ACCSCIENCE PUBLISHING

Asian Journal of Water, Environment and Pollution. Vol. 23, No. 1 (2026), pp. 331-339.
doi: 10.36922/AJWEP025050028

SHORT COMMUNICATION

Next-generation residential energy management: A web-based self-hosting solution

Jyothilal Nayak Bharothu[1], Suma Satyanarayan[2], Rayalapalli Ganesh Manikanta[1], Pothala Jagadeesh[1], Lakshmi Narasimha Sastry Varanasi[3], and Shravani Kanaka Kumari Palla[1]*

[1]Department of Electrical and Electronics Engineering, Rajiv Gandhi University of Knowledge and Technologies Nuzvid, Andhra Pradesh, India
[2]Department of Computer Science and Engineering, CMR Technical Campus, Hyderabad, Telangana, India
[3]Department of Electrical Engineering, National Institute of Technology Andhra Pradesh, Tadepalligudem, Andhra Pradesh, India

(This article belongs to the *Special Issue: Renewable Energy Systems and Strategies in Smart Grids and Smart Cities Development*)

***Corresponding author:** Shravani Kanaka Kumari Palla (shravanivenugopal@rguktn.ac.in)

Received: January 30, 2025; 1st revised: May 2, 2025; 2nd revised: May 17, 2025; 3rd revised: May 31, 2025; 4th revised: June 11, 2025; Accepted: June 12, 2025; Published online: November 17, 2025

Abstract: Rising residential electricity demand, driven by population growth and appliance proliferation, strains non-renewable energy resources and increases carbon emissions. In India, households consume ~25% of total electricity, yet most users receive only monthly aggregated bills without real-time insights or control over individual loads. Existing commercial energy monitors are costly (>USD 100) and lack remote access, whereas open-source alternatives (Raspberry Pi, Arduino) suffer from complexity, high power use, or missing cloud integration. This work presents a low-cost (<USD 20), web-based, self-hosted energy management system using NodeMCU and PZEM-004T V3.0, enabling real-time monitoring, remote control, and data-driven optimization for sustainable residential energy use. This study presents an efficient energy management system for residential buildings, addressing the increasing demand for electrical energy and the need for improved management practices. The system integrates a NodeMCU (ESP8266) microcontroller and a PZEM-004TV3.0 module to monitor and manage electrical energy loads. Real-time data on voltage, current, power, and energy were acquired and transmitted to Google Firebase via the MQTT protocol, enabling remote monitoring and control. Users can manage appliances locally through manual switches and light-emitting diode indicators or remotely through a web interface. The system achieved a calibration efficiency of 98.58% compared to standard instruments, with voltage and current errors of 0.15% and 3.12%, respectively. By providing detailed insights into energy usage and control capabilities, this cost-effective solution empowers users to optimize consumption, reduce wastage, and contribute to energy sustainability efforts.

Keywords: Residential energy management; NodeMCU; PZEM-004TV3.0; MQTT Protocol; Google Firebase Internet of Things-based energy control; Cost-effective energy management

1. Introduction

The escalating demand for electrical energy in residential buildings, coupled with inadequate management practices, is driving an impending energy crisis. This challenge primarily arises from a heavy reliance on non-renewable energy sources and inefficient energy utilization. Currently, utility companies provide only aggregated consumption data, such as the total energy used by a building, without offering real-time insights or control over individual electrical loads. This limitation restricts users' ability to access detailed information and manage energy effectively, thereby hindering efforts to optimize consumption and reduce wastage.

Although numerous studies have addressed this issue, many existing solutions are either costly or provide limited accessibility. Furthermore, systems employing controllers such as Raspberry Pi, Arduino Uno, or WeMos often suffer from inefficiencies or lack intuitive user interfaces, making them less suitable for widespread residential adoption. This highlights the need for a practical, cost-effective, and efficient energy management system tailored to the specific requirements of household use.

To address these challenges, this research proposes a cost-effective solution that enables both local and remote monitoring and control of electrical appliances in residential buildings. The system leverages a NodeMCU (ESP8266) microcontroller and a PZEM-004TV3.0 module to provide real-time energy usage data and bidirectional control between users and appliances. Local control is facilitated through manual switches and light-emitting diode (LED) indicators, whereas remote control is achieved through notifications and a web-based interface accessible from anywhere in the world.

This system aims to enhance energy management functionality while minimizing costs and maximizing effectiveness. By providing detailed insights into energy consumption, the system empowers users to manage their energy usage more effectively, improve efficiency in residential buildings, and contribute to broader efforts to reduce dependence on non-renewable energy sources. Ultimately, this research aims to advance residential energy management systems, offering a solution that is practical and accessible for everyday consumers.

