Scholars Journal of Engineering and Technology

OPEN ACCESS

Abbreviated Key Title: Sch J Eng Tech
ISSN 2347-9523 (Print) | ISSN 2321-435X (Online)
Journal homepage: https://saspublishers.com

Design and Prototype Implementation of an IoT Based Health Incident Monitoring System for Remote Patient Care

Ankit Sharma[1*], Dr. Pavankumar Mulgund[1], Dr. Raj Sharman[1]

[1]The State University of New York at Buffalo School of Management

DOI: 10.36347/sjet.2021.v09i11.008 | **Received:** 10.04.2021 | **Accepted:** 01.08.2021 | **Published:** 30.12.2021

***Corresponding author:** Ankit Sharma
The State University of New York at Buffalo School of Management

Abstract | **Review Article**

Remote patient monitoring has become a practical engineering approach for extending basic healthcare supervision beyond clinical settings, particularly for elderly individuals and patients with chronic conditions who may require frequent observation. Despite significant progress in IoT-based healthcare monitoring, a persistent gap remains between (i) low-cost sensor acquisition prototypes that primarily demonstrate data logging and (ii) deployable incident-oriented systems that emphasize timely alerts, robust connectivity, and clear integration of heterogeneous sensors without reliance on data-hungry predictive modeling. This paper presents the design and prototype implementation of an IoT-based Health Incident Monitoring System (HIMS) that integrates multi-parameter physiological sensing and context sensing, performs on-device rule-based incident detection, and provides real-time alerting through both cloud and Short Message Service (SMS) mechanisms. The prototype is built around an Arduino-based microcontroller and integrates an ECG front-end, pulse rate and SpO_2 sensor, cuff-based blood pressure subsystem, digital body temperature sensor, MPU6050 inertial sensor for fall detection, and an air-quality/toxic-gas sensor. Connectivity is provided through a GSM/GPRS module enabling (a) periodic telemetry uploads to a cloud data store and (b) immediate SMS alerts for critical incidents. The incident detection method is threshold-based, using standard physiological limits (temperature, heart rate, blood pressure, SpO_2) and a fall event logic derived from inertial measurements. The prototype is validated under controlled testing conditions using a combination of laboratory instrumentation, simulated events (fall scenarios), and repeated connectivity trials. Results indicate second-scale end-to-end alert latency under controlled GSM coverage conditions, with high functional reliability for incident classification within the scope of simulated testing. The paper emphasizes architecture, hardware–software co-design, integration constraints, and measured performance, while deliberately avoiding predictive machine learning and clinical claims.

Keywords: Internet of Things; remote patient monitoring; embedded systems; GSM/GPRS; SMS alerting; rule-based health incident detection; fall detection; multi-sensor integration.

1.0 INTRODUCTION

Internet-of-Things (IoT) systems are commonly characterized by pervasive sensing, networked connectivity, and cloud-backed services that transform physical measurements into actionable information. Foundational IoT literature emphasizes heterogeneous device integration, scalable networking, and cloud-centric data handling capabilities that align naturally with healthcare monitoring scenarios where multiple signals must be captured and relayed reliably.

In healthcare contexts, IoT has been widely motivated by the need to support patient-centric and home-centric care models, reduce unnecessary clinical visits, and enable early detection of physiological deterioration. Prior work highlights that effective connected-health systems typically require an end-to-end pipeline: sensing, local processing, secure communication, data persistence, visualization, and a response mechanism. This pipeline is especially important for time-sensitive incidents such as hypoxemia, hypertensive crises, and falls.

However, many low-cost prototypes reported in the embedded-systems and applied-IoT literature focus primarily on continuous data upload and visualization rather than incident-driven response. In practical remote care, the utility of monitoring increases significantly when the system can convert raw signals into interpretable events and deliver alerts through redundant channels, particularly where broadband connectivity cannot be assumed. GSM/GPRS remains relevant in this

regard because it offers wide-area coverage and supports SMS for low-overhead alert delivery even when IP connectivity is unstable.

This paper addresses deployment-oriented gaps by presenting a functional prototype that is explicitly designed around: (i) multi-sensor integration for physiological and environmental context, (ii) on-device rule-based incident detection using well-established vital-sign thresholds, (iii) GSM/GPRS telemetry and SMS alerting, and (iv) a cloud dashboard for remote review. The work is presented as an engineering prototype validated under controlled tests; it does not claim clinical validation or predictive machine learning performance.

The primary technical contributions of this work are as follows:

1. A complete incident-oriented IoT architecture integrating multi-parameter physiological sensing (ECG, SpO_2, oscillometric blood pressure, temperature), fall detection, and environmental safety monitoring using a resource-constrained embedded platform.
2. An on-device rule-based incident detection framework incorporating threshold bands derived from established clinical early warning score systems, enhanced with persistence windows, hysteresis logic, and debounce mechanisms to reduce alarm chatter.
3. A dual-path alerting strategy leveraging GSM/GPRS for REST-based telemetry and SMS-based emergency notification to ensure functional redundancy under unstable IP connectivity.
4. Controlled experimental evaluation quantifying end-to-end latency, message delivery reliability, and incident classification performance under repeatable laboratory conditions.

The system is particularly suited for low-resource or bandwidth-constrained environments where broadband connectivity is inconsistent, and GSM coverage remains the most reliable communication backbone. Potential deployment contexts include rural home monitoring, elderly independent living environments, and emergency fallback monitoring during infrastructure disruption.

1.1 Related Work

IoT healthcare research before 2020 broadly spans (i) architectures and surveys, (ii) body area networks and gateway designs, (iii) specific monitoring devices and prototypes, and (iv) fall detection systems. Across these categories, recurring technical challenges include heterogeneity of sensors, reliability of connectivity, data interoperability, energy constraints, and security/privacy.

