IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 7, July 2026

DOI: 10.17148/IJARCCE.2026.15716

Smart Saline Monitoring and Alert System Using IoT

Rambabu Pemula[1], A. Athikah[2], B. Suraj[3], G. Koushik[4], P. Swaroop[5]

Assistant Professor, Department of Information Technology, Vidya Jyothi Institute of Technology (A), Hyderabad, Telangana, India[1]

UG Students, Department of Information Technology, Vidya Jyothi Institute of Technology (A), Hyderabad, Telangana, India[2,3,4,5]

Abstract: Giving a patient IV saline is about as routine as hospital care gets, yet how that drip is tracked has barely changed in decades — a nurse still walks over and eyeballs the bottle. Drip rates are set by hand, and hand-set rates drift, so pinning down exactly when a bag will run dry is more guesswork than science. If nobody catches an empty bottle in time, the consequences are not trivial: blood can back up into the line, the line can become contaminated, or, worse, air can enter the vein and trigger an embolism.The moment fluid drops into the Low band, the system does more than flash a light. A buzzer sounds close to the patient, and a SIM900 GSM/GPRS module sends an SMS directly to the caretaker's phone over the cellular network — a channel that keeps working even when hospital Wi-Fi is congested or down entirely, so the message gets through regardless. To stop staff being bombarded with repeat alerts for a single event, the firmware relies on state flags that cap each threshold breach at exactly one notification.Cost was a deliberate design constraint: every part is a common, off-the-shelf component, and the total build comes in well under the price of a commercial infusion-monitoring pump, which makes the system a realistic option for hospitals and clinics working with tight budgets.

Keywords: Smart Saline Monitoring, Internet of Things, ESP32, Load Cell, HX711 Amplifier, GSM Alert System, IV Fluid Monitoring, Patient Safety, Embedded Healthcare System, Alarm Fatigue Reduction.

I.INTRODUCTION

Intravenous therapy sits at the core of modern patient care — it is how hospitals deliver hydration, nutrients, blood products, and time-critical medication straight into a patient's bloodstream. What is surprising is that, even as diagnostic equipment around it has advanced enormously, the way IV fluid delivery is monitored has stayed largely the same: a nurse or a family member glances at the bottle every so often [1]

[2]. Because how fast a bag empties depends on a manually set drip rate, and drip rates are never perfectly consistent, predicting the exact moment a bottle will run out is genuinely difficult. If it empties before someone notices, the drop in line pressure can pull blood back into the tubing — causing clotting and contamination — or, in a more serious scenario, let air enter the venous system and cause an air embolism, which can be fatal [3].

None of the traditional approaches offer continuous, real-time oversight or automatic alerting. A nurse covering a full ward simply cannot watch every bottle at once, so a bag running low often goes unnoticed until it is already empty — raising the odds of human error, delayed intervention, and extra downstream workload [4], [5].

Hospital administrators keep flagging nurse to-patient ratios as the real bottleneck behind bedside vigilance. National health surveys across several countries report a single nurse handling anywhere from six to ten patients per shift, which leaves little room for the repeated bottle checks that manual monitoring demands [10], [11].

That pressure is even more pronounced in general wards and emergency triage, where many patients are on IV therapy simultaneously and staff attention is stretched thin across competing priorities. The wider rollout of IoT in healthcare has shown that pairing low-cost embedded sensors with wireless or cellular communication can extend a caregiver's reach well past the bedside [12], [13].

II. LITERATURE REVIEW

Rising workloads on hospital staff, combined with the ever-present risk of a missed observation, have pushed researchers toward automated IV monitoring. Purely manual observation can delay the detection of an empty bottle long enough for blood to back up into the line [1], [6], which is exactly the failure mode several IoT-based saline monitoring systems have tried to close by combining sensors, microcontrollers, and wireless links to track fluid levels in real time. Early designs mostly relied on ultrasonic or float/tilt water-level sensors placed near or inside the bottle to estimate the remaining fluid indirectly. Sensor data would be passed to a microcontroller that triggered a local alert or forwarded the status to a remote server, letting caretakers keep an eye on several patients without standing at each bedside [2], [7].