1.1. Literature review

Recent studies have explored Internet of Things (IoT)-based energy management systems to address the growing demand for efficient energy utilization in residential settings. Tsai *et al*.[1] developed a wireless power monitoring system for air-conditioning appliances using IoT, achieving high accuracy; however, it required costly hardware, limiting its accessibility for widespread adoption. Similarly, Suryono *et al*.[2] proposed a system using WeMos D1 Mini for monitoring electricity consumption through smartphones, but its user interface was less intuitive for non-technical users. Kamal *et al*.[3] introduced an IoT-based smart electric meter, focusing on grid-level applications rather than household-specific control, which limited its applicability for residential users. Many existing solutions are either costly or provide limited accessibility.[4]

In addition, Lestari *et al*.[5] presented a prototype for energy consumption monitoring in a university building, but the system lacked remote control capabilities and relied on complex hardware setups. Jonathan and Putri[6] developed a home power monitoring tool using ESP32, which provided real-time data but did not integrate cloud-based storage for long-term analysis. Macheso and Thotho[7] and Andrei *et al*.[8] explored ESP32- and Arduino-based solutions, respectively, but faced challenges with scalability and user-friendliness. Chooruang and Meekul[9] proposed an IoT energy monitoring system; however, its high implementation cost made it less viable for residential use.

To address the need for a quantitative comparison with similar systems, Table 1 summarizes the performance of our system against Raspberry Pi- and Arduino Uno-based solutions. For instance, a Raspberry Pi-based system was reported with 97.5% calibration efficiency, 0.2% voltage error, and a cost of approximately USD 100, limiting its affordability, and described an Arduino Uno-based system with a 3.5% current error and a cost of USD 50, but it lacked cloud integration. Our system, which utilizes a NodeMCU microcontroller and PZEM-004TV3.0 module, achieves 98.58% calibration efficiency, 0.15% voltage error, 3.12% current error, and a cost of USD 20, offering superior accuracy and affordability for residential use.

Advanced approaches incorporating machine learning and smart grid technologies have also been studied. Stone and Thotho[10] utilized machine learning integrated with IoT for smart building energy management, achieving improved efficiency but requiring significant computational resources. Similarly, Mrudul *et al*.[11] applied machine learning to optimize energy efficiency in smart grids, focusing on large-scale systems rather than residential applications. Rao *et al*.[12] provided practical guidelines for household energy conservation but did not propose a technical solution for real-time monitoring and control.

Table 1. Comparison of Internet of Things-based energy management systems

System	Controller	Calibration efficiency (%)	Error (voltage; current)	Cost (USD)
Tsai *et al.*[1]	Raspberry Pi	97.50	0.2%; 2.8%	~100
Andrei *et al.*[8]	Arduino Uno	96.8	0.3%; 3.5%	~50
Proposed system	NodeMCU	98.58	0.15%; 3.12%	~20

In contrast, our system combines cost-effective hardware (NodeMCU and PZEM-004TV3.0) with a user-friendly web interface and cloud-based storage, enabling both real-time monitoring and control of multiple household appliances. Unlike previous solutions, it balances affordability, accessibility, and functionality, offering granular insights and remote control capabilities to enhance energy efficiency without compromising occupant comfort.[13]

2. System design

The proposed system aims to sense and control electrical parameters, including voltage, current, power, and energy, using simple hardware components. The system enables local monitoring through a liquid crystal display (LCD) in the residential environment and global monitoring through Google Firebase Cloud. It also provides flexibility in managing electrical loads and offers local notification through LED indicators. The complete outline of the system setup is illustrated in Figure 1.

2.1. Hardware configuration

Decisions regarding the components used in the hardware setup were influenced by goals such as low power consumption, user interface capability, and high measurement precision. The key hardware components integrated into the system include the NodeMCU microcontroller, PZEM-004TV3.0 module, LEDs, I2C organic LED display, 3.3V lithium battery, relay module, and a two-way switch.

2.1.1. PZEM-004T V3.0

The PZEM-004TV3.0 module enables the measurement of electrical parameters such as voltage, current, power, and energy with high accuracy. It transmits this data to Firebase through NodeMCU for continuous tracking and evaluation. Known for its high precision in energy monitoring, the PZEM-004TV3.0 serves as the primary data source for the system. This precise monitoring enables users to identify inefficient appliances with high energy consumption, allowing targeted control to minimize waste and enhance overall energy efficiency.

