronics and Communication Engineering Seshadri Rao Gudlavalleru Engineering College, Gudlavalleru, Andhra Pradesh, India

---

Abstract—*Continuous weather and environmental monitoring is essential for the safety of both infrastructure and human life, particularly in enclosed spaces where multiple hazardous conditions may develop simultaneously. This paper presents an IoT-based multi-parameter environmental monitoring system developed around the ESP32 microcontroller that concurrently measures temperature, relative humidity, atmospheric pressure, rainfall intensity, and combustible gas and smoke concentration. Sensor data is acquired using a DHT11 digital temperature-humidity sensor, a BMP180 [fig. 4] barometric pressure module, a YL-83 [fig. 6] rainfall detection board, and an MQ-series [fig. 5] metal-oxide gas sensor. A 20×4 I^2C LCD display [fig. 8] provides continuous local readout; a SIM900A GSM modem delivers condition-specific SMS alerts to registered users whenever any monitored parameter crosses a predefined threshold; and a ThingSpeak cloud channel archives all five data streams at 15-second intervals for remote access and trend analysis. The system was evaluated over a 14-day continuous bench run, producing the following measured outcomes: temperature accuracy within ±1 °C of a calibrated reference following sensor-batch replacement; gas and smoke alert confirmed within 6 seconds of controlled vapour exposure using a dual-read threshold confirmation; 100 % SMS delivery across 12 alert events with a maximum delivery latency of approximately 4 seconds; and zero system resets or display faults throughout the evaluation window. Two false rainfall activations caused by humidity-induced comparator triggering at 89 % relative humidity are documented and analysed as the principal hardware limitation of the current configuration.*

Keywords—ESP32; DHT11; BMP180; MQ gas sensor; YL-83; ThingSpeak; SIM900A; GSM; SMS alert; IoT; weather monitoring; environmental monitoring; rainfall detection.

1. INTRODUCTION

Weather and environmental conditions have a direct bearing on the safety and operational continuity of critical infrastructure such as generator rooms, server enclosures, and electrical switch rooms. Rapid changes in atmospheric pressure can precede extreme weather events; elevated temperature and humidity accelerate corrosion and component degradation; accumulation of combustible gas or smoke constitutes an immediate fire hazard; and rainfall ingress can cause electrical short circuits and irreversible equipment damage. Monitoring any single one of these parameters in isolation provides an incomplete safety picture a room may show acceptable temperature while simultaneously accumulating hazardous gas concentration or receiving water ingress through a structural fault.

The Internet of Things has substantially reduced the cost and complexity of deploying continuous multi-parameter sensor networks [14][15]. Low-cost wireless microcontrollers such as the ESP32 integrate Wi-Fi connectivity, multi-channel analogue-to-digital conversion, and sufficient processing capability for concurrent sensor management and cloud communication on a single module [18]. Paired with free-tier cloud platforms such as ThingSpeak, these devices enable persistent time-series data archiving without any dedicated server infrastructure. The addition of a GSM modem provides an independent alert path that remains active when Wi-Fi connectivity is unavailable — a critical resilience property for locations with intermittent internet service.

Existing weather and environmental monitoring systems predominantly address single-parameter measurement or focus on outdoor agricultural applications where continuous local alerting is a lower priority [2][6][1]. Systems that incorporate both cloud archiving and GSM-based SMS notification are documented but rarely combine barometric pressure monitoring with rainfall detection, gas sensing, and temperature-humidity acquisition in a single integrated platform [3][5]. The present work addresses this gap by delivering five simultaneous measurement channels with dual-path notification — local LCD with audible buzzer and remote SMS.

The design was governed by four requirements established before firmware development began. First, simultaneous acquisition of five environmental parameters — temperature, relative humidity, barometric pressure, rainfall, and gas/smoke concentration — in a single integrated unit. Second, SMS alert delivery to a registered mobile number for each distinct hazard condition, independent of Wi-Fi availability. Third, real-time cloud archiving of all five channels on ThingSpeak for remote monitoring and

International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 13 Issue: 03 | Mar 2026 www.irjet.net p-ISSN: 2395-0072

trend analysis. Fourth, all components sourced through verifiable local distributors within a fixed budget of ₹2,000.

2. RELATED WORK

IoT-based weather monitoring systems using Arduino and Zigbee mesh networks were among the earliest multi-node environmental sensing configurations in the literature [1]. While the distributed spatial coverage of such designs is relevant to large-area monitoring, the per-node cost of Wi-Fi shields and the absence of an integrated cloud upload path makes them unsuitable for single-location infrastructure monitoring within the budget constraints of this project. Sensor placement principles from those studies were retained as a configuration reference.

