American Journal of Smart Technology and Solutions (AJSTS)

ISSN: 2837-0295 (ONLINE)

VOLUME 5 ISSUE 1 (2026)

PUBLISHED BY
E-PALLI PUBLISHERS, DELAWARE, USA

American Journal of Smart
Technology and Solutions (AJSTS)

Volume 5 Issue 1, Year 2025
ISSN: 2837-0295 (Online)
DOI: https://doi.org/10.54536/ajsts.v5i1.7146
https://journals.e-palli.com/home/index.php/ajsts

Development of an Energy Monitoring and Remote Load Control System

OLADIRAN. Sammy Olusola[1], Edison Nosakhare George[2*]

Article Information

Received: November 10, 2025

Accepted: February 26, 2026

Published: May 08, 2026

Keywords

Environmental Automation. Internet Of Things (IoT), Real-Time Energy Monitoring, Remote Load Control, Smart Energy Meter

ABSTRACT

Abstract - The management of electricity in Nigeria face challenges, which include issues with inaccurate billing, a lack of consumer awareness with regard to electricity consumption, and a general lack of control over domestic appliances. Even though prepaid metering technology provides greater billing precision, there is a lack of real-time feedback and limited load control functionality. In this paper, the design and development of an IoT-based smart energy metering and automatic load controlling system at an affordable cost is presented. This IoT-based system uses current sensors (ACS712) and voltage sensors (ZMPT101B), which work in conjunction with the NodeMCU ESP8266 microcontroller to sense the real-time values of the electrical parameters, namely the voltage, current, power, and energy. These values are further sent via Wi-Fi connectivity to the Arduino IoT Cloud, where they can be displayed using the Arduino IoT Remote application, which further allows remote control functionality through the relay. For efficient energy consumption, environmental automation has been implemented by the addition of a BH1750 light sensor and a DHT11 humidity sensor. The former can self-regulate a lighting load depending on the surrounding light conditions, whereas the latter can control a socket outlet connected to a fan depending on the humidity. Performance analysis has been carried out by the use of domestic appliances like lighting and plug-in appliances. Results have confirmed the measurement accuracy within a tolerance of ±2.5 V and ±0.2 A compared to a reference multimeter. Data updates were completed within a time span of 2.0–3.5 seconds, whereas the relay control response time ranged from 1.0–3.0 seconds. The developed prototype proves that the integration of smart metering and environment-aware automation through the IoT has the capability to offer efficient and effective energy management. The system has the ability to offer a suitable means to reduce wastages of energy and cut down on the costs of electricity consumed by Nigerian homes.

INTRODUCTION

Smart meters are changing the face of electricity consumption with their capabilities for near-real-time monitoring, improved billing accuracy, and better load management. Unlike traditional electric meters that only offer rudimentary feedback and require manual reading, smart electric meters allow customers to monitor energy consumption at an appliance level, manage peak demand, and even prevent energy theft. By 2020, the number of smart electric meters increased substantially in the US, with over 50% of the approximate 150 million electric meters in use at that time being smart electric meters, having started with less than double that number in 2011 (Federal Energy Regulatory Commission, 2020) .

For instance, the power industry in Nigeria has been gradually moving away from estimated metering towards prepaid meters, with about 4 million units installed as of 2023 (Nigerian Electricity Regulatory Commission, 2023). The use of the prepaid meter promotes the pay-before-use culture and the aspect of accountability but does not offer the real- time usage of the appliances.

The Internet of Things (IoT) offers an opportunity in the energy management field. IoT-based meters allow remote surveillance and control of electrical appliances through the use of sensors and cloud technology. The sensing capabilities offered by environmental sensors such as light and humidity sensors make it possible to automate loads in an intelligent manner. Many research studies have explored the development of smart metering systems based on IoT technology. These include designs based on microcontrollers and Wi-Fi connectivity for smart surveillance at the appliance level (Muralidhara *et al.*, 2020; Gavhane *et al.*, 2021; Kumar *et al.*, 2021), as well as platforms offering dynamic billing, tampering detection, and alerts in real-time (Mahmood *et al.*, 2020; Komathi *et al.*, 2021; Aniedu *et al.*, 2017).

Despite these improvements, most of the smart meters in Nigeria are still controlled by the power firm, thereby limiting user engagement. Consumers cannot in any way know their current power usage per appliance, l*et al*one control appliances from a distance. Also, the current technology hardly uses environmental factors in enabling automatic load management, thus resulting in wastage of energy.