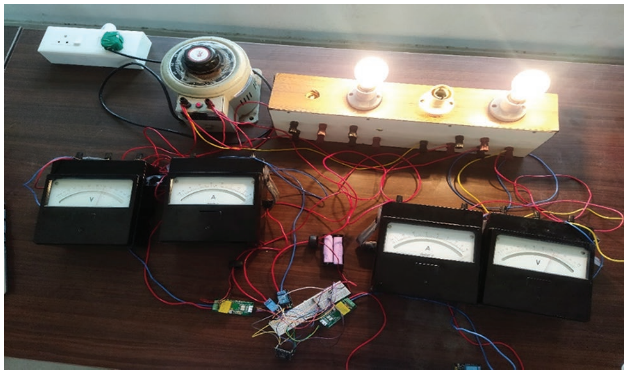


Figure 1. Circuit setup

2.1.2. Two-way switch

The two-way switch enables manual control of electrical loads in case of mechanical faults or Internet connectivity issues. This feature allows users to manage appliances directly, avoiding the relay module when necessary, thereby ensuring the system remains operational during such conditions.

2.1.3. Relay module

The relay module acts as a switch controlled by the NodeMCU.It consists of terminals labeled VCC, GND, IN1, IN2, IN3, and IN4. Whenever the NodeMCU sends a 5V control signal, the relay toggles the state of the connected electrical loads, enabling automated switching of appliances as required by the system.

2.1.4. NodeMCU

The NodeMCU microcontroller serves as the central interface and features an in-built Wi-Fi chip that establishes a connection with the Firebase cloud for online data management and remote monitoring. This microcontroller collects information from the PZEM-004TV3.0 module and transmits it to Firebase using the MQTT protocol. In addition, the NodeMCU receives control signals from Firebase to manage electrical loads through the relay module, enabling automated device control.

2.1.5. LED indicators

LED indicators were used to provide visual indications of the relay status (ON/OFF). They also function as alarm indicators whenever an abnormal voltage or current is detected, ensuring that the user is informed of the system's condition, even if the user cannot constantly monitor the cloud interface or the LCD.

2.1.6. I2C liquid crystal display

In the system, an I2C LCD was used, which is interfaced with the NodeMCU, to display live voltage, current, and power measurements. This makes it possible to monitor electrical parameters at the local level, especially in places where an Internet connection may not be accessible. This component is effective in providing an instant means of tracking energy usage by individual users.

2.2. Software configuration

Different software tools were incorporated into the system, which enhanced the user interface and improved the reliability of data management.

2.2.1. Firebase

Firebase serves as the cloud database and is responsible for storing data transmitted from the NodeMCU in real time. It allows remote monitoring and control through the Firebase Real-time Database, which synchronizes data across devices. This integration enables users to monitor energy data and manage electrical consumption through the cloud (Figure 2).

2.2.2. Google Apps script

Google Apps Script was used to interface with other Google services, including Firebase and IFTTT. It allows the system to receive data from NodeMCU and analyze the received information to perform actions based on it, which can control the hardware and other cloud services depending on the real-time data received.

2.2.3. Arduino integrated development environment

Arduino Integrated Development Environment was adopted to load the required libraries into the NodeMCU (ESP8266) microcontroller. It coordinates the operation and communication protocols between the PZEM-004TV3.0 sensor, the relay module, and the cloud-based data storage platform, Firebase.

2.2.4. Visual studio code

The web interface was developed using Visual Studio Code for system development, which ensures features for the provision of code editing and the establishment of extensions for HTML, CSS, and JavaScript. This makes the development of an optimal and user-friendly human interface possible to enable interaction with the system by applying graphical and numerical representations of power characteristics, as illustrated in Figure 3.

3. System implementation

The system was specifically designed for the monitoring of electrical parameters and the control of electrical loads, with the simultaneous provision of an intuitive user interface. This is achieved using Wi-Fi connectivity to display readings on a webpage, which, with the help of a server that downloads and uploads historical data, is updated in real time, with all readings recorded in an Excel sheet (Figure 4).