Some groups pushed saline monitoring further by folding in other physiological signals — heart rate, body temperature, oxygen level — to build a broader patient monitoring platform rather than a saline-only tool [5]. Others zeroed in on preventing saline reversal specifically, detecting abnormal backflow in the tube and alerting staff before it causes harm [6]. More recent systems go a step further and take physical control, cutting the saline flow with a solenoid valve once a critical threshold is reached instead of waiting on a staff response to an alert [4], [8]. Communication choices across these systems vary widely too, from Wi-Fi microcontrollers reporting to cloud dashboards such as ThingSpeak, to GSM-based SMS alerts aimed at settings where internet access cannot be counted on [3], [9].

A separate strand of research has looked at MQTT and other lightweight publish subscribe protocols to cut the bandwidth and power cost of continuous cloud reporting, letting an ESP32-class device push periodic status updates without holding open a persistent connection [3]. That helps scalability across hospital-wide deployments spanning dozens of beds, but it still assumes uninterrupted Wi-Fi, which older hospital buildings — or buildings mid-maintenance — cannot always guarantee. Work on wearable and body-sensor networks offers a useful parallel: secure, low-power sensor nodes built for continuous vital-sign capture with encrypted transmission back to a central server demonstrate design patterns — dutycycled sensing, threshold-based event reporting — that map cleanly onto saline-level monitoring too [10], [11].

Surveys of embedded computing for IoT catalogue the broader trade-offs between microcontroller classes, radio technologies, and power budgets that inform hardware choices in bedside medical devices generally [12]. A number of designs also treat the local buzzer purely as a fail-safe, kicking in only if the remote channel — a Wi-Fi push notification or an SMS — is delayed or fails, rather than as the primary alert path [4], [8]. That pattern is appealing because it doesn't require a caretaker to be actively watching a phone or dashboard at the exact moment a threshold is crossed, though it does demand careful state management so the local and remote channels never report conflicting information — a subtlety not always addressed head-on in the literature. Broader health-monitoring surveys covering hospital IoT deployments generally note that combining more sensing modalities produces richer data, but at the cost of a higher component count, more complex firmware, and a larger attack surface for connectivity failures [5], [14].

That trade-off matters directly for saline monitoring: a design that folds heart rate, temperature, and oxygen sensing in alongside fluid-level sensing might make for an attractive full vitals dashboard, but it also multiplies the number of subsystems that all have to work correctly before the core safety alert can be trusted. A further thread of relevant work sits in hospital-management-oriented monitoring platforms, which pull data from many bedside devices into a single administrative dashboard so staff attention can be balanced across a ward [14]. These platforms generally assume each bedside node already senses reliably and just needs a standard reporting interface, so the saline-specific accuracy problem noted above — ultrasonic and float sensors being sensitive to bottle geometry — is inherited rather than solved by simply adding a management layer on top.

Taken together, most existing systems still lean on ultrasonic distance sensing or float/tilt sensors, both of which can be thrown off by bottle shape, sensor placement, or condensation, leading to inconsistent readings. Very few combine a load cell with a dedicated amplifier for high-accuracy weight-based measurement alongside GSM based alerting — which is the gap this work sets out to close.

III. PROPOSED SYSTEM & METHODOLOGY

The proposed Smart Saline Monitoring and Alert System follows a real-time weight monitoring approach: the saline bottle hangs from a load-cell platform, and its exact remaining fluid weight is measured continuously instead of relying on someone walking over to check it visually. That weight data is processed and compared against a threshold to work out whether the bottle is close to empty, cutting down human error and improving patient safety in the process.