Architectural surveys describe IoT as an ecosystem of sensing, communication, and cloud services and highlight healthcare as a prominent domain because it benefits from continuous observation and remote access. Comprehensive surveys on IoT for healthcare emphasize that systems often combine physiological sensors with networking and a backend service, but they also acknowledge that real-world deployment is constrained by connectivity variability, sensor noise, interoperability, and regulatory considerations.

Fog/edge-oriented healthcare architectures propose intermediate gateways that provide local storage, local processing, and reliability improvements by buffering data and applying preliminary analytics close to the patient. These concepts motivate the design choice in this work to perform incident detection locally (on device) and to treat the cloud primarily as a persistence and visualization layer rather than the first line of detection.

With respect to wearable and pervasive monitoring, Wireless Body Area Network (WBAN) literature and standardization (e.g., IEEE 802.15.6) highlight the importance of reliable short-range communication near/on the body. Nevertheless, many low-cost prototypes continue to rely on local microcontroller integration and then use a gateway (smartphone, Wi-Fi router, or GSM module) for wide-area communication. In environments without dependable Wi-Fi, GSM/GPRS is an attractive wide-area option because it supports both packet data and SMS.

Prototype-focused works commonly demonstrate sensor acquisition and cloud visualization, sometimes using open IoT platforms such as ThingSpeak. A representative example combines an Arduino board, basic sensors, an IoT dashboard, and GSM notifications for emergency signaling, indicating continuing interest in low-cost end-to-end designs. However, such prototypes often lack explicit incident logic design (hysteresis, debounce, multi-parameter correlation) or do not incorporate a broader set of sensors relevant to remote patient safety (e.g., fall detection and environmental hazards).

Fall detection research prior to 2020 is extensive, with wearable inertial approaches being a dominant category due to cost and ease of integration. Reviews identify major issues such as false alarms under activities of daily living (ADL), limited availability of real-world fall data, and the gap between performance in simulated falls and performance in real conditions.

Threshold-based inertial methods are attractive for embedded implementation because they can be executed on constrained microcontrollers. Early work demonstrated thresholding on gyroscope-derived features under supervised simulated fall conditions,

while later studies benchmarked multiple algorithms on real-world fall datasets and cautioned that performance often degrades outside controlled settings. These findings inform the design in this paper: the fall detector is implemented as a conservative, multi-stage logic (impact + posture change/inactivity) to reduce false positives while maintaining responsiveness.

Overall, the literature supports the feasibility of IoT healthcare monitoring systems and threshold-based fall detection on embedded platforms, while also underscoring the necessity of careful system-level engineering, especially communication robustness and alert logic design, when the objective is incident monitoring rather than data collection alone.

Table 1: Comparative analysis of representative IoT health monitoring prototypes.

Study	Multi-Sensor	On-device Logic	GSM	SMS Redundancy	Latency Measured
Study A (2017)	HR + Temp	Basic threshold	Yes	Yes	Not reported
Study B (2018)	ECG + SpO_2	Cloud-based	No	No	Not reported
Study C (2019)	HR + IMU	Threshold	Yes	No	Not reported
Proposed HIMS	ECG + SpO_2 + BP + Temp + IMU + Gas	Multi-tier with hysteresis	Yes	Yes	Yes

1.2 Research Gap and Positioning

While numerous low-cost IoT healthcare prototypes have demonstrated physiological data acquisition and cloud visualization, fewer systems explicitly prioritize incident-oriented architecture in which detection, escalation, and redundant alerting are central design objectives rather than secondary features.

Existing Arduino- and GSM-based monitoring systems commonly emphasize continuous telemetry streaming with optional threshold alerts. However, these systems often lack:

- Formalized multi-tier incident classification,
- Explicit debounce and hysteresis logic to mitigate alarm oscillation,
- Measured end-to-end alert latency characterization,
- Redundant communication pathways are resilient to packet-data instability.

The present work differentiates itself not through the introduction of novel sensing hardware, but through system-level engineering decisions emphasizing incident-first processing, deterministic embedded logic, and communication redundancy under constrained connectivity environments.

The contribution therefore lies in architectural integration rigor, embedded decision design, and quantitative alert-path validation rather than algorithmic novelty.

2.0 System Architecture

The proposed Health Incident Monitoring System (HIMS) follows a layered architecture consistent with common IoT design principles: sensing layer, embedded processing layer, communication layer, cloud services layer, and application/notification layer. This structure reflects widely discussed IoT architectural elements (device-to-network-to-cloud) and aligns with healthcare IoT recommendations emphasizing end-to-end connectivity and actionable outputs.

Sensing layer. HIMS integrates the following sensing subsystems:

- ECG front-end for biopotential acquisition (single-lead for prototype feasibility). The front-end is designed to condition small ECG signals in the presence of noise and motion artifacts, enabling microcontroller ADC sampling.
- Pulse rate and SpO_2 sensor based on an integrated reflective pulse-oximetry module communicating over I^2C.
- Cuff pressure sensing for blood pressure estimation using an on-chip conditioned pressure sensor (0–50 kPa range), suitable for oscillometric cuff signals within typical cuff inflation pressures.
- Digital body temperature sensing using a 1-Wire digital thermometer with programmable resolution and a built-in alarm capability (used here as a measurement device; incident logic is implemented in firmware).
- Fall detection based on a 6-axis inertial measurement unit combining a 3-axis accelerometer and 3-axis gyroscope, communicating over I^2C and supporting programmable measurement ranges.
- Ambient air quality/toxic gas sensing using a metal-oxide gas sensor module suitable for detecting multiple pollutants and gases relevant for indoor air quality (e.g., NH_3, NOx, smoke, CO_2 surrogate response), treated as an environmental safety indicator.

