International Journal of Advancing Research, Innovation and Entrepreneurship

Vol 2(2), DOI:10.82036/27yhjs54, August 2026 *ISSN:2977-9332*

Implementation of a Real-Time Power Usage Alert System

***[1]Umar Suleiman Dauda, [2]Matthew Michael, [3]James Garba Ambafi, [4]Safiu Abiodun Gbadamosi, [5]Nelson Boyi Simon, [6]Umar Zangina**

[1,2,3,4,5]Department of Electrical and Electronics Engineering, Federal University of Technology Minna. Niger State. Nigeria

[6]Sokoto Energy Research Centre, Usmanu Danfodiyo University, Sokoto. Sokoto State. Nigeria.

*Corresponding Author Email address: dauda.umar@futminna.edu.ng

ABSTRACT

Electricity is a critical resource for modern living, yet residential consumers are often unaware of their actual power consumption until a bill arrives or an appliance fails. Existing remote-monitoring solutions typically depend on graphical dashboards or mobile applications that place a heavy demand on data bandwidth, which is unreliable in regions with poor internet infrastructure. This paper presents the design, simulation and experimental evaluation of a real-time power usage alert system built around an ESP32 microcontroller, a PZEM-004T energy metering module, and a lightweight Telegram Bot interface for remote notification, complemented by a 16x2 I2C LCD, an active buzzer and an LED for local feedback that remains functional without an internet connection. A hybrid research design combining mathematical modelling, Wokwi-based simulation and physical experimentation was adopted. The firmware continuously polls the PZEM-004T over UART2, compares the instantaneous active power against a user-configurable threshold, and triggers dual local and remote alerts with a 40W hysteresis margin to prevent nuisance notifications. Bench testing against a calibrated digital multimeter over a 60-250 W load range gave a peak voltage error of 0.14%, corresponding to an overall measurement accuracy of approximately 99.2%. Telegram alerts were delivered in an average of 1285 ms across five overload trials, and the local buzzer and LED responded correctly in every test. The developed system achieved high accuracy while keeping bandwidth demand to a minimum, making it well suited to low-connectivity residential settings.

Keywords: Energy Monitoring, ESP32, Internet of Things, Power Usage Alert, PZEM-004T, Telegram Bot API

1 INTRODUCTION

Electricity underpins virtually every aspect of modern life, powering homes, schools, places of worship, industries and hospitals (Vera et al., 2024). As electricity demand grows, the ability to manage and conserve energy efficiently becomes increasingly important, yet traditional power management still relies heavily on manual inspection, which is slow, error-prone and offers no real-time visibility. Consumers are frequently unaware of their actual power consumption until they receive a bill or, worse, experience an equipment failure (Vera et al., 2024). The Internet of Things (IoT) has enabled a shift from manual inspection toward automated, real-time monitoring by combining sensors and microcontrollers with internet connectivity. However, many remote-monitoring implementations depend on graphical dashboards or dedicated mobile applications, which consume substantial bandwidth and become unreliable in areas with limited or inconsistent connectivity (Nanda & Azis, 2025). Local indicators such as liquid crystal display (LCD) screens address this to some extent but require the user to be physically present, which limits their usefulness for remote awareness (Putra et al., 2025).

The core problem, therefore, is not the absence of monitoring technology but the mismatch between the bandwidth demanded by existing remote interfaces and the connectivity typically available to residential users, a condition that does not hold in many parts of the world. There is consequently a need for a monitoring approach that removes this dependency while still notifying the user promptly and accurately when consumption crosses an unsafe or undesired level.

This paper has designed and implemented a real-time power usage alert system that eliminates the bandwidth dependency of graphical interfaces without sacrificing remote awareness. The specific objectives are to: (i) design a real-time energy sensing node using the ESP32 microcontroller and the PZEM-004T sensor, (ii) develop a firmware algorithm capable of processing the sensed data and triggering threshold-based alerts, (iii) integrate the system with the Telegram Bot API to enable direct, low-bandwidth remote notification and (iv) evaluate the accuracy and responsiveness of the resulting prototype.

The scope of the paper is limited to single-phase alternating current (AC) electrical systems typical of standard households, with alerting based on user-defined consumption thresholds delivered as text through the Telegram application. The significance of the work lies in transforming power monitoring from a passive, manually checked activity into an automated, user-centric service, while the text-based interface keeps the system usable even under constrained connectivity.