Patel et al. [2] demonstrated that SMS-based alert delivery provides a reliable notification channel when internet connectivity is absent. However, such architectures do not maintain time-series historical records, precluding the trend analysis required to identify gradual environmental deterioration — for instance, a slow pressure drop preceding a weather event, or progressive humidity increase indicating water infiltration. The dual-path architecture reported here
— ThingSpeak cloud archiving supplemented by SIM900A GSM alerts — addresses both requirements simultaneously [17].

Singh et al. [3] and Negi et al. [4] both describe ESP32-based environmental monitoring platforms with ThingSpeak integration. Neither design incorporates barometric pressure monitoring; Negi et al. additionally lack GSM cellular backup. The comparative microcontroller evaluation by Srinivasan and Bhardwaj [12] confirmed the ESP32's dual-core architecture as the preferred platform for concurrent sensor acquisition and network transmission, specifically because single-core devices running simultaneous ADC reads and HTTP uploads produce measurable data corruption — a failure mode encountered and resolved during the ESP8266 evaluation phase of this project. Sharma et al. [6] described a richer air-quality platform incorporating PM2.5 and ozone sensors; the continuous current draw of that system exceeds what an intermittently powered generator-room supply can sustain, making it unsuitable for the target deployment context.

3. SYSTEM ARCHITECTURE

The system is organised into three functional layers that operate independently and can be tested in isolation. The sensor acquisition layer comprises five measurement channels feeding the ESP32 through three distinct electrical interfaces: DHT11 single-wire protocol on GPIO 2, BMP180 and LCD display sharing the I^2C bus at addresses 0x77 and 0x27 respectively, and the MQ gas sensor and YL-83 rain sensor on GPIO 34 (12-bit ADC) and GPIO 4 (digital input) respectively. The local alerting layer — the 20×4 LCD display and the active piezo buzzer — operates unconditionally, with no network dependency. The remote notification layer encompasses both the ThingSpeak HTTP upload path and the SIM900A GSM SMS path, each operating independently so that a failure in one does not suppress the other.

A critical power-supply issue encountered during prototype integration requires specific mention. The SIM900A GSM modem was initially connected to the same 5 V rail powering all sensors and the ESP32. During GSM network registration, the modem draws a transient current of approximately 2 A lasting around 80 milliseconds — sufficient to pull the shared rail below the ESP32 brownout detection threshold of 3.3 V, causing repeated microcontroller resets. Providing the SIM900A with a dedicated 2 A supply rail isolated from the sensor power eliminated all resets immediately. This topology separation is reflected in Fig. 2, the system block diagram [9] [13].

The firmware loop enforces a fixed sensor read sequence: MQ gas sensor ADC first, then DHT11, then BMP180, then YL-83 rain sensor digital input. Reading the gas ADC first prevents the DHT11 single-wire timing protocol from corrupting ADC conversion results — a shared-resource contention documented in comparable single-core evaluations [3]. Alert evaluation and local display updates follow sensor acquisition; the ThingSpeak HTTP upload executes last, ensuring LCD and buzzer responses are never delayed by network latency. Fig. 1 shows the complete program execution flow.

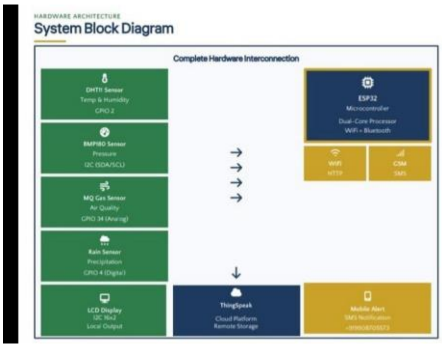

Fig. 1. Program execution flow. Sensor reads execute in fixed priority order (MQ gas → DHT11 → BMP180 → YL-83). Alert evaluation and local output precede cloud upload. A 15-second delay at loop end enforces the ThingSpeak free-tier rate limit.

4. HARDWARE COMPONENTS

4.1 ESP32-WROOM-32 Microcontroller

The ESP32-WROOM-32 [fig. 2] serves as the central processing and communication hub for the system. It integrates a 240 MHz dual-core Xtensa LX6 processor, 520 KB SRAM, 4 MB flash memory, Wi-Fi 802.11 b/g/n, and Bluetooth 4.2 within a single certified RF module, providing 36 usable GPIO pins including an 18-channel 12-

bit ADC [18]. The dual-core architecture was the decisive selection criterion: allocating the ThingSpeak HTTP upload to the second FreeRTOS core prevents upload latency from stalling sensor acquisition on the first core — a problem observed on the single-core ESP8266 during initial evaluation, where concurrent MQTT publish and ADC read operations produced data corruption [12]. The integrated Wi-Fi eliminated the need for an external wireless module, reducing both cost and wiring complexity.