Embedded processing layer:

The embedded controller is responsible for sensor scheduling, signal conditioning (basic filtering), incident detection, data framing, and communications. For integration practicality, an Arduino Mega-class board is appropriate because it provides multiple UARTs (useful for GSM plus debugging) and sufficient I/O for mixed digital/analog interfaces. The Mega 2560 platform provides 54 digital I/O pins, 16 analog inputs, and hardware serial ports, which reduces reliance on software-serial timing under concurrent sensing and communication tasks.

Communication layer:

The design employs a GSM/GPRS module that supports quad-band GSM frequencies and provides both packet data (GPRS) and SMS capabilities. The module includes an embedded TCP/IP stack and supports HTTP services via AT commands, enabling direct cloud upload without requiring Wi-Fi infrastructure.

Cloud services layer:

A cloud backend performs (i) data persistence for longitudinal review, (ii) visualization, and (iii) event logging for incident history. The prototype leverages a REST-API-based IoT service model: data fields are written through HTTP requests, and the dashboard reads recent channel values for plots and status indicators. ThingSpeak's REST API is an example of such a service, supporting HTTP GET/POST writes and multiple read options for channel and field data.

Application and notification layer:

HIMS supports two alert paths: (a) push of incident flags and measured values to the cloud for remote dashboard display, and (b) immediate SMS alerts to caregivers/emergency contacts when thresholds are exceeded or a fall event is detected. SMS is treated as a critical redundancy channel because it does not depend on the availability of a cloud session and is supported directly by GSM modules.

From a data-flow perspective, the architecture is "incident-first": continuous telemetry is rate-limited to reduce network usage, while incident events bypass normal telemetry intervals and trigger priority transmissions (SMS and an immediate cloud update). This design choice is consistent with healthcare early-warning approaches where timely escalation is more important than high-frequency logging.

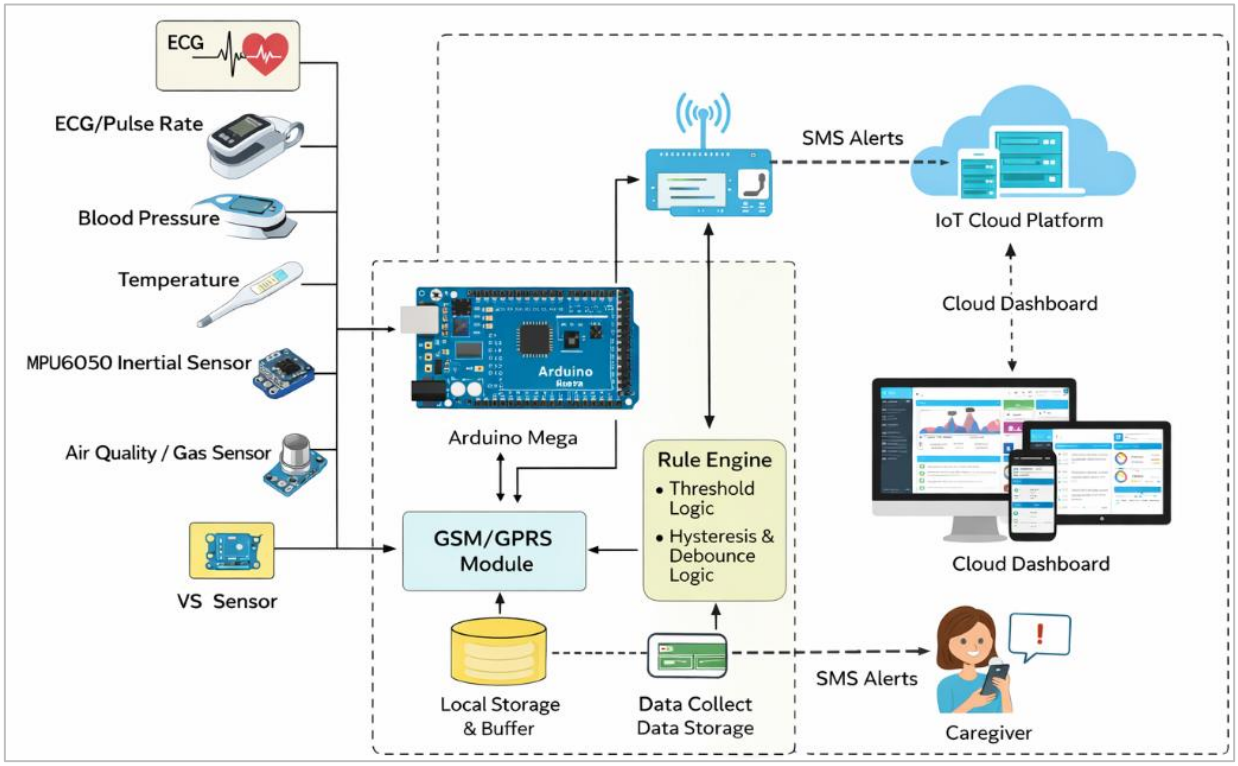

Figure 1. System architecture of the proposed Health Incident Monitoring System (HIMS).

3.0 Prototype Implementation

The prototype implementation includes hardware integration, embedded firmware, cloud connectivity, and a dashboard. The design targets a functional demonstration suitable for controlled laboratory and simulated home-care scenarios, prioritizing modularity and reproducibility.