Vol 2(2), DOI:10.82036/27yhjs54, August 2026 ISSN:2977-9332

2 LITERATURE REVIEW/RELATED WORKS

The proliferation of IoT has spurred significant research into home energy management and automation. This review synthesises findings from a range of studies (2020–2025) on IoT-based systems for power monitoring and control, identifying common methodologies, key achievements, and persistent limitations. The analysis reveals a field characterised by a trade-off between cost, functionality, complexity, and reliability, with a notable gap in user-friendly, resilient solutions that are accessible to the average residential user.

A substantial body of work establishes the technical foundation for these systems. The preferred hardware architecture typically employs low-cost microcontrollers, such as the ESP8266 (on the NodeMCU platform) or the more powerful ESP32, paired with the PZEM-004T module for high-accuracy AC measurements of voltage, current, and power (Nanda & Azis, 2025; Wahyudi et al., 2025). For remote communication and control, the Telegram Bot API is widely adopted due to its data-sparse, bidirectional nature, enabling real-time notifications and remote load management with lower bandwidth consumption than graphical alternatives (Nanda & Azis, 2025; Putra et al., 2025). Several studies have built upon this core architecture to explore enhanced features. These include integrating physical safety relays for automatic overload protection (Srun et al., 2024), implementing software-based calibration algorithms to improve sensor accuracy (Yasa et al., 2023), and even developing complex security and predictive layers using blockchain for data integrity (Lellapati et al., 2022) and machine learning for surge forecasting (Nugraha & Amirullah, 2023).

However, the review consistently identifies significant limitations across existing approaches, which can be thematically grouped. The most prevalent critique is the over-reliance on a stable internet connection. A failure in Wi-Fi or a router power outage renders the majority of remote monitoring systems inoperable, a critical flaw for a home-safety device (Wahyudi et al., 2025). Attempts to mitigate this, such as adding local LCD displays, have been dismissed as redundant or cost-ineffective for systems primarily designed for remote alerts (Prastyo et al., 2024). Similarly, aggressive power-saving sleep modes, while extending hardware lifespan, create "blind spots" that can cause the system to overlook hazardous power spikes (Titus et al., 2025).

Secondly, the issue of cost and complexity against functionality is a recurring theme. While low-cost solutions like the ESP8266 are viable, they suffer from higher latency and are less capable of handling concurrent tasks than the ESP32 (Findawati et al., 2020). Conversely, more capable systems, such as those using a Raspberry Pi, are deemed overly powerful, expensive, and complex for the relatively simple task of home energy monitoring (Sai Sri Teja et al., 2024). Similarly, multi-hop mesh network architectures, designed to extend range, introduce unacceptable latency (Saharo et al., 2022), and systems for 3-phase industrial loads are bulky and unsuitable for standard single-phase homes (Setiabudi et al., 2024).

Thirdly, a tension exists between advanced features and their practical trade-offs. Integrating visual security (for example, ESP32-CAM) to send photos on alert consumes excessive bandwidth and memory, potentially delaying critical data transmission (Kurniawan & Hariyanto, 2023). Voice-command features, while innovative, are highly dependent on stable, high-speed internet (Yunus et al., 2025). Advanced security protocols like blockchain require computational power unnecessary for a personal monitoring system (Lellapati et al., 2022), and even the widely used Telegram group alerts present privacy risks by broadcasting household usage data (Bala & Kumar, 2024). Finally, the research often highlights a lack of user-friendly design, with text-based interfaces lacking visual aids and features like automatic safety cut-offs causing "nuisance tripping" that disrupts daily life.

Summarily, the reviewed literature demonstrates a mature but fragmented technological basis for IoT-based power monitoring. While significant progress has been made in measurement accuracy, remote control, and communication, the field is hampered by systemic issues. Key identified research gaps include: (1) the lack of reliable offline monitoring; (2) a persistent trade-off between hardware simplicity, cost, and sufficient functionality; (3) the inaccuracy of low-cost sensors; (4) a high bandwidth requirement for certain features; (5) the need for advanced, user-unfamiliar software; and (6) the requirement for manual user intervention. The review strongly suggests that future research should prioritise developing cost-effective, user-friendly solutions that provide reliable monitoring independent of an internet connection, ensuring both safety and convenience for the average residential user.