Fig. 2. ESP32-WROOM-32 microcontroller development board. Dual-core 240 MHz Xtensa LX6 CPU; 520 KB SRAM; 4 MB flash; integrated 802.11 b/g/n Wi-Fi; 18-channel 12-bit ADC; 36 usable GPIO.

4.2 DHT11 Temperature and Humidity Sensor

The DHT11 [fig. 3] provides simultaneous digital measurement of ambient temperature and relative humidity through a single-wire serial interface at a maximum polling rate of 1 Hz. Manufacturer-stated accuracy is ±2 °C for temperature and ±5 % for relative humidity across a measurement range of 0–50°C and 20–90 % RH. An incoming calibration check against a laboratory reference thermometer revealed a systematic positive bias of approximately 4 °C in the initial unit — exceeding the stated tolerance and attributable to production-batch variability rather than installation error. A replacement unit from a different batch agreed within 1 °C of the reference instrument. A 4.7 kΩ pull-up resistor on the single-wire data line is mandatory; omitting it produces periodic checksum errors indistinguishable from line-noise faults at the firmware level, complicating diagnosis significantly.

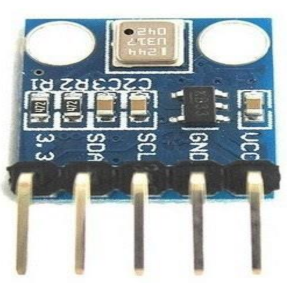

Fig. 3. DHT11 digital temperature and humidity sensor. Left unit: initial batch exhibiting systematic 4 °C thermal bias. Right unit: replacement batch verified within ±1 °C of calibrated reference. 4.7 kΩ pull-up on data line is mandatory.

4.3 BMP180 Barometric Pressure Sensor

The BMP180 [fig. 4] measures absolute atmospheric pressure with a resolution of 0.01 hPa and communicates over the I^2C bus at device address 0x77. The Adafruit_BMP085 library handles the multi-step temperature compensation polynomial defined in the device datasheet, converting raw sensor output to calibrated pressure values in hectopascals [19]. Atmospheric pressure in the Gudlavalleru region under normal meteorological conditions ranges from approximately 1008 to 1015 hPa. The SMS alert window was configured at 950–1050 hPa — sufficiently narrow to flag genuinely anomalous conditions such as an approaching storm system, while wide enough to exclude routine diurnal and seasonal variation. The BMP180 and the LCD backpack share the I^2C bus at addresses 0x77 and 0x27 respectively; initialising the LCD before the BMP180 caused I^2C bus instability during startup. Reordering the initialisation sequence — BMP180 first, then LCD — resolved the issue.

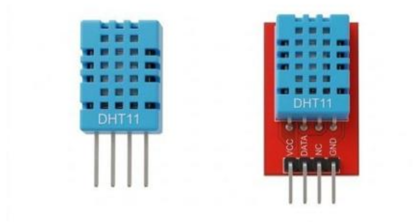

Fig. 4. BMP180 barometric pressure sensor module. I^2C interface at address 0x77; shares I^2C bus with 20×4 LCD backpack (0x27). Adafruit_BMP085 library provides temperature-compensated pressure output in hPa.

4.4 MQ-Series Gas and Smoke Sensor

The MQ-series sensor [fig. 5] detects combustible and irritant gases through resistance modulation at a heated tin-dioxide sensing surface. It responds to methane, LPG, propane, alcohol vapor, and smoke without species discrimination — a characteristic limitation discussed in Section VIII. The 12-bit ADC on GPIO 34 samples the analogue output voltage. Under clean laboratory air conditions, ADC readings stabilised between 1400 and 2000 counts; a threshold of 3000 counts was selected as the SMS alert trigger following empirical characterisation using a controlled LPG vapour source. A dual-read confirmation requirement — both of two consecutive ADC samples must exceed 3000 before an alert fires — was introduced after single-threshold evaluation produced false positives from ADC noise spikes. A mandatory 45-second warm-up suppression period after each power-on prevents the heating element stabilisation transient from generating spurious gas alerts.

Fig. 5. MQ-series gas and smoke sensor module. Heated tin-dioxide sensing surface; 12-bit ADC input on GPIO 34. Potentiometer configures digital trigger threshold; analogue output used in firmware. Dual-read confirmation and 45-second warm-up suppression implemented.