In this background section, a low-cost IoT system for energy monitoring and load control based on the NodeMCU ESP8266 microcontroller has been proposed. The system measures the crucial parameters like voltage, current, power, and energy, and can control appliances remotely through a mobile application. Environmental sensing using a BH1750 light sensor and a DHT11 humidity sensor enables the autonomous control of certain appliances depending upon the environmental conditions. Thus, the proposed system combines remote

---

[1] Department of Information and Communication Engineeering Federal University of Technology, Akure, Nigeria
[2] Department of Information and Communication Engineering Federal University of Technology Akure, Nigeria
[*] Corresponding author's e-mail: edisonnosakhare@gmail.com

control and environmental sensing for a cost-effective solution for energy management purposes. The structure of the paper is as follows: Section II discusses the related work. Section III discusses the system architecture and research methodology. Section IV describes the implementation design and experiment setup. Section V presents results and discussions. Section VI concludes the paper and suggests future work.

LTERATURE REVIEW

There has been extensive research work on smart energy monitoring as well as IoT-based load control. The following are the in-depth reviews of some of the relevant work:

In a study by Suryachakam *et al.* (2019), a low-cost home automation system was designed and developed using NodeMCU and Android mobile application capability for appliance control as scheduled or based on presence and environmental conditions. The system was capable of real-time monitoring but required complete dependence on the internet. As a contrast to Bluetooth range-based control systems that require pairing for control and remain relevant for short distances only, the Wi-Fi-based system ensures longer distances and thus effectively works as a suitable option for distant energy monitoring related tasks.

Similarly, Anupama and Mahadevaswamy (2018) designed a Bluetooth-based energy monitoring system that monitored voltage, current, and active power and was capable of interfacing with the Android mobile app through BLE technology. Though simple and energy-efficient with a range relevant for this task, this system requires a pairing process and supports control distances relevant for short range monitoring only.

Kumar (2014) designed a smart home automated system with Arduino having Ethernet connectivity as a web server and was capable of appliance control through HTTP requests as interfaced with Android apps. However, continuous dependence on the internet would make this system restricted for use in regions with poor bandwidth.

Reddy *et al.* (2016) designed a comprehensive IoT-based home automation system built with Arduino Mega and Wi-Fi connectivity for manual or automated switches and capable of monitoring appliance status in real-time. The system was primarily focused on automated control rather than total energy monitoring analysis at the appliance side as intended here.

Khunchai and Thongchaisuratkrul (2019) accomplished a system integration of NodeMCU with motion sensor and temperature sensor through NETPIE IoT platforms for cloud-based appliance control advantages. Though successfully accomplished for automated control of appliances at houses, this preceding research work was not primarily focused on energy monitoring analysis as desired here.

Bhatt and Patoliya (2016) proposed a cost-effective automated control system for homes based on MQTT messaging protocols over Wi-Fi-enabled devices; as a benefit of this system, the energy consumption was minimized as was human intervention as required; yet this system supported dependence on home servers serving as a setback here for practical usage.

Samson *et al.* (2019) presented a Raspberry Pi-based IoT energy monitoring system with mobile application support for monitoring and billing. Although effective for real-time monitoring, the system lacked a strong focus on appliance-level load control and was constrained by deployment limitations related to its communication architecture.

Zakirulla *et al.* (2020) proposed a smart meter with voice notifications for visually impaired people to increase inclusiveness, although without extensive testing. Also, Karthick *et al.* (2021) designed a miniature smart energy meter based on ESP8266 with a relay-switching function and peak-demand notifications, but the reliance on the cloud undermined fully functional performance even with detailed monitoring.

Rupesh and Selvan (2021) designed a low-cost and modular smart metering system based on Arduino and ESP8266 with data uploading to ThingSpeak for scalability, which remained dependent on the cloud infrastructure.

Gavhane *et al.* (2021) used Arduino Mega, sensors, and relays with ESP8266 to support local and cloud monitoring with flame detecting and voice-controlling capabilities for an automation, safety, and energy management system, which was mainly for household applications.

Alaudin *et al.* (2018) designed a real-time monitoring system based on Arduino UNO and NodeMCU for voltage and current measurement and uploading to ThingSpeak for analysis, without any interactive functions for far-end switching control.

Yaghmaee and Hejazi (2018) designed a smart meter with a relay-switching function for load control and environmental analysis, but with a system complexity and multi-sensor and infrastructure requirement that increases the implementation and operational costs.

Thakare *et al.* (2016) designed an energy meter based on Arduino Nano and ESP8266 with a measurement accuracy of 92.6% for IoT smart energy management, without direct voltage measurement and appliance-switching notifications.