3.1. Data acquisition from electrical loads

The PZEM-004T V3.0 module acquires voltage and current through the input voltage terminals and the current transformer coil. Other parameters, such as power and energy, are calculated based on these

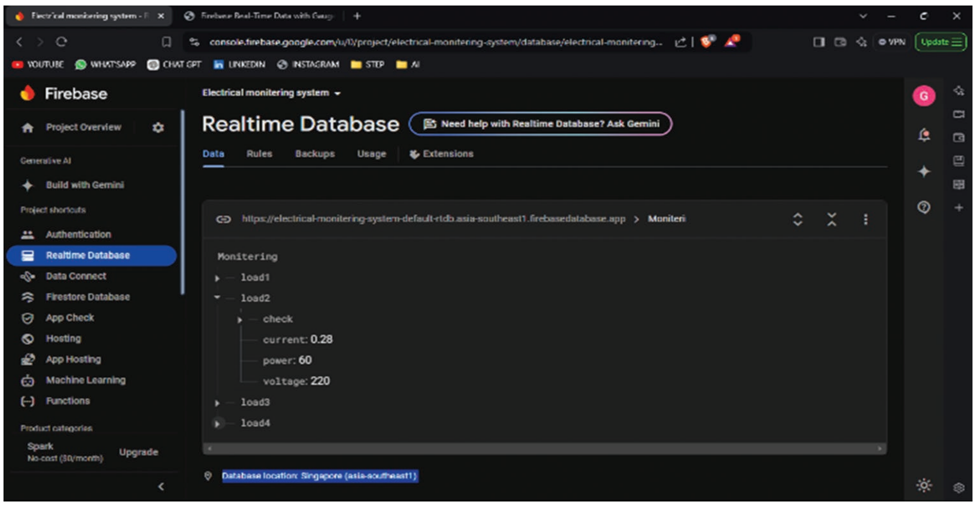


Figure 2. Data transmission from NodeMCU to Firebase

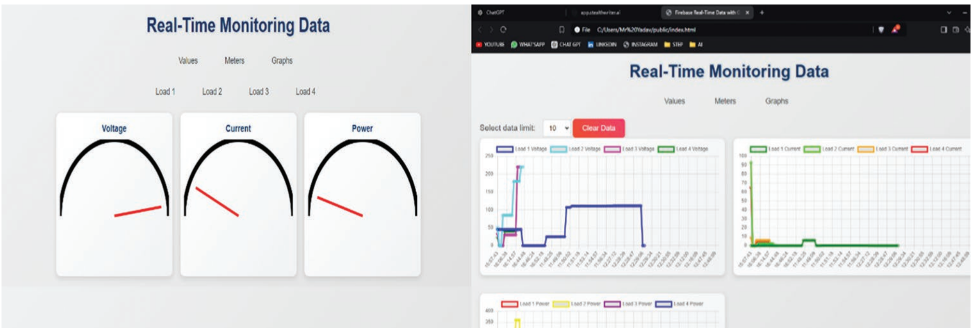


Figure 3. Design of the webpage gauge indication using Visual Studio Code and a graphical representation of data from electrical loads

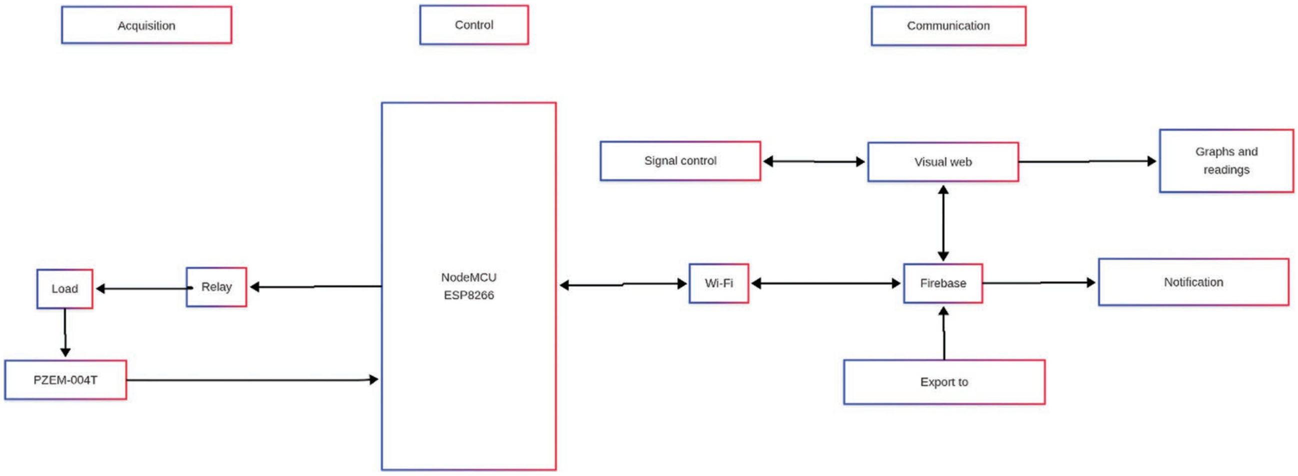


Figure 4. Block diagram of the system

electrical measurements. The acquired readings are then transmitted to the subsequent processing phase, the NodeMCU, with the aid of a serial communication interface.