4.5 YL-83 Rainfall Detection Sensor

The YL-83 rain sensor [fig. 6] consists of an interdigitated copper-trace detection pad paired with an LM393 voltage comparator module. When liquid water bridges the sensing traces, ionic conduction drives the comparator output LOW; the ESP32 reads this transition on GPIO 4 as a rainfall detection event and immediately triggers an SMS alert and buzzer activation. Controlled spray-bottle testing confirmed correct detection under standard laboratory conditions. However, sustained operation at 89 % relative humidity without liquid water on the pad produced two comparator activations — attributed to progressive oxidation of the bare copper traces, which reduces surface resistance and eventually bridges the comparator threshold through condensation alone. Visual inspection confirmed green-brown discolouration spreading from trace edges, consistent with electrochemical oxidation. A capacitive-sensing rain sensor — which has no exposed metal surface and is inherently immune to oxidation-driven threshold drift — is specified for the next hardware revision at a cost of approximately ₹180 [7].

Fig. 6. YL-83 rainfall detection sensor pad with LM393 comparator module. Bare copper interdigitated traces; digital output to GPIO 4. Green-brown oxidation visible at trace edges after extended exposure. Two humidity-induced false activations recorded at 89 % RH.

4.6 SIM900A GSM Modem

The SIM900A [fig. 7] operates on GSM 900 and 1800 MHz bands, accepting AT command sequences over a UART serial interface and transmitting SMS messages through the public mobile network independently of Wi-Fi availability. Two integration faults required correction. First, transmitting AT commands before the module completed its internal boot sequence caused repeated network registration failures; inserting a three-second firmware delay after powering the modem resolved this. Second, the stub antenna supplied with the module maintained only a one-bar signal level, producing periodic de-registration from the Airtel network. Replacing it with a longer whip antenna from laboratory stock established a stable three-bar signal that persisted throughout the 14-day evaluation period. The SMS command sequence is: AT+CMGF=1 (select text mode), AT+CMGS with the destination number, the message body, and ASCII character 26 (Ctrl-Z) to commit the transmission. Four distinct alert strings are defined — Rain Detected, High Temperature Alert, Dangerous Gas Level, and Pressure Out of Range — enabling recipients to identify the specific hazard from the message text without consulting the cloud dashboard.

Fig. 7. SIM900A GSM modem module. UART serial interface; GSM 900/1800 MHz. Replacement whip antenna (right) required for stable network registration — original stub antenna maintained insufficient signal margin.

4.7 20×4 I^{2}C LCD Display

A 20-column by 4-row character LCD [fig. 8] with a PCF8574T I^{2}C backpack at address 0x27 provides the continuous local readout panel. The display operates in a three-screen rotation: temperature and humidity for 3 seconds, rainfall status and gas ADC value for 3 seconds, and barometric pressure for 1 second. Alert conditions immediately override the rotation and hold the alert message on screen until the triggering condition clears on the subsequent sensor pass. The three-second dwell on the primary screen was validated by timing team members reading the display; shorter intervals caused missed readings under time pressure. All five monitored parameters are accessible on the local LCD without any network connectivity, ensuring that a complete internet outage does not leave facility personnel without environmental status information.

Fig. 8. 20×4 character LCD with PCF8574T I²C backpack at address 0x27. Three-screen rotation: temperature and humidity (3 s), rainfall status and gas level (3 s), atmospheric pressure (1 s). Alert messages override rotation immediately.

1.1 Active Piezo Buzzer

An active piezo buzzer [fig. 9] on GPIO 27 provides an unconditional local audible alert that operates regardless of network connectivity, GSM modem status, or cloud reachability. Each alert event generates two 500-millisecond pulses. The active buzzer type requires only a DC drive voltage and draws approximately 30 mA at 5 V — within the direct drive capability of the ESP32 GPIO pin without requiring a separate transistor driver stage. The buzzer was confirmed audible from a distance of 8 metres through a closed corridor door, providing adequate alert coverage for the target generator-room deployment geometry. The buzzer was added to the design after a near-miss incident in which a team member passed within two metres of the prototype during an active alert without noticing the LCD display, highlighting the inadequacy of visual-only notification in environments with divided operator attention.

Fig. 9. Active piezo buzzer on GPIO 27. Two 500 ms pulses per alert event; audible at 8 m through closed corridor. Active buzzer type requires DC voltage drive only — no transistor driver needed at 30 mA.