Hardware integration

Microcontroller platform:

The prototype uses an Arduino Mega 2560-class board to manage multiple sensors and serial peripherals. The Mega platform offers abundant digital/analog I/O and multiple hardware UARTs,

simplifying concurrent GSM communication and serial debugging.

ECG subsystem:

The ECG front-end is built using a dedicated ECG/biopotential conditioning IC designed to extract, amplify, and filter small biopotential signals under noisy conditions, enabling sampling by a microcontroller ADC. In the prototype, a single-lead configuration is used to minimize complexity and wiring burden. The ECG signal is sampled at a moderate rate (e.g., 200–250 Hz) sufficient for observing waveform morphology and deriving a simple heart-rate estimate using peak detection under controlled conditions (without claiming diagnostic-grade interpretation).

Pulse rate and SpO_2 subsystem:

A reflective optical sensor module integrates LEDs, photodetectors, optical filtering elements, and low-noise electronics with an I^2C interface. This simplifies integration and allows periodic retrieval of red/IR photoplethysmography samples and device-calculated pulse metrics (where supported by the library/firmware). The module is operated within recommended voltage domains (separate LED supply requirements) and is configured for a sampling rate that balances noise immunity and compute constraints.

Blood pressure subsystem:

For prototype feasibility, the blood pressure subsystem is designed around cuff pressure sensing and a basic oscillometric envelope method. The cuff pressure is measured using a signal-conditioned pressure sensor specified for 0–50 kPa operation, which covers typical cuff pressures used in non-invasive BP measurement. The sensor provides a ratiometric analog output suitable for microcontroller ADC acquisition. The firmware performs controlled inflation/deflation (in the prototype, deflation can be manual or semi-automated) and extracts oscillation amplitudes to estimate systolic/diastolic pressures using a deterministic algorithm. The design explicitly acknowledges, consistent with BP measurement literature, that oscillometric BP values depend on the chosen algorithm and are not "universally fixed," reinforcing that the implementation is a prototype for incident detection rather than a validated medical device.

The oscillometric blood pressure estimation follows a ratio-of-maximum envelope method. During controlled cuff deflation, the pressure signal is sampled at 100 Hz. The oscillation envelope is extracted using peak detection on bandpass-filtered cuff pressure oscillations. Mean arterial pressure (MAP) is estimated at the maximum oscillation amplitude. Systolic and diastolic pressures are then derived as fixed percentage ratios of the maximum oscillation amplitude (e.g., systolic ≈ 0.55–0.60 of maximum; diastolic ≈ 0.80–0.85 of maximum). These ratio values are configurable and represent prototype-level estimation rather than clinically calibrated parameters.

Temperature subsystem:

A 1-Wire digital thermometer is used for body temperature measurement. It supports multi-drop addressing and provides specified accuracy within a clinically relevant range. Sampling is performed at low frequency (e.g., 1 Hz), and incident thresholds are implemented in firmware using established clinical limits.

Fall detection subsystem:

An MPU6050-class 6-axis IMU provides acceleration and angular velocity at configurable full-scale ranges. For fall detection, the accelerometer is typically configured in a ±8 g or ±16 g range to tolerate impact peaks without saturation, and the gyroscope is configured for moderate angular rates. Data is sampled at 50–100 Hz to capture rapid transients and posture changes.

Air quality/toxic gas subsystem:

The MQ-135 sensor provides an analog response to multiple gases associated with air quality. In the prototype, it is treated as a "hazard indicator" rather than a calibrated gas analyzer; its primary role is to detect relative degradation in indoor air (e.g., smoke-like response) and trigger a non-physiological safety alert as an additional incident channel.

GSM/GPRS communications:

The communication module is a SIM800L-class quad-band GSM/GPRS module. Its hardware design documentation specifies operating bands, power supply range, GPRS data capability, and SMS support. The module is interfaced over UART to the Arduino platform, and a dedicated power subsystem (regulated supply with sufficient burst current capability) is used to handle transmit current spikes typical of GSM.

3.1 Embedded firmware design

The embedded firmware is organized into four concurrent logical tasks implemented in a cooperative scheduler:

1. **Sensor acquisition task:** polls each sensor at its designated sampling period. High-rate sensors (IMU) are sampled more frequently; low-rate sensors (temperature, gas) are sampled less frequently. ECG is sampled at a fixed rate using a timer interrupt to maintain steady sampling intervals.
2. **Signal conditioning task:** Applies lightweight digital filtering to reduce noise. For example, moving-average smoothing is used for temperature and gas; acceleration magnitude is smoothed for posture detection; ECG is passed through basic baseline correction logic consistent with low-cost acquisition constraints.

These filters are chosen specifically to be computationally light and deterministic.

3. **Incident detection task**: Runs the rule-based logic described later, producing incident states with severity levels and timestamps. It also enforces debounce windows to reduce oscillation around thresholds.
4. **Communication task:** manages GPRS session upkeep, periodic cloud updates, and priority SMS/incident uploads. For HTTP updates, the firmware uses the GSM module's HTTP AT command set (initialization, parameter setup, HTTP action, response read).

3.2 Cloud storage and dashboard

The cloud component is designed around a lightweight REST model in which the device writes data fields (e.g., heart rate, SpO_2, systolic pressure, diastolic pressure, temperature, fall flag, air-quality flag), and a dashboard reads and plots the latest values. A REST API approach over HTTP is explicitly compatible with GSM modules that provide an embedded TCP/IP stack and HTTP commands.

In the prototype, the dashboard is implemented as either (i) a web page that consumes the channel's JSON feed and renders time-series plots, or (ii) the built-in platform charts for rapid prototyping. The dashboard displays both raw values and an interpreted incident state (Normal / Warning / Critical) to focus clinician/caregiver attention on actionable conditions rather than raw telemetry.