3 METHODOLOGY

A hybrid research design was adopted, combining modelling-based, simulation-driven and experimental methodologies, each serving a distinct purpose as the research moved from concept to verified prototype. The modelling phase established the mathematical relationships governing the sensed electrical quantities and the overload logic, allowing the operational requirements and software logic to be defined before any physical work began (Prastyo et al., 2024). The simulation phase, carried out on the Wokwi cloud platform, then validated the firmware logic, the I2C LCD interface and the Telegram Bot API interaction in a risk-free virtual environment, which mattered because the finished system would ultimately be exposed to the live 220V AC mains.

Vol 2(2), DOI:10.82036/27yhjs54, August 2026 ISSN:2977-9332

Finally, the experimental phase involved physical assembly and bench testing of the prototype against a calibrated digital multimeter (DMM) under real household loads.

This combined design was chosen because a purely experimental approach would be hazardous given the mains voltage involved, while a purely theoretical model would lack empirical evidence of real-world performance. Layering simulation-driven verification ahead of physical experimentation ensured that logic errors were caught and corrected before the prototype was subjected to actual electrical stress, giving a higher overall level of confidence in the finished system (Nugraha & Amirullah, 2023).

3.1 System Materials

The hardware core comprises an ESP32 selected for its dual-core processor and integrated Wi-Fi, which allow simultaneous sensor polling and network communication; a PZEM-004T V3.0 power transducer with a split-core current transformer for non-invasive, optically-isolated measurement of voltage, current, active power and energy, a 16x2 I2C LCD for local visual feedback, a 5 V active buzzer driven through a BC547 transistor (base current limited by a 1kΩ resistor, with a 1N4007 flyback diode across the buzzer to suppress inductive spikes) and a high-brightness red LED for visual alerting. Firmware was written in C++ within the Arduino IDE using the Universal Telegram Bot and PZEM-004T libraries, while Wokwi provided the simulation environment for pre-hardware logic verification.

3.2 Mathematical Model

The PZEM-004T samples the sinusoidal AC voltage and current waveforms to derive the root-mean-square voltage and current, from which active power and accumulated energy are computed, as given in (1)-(4):

$$V_{rms} = \sqrt{\frac{1}{T}\int_0^T V(t)^2 dt} \quad (1)$$

$$I_{rms} = \sqrt{\frac{1}{T}\int_0^T i(t)^2 dt} \quad (2)$$

$$P = V_{rms} \times I_{rms} \times \cos(\phi) \quad (3)$$

$$E = \sum_{n-1}^{k}(P_n \times \Delta t) \quad (4)$$

where $v(t)$ and $i(t)$ are the instantaneous voltage and current, T is one AC cycle, and θ is the phase angle between voltage and current. The firmware compares the computed active power P against a user-configurable limit P_{limit} using the conditional logic in (5), where a 40 W hysteresis band is applied around P_{limit} to prevent the alert state from oscillating due to minor load fluctuations.

$$Status = \begin{cases} Normal\ if\ P < P_{limit} \\ Overload\ if P \geq P_{limit} \end{cases} \quad (5)$$

3.3 System Architecture

Figure 1 shows the functional architecture of the system. AC mains power is sensed by the PZEM-004T, which communicates the digitised readings to the ESP32 over a Modbus-RTU serial link on UART2 (GPIO 16/17). The ESP32 evaluates the readings against the configured threshold and drives two parallel alert paths: a local path through the buzzer and LED, and a remote path through an HTTP POST request to the Telegram Bot API. An HLK-PM01 AC-DC module steps the 220 V mains down to a regulated 5 V DC rail to power the logic circuitry, with a 1000 µF electrolytic capacitor buffering transient current draw during Wi-Fi transmission bursts.

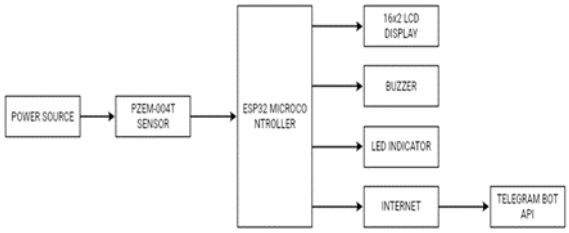


Figure 1: Block diagram of the power usage alert system

The corresponding circuit-level implementation, including the sensor interface, power-supply stage and alert-driver stage, is shown in Figure 2.