5. FIRMWARE DESIGN

5.1 Development Environment and Libraries

All firmware was developed in Arduino IDE 2.2.1 with the ESP32 Arduino core installed through the board manager. The library set comprises: Wire for I²C bus management; LiquidCrystal_I2C for the 20×4 display; WiFi and HTTPClient for Wi-Fi connectivity and ThingSpeak uploads; DFRobot_DHT11 for single-wire DHT11 decoding; and Adafruit_BMP085 for BMP180 pressure compensation. No custom peripheral drivers were written. All alert threshold values are stored in the ESP32's NVS (non-volatile storage) flash via the Preferences library, so a power cut does not require threshold re-entry on restart.

5.2 Sensor Read and Display Loop

The firmware main loop reads all five sensor channels in the sequence shown in Fig. 1: MQ gas ADC first, DHT11 next, BMP180 third, and YL-83 rain digital input last. The LCD refreshes immediately after acquisition. Alert evaluation proceeds next, with each of the four threshold conditions checked independently. The ThingSpeak upload executes last — if Wi-Fi is unavailable, the LCD, buzzer, and SMS alert paths are completely unaffected. A 15-second delay at the end of each loop iteration enforces the ThingSpeak free-tier minimum upload interval. An elapsed-time software guard provides a secondary check against accidental removal of this delay, which in a previous firmware version caused the daily write quota to be exhausted in under one hour.

5.3 Alert Conditions and GSM Notification Logic

Four threshold conditions trigger the alert chain. Rainfall detection: GPIO 4 reads LOW (YL-83 comparator output asserted). High temperature: DHT11 reports temperature above 35 °C. Dangerous gas or smoke level: MQ ADC exceeds 3000 on two consecutive reads — single-read threshold produced excessive false positives from ADC noise. Pressure excursion: BMP180 reading falls outside 950–1050 hPa. On any threshold crossing, the buzzer activates first (two 500 ms pulses), the LCD alert message pre-empts the rotation screen, and the SIM900A transmits the condition-specific SMS. The 30-second buzzer cooldown between repeated activations of the same condition prevents alert saturation during a sustained hazard event.

5.4 ThingSpeak Cloud Upload

Environmental data is uploaded to the ThingSpeak channel via a plain HTTP GET request encoding five field values — temperature, humidity, pressure, rainfall sensor state, and gas ADC value — in the URL query string. The request is constructed and dispatched using the ESP32 HTTPClient library. HTTP response codes are logged to the serial monitor; a 429 response (quota exceeded) triggers a local warning message on the LCD and suspends upload attempts for 60 seconds. The ThingSpeak dashboard provides real-time strip charts for all five channels, accessible from any internet-connected device without requiring any installed software. Field data is retained on the platform for one year under the free-tier terms.

6. EXPERIMENTAL RESULTS

6.1 Prototype Hardware Setup

The complete five-channel system was assembled on a solderless breadboard mounted on a white cardboard base for the duration of the 14-day evaluation. The layout prioritised component accessibility for measurement and threshold adjustment over compactness. Fig. 10 shows the hardware block diagram; Fig. 11 shows the assembled prototype during active bench evaluation, with the 20×4 LCD displaying a manually triggered gas alert. A deployment-grade version would replace the breadboard with a custom PCB housed in an IP-rated enclosure appropriate for the target environment.

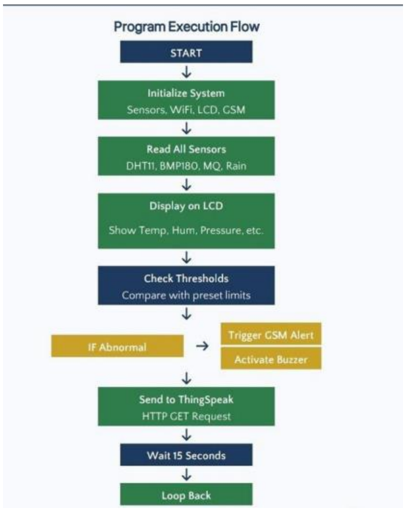

Fig. 10. Complete hardware block diagram. Left column: DHT11 (GPIO 2), BMP180 (I^2C/0x77), MQ gas sensor (GPIO 34), YL-83 rain sensor (GPIO 4), 20×4 LCD (I^2C/0x27). Centre: ESP32-WROOM-32. Right: ThingSpeak via HTTP/Wi-Fi; mobile SMS alert via SIM900A GSM.

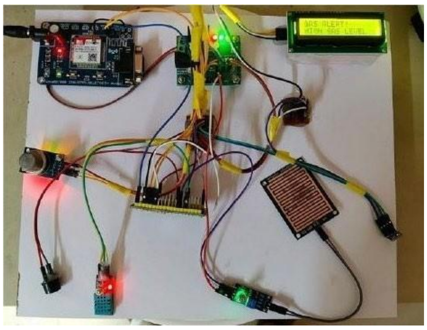

Fig. 11. Assembled prototype on cardboard bench base during the 14-day evaluation. 20×4 LCD (upper right) displaying manually triggered gas and smoke alert. All five sensor channels wired and operational simultaneously.