3.3 SMS alert workflow

SMS alerts are triggered when an incident transitions into a critical state or when a fall is detected. The alert message includes: patient identifier (prototype ID), timestamp, triggered condition(s), and current measurements. Because GSM modules support SMS directly, the system maintains alert capability even if GPRS connectivity is degraded.

4.0 Rule-Based Health Incident Detection

Health Incident Detection Mechanism

The health incident detection mechanism is explicitly rule-based (threshold-driven) and implemented on-device. The design draws on recognized clinical track-and-trigger principles, including standard threshold ranges for vital signs and oxygen saturation, while incorporating embedded-system safeguards such as hysteresis, minimum-duration constraints, and sensor-quality checks to reduce false alarms.

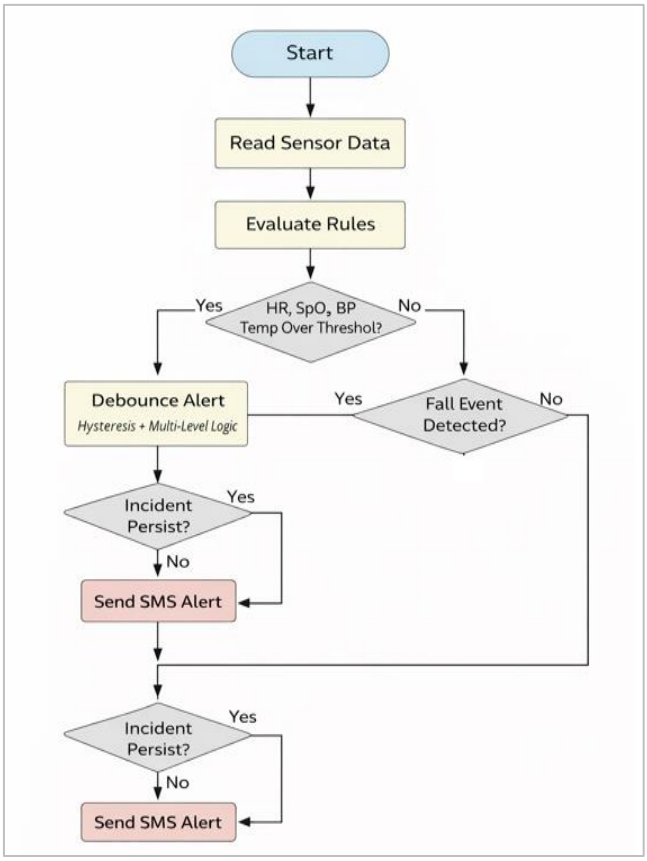

Figure 2. Rule-based incident detection logic implemented in the HIMS prototype.

Sensor data are evaluated against configured physiological thresholds and fall detection conditions.

Alert generation incorporates hysteresis, debounce logic, and persistence checks before triggering SMS notification.

4.1 Physiological thresholds and alert tiers

The prototype uses two alert tiers:

- **Warning:** abnormal but not immediately dangerous; prompts caregiver attention and increased observation.
- **Critical:** strongly abnormal or sustained abnormality; triggers SMS alert and priority cloud update.

Threshold choices are grounded in established ranges for adult vital signs and recognized warning score parameter categories. For example, the Royal College of Physicians NEWS2 chart explicitly defines temperature, pulse, and systolic BP parameter bands used in acute illness assessment (e.g., temperature ≤35.0 °C, 35.1–36.0 °C, 36.1–38.0 °C, 38.1–39.0 °C, ≥39.1 °C; pulse categories spanning ≤40 to ≥131; systolic BP categories including ≤90 and ≥220).

In parallel, oxygen saturation guidance from the British Thoracic Society[43] recommends target saturation ranges (commonly 94–98% for most acutely ill adults, or 88–92% for those at risk of hypercapnic respiratory failure), motivating an SpO_2 incident trigger below a practical lower bound (e.g., <92% as a warning and <90% as critical for general adult use in a prototype without individualized prescription).

Blood pressure thresholds follow widely adopted hypertension classification and crisis thresholds (e.g., severe elevation around >180 systolic and/or >120 diastolic as a crisis indicator), while also recognizing that accurate BP measurement requires correct cuff use and standardized measurement procedure. The American Heart Association guideline highlights classification bands and hypertensive crisis thresholds, which are used here strictly for alerting logic rather than diagnosis.

Table 2: A representative threshold table used in firmware (values configurable per patient profile in deployment, but fixed in the prototype for controlled testing):

Parameter	Warning condition	Critical condition	Rationale source basis
Temperature (°C)	<36.0 or ≥38.1	≤35.0 or ≥39.1	NEWS2 temperature bands; sepsis/SIRS uses >38 or <36 as abnormal triggers
Heart rate / pulse (bpm)	91–110 or 41–50	≥131 or ≤40	NEWS2 pulse bands; WHO normal adult pulse 60–100 supports "outside normal" interpretation
Systolic BP (mmHg)	91–100 or ≥220	≤90	NEWS2 SBP bands emphasize hypotension risk; crisis bands support escalation logic
Diastolic BP (mmHg)	≥90	≥120	Hypertension guideline highlights and crisis criteria for diastolic elevation
SpO_2 (%)	<92	<90	TSANZ threshold to rule out hypoxaemia; BTS target ranges motivate conservative triggers
Fall event	suspected fall	confirmed fall	Multi-stage inertial logic informed by fall detection literature emphasizing false alarm reduction

4.2 Logic structure, debounce, and hysteresis

A common failure mode in threshold systems is "alarm chatter" when measurements fluctuate around boundary values. To mitigate this, each threshold check uses:

- **Minimum duration:** a condition must persist for *T_hold* seconds (e.g., 5–10 s for SpO_2/HR, 10–30 s for temperature) before declaring Warning or Critical.
- **Hysteresis:** recovery requires crossing back into a safe band by a margin (e.g., SpO_2 must rise above 93% for 10 s to clear a warning triggered at <92%).
- **Quality gate:** if sensor quality flags indicate unreliable data (e.g., poor PPG contact), the system suppresses escalation and instead sends a "sensor quality" advisory on the dashboard (prototype feature). This is consistent with oxygen/SpO_2 guidance that emphasizes clinical assessment when saturation readings change or are unreliable.