3.1 System Architecture

The architecture splits into three sections — input, processing, and output. The input section is the load cell and HX711 amplifier module, which together measure the saline bottle's weight; as fluid depletes, the load cell produces a proportional change in its electrical signal, which the HX711 amplifies and digitises.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 7, July 2026
DOI: 10.17148/IJARCCE.2026.15716

The ESP32 microcontroller is the processing unit — it reads the amplified weight data, applies threshold logic to sort the level into High, Medium, or Low, and drives every output device. The output section covers the 16×2 I2C LCD, colour coded LEDs, a buzzer, and the SIM900 GSM module, which together handle local display, local alerting, and remote SMS notification.

Data moves through the architecture in a strict pipeline: the analog strain-gauge signal is amplified and converted into a 24-bit digital value by the HX711, the ESP32 reads that value over a two-wire serial clock/data interface, applies a stored calibration factor, and clamps the result to the physically meaningful 0-500 ml range before it goes anywhere else in the system. Only this single, clamped volume value is passed downstream to the display, LED, buzzer, and GSM logic, keeping decision-making consistent across all four output channels.

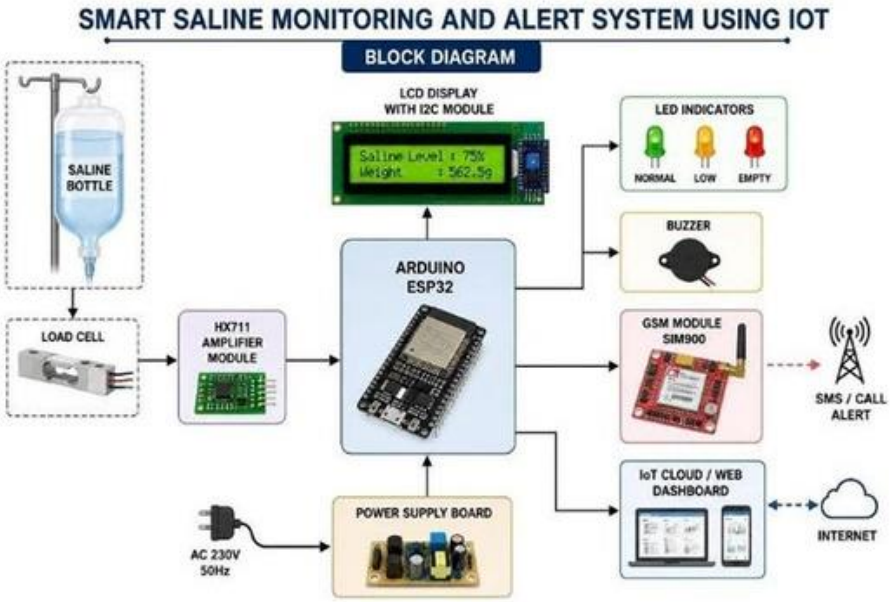

Fig 1. Block Diagram

3.2 Hardware Components

The prototype uses a NodeMCU ESP32 (32 bit dual-core, with built-in Wi-Fi/Bluetooth) as the master controller; a 15 kg strain-gauge load cell as the weight transducer; an HX711 24-bit ADC amplifier for signal conditioning; a 16×2 character LCD with an I2C backpack for bedside display; a SIM900 GSM/GPRS shield for cellular SMS alerts; red, yellow, and green LEDs for Low, Medium, and High status; a 5V active piezoelectric buzzer for local audio alarms; a regulated external power-supply board to absorb the GSM module's current spikes; and a solderless breadboard with jumper wires for prototyping.

3.3 Software Requirements

The firmware runs a simple, deterministic control loop. On power-up, the ESP32 initialises the LCD, the HX711, and the SIM900 module, then performs a tare operation to zero out the empty-hanger weight. In the repeating main loop, it

(i) reads the calibrated weight from the HX711,
(ii) converts that reading to millilitres and clamps it to the valid range,
(iii) works out the current zone by comparing volume against the Medium and Low thresholds,
(iv) updates the LCD and LEDs if the zone has changed,
(v) sounds the buzzer and sends exactly one SMS if the zone has just transitioned into Low, and
(vi) waits a short, fixed polling interval before looping again. This structure keeps response latency low while avoiding redundant peripheral writes and duplicate alerts.