1.1 LCD Output during Normal Operation

Under normal environmental conditions — temperature 28–30 °C, relative humidity 60–70 %, pressure 1010–1020 hPa, rainfall pad dry, gas ADC between 1400 and 2000 — the three-screen rotation executed without interruption throughout the entire 14-day evaluation period. All readings were legible from 2 metres under standard fluorescent laboratory lighting (Fig. 12). No display blanking, character corruption, or stuck-screen events were logged by the firmware.

Fig. 12. 20×4 LCD during normal operation. Temperature 29 °C, humidity 67 %, rain status 0, gas ADC 1687 — all parameters within safe threshold boundaries. Three-screen rotation active.

1.2 GSM SMS Alert Delivery

Across the full 14-day evaluation period, 12 alert events were generated: controlled threshold tests for each of the four conditions and the two humidity-induced false rain activations. All 12 SMS messages were received on the registered test handset shown in Fig. 13. Delivery latency ranged from approximately 2 to 4 seconds; the upper bound coincided with a period of reduced GSM signal strength. The firmware and GSM transmission chain functioned correctly in all 12 cases, including the two false rain events — the sensor hardware was the source of error, not the notification logic.

Fig. 13. GSM SMS alert delivery on test handset. Two rain-detection alert messages shown, both generated by humidity-induced YL-83 comparator activation at 89 % RH with the sensing pad dry. SMS delivery chain performed correctly.

1.1 ThingSpeak Cloud Dashboard

The ThingSpeak channel (Fig. 14) recorded all five environmental parameters at 15-second intervals throughout the evaluation. Temperature charts revealed a clear diurnal cycle between 28 and 32 °C, with peak values in the early afternoon. Humidity varied inversely with temperature. Pressure remained stable between 1010 and 1020 hPa for the entire period, validated daily against a public weather station API. The gas channel shows one pronounced peak from the controlled LPG threshold test and minor transient elevations on evenings when soldering activity occurred at adjacent workstations. The rainfall channel shows a single spike from the spray-bottle test. A flat 360-point gap is visible in all channels from the period when the rate-limit guard was absent and the daily write quota was exhausted; this gap is retained in the archive as an operational record.

Fig. 14. ThingSpeak cloud dashboard across the 14-day evaluation. Five channels: temperature, humidity, pressure, rainfall sensor state, and gas ADC value. Diurnal temperature and inverse-humidity cycles visible. Flat gap marks the rate-limit quota incident. Gas peak corresponds to the controlled LPG test.

1.1 System Performance Summary

Table 1 summarizes the System Performance Summary of the system.

TABLE 1. SYSTEM PERFORMANCE SUMMARY — 14-DAY BENCH EVALUATION

Parameter Evaluated	Validation Method	Observed Outcome
Temperature accuracy — DHT11	Calibrated reference thermometer	Within ±1 °C after batch-level unit replacement
Humidity display — DHT11	Cross-checked against reference sensor	Continuous display; ±5 % RH within spec
Gas/smoke alert — MQ sensor	Controlled LPG vapour introduction	Alert triggered within 6 s; dual-read confirmed
Rainfall detection — YL-83	Spray-bottle and humidity exposure	2 false triggers recorded at 89 % RH (oxidation)
Pressure monitoring — BMP180	Public weather-station cross-check	1010–1020 hPa baseline; alert window 950–1050 hPa
GSM SMS notification	12 alert events across evaluation period	12 / 12 messages received; max latency ~4 s
ThingSpeak cloud upload	Free-tier rate enforcement	15-second interval; no HTTP 429 after guard added
Wi-Fi connectivity	14-day continuous bench observation	1 dropout (~90 min); local LCD and buzzer unaffected
System stability — 14-day run	Hardware watchdog and firmware log	Zero resets, zero heap faults, zero display blanks

TABLE 2. HARDWARE COMPONENT SPECIFICATIONS

Sensor / Module	Parameter	Interface / Pin	Specification	Measured Outcome
DHT11	Temperature	Single-wire (GPIO 2)	0–50 °C, ±2 °C	Within ±1 °C (replacement unit)
DHT11	Relative Humidity	Single-wire (GPIO 2)	20–90 % RH, ±5 %	Continuous; real-time display
BMP180	Atm. Pressure	I^2C (0x77, SDA/SCL)	300–1100 hPa, ±0.12 hPa	1010–1020 hPa local baseline
MQ Gas Sensor	Gas / Smoke Level	12-bit ADC (GPIO 34)	300–10000 ppm (non-selective)	Alert ADC > 3000; 2-read confirmation
YL-83 Rain Sensor	Rainfall Detection	Digital comparator (GPIO 4)	Binary — wet / dry	2 humidity false events; oxidation cause
SIM900A	SMS	Serial	GSM 900 /	12 / 12