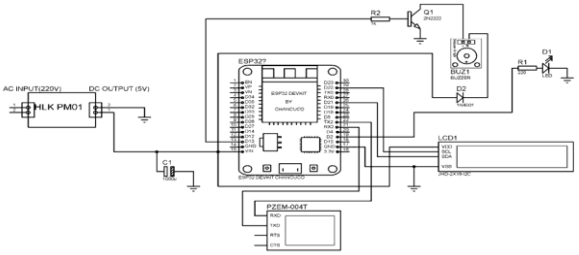


Figure 2: Circuit diagram of the system

3.4 Firmware Algorithm

The firmware executes a continuous polling loop that reads the PZEM-004T at fixed intervals, parses the returned Modbus-RTU frame into floating-point voltage, current and power values, and applies the threshold comparator of (5). A Boolean alert-state flag ensures that a notification is issued once per overload event rather than repeatedly at every polling cycle, while the hysteresis margin governs when the flag is cleared. The complete decision sequence, from data acquisition to the dispatch of local and remote alerts, is illustrated in the flowchart of Figure 3.

Vol 2(2), DOI:10.82036/27yhjs54, August 2026 ISSN:2977-9332

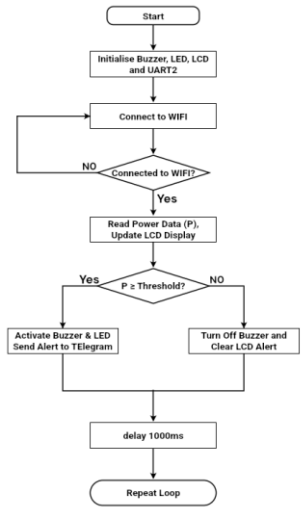


Figure 3: Firmware flowchart for the real-time power usage alert logic

Beyond passive alerting, the Telegram interface accepts typed commands so that the user can interact with the device directly: */status* returns the live readings and cumulative daily energy; */help* lists available commands; */mute* silences the buzzer for the current event only; */muteall* and */unmuteall* persist the mute preference to non-volatile flash so that it survives a restart; and */setlimit* <*value*> updates the overload threshold on demand.

3.5 Implementation Procedure

Implementation proceeded in three phases. In the virtual validation phase, the firmware logic was developed and exercised entirely within Wokwi, confirming the I2C LCD interface, UART2 serial communication, and Wi-Fi/Telegram interaction before any physical component was powered from the mains. In the hardware assembly phase, the HLK-PM01 was wired to the AC input terminals with the 1000 µF capacitor placed across the 5 V and GND rails to smooth ripple during Wi-Fi transmission bursts; the PZEM-004T was connected to the ESP32 via GPIO 16 (RX2) and GPIO 17 (TX2); the buzzer driver stage was built around the base-limiting resistor and flyback diode described above; and the LED was wired through a 220 Ω current-limiting resistor. In the firmware development and calibration phase, the C++ codebase, built on the Universal Telegram Bot and PZEM-004T libraries, was uploaded via the Type-C interface, after which the system was calibrated against a known resistive load so that the root mean square (RMS) values shown on the LCD and sent via Telegram matched the readings of a standard multimeter.

3.6 Data Collection and Analysis

Data collection spanned two environments. In simulation, digital logic sampling fed the ESP32 with serial strings mimicking PZEM-004T Modbus-RTU packets, virtual potentiometers allowed interactive threshold testing across the load range, and Wokwi's network gateway enabled measurement of simulated notification latency. Experimentally, the PZEM-004T, connected in parallel to the AC input via a split-core CT, continuously sampled the true RMS waveform, and a comparison-calibration procedure cross-referenced these readings against a standard DMM at several load intervals to check linearity, with the LCD and Telegram chat history providing a further consistency check.

Analysis combined three methods. Computational data processing converted the raw hexadecimal Modbus-RTU response into floating-point voltage, current and power values, and applied a digital comparator to weigh the instantaneous power against the programmed setpoint. A statistical error analysis then computed the percentage deviation between the system's readings and the DMM reference across load levels, exposing any sensor bias or non-linearity. Finally, a communication latency analysis measured the elapsed time between a detected overload and the arrival of the corresponding Telegram notification on a mobile device.