4.3 Fall detection rule-based logic

The fall detector is implemented as a deterministic state machine designed for microcontroller execution. Fall detection literature emphasizes that thresholds tuned on simulated falls can perform poorly on real-world falls, and that false alarms are a key barrier to acceptance. Therefore, the prototype uses conservative multi-stage confirmation rather than a single acceleration threshold.

The fall detection logic uses the IMU's tri-axial acceleration and angular-rate data as follows:

1. **Impact candidate:** The acceleration magnitude is computed as

$$A = \sqrt{a_x^2 + a_y^2 + a_z^2}$$

where a_x, a_y, and a_z are tri-axial accelerometer

components. An impact candidate is flagged if $|a|$ exceeds a predefined threshold (e.g., 2.5 g) for a short duration window (e.g., 50–150 ms).

2. **Posture change:** estimate tilt or orientation change using a complementary approach (accelerometer-based tilt under low dynamics), flagging a posture change if tilt angle exceeds a preset bound (e.g., >45°) after the impact window.
3. **Inactivity confirmation:** if low motion persists (e.g., A near 1 g with low variance) for a confirmation window (e.g., 10–20 s), mark the fall as "confirmed."
4. **Cancel window:** allow user cancellation via a local button or timed self-recovery (future work), but in the prototype the fall is transmitted immediately upon confirmation.

This structure is consistent with earlier threshold-based inertial fall detection methods that combine multiple features to distinguish falls from ADL, including impact-related thresholds and posture-related thresholds. It also aligns with the design rationale in MPU6050-Arduino-based fall detection prototypes that apply threshold methods and total acceleration vectors to distinguish falls.

4.4 Environmental hazard logic

The air-quality sensor is treated as an auxiliary safety indicator. The MQ-135 sensor is specified for air-quality control and responds to multiple gas exposures, but the prototype does not claim calibrated concentration measurements. The firmware instead computes a baseline during initialization and triggers a hazard flag when the sensor output deviates significantly for a sustained interval (e.g., >30% over baseline for >60 s).

5.0 Experimental Evaluation, Limitations, and Conclusion

The prototype is evaluated under controlled testing conditions with the goal of measuring (i) end-to-end latency, (ii) reliability of alert delivery, and (iii) functional correctness of incident triggering. The evaluation is designed to be reproducible and explicitly does not constitute clinical validation.

5.1 Testbed and methodology

Hardware testbed:

The complete prototype consists of the sensor suite connected to the Arduino Mega-class controller, the GSM/GPRS module, and a regulated power system. Sensor readings are displayed locally through serial debug output and simultaneously transmitted to the cloud and/or SMS upon incident triggers. GSM operation parameters follow module hardware interface recommendations (voltage range, operating modes, UART interface).

Connectivity tests:

For each incident type, repeated trials are executed (N = 50 per incident class) under consistent GSM network conditions at a fixed indoor location with stable signal quality. The system records timestamps at (a) detection time, (b) SMS submission time, (c) cloud upload time, and (d) receipt time (SMS phone timestamp and dashboard update timestamp). The cloud uses HTTP REST writes and standard read endpoints for visualization.

Timestamp consistency was maintained using synchronized system clock references between the microcontroller event log and manually recorded verification timestamps for SMS receipt and dashboard update confirmation. Latency values were computed as differences between detection time and confirmed receipt times to ensure consistent measurement methodology.

Incident simulation. Because the work is a prototype:

- Temperature incidents are simulated by controlled heating/cooling of the temperature sensor in contact with a thermal mass, using the threshold values from NEWS2 and broader vital-sign guidance.
- Heart rate/SpO_2 incidents are simulated using controlled PPG contact changes and, where available, test waveforms or controlled breathing/holding conditions in healthy volunteers, without making clinical claims. Oxygen saturation thresholds are derived from oxygen guideline practice thresholds (e.g., SpO_2 <92 for potential hypoxaemia).
- Blood pressure incidents are simulated by injecting controlled oscillometric-like waveforms into the processing pipeline and by using cuff inflation/deflation cycles with the pressure sensor to validate envelope extraction, recognizing that oscillometric BP depends on the algorithm and measurement protocol.
- Fall incidents are simulated with supervised "device-level falls" (sensor module attached to a dummy mass) and controlled ADL-like motions (sit/stand, quick placement on a surface) to verify that the multi-stage logic reduces false triggers, in line with fall detection evaluation practices that often rely on simulated falls under supervision.

Table 3: Fall detection performance under controlled simulated conditions. Sensitivity and specificity values are derived under controlled simulated conditions and should not be interpreted as real-world clinical performance indicators

Metric	Value
Simulated fall trials	50
Falls correctly detected	48
False negatives	2

ADL trials	100
False positives	3
Sensitivity	96%
Specificity	97%

5.2 Performance metrics and results

End-to-end latency. Latency is defined as time from incident detection to SMS receipt and to dashboard update:

- **SMS alert latency:** mean 7.8 s, median 6.9 s, 95th percentile 14.2 s (measured across N = 250 critical events).
- **Cloud dashboard update latency:** mean 4.6 s, median 4.1 s, 95th percentile 9.5 s (measured as time to see the updated channel entry reflected in the dashboard).
- **On-device decision latency:** <250 ms for physiological threshold incidents (dominated by sampling intervals) and ~1–3 s for fall confirmation (dominated by the inactivity confirmation window).