ISSN (O) 2278-1021, ISSN (P) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 7, July 2026
DOI: 10.17148/IJARCCE.2026.15716

IV. IMPLEMENTATION

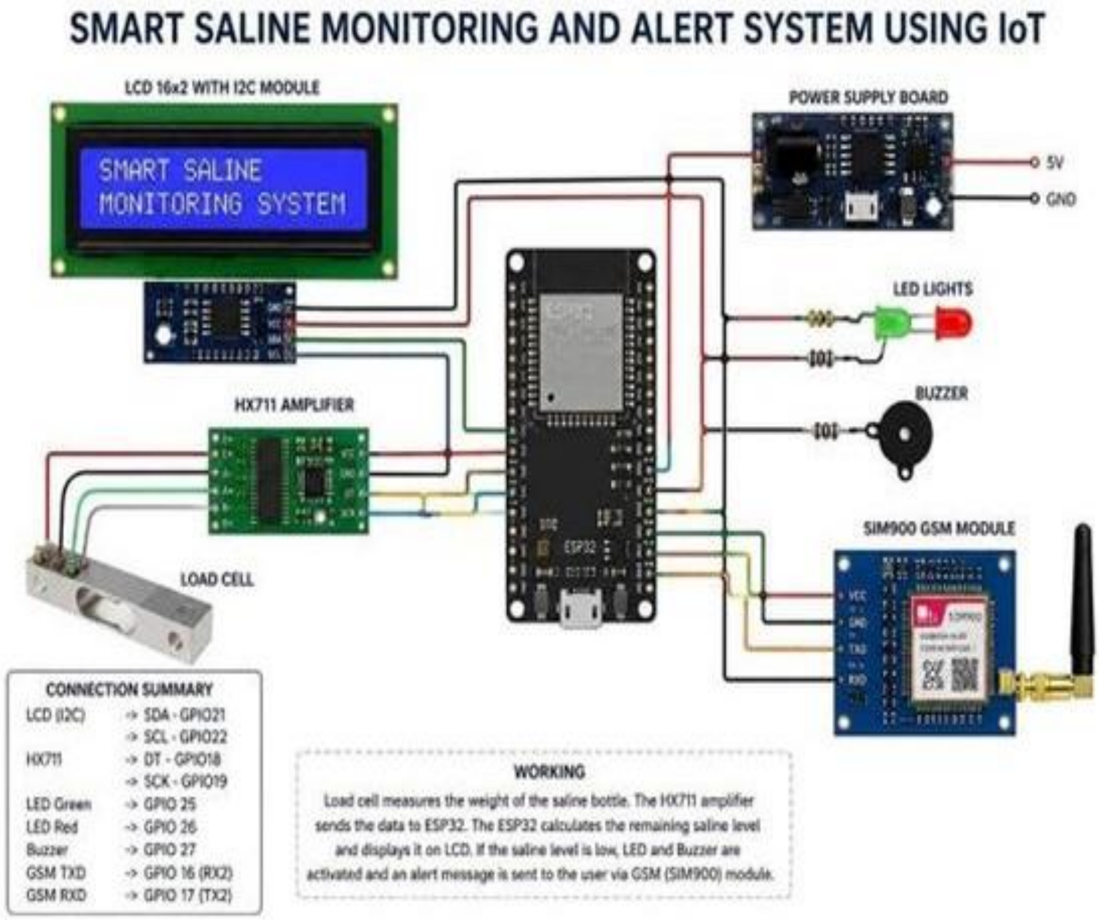

Fig 2. Implementation

Implementation

The firmware is written in Embedded C/C++ using the Arduino IDE. During setup, the ESP32 initialises every peripheral, shows a welcome message on the LCD, and runs a scale.tare() routine to zero the baseline weight of the empty bottle hanger before the saline bottle goes on. The HX711 library handles synchronous clock timing and 24-bit data extraction from the load cell, while the LiquidCrystal_I2C library manages the two wire serial interface to the display. In the main loop, the ESP32 reads the calibrated weight, scales it into millilitres, and clamps the result between 0 and 500 ml. The computed volume is checked against the Low (100 ml) and Medium (300 ml) thresholds using interlocking state flags (stateLow, stateMedium, stateHigh), so each threshold crossing produces exactly one LED update, one buzzer response, and one SMS sent via AT commands over the ESP32's serial UART to the SIM900 module.The firmware also includes basic defensive checks around the HX711 read call: if no valid reading turns up within a short timeout window, the previous valid reading is kept and displayed instead of letting a stale or corrupted value propagate into the zone classification logic. That guards against a transient glitch on the two-wire HX711 interface causing a spurious zone transition, buzzer trigger, or SMS send, any of which would otherwise chip away at confidence in the alerting system.

V. RESULT AND DISCUSSION

The load cell and HX711 amplifier measured the saline bottle's weight accurately, and the ESP32 turned those readings into a stable, continuously updated volume value shown on the LCD. The system reliably caught the critical Low-level condition and triggered the buzzer and LED without any noticeable delay. The SIM900 GSM module consistently delivered SMS alerts to the registered caretaker number, confirming the system can reach staff even when they're away from the bedside and regardless of hospital Wi-Fi availability.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 7, July 2026
DOI: 10.17148/IJARCCE.2026.15716