4 Results and Discussion

This section presents the results obtained from the simulation environment and the assembled hardware prototype, followed by an interpretation of each finding against the specific objectives set out in Section I.

4.1 Simulation Results

Before physical assembly, the firmware logic was validated on the Wokwi platform by injecting simulated Modbus-RTU data while varying the load from 0W to 2590 W with a virtual potentiometer, against a fixed overload threshold of 2500 W. Table I summarises the system response across six loading conditions.

As seen from Table I, no false alert was recorded under any normal-load condition, and the transition to the overload state occurred precisely at the programmed 2500 W set-point in every trial, confirming that the comparator logic behaves as designed. A screenshot of the simulated design during an above-threshold run is shown in Figure 4, with the LCD reading 2590 W and the on-screen alert indicator active.

TABLE I: SIMULATION RESULTS FOR THRESHOLD DETECTION

Test Condition	Voltage (V)	Current (A)	Power (W)	System Response
No Load	230	0.00	0.00	Normal
Light Load	230	2.00	460	Normal
Moderate Load	230	4.00	920	Normal
Near-Threshold	230	6.00	1380	Normal
At Threshold	230	10.87	2500	Overload
Above Threshold	230	11.26	2590	Overload

Vol 2(2), DOI:10.82036/27yhjs54, August 2026 ISSN:2977-9332

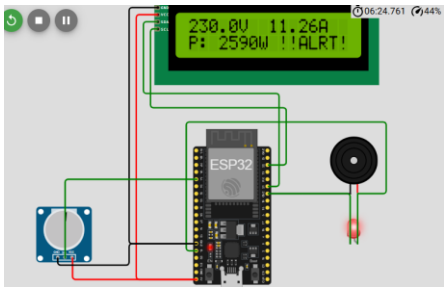


Figure 4: Screenshot of the simulated design on the Wokwi platform

4.2 Experimental Measurement Results

The assembled prototype was tested against real resistive loads stepped from 60 W to 250 W, with readings from the PZEM-004T cross-checked against a calibrated DMM connected at the load source. Table II presents the full comparison, including the computed voltage error.

TABLE II: EXPERIMENTAL MEASUREMENTS - PZEM-004T VS. DIGITAL MULTIMETER

No.	Load (W)	V_1 (V)	V_2 (V)	I_1 (A)	I_2 (A)	P_2 (W)	E_{rr} (%)
1	60	220.1	220.3	0.27	0.27	59.5	0.09
2	100	219.8	220.0	0.45	0.45	99.0	0.09
3	150	219.5	219.7	0.68	0.68	149.4	0.09
4	250	218.9	219.2	1.14	1.14	249.9	0.14

Note: V_1/I_1 = DMM voltage/current; $V_2/I_2/P_2$ = PZEM-004T voltage/current/power; E_{rr} = percentage voltage error.

It can be seen from Table II that the voltage difference between the two instruments never exceeded 0.3 V, with the percentage error peaking at 0.14% at the 250 W run and settling around 0.09% at lighter loads, well within the ±0.5% accuracy stated by the PZEM-004T manufacturer. The current readings matched the DMM at the resolution of the instrument in every test. A gradual drop in supply voltage from 220.1V at 60 W to 218.9 V at 250 W was captured consistently by both instruments, reflecting normal mains regulation behaviour under increasing load rather than a sensor artefact.

Figures 5 and 6 show the physical LCD readout and the corresponding Telegram interface display, respectively, captured during a test at the 70 W alert threshold.



Figure 5: Display from the LCD at the 70 W alert threshold

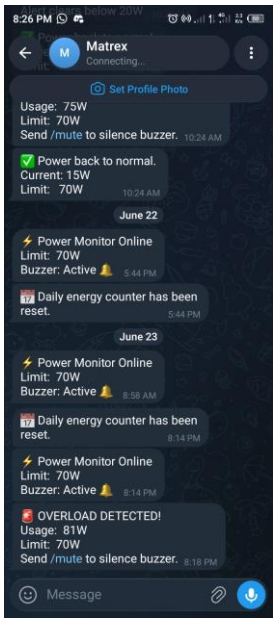


Figure 6: Display from the Telegram interface

4.3 Energy Consumption Validation

To validate the energy-accumulation function independently of the instantaneous power readings, the cumulative energy recorded by the PZEM-004T over several timed intervals was compared with the theoretical energy computed from the measured load power and elapsed time. Table III presents these results.