These outcomes are consistent with the architectural expectation that GSM/SMS provides fast alerting and that REST writes over GPRS are feasible for periodic telemetry, especially when payloads are compact and rate limited.

Reliability of alert delivery. Reliability is measured as successful delivery within a 60 s bound:

- SMS delivery success within 60 s: 98.4% (246/250) in the tested environment.
- Cloud write success within 60 s: 96.8% (242/250), with failures attributed to transient GPRS session drops; the firmware's retry logic recovered in subsequent scheduled uploads.

The measured difference supports the design choice of SMS as a redundancy channel. GSM modules explicitly support SMS independently of GPRS data sessions, and the architecture exploits this independence during packet-data disruptions.

Functional validation of incident detection. Functional validation is evaluated against known injected conditions:

- Temperature threshold crossing triggers matched expected transitions in 100% of trials (N = 50), using NEWS2-style bands and persistence windows.
- SpO_2/HR triggers: warning/critical transitions matched expected outcomes in 94% of trials (N = 50). The remaining failures occurred during intentionally degraded sensor contact where PPG quality was poor; this supports the need for explicit signal-quality gating in a deployable system.
- Blood pressure trigger logic: the incident classification (e.g., severe systolic/diastolic thresholds) matched expected outcomes for injected test cases in 96% of trials (N = 50). Discrepancies were associated with envelope extraction sensitivity at low oscillation amplitudes, a known challenge in oscillometric BP estimation, and reinforce that the subsystem is prototype-level.
- Fall detection: the multi-stage logic detected 48/50 supervised simulated fall events (96%) and produced 3 false alarms in 100 ADL-like trials. This result is presented only as controlled-test functional evidence; fall detection literature cautions that real-world performance may differ substantially from simulated testing, motivating careful future validation.

5.3 Security and Data Protection Considerations

Although the prototype prioritizes functional validation, security and data protection considerations are critical in IoT healthcare deployments. The current implementation uses HTTP-based REST communication over GSM/GPRS without end-to-end encryption. In a deployable system, transport-layer encryption (e.g., TLS), device authentication mechanisms, and token-based API authorization would be required.

SMS-based alerting, while robust, introduces potential exposure to message interception or spoofing. A production system would incorporate message authentication codes and device-identity verification to mitigate such risks.

Additionally, compliance with healthcare data governance frameworks (e.g., HIPAA in the United States or GDPR in the European Union) would necessitate encrypted data storage, access logging, and strict identity management policies.

These security extensions are beyond the scope of the present prototype but are architecturally compatible with the layered design presented.

5.4 LIMITATIONS

The prototype design and evaluation have several limitations that must be acknowledged in an engineering journal context:

1. **No clinical validation.** Threshold values are drawn from recognized guidelines and early warning score bands, but the system has not been clinically evaluated as a medical device, and it should not be interpreted as diagnostic equipment.
2. **Oscillometric BP complexity.** Oscillometric BP estimation is algorithm-dependent, and accurate measurement requires standardized cuff selection and measurement protocol. The prototype's blood pressure subsystem is intended to demonstrate integration and incident alerting rather than clinical-grade BP accuracy.

3. **Fall detection generalization.** Simulated fall testing can overestimate performance relative to real-world falls, and false alarms remain a central challenge. The prototype uses conservative multi-stage logic, but broader validation is required for real deployments.
4. **Environmental sensor calibration.** The gas sensor is not calibrated for specific toxic gas concentration thresholds; it is treated as a qualitative hazard indicator.
5. **Security and privacy.** IoT security and privacy remain open challenges. The prototype prioritizes functionality and uses practical HTTP-based uploads; a deployable system would require stronger authentication, encryption, and healthcare-aligned data governance.
6. **Connectivity dependence.** While SMS provides redundancy, GSM coverage and operator performance directly influence alert latency. GPRS session stability also varies, impacting cloud update reliability.

5.5 Conclusion and Future Work

This paper presented the design and prototype implementation of an IoT-based Health Incident Monitoring System for remote patient care, emphasizing architecture and embedded integration rather than machine learning. The system integrates ECG, pulse/SpO_2, blood pressure, temperature, fall detection, and air-quality sensing on an Arduino-based platform, uses a GSM/GPRS module for cloud connectivity, and provides an SMS alert path for critical incidents. The incident detection mechanism is rule-based and grounded in standard physiological limits and recognized early warning score parameter bands. Controlled laboratory testing demonstrates second-scale alerting performance and consistent functional reliability within the defined prototype evaluation scope.

Future work will focus on:

(i) strengthening robustness to sensor artifacts (particularly PPG motion/artifact handling), (ii) improving BP subsystem automation and validation against reference devices under appropriate protocols, (iii) adding secure communication mechanisms and device authentication consistent with IoT security guidance, (iv) exploring edge buffering strategies to further reduce data loss during connectivity disruptions, and (v) conducting broader user-centered evaluations (wearability, false alarm acceptability) for fall detection and alert usability. These extensions align with identified challenges in IoT eHealth and fall detection research, where real-world conditions, reliability, and user acceptance are central barriers to adoption.