3.2. Communication bridge through NodeMCU

The NodeMCU microcontroller serves as the communication gateway between the PZEM module and Firebase. Using the MQTT protocol, NodeMCU transmits voltage, current, power, and energy data obtained from the PZEM module to Firebase, utilizing an application programming interface key and the database uniform resource locator. Moreover, data from Firebase can also be read by NodeMCU to perform both data transmission and reception in real-time.

3.3. Load control mechanism

Control commands from Firebase are processed by the NodeMCU, which sends 5V signals to the relay module to switch appliances on or off. This functionality allows users to remotely manage inefficient appliances, such as turning off high-consumption devices left operating unnecessarily, thereby reducing energy waste and improving overall efficiency.

3.4. Web hosting and deployment

Firebase hosts the webpage, which consists of real-time data from the Firebase Real-time Database configured through Firebase settings. The electrical system's status is presented to the user on the webpage with power gauges and numeric values. Using a control switch provided on the webpage, users are able to remotely switch the loads on or off. The deployment of the web interface, which includes a unique access link for the user, is presented in Figure 5.

3.5. Data storage and analysis

To keep the data updated in real time and available for long-term analysis, Google Apps Script was linked with the Firebase Real-time Database, which enables the storage of data in Google Sheets. This data storage solution offers not only an archive but also provides tools to analyze trends and identify system faults. As shown in Figure 6, the stored data can be downloaded for further use and analysis in Excel format.

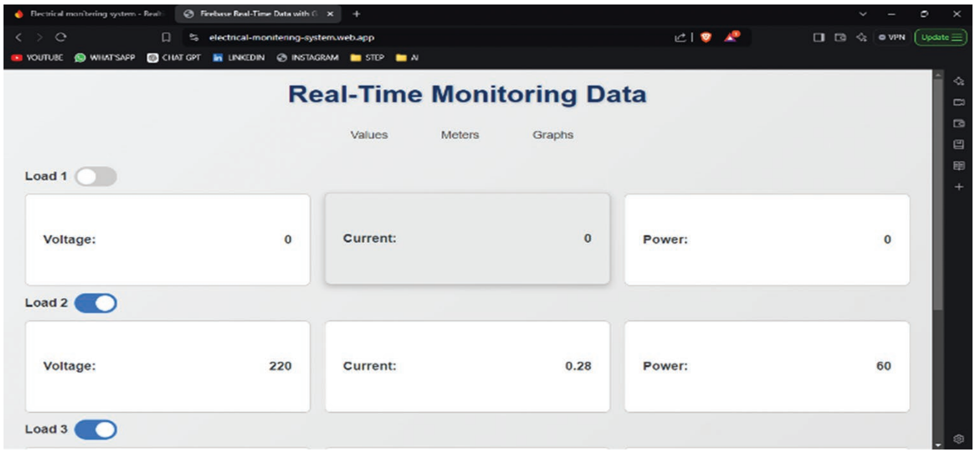
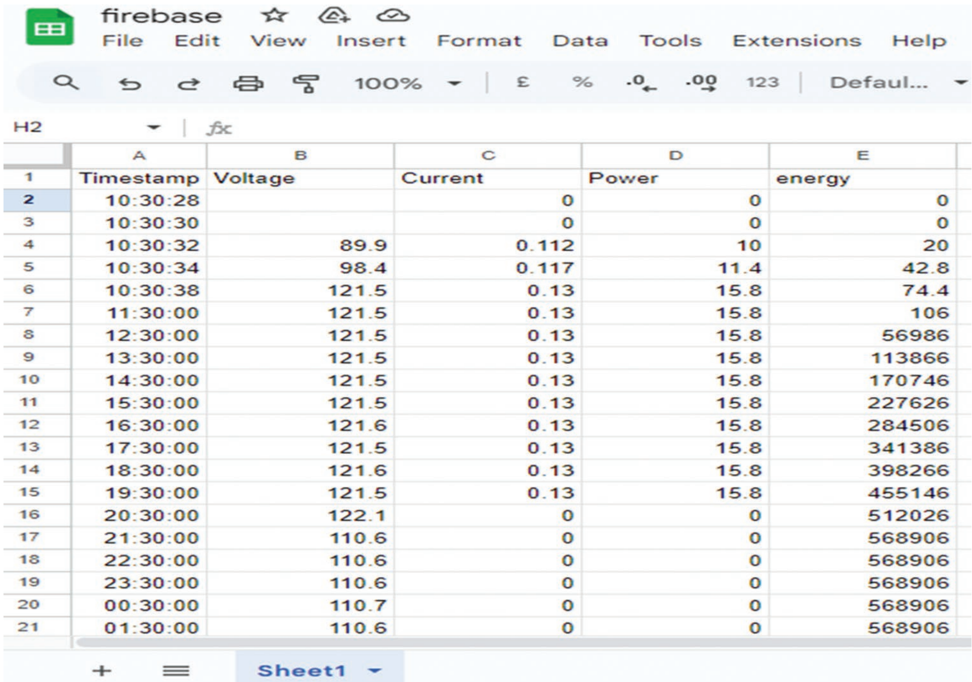