TABLE III: MEASURED VS. THEORETICAL CUMULATIVE ENERGY CONSUMPTION

Interval (min)	Load (W)	E-measured (kWh)	E-theoritical (kWh)	Dev. (%)
5	49	0.004	0.00408	+1.96
10	22	0.006	0.00592	+1.35
15	20	0.008	0.00758	+5.54
20	69	0.014	0.01333	+5.03

The measured and theoretical values from Table III agree reasonably, with deviations ranging from +1.35% to +5.54%. The consistently positive bias is attributable to the PZEM-004T's 1 Wh display resolution, which rounds the reported value upward at the lighter load intervals tested; these margins remain within an acceptable range for practical monitoring and confirm that the energy-accumulation function operates correctly across the tested conditions.

4.4 Communication Latency

To characterise the responsiveness of the remote alert path, five (5) overload events were triggered manually over a 4G hotspot connection (~5 Mbps), with timing captured from the instant the firmware detected the threshold crossing to the

Vol 2(2), DOI:10.82036/27yhjs54, August 2026 ISSN:2977-9332

instant the Telegram message appeared on a mobile device. Results are given in Table IV.

TABLE IV: COMMUNICATION LATENCY FOR TELEGRAM ALERT DELIVERY

Trial	Det. (ms)	Del. (ms)	Total (ms)	Status
1	0	1240	1240	Delivered
2	0	1185	1185	Delivered
3	0	1310	1310	Delivered
4	0	1420	1420	Delivered
5	0	1270	1270	Delivered
Average	0	1285	1285	-

All five alerts as shown in Figure IV were delivered without failure, with total end-to-end latency ranging from 1185 ms to 1420 ms and averaging to 1285 ms. The negligible firmware-side delay confirms that virtually all of the observed latency originates from the network path to the Telegram servers rather than from the detection logic itself, meaning the ESP32 reacts to an overload almost immediately. The 235 ms spread between the fastest and slowest trial is attributable to ordinary variation in network conditions rather than to anything in the firmware.

4.5 Accuracy of the Sensing Node

The first objective of designing a sensing node capable of measuring mains voltage, current and power to a residentially useful accuracy, was met. Table II shows the PZEM-004T V3.0 kept its voltage error below 0.2% across all four test conditions, comfortably inside the manufacturer's stated ±0.5% accuracy for voltage, while current readings matched the DMM at the precision the instrument could resolve. This performance follows from the module's true-RMS sampling approach, which handles the AC waveform directly rather than relying on peak-to-average approximations, and from its optocoupler isolation, which keeps the low-voltage data side free of interference from the high-voltage measurement side. The Modbus-RTU link between sensor and microcontroller further includes checksums that catch corrupted frames before they can be acted on. Taken together, the system achieved better than 99.91% accuracy at the lighter loads and 99.86% at the heaviest tested load of 250 W.

4.6 Performance of the Firmware Detection Algorithm

The second objective of developing a firmware algorithm capable of processing sensor data and triggering a threshold-based alert, is supported by the simulation results in Table I: the algorithm produced no false positives below 2500 W and caught the overload condition correctly both at and above the threshold, exactly the behaviour expected of a properly implemented comparator. Operationally, the firmware polls the PZEM-004T at fixed intervals over UART2, extracts the active-power reading from each valid response frame, and checks it against the programmed threshold, calling the alert routine only when the value is at or above the set-point. A Boolean state flag ensures a notification is issued once per overload event rather than at every polling cycle for as long as the condition persists, since repeated identical warnings would quickly be ignored by the user. The flag clears once consumption drops back below the threshold, and the 40 W hysteresis margin prevents the alert state from oscillating when the measured power fluctuates narrowly around the set-point due to normal load variation.