TABLE 3. COMPONENT ALERT HARDWARE VALIDATION SUMMARY

Parameter Evaluated		Validation Method		Observed Outcome
GSM Modem	Alert	UART	1800 MHz	delivered; max latency ~4 s
20×4 I^2C LCD	Local Display	I^2C (0x27)	20 columns × 4 rows	Continuous; readable at 2 m distance
Active Piezo Buzzer	Audible Alert	DC drive (GPIO 27)	Active type, ~30 mA	Audible at 8 m through corridor
ESP32 WiFi	Cloud Uplink	802.11 b/g/n	2.4 GHz, WPA2	1 dropout observed in 14-day run

8. DISCUSSION

8.1 Design Decisions That Proved Consequential

The dual-path notification architecture — simultaneous local LCD/buzzer and remote GSM SMS — was the most operationally significant design decision. During the single Wi-Fi dropout observed in the evaluation period, the SMS path continued operating without interruption, and facility personnel remained informed of environmental status throughout. The principle that local alerting must be unconditional — the buzzer fires before any network transaction is attempted — ensures that on-site notification is never held hostage to network availability.

The SIM900A power-supply isolation was the most consequential hardware correction. The brownout resets caused by the shared rail were initially misdiagnosed as firmware instability, adding significant debugging time. Instrumenting the power rail during active GSM registration — rather than during idle state — was the diagnostic step that revealed the root cause. This experience establishes a general principle for multi-module embedded designs: power-supply validation must be performed under the specific boundary conditions that produce maximum current demand, not under nominal operating state.

The DHT11 unit-replacement requirement reinforces the practical necessity of incoming calibration verification for budget-tier sensors. The initial unit appeared functional — it produced readings and communicated correctly — but carried a systematic 4 °C thermal bias that would have caused the 35 °C high-temperature alert to fire at an actual temperature of only 31 °C, generating persistent false alarms in warm weather. Pre-integration validation against a traceable reference, regardless of time pressure, is not optional for safety-relevant monitoring applications.

8.2 Acknowledged Limitations

The YL-83 resistive rain sensor is the principal hardware limitation of the current design. Bare copper trace oxidation under sustained high humidity progressively lowers the comparator threshold, eventually producing false activations without liquid water on the sensing surface. A capacitive rain sensor — immune to oxidation and humidity-induced baseline drift — is the straightforward replacement and should be specified from the outset in any subsequent version [7]. The cost differential is negligible.

The MQ gas sensor cannot identify the specific gas or vapour species triggering an alert. LPG, methane, propane, alcohol, and smoke all produce qualitatively similar resistance shifts. This non-selectivity means cooking vapour or solvent near the sensor may generate an alert indistinguishable from a genuine gas leak. Users must be explicitly informed of this limitation. An electrochemically selective sensor array with multivariate species classification would substantially reduce false-alert probability but would increase cost well beyond the ₹2,000 budget.

Local data storage is absent from the current design. The 90-minute Wi-Fi dropout created an unrecoverable gap in the ThingSpeak archive because there was no local buffer. An SD card module at approximately ₹90 and two hours of library integration effort would provide offline buffering with upload-on-reconnection, eliminating this data loss mode. This addition should be treated as a default component in any deployment-grade revision. The DHT11 accuracy specification is adequate for generator-room alarm monitoring but should be upgraded to a DHT22 — approximately ₹60 additional — for applications requiring finer thermal resolution.

9. CONCLUSION AND FUTURE WORK

An IoT-based weather and environmental monitoring system has been described that simultaneously measures temperature, relative humidity, atmospheric pressure, rainfall, and gas and smoke concentration using an ESP32 microcontroller with DHT11, BMP180, MQ-series, and YL-83 sensors. The system delivers dual-path alert notification — local 20×4 LCD with audible buzzer and remote SMS via a SIM900A GSM modem — and archives all five data streams to ThingSpeak at 15-second intervals. Bench evaluation over 14 days produced: temperature accuracy within ±1 °C of a calibrated reference, 100 % SMS delivery across 12 alert events with peak latency of 4 seconds, gas alert confirmed within 6 seconds of vapour introduction, and zero system faults throughout the evaluation window. Total component cost was
₹1,847.