From the above reviews, it is clear that the current available systems either involve monitoring alone, control alone, rely greatly on internet/cloud connectivity, or are complex with multiple devices. There is a gap in the design and implementation of an easy and affordable system, mobile device-centered, incorporating real-time monitoring, relay-controlled functions, and environment automation with limited internet connectivity. To bridge this gap, this study aims to use NodeMCU ESP8266 together with an ACS712 current sensor, ZMPT101B voltage sensor, BH1750 light sensor, DHT11 humidity sensor, and an interactive mobile application for both local and remote control functions.

MATERIALS AND METHODS

Overview of the Proposed System

The proposed system is a single-phase smart energy metering system with remote control capabilities for connected electrical devices. The methodology is based on developing a smart device that incorporates the concepts of electrical energy measurement, environmental measurement, control, as well as supervision using cloud services into a small-form-factor device. The developed system can monitor the voltage, current, power, as well as energy consumed by the devices connected to it. Additionally, the system can control the connected devices manually or automatically depending on the changes in the environment. The methodology ensures that the developed system is cost-effective by using a single microcontroller to perform all the functions required by the system. Therefore, the developed system can replace traditional smart meters since they lack the capability to monitor the devices connected to them or control these devices automatically depending on changes in the environment.

System Architecture

The system architecture consists of four primary functional subsystems: electrical measurement, environmental sensing, control and actuation, and communication/interface. These systems function in a coordinated manner with a central processing unit, thus providing a comprehensive smart energy management system.

The measurement subsystem is responsible for measuring the electrical parameters of the attached load. Voltage sensing is achieved through the ZMPT101B module, which offers galvanic isolation between the mains supply, which is at a high voltage, and the electronic circuits, which operate at a low voltage. The current is sensed using the ACS712 Hall effect-based current sensor, which offers non-intrusive current sensing. Due to the single-channel Analog-to-Digital Converter (ADC) of the ESP8266, the voltage and current signals are sampled separately to prevent interference between the signals.

The environmental sensing subsystem is comprised of a digital light intensity sensor BH1750 and a humidity sensor DHT11. The environmental sensors provide real-time illumination and humidity level data to facilitate autonomous control of lighting and ventilation loads.

The control subsystem in the system is designed to use electromagnetic relay modules for the switching of electric devices. This provides electric isolation between the microcontroller and the electric devices.

Moreover, the communication interface subsystem is enabled through the Wi-Fi capabilities of the NodeMCU ESP8266 board, which supports bidirectional communication with the Arduino IoT cloud. A block diagram representing the interaction of these sub-systems is shown in Fig. 1, while Fig. 2 shows the major component the NodeMCU ESP8266.

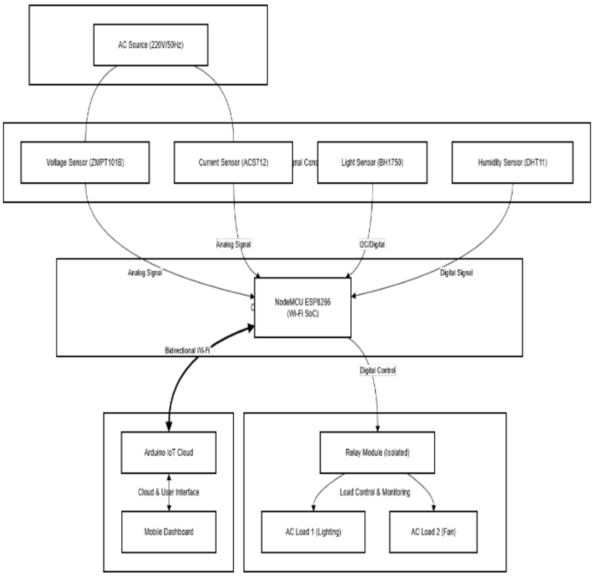

Figure 1: Block Diagram of Proposed System

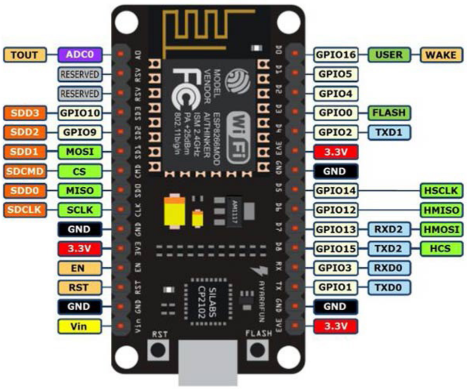

Figure 2: NodeMCU 8266

Hardware Design and Interfacing

The hardware design focuses on measurement accuracy, system reliability, and cost efficiency. The NodeMCU ESP8266 microcontroller was used due to the built-in Wi-Fi support, the number of GPIO pins, power consumption, and the support provided by the developer community, making it suitable for IoT applications.