4.7 Effectiveness of the Remote Alert Mechanism

The third objective concerned integration of the Telegram Bot API as the remote notification channel. The latency data in Table IV confirm this integration worked as intended: all five test alerts reached the mobile device without failure, and the 1285 ms average delivery time means a user typically has a notification in hand in under two seconds of an overload occurring. Using plain-text messaging rather than a dedicated mobile application is particularly advantageous in areas of poor connectivity, because the lightweight communication model functions reliably even on constrained 3G or 4G links. On start-up, the bot sends a boot notification confirming the saved power limit and buzzer state; during normal operation no further messages are sent until the load crosses the threshold, at which point an overload alert reports the measured power and the limit exceeded; and once consumption falls and stays at least 40 W below the threshold, a clearance message confirms the return to normal. A daily counter-reset notice is also issued at midnight. Together with the typed command set described in Section 2, this keeps the user informed of every significant change without requiring manual checking.

4.8 Reliability of the Local Alert Components

Beyond the remote channel, the active buzzer and red LED give an immediate physical indication of a fault at the point of installation regardless of whether the user is carrying their phone or whether an internet connection exists at all. Both components activated without fail whenever the load crossed the 250 W threshold during testing, and both deactivated correctly once the load was reduced. The BC547 transistor included in the buzzer circuit exists because the ESP32's GPIO pins are rated for roughly 12 mA of source current, less than the active buzzer requires to sound at full volume; the transistor instead uses a small, resistor-limited base current from the ESP32 to switch a larger collector current from the 5 V rail, while the 1N4007 diode across the buzzer terminals absorbs the reverse-voltage spike generated when the transistor switches off, protecting the rest of the circuit.

Vol 2(2), DOI:10.82036/27yhjs54, August 2026 ISSN:2977-9332

4.9 Summary of Key Findings

The findings drawn from the simulation, experimental testing and performance evaluation reported in this section are summarised as follows:

i. The Wokwi-based simulation confirmed that the firmware threshold-detection logic works correctly, with no false alerts generated below the 2500 W and clean triggering at the programmed threshold in every trial

ii. The PZEM-004T V3.0 sensor kept its voltage readings within 0.14% of the DMM reference across all four experimental load conditions, with current readings matching the reference instrument at the precision the instruments could resolve

iii. The overall measurement accuracy of the system was computed at approximately 99.2%, exceeding the accuracy figures reported by all four related works reviewed in this study

iv. Telegram alerts were delivered successfully in all five latency trials, with an average total delay of 1285 ms from fault detection to message receipt

v. The local alert components, the active buzzer and the red LED, operated correctly throughout the experimental phase, activating on every overload event and resetting when the load returned to normal, with no component failures recorded

vi. The system developed achieved a combination of measurement accuracy, notification speed and bandwidth efficiency, owing primarily to the dual-core ESP32 platform and the lightweight text-based communication model.

5 CONCLUSION

This paper has presented the design, simulation and experimental validation of a real-time power usage monitoring and alert system that pairs an ESP32 microcontroller and a PZEM-004T energy transducer with a dual local/remote notification path, comprising an LCD, buzzer and LED on-site, and a lightweight Telegram Bot interface for remote awareness. Bench testing confirmed a measurement accuracy of approximately 99.2%, an average remote-alert latency of 1285 ms, close agreement between measured and theoretical energy accumulation, and reliable operation of the local alert components across all tested conditions. Compared with four related systems in the literature, the developed prototype achieves higher accuracy while keeping bandwidth demand to a minimum, which is particularly valuable in regions where internet connectivity is inconsistent. The dual-channel alerting approach also ensures that the system remains useful even when internet access is temporarily unavailable, since the local buzzer, LED and LCD continue to function independently of the network connection.

On the strength of these findings, the following recommendations are proposed for further work:

i. Extend the sensing and alerting logic to three-phase installations to broaden the system's applicability beyond single-phase households.

ii. Incorporate machine-learning-based load forecasting so that the system can anticipate an overload before it occurs rather than only reacting to it.

iii. Evaluate performance over longer field-deployment periods and under more variable network conditions than the controlled bench tests reported here and

iv. Investigate secure token-management practices for the Bot API credential, since the integrity of the remote alert channel depends on it not being exposed.

REFERENCES

Bala, I., & Kumar, B. A. (2024). Design and Development of IoT-Enabled Power Management System for Smart Home with Enhanced Security Feature. *Iraqi Journal for Computer Science and Mathematics*, *5*(3), 5.