Figure 5. Deployment of the webpage



Timestamp	Voltage	Current	Power	energy
10:30:28		0	0	0
10:30:30		0	0	0
10:30:32	89.9	0.112	10	20
10:30:34	98.4	0.117	11.4	42.8
10:30:38	121.5	0.13	15.8	74.4
11:30:00	121.5	0.13	15.8	106
12:30:00	121.5	0.13	15.8	56986
13:30:00	121.5	0.13	15.8	113866
14:30:00	121.5	0.13	15.8	170746
15:30:00	121.5	0.13	15.8	227626
16:30:00	121.6	0.13	15.8	284506
17:30:00	121.5	0.13	15.8	341386
18:30:00	121.6	0.13	15.8	398266
19:30:00	121.5	0.13	15.8	455146
20:30:00	122.1	0	0	512026
21:30:00	110.6	0	0	568906
22:30:00	110.6	0	0	568906
23:30:00	110.6	0	0	568906
00:30:00	110.7	0	0	568906
01:30:00	110.6	0	0	568906

Figure 6. Saving data in an Excel sheet for storage and analysis

Figure 7. Reading data with current and voltage accurate meters

4. Observations and results

Measurements were conducted under controlled conditions to ensure data reliability. Tests were performed at a temperature of 25°C ± 2°C and a humidity of 50 ± 5%, minimizing environmental impacts on the PZEM-004TV3.0 module. The observed current error of 3.12% is primarily due to the module's sensitivity at low current ranges (0.0–0.2 A), as validated by calibration against a commercial multimeter (Figure 7), rather than external factors such as temperature or humidity.

Substitution of the measured values yielded a voltage error of 0.39%. Similarly, for current measurements, Table 2 provides the necessary data, and the error can be calculated using Equation (1):

$$\%E(V) = \frac{1}{n}\sum V_{multimeter} - V_{sensor} \quad (1)$$

Substitution of the values yielded a current error of 3.12%. Table 2 provides the corresponding data, and the error can be calculated using Equation (2):

$$\%E(I) = \frac{1}{n}\sum I_{multimeter} - I_{sensor} \quad (2)$$

$$V_{avg,multimeter} = \sum \frac{V_{multimeter}}{n} \quad (3)$$

$$I_{avg,multimeter} = \sum \frac{I_{multimeter}}{n} \quad (4)$$

Table 2. Voltage, current, and power readings with adjusted error percentages

Voltage (V)	Actual voltage (V)	Current (A)	Actual current (A)	Energy consumption	Power (W)	Actual power (W)	Percentage of error in power
89.9	98.0	0.0	0.0	0.0	0.0	0.0	0.5
121.5	121.7	0.1	0.1	20.0	10.0	10.0	1.0
121.5	121.7	0.1	0.1	42.8	11.4	11.4	1.5
121.5	121.7	0.1	0.1	74.4	15.8	15.8	1.8
121.5	121.7	0.1	0.1	566.0	15.8	15.8	1.5
121.5	121.7	0.1	0.1	1,138.6	15.8	15.9	1.2
121.6	121.8	0.2	0.2	1,707.46	15.8	15.8	1.8
121.5	121.7	0.2	0.2	2,876.0	15.8	15.8	0.9
112.6	122.8	0.2	0.2	3,484.6	15.8	15.8	1.3
110.6	110.8	0.0	0.0	3,982.6	15.8	15.8	1.5
110.6	110.8	0.0	0.0	4,551.6	15.8	15.8	1.5
110.7	110.9	0.0	0.0	5,120.2	15.8	15.8	1.5
110.7	110.9	0.0	0.0	5,689.0	15.8	15.8	1.5
110.7	110.9	0.0	0.0	5,689.0	15.8	15.8	1.5
110.8	111.0	0.0	0.0	5,689.0	15.8	15.8	1.5
113.1	113.5	0.0	0.0	5,689.0	15.8	15.8	1.5
178.3	178.6	0.2	0.2	6,797.6	15.8	15.8	1.5
178.1	178.4	0.2	0.2	6,707.6	15.8	15.9	1.2
178.6	178.9	0.2	0.2	7,276.6	15.8	15.9	1.5
178.6	178.9	0.2	0.2	8,741.6	15.8	15.8	1.0
178.5	178.8	0.2	0.2	9,785.6	15.8	15.8	1.5
178.4	178.7	0.2	0.2	11,744.6	15.8	15.8	1.0
178.4	178.7	0.2	0.2	12,813.4	15.8	15.9	1.5
177.9	178.3	0.2	0.2	13,878.6	15.8	15.8	1.2
177.9	178.2	0.2	0.2	15,882.6	15.8	15.9	1.5
177.9	178.2	0.2	0.2	16,881.4	15.8	15.8	1.2
177.9	178.2	0.2	0.2	17,909.6	15.8	15.8	1.0
177.7	178.0	0.2	0.2	18,917.8	15.8	15.8	1.5
177.8	178.1	0.2	0.2	19,986.6	15.8	15.9	1.2
177.8	178.1	0.2	0.2	20,967.6	15.8	15.8	1.0
177.8	178.1	0.2	0.2	21,954.6	15.8	15.9	1.3