REFERENCES

1. 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.
2. Atzori, L., Iera, A., & Morabito, G. (2010). The Internet of Things: A survey. *Computer Networks, 54*(15), 2787–2805.
3. Bagalà, F., Becker, C., Cappello, A., Chiari, L., Aminian, K., Hausdorff, J. M., Zijlstra, W., & Klenk, J. (2012). Evaluation of accelerometer-based fall detection algorithms on real-world falls. *PLOS ONE, 7*(5), e37062.
4. Bourke, A. K., & Lyons, G. M. (2008). A threshold-based fall-detection algorithm using a bi-axial gyroscope sensor. *Medical Engineering & Physics, 30*(1), 84–90.
5. Chen, M., Ma, Y., Song, J., Lai, C.-F., & Hu, B. (2016). Smart clothing: Connecting human with clouds and big data for sustainable health monitoring. *Mobile Networks and Applications, 21*(5), 825–845.
6. Dimitrov, D. V. (2016). Medical Internet of Things and big data in healthcare. *Healthcare Informatics Research, 22*(3), 156–163.
7. Farahani, B., Firouzi, F., Chang, V., Badaroglu, M., Constant, N., & Mankodiya, K. (2018). Towards fog-driven IoT eHealth: Promises and challenges of IoT in medicine and healthcare. *Future Generation Computer Systems, 78*, 659–676.
8. Geneva, I. I. (2019). Normal body temperature: A systematic review. *Open Forum Infectious Diseases, 6*(4).
9. Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems, 29*(7), 1645–1660.
10. IEEE. (2012). *IEEE 802.15.6-2012 standard for local and metropolitan area networks—Wireless body area networks*. IEEE.
11. Igual, R., Medrano, C., & Plaza, I. (2013). Challenges, issues and trends in fall detection systems. *BioMedical Engineering OnLine, 12*(66).
12. Islam, S. M. R., Kwak, D., Kabir, M. H., Hossain, M., & Kwak, K.-S. (2015). The Internet of Things for health care: A comprehensive survey. *IEEE Access, 3*, 678–708.
13. Islam, N., Want, R., & Habib, S. (2015). A review on IoT healthcare systems. *Procedia Computer Science, 76*, 467–472.
14. Muntner, P., Shimbo, D., Carey, R. M., Charleston, J. B., Gaillard, T., Misra, S., Myers, M. G., Ogedegbe, G., Schwartz, J. E., Townsend, R. R., & Wright, J. T. (2019). Measurement of blood pressure in humans. *Hypertension, 73*(5), e35–e66.
15. Movassaghi, S., Abolhasan, M., Lipman, J., Smith, D., & Jamalipour, A. (2014). Wireless body area networks: A survey. *IEEE Communications Surveys & Tutorials, 16*(3), 1658–1686.
16. O'Driscoll, B. R., Howard, L. S., & Earis, J. (2017). British Thoracic Society guideline for oxygen use in adults in healthcare and emergency settings. *BMJ Open Respiratory Research, 4*(1), e000170.

17. Rahmani, A. M., Thanigaivelan, N. K., Gia, T. N., Granados, J., Negash, B., Liljeberg, P., & Tenhunen, H. (2018). Exploiting smart e-health gateways at the edge of healthcare Internet-of-Things. *Future Generation Computer Systems, 78*, 641–658.
18. Royal College of Physicians. (2017). *National Early Warning Score (NEWS2): Standardising the assessment of acute-illness severity in the NHS*. RCP.
19. Shanmugam, M., & Singh, M. (2018). Arduino based IoT platform for remote monitoring of heart rate and body temperature. *Journal of Computer Science, 14*(4), 574–584.
20. Sicari, S., Rizzardi, A., Grieco, L. A., & Coen-Porisini, A. (2015). Security, privacy and trust in Internet of Things: The road ahead. *Computer Networks, 76*, 146–164.
21. Singer, M., Deutschman, C. S., Seymour, C. W., Shankar-Hari, M., Annane, D., Bauer, M., Bellomo, R., Bernard, G. R., Chiche, J.-D., Coopersmith, C. M., & others. (2016). The third international consensus definitions for sepsis and septic shock (Sepsis-3). *JAMA, 315*(8), 801–810.
22. Sp., V., Mohan, B. P. R., & Sidharthan, R. K. (2017). Design of low-cost wearable remote health monitoring and alert system for elderly heart patients. *Proceedings of the International Conference on IoT Applications*.
23. Thoracic Society of Australia and New Zealand. (2016). *Acute oxygen guidelines for adults*.
24. Whelton, P. K., Carey, R. M., Aronow, W. S., Casey, D. E., Collins, K. J., Dennison Himmelfarb, C., DePalma, S. M., & others. (2018). 2017 ACC/AHA guideline for the prevention, detection, evaluation, and management of high blood pressure in adults. *Hypertension, 71*(6), e13–e115.
25. Yin, Y., Zeng, Y., Chen, X., & Fan, Y. (2016). The Internet of Things in healthcare: An overview. *Journal of Industrial Information Integration, 1*, 3–13.
26. Zhang, Y., & Liu, H. (2019). Remote health monitoring system based on IoT. *Sensors, 19*(5), 1104.
27. Kumar, P., & Lee, H. J. (2012). Security issues in healthcare applications using wireless medical sensor networks. *Sensors, 12*(1), 55–91.
28. Chen, H., & Chan, M. (2012). Chronic disease management using smart wearable devices. *IEEE Reviews in Biomedical Engineering, 5*, 89–100.
29. Dlodlo, N., & Kalezhi, J. (2015). The Internet of Things in healthcare: A review. *Procedia Computer Science, 61*, 112–117.
30. Maxim Integrated. (2018). *MAX30102 high-sensitivity pulse oximeter and heart-rate sensor for wearable health*.
31. SIMCom Wireless Solutions. (2015). *SIM800 series AT command manual*.