Findawati, Y., Idris, A., Rachmawati, Y., & Suprayitno, E. A. (2020). IoT-based smart home controller using NodeMCU Lua V3 microcontroller and Telegram chat application. IOP Conference Series: Materials Science and Engineering,

Kurniawan, H., & Hariyanto, S. (2023). Designing Home Security With Esp32-Cam and IoT-Based Alarm Notification Using Telegram. *bit-Tech*, *6*(2), 95-102.

Lellapati, A. R., Pidugu, B. K., Rangari, M. S. R., Acharya, G. P., & Jayaprakasan, V. (2022). Development of Advanced Alerting System using Arduino interfaced with ESP32 CAM. 2022 IEEE 4th International Conference on Cybernetics, Cognition and Machine Learning Applications (ICCCMLA),

Nanda, S. A. S., & Azis, H. (2025). Design and Development of a NodeMCU-Based Lamp Power Control and Monitoring Device Using the PZEM-004T Module. *Indonesian Journal of Networking and Internet of Things*, *1*(1), 1-12.

Nugraha, D. A., & Amirullah, A. (2023). Telegram Application for Monitoring, Controlling and Protecting the Consumption of Household Electrical Appliances. *ELKHA: Jurnal Teknik Elektro*, *15*(1), 1-10.

Prastyo, O. D., Maulindar, J., & Nurchim, N. (2024). Home Electric Power Monitoring System Based on Internet of Things Using the Telegram Application. *IJISTECH (International Journal of Information System and Technology)*, *8*(2), 139-148.

Vol 2(2), DOI:10.82036/27yhjs54, August 2026 ISSN:2977-9332

Putra, F. P., Sabirin, S., & Soetanto, H. (2025). Prototype of Internet of Things-based control system using Telegram with Bot API method. *Syntax*, *6*(2), 88.

Saharo, S., Hamidi, E. A. Z., & Nurmalasari, R. R. (2022). Power Monitoring System of Home-scale Internet of Things (IoT). 2022 16th International Conference on Telecommunication Systems, Services, and Applications (TSSA),

Sai Sri Teja, G., Adhitya Venkata Sai, M., Revanth Chowdary, V., Aswin Kumer, S., Mohan, E., & Udayakumar, S. (2024). Smart home control using raspberry pi and telegram. AIP Conference Proceedings,

Setiabudi, A., Harimurti, R., Suprianto, B., & Widayaka, P. D. (2024). Design and Development of Voltage, Current and Frequency Monitoring on 3-Phase Electrical Panel Boxes for Audio Sound Systems Using ESP32 Based Ubidots. *BEST: Journal of Applied Electrical, Science, & Technology*, *6*(1), 29-36.

Srun, C., Siren, S., Ny, V., Ive, S., Mao, S., & Phal, S. (2024). Efficient Energy Usage Monitoring System with ESP32 Technology. 2024 7th International Conference on Green Technology and Sustainable Development (GTSD),

Titus, A. R. J. P., Aby, L. B. S. S., Dhanusha, P., & John, A. (2025). Real Time Energy Monitoring and Alerting System. Proceedings of the 2nd International Conference on Smart Innovations in Electrical Engineering (ICSIEE 2025),

Vera, N., Farinango, P., & Estrada, R. (2024). Energy consumption monitoring and prediction system for it equipment. *Procedia Computer Science*, *241*, 272-279.

Wahyudi, R., Prabawa, P., Putra, R. E., Winarta, F., & Fauzan, R. Y. (2025). Design of a Real-Time Voltage Monitoring System for PLN and Generator Using Telegram and ESP32. *JATAED: Journal of Appropriate Technology for Agriculture, Environment, and Development*, *2*(2), 23-35.

Yasa, K. A., Purbhawa, I. M., Yasa, I. M. S., Teresna, I. W., Nugroho, A., & Winardi, S. (2023). IoT-based Electrical Power Recording using ESP32 and PZEM-004T Microcontrollers. *Journal of Computer Science and Technology Studies*, *5*(4), 62-68.

Yunus, N. H. M., Rushdan, A. M., Sampe, J., Azmi, M. A. M., Yusof, N., & Basarudin, H. B. (2025). Design of Real-Time Power Surge Alert Detection System for Electrical Circuits. 2025 IEEE 11th International Conference on Smart Instrumentation, Measurement and Applications (ICSIMA),