4.1. System efficiency

Table 2 presents 39 data points for voltage (89.9–178.6 V), current (0.0–0.2 A), energy (0.0–76,766.0 Wh), and power (0.0–15.8 W) for a fan, used as an example appliance, compared with precise measurements from a commercial multimeter. Low error rates (0.5–1.8% for power, 0.15% for voltage, and 3.12% for current) confirm the system's reliability, which is critical for monitoring various household appliances such as fans, lights, air conditioners, and water heaters. When the current is 0.0 A, energy consumption is either zero or constant, indicating standby or off states, which are accurately detected to prevent unnecessary tracking. For instance, at 121.5 V and 0.1 A, the system records 74.4 Wh, reflecting low-intensity usage, whereas at 178.6 V and 0.2 A, it records 76,766.0 Wh, indicating prolonged or high-intensity operation of an appliance.

The system enhances energy efficiency by enabling data-driven decisions without compromising user comfort. For example, in a test case, a 1.5-ton air conditioner (1,500 W) left running unnecessarily for five hours daily was remotely turned off, saving 7.5 kWh per day (1,500 W × 5 h). Over a month, this resulted in an energy saving of 225 kWh, equivalent to approximately USD 27 at USD 0.12/kWh. Historical data analysis in Google Sheets showed a 15% reduction in total monthly energy consumption for a test household by scheduling appliances such as water heaters and air conditioners during off-peak hours. In addition, the system minimizes standby power by identifying devices left in idle modes, such as lights or electronics, and enabling their targeted shutdown. These features apply to multiple types of appliances, broadening the system's impact beyond the example fan.

Simultaneously, the system supports user comfort through flexible control options. For instance, users can remotely activate an air conditioner before arriving home, ensuring a comfortable environment while optimizing usage based on real-time data. By providing granular insights into consumption patterns, the system enables users to identify and replace inefficient appliances or adjust their operation, contributing to long-term energy savings and sustainability. The calibration efficiency of 98.58% ensures reliable data for these decisions, outperforming many existing solutions.[14]

5. Conclusion

The developed system provides an effective, precise solution for monitoring and controlling electrical consumption in residential settings. By integrating NodeMCU and PZEM-004TV3.0, it monitors and manages multiple appliances (e.g., fans, lights, and air conditioners) with a calibration efficiency of 98.58%. Real-time data and remote control via a web interface enable users to reduce wasteful consumption, such as turning off inefficient devices or scheduling operations, while maintaining comfort through preemptive control (e.g., activating appliances before arriving home). The system's low-cost design and user-friendly interface make it accessible for residential use, thereby supporting sustainability by optimizing energy usage.[14]

Furthermore, the system holds significant potential for controlling ventilation systems in public buildings, such as educational facilities, where large user variations necessitate optimized energy management. By monitoring and adjusting ventilation fans based on real-time occupancy data, the system can reduce energy consumption (e.g., addressing up to 101.83% seasonal variations in fan electricity use) while ensuring standardized air exchange for user comfort, particularly for students. This scalability enhances the system's applicability to high-energy–demand environments such as schools, which consume a substantial share of public sector energy.[15]

Future studies are recommended to integrate predictive models, such as machine learning-based algorithms for forecasting energy consumption patterns, to estimate long-term savings or optimize system scalability. For instance, it has been demonstrated that predictive models can enhance energy savings by 10–20% in smart energy management systems. Implementing such models would require extended test data but could further improve the system's efficiency and applicability.