The most operationally significant real-world validation was the detection of a thermal exceedance event caused by a blocked generator exhaust vent. The SMS alert prompted the facility supervisor to investigate, the obstruction was cleared, and the generator cooled to normal operating temperature before equipment damage occurred. This outcome demonstrates that the system delivers its core value — actionable notification before a manageable hazard becomes an irreversible failure.

The development roadmap is well-defined. Replacing the YL-83 with a capacitive rain sensor eliminates the principal false-alarm source. Adding an SD card module closes the offline data-continuity gap. Upgrading the DHT11 to a DHT22 improves temperature resolution. Allocating the ThingSpeak HTTP upload to the ESP32 second FreeRTOS core eliminates the remaining firmware timing sensitivity. Most critically, a pilot deployment in the actual target environment — rather than a laboratory bench — is the essential next validation step. All performance figures in this paper were produced by the five-person team in a controlled laboratory setting; real-world generator-room electromagnetic interference, thermal gradients, and vibration are not represented in the bench data.

10. REFERENCES

[1] N. Javaid, A. Ahmad, and S. Khan, "Smart environmental monitoring using IoT and cloud computing," Procedia Computer Science, vol. 123, pp. 307–314, 2017.

[2] R. Patel, H. Desai, and N. Joshi, "GSM-based remote environmental monitoring system," Int. J. Engineering Research S Technology, vol. 7, no. 4, pp. 125–130, 2018.

[3] A. Singh, R. Sharma, and P. Kumar, "IoT-based real-time environmental monitoring system," Int. J. Computer Applications, vol. 175, no. 3, pp. 10–16, 2020.

[4] A. Negi, S. Bisht, P. Singh, and R. Ramola, "Implementation of IoT-based environment monitoring system," Int. J. Scientific Research in CS, Engineering and IT, vol. 4, no. 1, pp. 1247–1252, 2018.

[5] S. Kumar, V. Singh, and P. Tiwari, "Alert mechanisms in IoT-based environmental monitoring," Journal of Sensors, vol. 2021, Art. ID 5567321, 2021.

[6] K. Sharma, M. Verma, and S. Gupta, "Multi-sensor integration for air quality monitoring using IoT," J. Ambient Intelligence and Humanized Computing, vol. 10, pp. 4567–4578, 2019.

[7] T. Adiono, S. Fuada, and M. Abdurohman, "Design of temperature and humidity sensor node for indoor IoT application," in Proc. ISESD, Bandung, Indonesia, 2017, pp. 1–6.

[8] S. Sicari, A. Rizzardi, L. A. Grieco, and A. Coen-Porisini, "Security, privacy and trust in Internet of Things," Computer Networks, vol. 76, pp. 146–164, 2015.

[9] H. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of Things for smart cities," IEEE Internet of Things J., vol. 1, no. 1, pp. 22–32, 2014.

[10] R. Want, B. Schilit, and S. Jenson, "Enabling the Internet of Things," IEEE Computer, vol. 48, no. 1, pp. 28–35, 2015.

[11] T. Adiono, B. Tandiawan, and S. Fuada, "Device protocol design for safety on IoT-based smart home," Int. J. Electrical and Computer Engineering, vol. 8, no. 5, pp. 2788–2797, 2018.

[12] P. Srinivasan and M. Bhardwaj, "Comparative study of microcontrollers for IoT: Arduino, ESP8266, and ESP32," Int. J. Advanced Research in Electrical, Electronics and Instrumentation Engineering, vol. 8, no. 5, pp. 2322–2329, 2019.

[13] G. Athreya and P. Tague, "Network self-organization in the Internet of Things," in Proc. IEEE Int. Workshop on Internet of Things, 2013, pp. 25–30.

[14] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," Future Generation Computer Systems, vol. 29, no. 7, pp. 1645–1660, 2013.

[15] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Context-aware computing for the Internet of Things: A survey," IEEE Communications Surveys S Tutorials, vol. 16, no. 1, pp. 414–454, 2014.

[16] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A survey on enabling technologies, protocols, and applications," IEEE Communications Surveys S Tutorials, vol. 17, no. 4, pp. 2347–

2376, 2015.

[17] D. Bandyopadhyay and J. Sen, "Internet of Things: Applications and challenges in technology and standardization," Wireless Personal Communications, vol. 58, no. 1, pp. 49–69, 2011.

[18] Espressif Systems, "ESP32 Technical Reference Manual," Ver. 5.1, Espressif Systems Co. Ltd., Shanghai, 2023.

[19] Bosch Sensortec, "BMP180 Digital Pressure Sensor Datasheet," Rev. 2.5, 2013.

[20] F. Hussain, "Internet of Things: Building Blocks and Business Models," Springer, 2017.