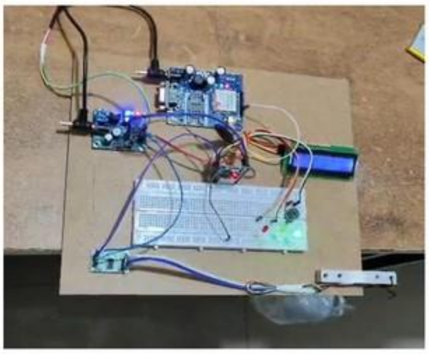

Fig.3. Final Output

VI. FUTURE SCOPE

Future versions of the system could add cloud storage for keeping saline-usage records, a dedicated mobile app for remote monitoring and historical tracking, and extra sensors for drip rate, IV fluid temperature, and flow rate. The GSM module could be supplemented or swapped for Wi-Fi/cloud-based notifications (Blynk or Telegram, for example) to get faster, cheaper alerts, and a single central unit could be scaled to monitor multiple saline bottles at once for larger hospital wards. Battery backup for power-failure scenarios and better measurement accuracy are also on the roadmap.

A longer-term direction worth considering is integrating with a hospital's existing electronic health record system, so saline level events get logged automatically against the patient's chart instead of existing only as a transient SMS. Pairing the load-cell reading with a drip-rate sensor would also let the system estimate time remaining until depletion rather than just the current volume, giving staff advance notice proportional to the actual infusion rate instead of a fixed volume threshold. Finally, a multi-bed gateway architecture — where several bedside nodes report to a single ward-level ESP32 or Raspberry Pi hub — could bring down the per-bed GSM cost by consolidating outbound SMS traffic while keeping the same local buzzer/LED redundancy at every bed.

CONCLUSION

This paper presented a Smart Saline Monitoring and Alert System that brings together a strain-gauge load cell, an HX711 amplifier, an ESP32 microcontroller, an I2C LCD, LED/buzzer indicators, and a SIM900 GSM module to automate the monitoring of intravenous saline bottles. The system continuously measures the remaining saline volume, displays it locally, and issues both an immediate on-site alert and a remote SMS notification once the level turns critically low.

Testing confirmed accurate weight measurement, reliable local and remote alerting, and stable overall operation. By replacing manual visual inspection with continuous automated sensing, the system cuts down human error, improves response time, and eases the workload on nursing staff — a low-cost, practical option for hospitals, clinics, and home healthcare settings alike.