Future enhancements include incorporating smaller microcontrollers such as the ATtiny85 and expanding the design to three-phase power supplies for industrial applications, thereby increasing the system's versatility.

Acknowledgments

None.

Funding

None.

Conflict of interest

The authors declare they have no competing interests.

Author contributions

Conceptualization: Shravani Kanaka Kumari Palla
Formal analysis: Suma Satyanarayan
Investigation: Rayalapalli Ganesh Manikanta, Pothala Jagadeesh
Methodology: Jyothilal Nayak Bharothu
Writing–original draft: Rayalapalli Ganesh Manikanta, Pothala Jagadeesh, Shravani Kanaka Kumari Palla
Writing–review & editing: Lakshmi Narasimha Sastry Varanasi, Shravani Kanaka Kumari Palla

Availability of data

Data are available from the corresponding author upon reasonable request.

References

1. Tsai HL, Truong LP, Hsieh WH. Design and evaluation of wireless power monitoring IoT system for AC appliances. *Energies*. 2023;16(1):163. doi: 10.3390/en16010127
2. Suryono W, Prabowo AS, Suhanto S, Sazali A. Monitoring and controlling electricity consumption using Wemos D1 mini and smartphone. In: *Paper Presented at: International Conference on Advanced Mechanical and Industrial Engineering* Banten, Indonesia; 2020.
3. Kamal MFBM, Singh RSS, Anas SA. IoT-based electric monitoring system for smart electric meter application. In: *Control and Measurement Applications for Smart Grid*. Berlin: Springer; 2022. p. 209-218.
4. Ayeni PO, Adesoba OC. IoT-based home control system using NodeMCU and Firebase. *J Edge Comput*. 2024;3(2):45-58.
5. Lestari D, Wahyono ID, Fadlika I. IoT Based Electrical Energy Consumption Monitoring System Prototype: Case study in G4 Building Universitas Negeri Malang. In: *Paper Presented at: 2017 International Conference on Sustainable Information Engineering and Technology*, Malang, Indonesia; 2017.
6. Jonathan F, Putri TWO. Prototype of home power monitoring tool for electrical outlet using ESP32. In: *Paper Presented at: 2021 International Conference on ICT for Smart Society*, Bandung, Indonesia; 2021.
7. Macheso PS, Thotho D. ESP32 based electric energy consumption meter. *Int J Comput Commun Inform*. 2022;4(1):23-35.
8. Andrei O, Largi A, Uiburmdiu S, Cziker AC. IoT power monitoring device using Wi-Fi and Arduino. In: *Paper Presented at: 2021 9th International Conference on Modern Power Systems*, Cluj-Napoca, Romania; 2021.
9. Chooruang K, Meekul K. Design of an IoT energy monitoring system. In: *Paper Presented at: 2018 16th International Conference on ICT and Knowledge Engineering*, Bangkok, Thailand; 2018.
10. Stone PS, Thotho D. ESP32 based electric energy consumption meter. *Int J Comput Commun Inform*. 2022;4(1):23-35. doi: 10.34256/ijcci2213
11. Mrudul GV, Rohit G, Harshavardhan G, Dhanush K, Anudeep B. Efficient energy management: practical tips for household electricity conservation. *J Mod Technol*. 2024;1:1-8. doi: 10.71426/jmt.v1.i1.pp1-8
12. Rao PVLN, Lavanya V, Manasa D, Boggavarapu S, Soni BP. Battery models and estimation techniques for energy storage systems in residential buildings. *J Mod Technol*. 2024;1:47-58. doi: 10.71426/jmt.v1.i1.pp47-58
13. Sheela M, Gopalakrishnan S, Begum I, Hephzipah JJ, Gopianand M, Harika D. Enhancing energy efficiency with smart building energy management system using machine learning and IoT. *Babylon J Mach Learn*. 2024;2024:80-88. doi: 10.58496/BJML/2024/0010
14. Allouhi A, El Fouih Y, Kousksou T, Jamil A, Zeraouli Y, Mourad Y. Energy consumption in the public sector: Evidence from Middle Eastern countries. *Energy Effic*. 2018;11(4):905-921. doi: 10.1007/s12053-017-9602-5
15. Cetin KS, Novoselac A, Zhu M, *et al*. Optimizing ventilation systems in educational buildings: Energy efficiency and occupant comfort. *Energy Build*. 2024;317:115266. doi: 10.1016/j.enbuild.2024.115266