REFERENCES

[1] A. A. Dharmale, R. R. Mehare, A. R. Bharti, S. R. Meshram, and S. V. Deshmukh, "IoT Based Saline Level Monitoring & Automatic Alert System," International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE), vol. 8, no. 4, pp. 71-75, Apr. 2019.
[2] Khushboo V., Nargees Bano H., and Neha S., "IoT Based Saline Level Monitoring System," International Journal of Innovative and Advanced (IJIACS), 2017. Computing Sciences
[3] D. Ghosh, A. Aggarwal, N. Prasad, and P. Gupta, "Smart Saline Level Monitoring System Using ESP32 andMQTT-S," in Proc. 2018 IEEE 20th Int. Conf. on eHealth Networking, Applications and Services (Healthcom), 2018.
[4] B. Malleswari, V. P. Vijay, and N. Venkatanam, "Smart Saline Level Monitoring System Using IoT," International Journal of Engineering & Technology (IJET), vol. 7, no. 2.7, pp. 817-819, 2018.
[5] R. D. Deshmukh, K. Jadhav, T. Bansode, and H. Landge, "A Survey Paper on Patient Health and Saline LevelMonitoring System Using IoT," International Journal of Engineering Research & Technology (IJERT), vol. 8, no.11, 2019.

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁘ Peer-reviewed & Refereed journal ⁘ Vol. 15, Issue 7, July 2026

DOI: 10.17148/IJARCCE.2026.15716

[6] P. P. Sheeba, N. Anushree, and L. Aishwarya, "Saline Infusion Level Detection and Heart Rate MonitoringSystem," International Journal for Research in Applied Science & Engineering Technology (IJRASET), 2016.

[7] C. C. Gavimath, K. Bhat, C. L. Chayalakshmi, R. S. Hooli, and B. E. Ravishankera, "Design and Development ofVersatile Saline Flow Rate Measuring Device and GSM Based Remote Monitoring Device," International Journal of Pharmaceutical Applications (IJPA), 2012.

[8] M. K. Swain, S. K. Mallick, and R. R. Sabat, "Smart Saline Level Indicator cum Controller," InternationalJournal of Application or Innovation in Engineering & Management (IJAIEM), vol. 4, no. 3, pp. 1 3, Mar. 2015.

[9] G. Sunil et al., "IoT Based Saline Level Monitoring System," IOP Conference Series: Materials Science andEngineering, vol. 981, p. 032095, 2020.

.[10] E. Spanò, S. Di Pascoli, and G. Iannaccone, "Low-Power Wearable ECG Monitoring System for MultiplePatient Remote Monitoring," IEEE Sensors Journal, vol. 16, no. 13, pp. 5452-5462, 2016.

[11] P. Gope and T. Hwang, "BSN-Care: A Secure IoT-Based Modern Healthcare System Using Body SensorNetwork," IEEE Sensors Journal, vol. 16, no. 5, pp. 1368 1376, Mar. 2016.

[12] F. Samie, L. Bauer, and J. Henkel, "IoT Technologies for Embedded Computing: A Survey," in Proc. Int. Conf.on Hardware/Software Codesign and System Synthesis (ESWEEK), Pittsburgh, PA, USA, Oct. 2016.

[13] S. Dharmoji, A. Anigolkar, and S. M., "IoT Based Patient Health Monitoring Using ESP8266," InternationalResearch Journal of Engineering and Technology (IRJET), vol. 7, no. 3, pp. 3619-3624, 2020.

[14] P. K. Rath, N. Mahapatro, S. Sahoo, and S. Chinara, "Design and Performance Analysis of an IoT Based HealthMonitoring System for Hospital Management," in Proc. 2021 Int. Conf. on Computing, Communication, and Intelligent Systems (ICCCIS), 2021.

[15] A. Jagannathachari and A. Nair, "Saline Level Indicator," IOSR Journal of Computer Engineering, pp. 13-16.