In order to reduce the effect of the single-channel ADC of the ESP8266, a sequential sampling strategy is implemented. In the proposed system, the voltage and current signals are sampled sequentially without allowing them to interfere with each other, ensuring a smooth data acquisition process. The signals from the environmental sensors are acquired digitally using the I²C and single-wire protocols, which operate concurrently without competing for the shared ADC.

The relay modules are digitally controlled by the microcontroller to enable both manual and automated control of the loads. A regulated DC power supply is used to provide the system with the necessary voltage for operation. In addition, the high-voltage (AC) and low-voltage (DC) signals are physically separated on the prototyping board to enhance safety and reduce electromagnetic interference.

The prototype was constructed on a veroboard during its development phase, while the final hardware configuration was installed in a ventilated cabinet with AC outlets, lamp sockets, and indicator lamps for feedback. Table I shows the hardware configuration used for the system.

Table 1: Hardware Configuration of Proposed System

Component	Sensor Type	NodeMCU Pin	Parameter
ZMPT101B	Voltage	A0	V_{rms}
ACS712	Current	A0	I_{rms}
BH1750	Light	D1 (SCL), D2 (SDA)	Lux
DHT11	Humidity	D 4 (GPIO2)	RH%
Relay 1 & 2	Actuator	D5, D6	Load Control

non-blocking loop programming paradigm for the purpose of continuous sensing, processing, control, and synchronization with the cloud.

Mathematical Modeling and Signal Processing

Energy monitoring in an accurate manner requires processing raw data from an ADC and converting it into useful electrical values. The voltage and current are sampled over a whole period of a 50 Hz AC signal (20 ms). A sampling frequency of approximately 4 kHz is used, yielding about 80 samples per cycle, which is sufficient to accurately capture the waveform for numerical processing. These samples are then used to compute the root mean square (RMS) values, representing the effective electrical quantities responsible for power transfer. The root mean square (RMS) values are calculated as:

$V_{rms}= \sqrt{(1/N)}\ \sum_{(n=1)}^{N} v\,[n]^2$ (1)

$I_{rms}= \sqrt{(1/N)}\ \sum_{(n=1)}^{N} i\,[n]^2$ (2)

Where N represents the number of samples acquired within one cycle.

Average apparent power is approximated, a unity power factor approximation was adopted for computational simplicity. This is acceptable for low-power domestic loads where reactive components are moderate; however, minor power estimation deviations may occur for inductive appliances. Energy consumption is obtained by integrating power over time:

$E_{((kWh))}= 1/3{,}600{,}000 \int_0^T P(t)dt \approx \sum_{(i=1)}^{k} ((V_{rms} \times I_{rms}) \times \Delta t)/3{,}600{,}000$ (3)

This formulation estimates apparent energy under the assumption of unity power factor and does not account for phase displacement or harmonic distortion; therefore, the system is intended for monitoring and comparative analysis rather than utility-grade billing.

This method allows for the constant monitoring of the energy consumption of the associated loads. It is notable that the method applies a discrete sum approximation in calculating the energy consumption. As shown in Equation 3, the integral of instantaneous power is converted to a digital format via the NodeMCU, where energy increments determined at constant Δt are accumulated during the operational period. Dividing the constant 3,600,000 is used to standardize the unit of energy from watt-seconds to kilowatt-hours.

Environmental Automation and Hysteresis Control

To avoid the frequent switching of the relays due to the noisy signal from the sensors and the small variations in the signal, the control strategy of the humidity-based ventilation subsystem was developed based on the concept of 'hysteresis control.' This strategy defines the state of the fan's switch, S_fan as follows:

$$S_{fan} = \begin{cases} 1, if\ RH \geq 60\% \\ 0, if\ RH \leq 50\ \% \\ S_{fan}(t-1), if\ 50\% < RH < 60\% \end{cases}$$

(4)

This method ensures the stable functioning of the hardware, thus avoiding the early degradation of the electromagnetic relay while maintaining the indoor comfort within the optimal relative humidity of 50% - 60%.

The system also includes a Daylight Harvesting mechanism, which ensures a constant level of total illuminance, denoted as L_{total}, while favoring natural lighting sources. The total amount of illuminance present in the workspace is a combination of the natural ambient lighting, denoted as $L_{natural,}$ and the artificial lighting supplied by the lighting system, denoted as $L_{artificial}$:

$L_{total}=L_{natural}+ L_{artificial}$ (5)

Since the artificial lighting load is controlled by a binary electromagnetic relay, the compensator is threshold-based. The artificial lighting load is switched off when natural illuminance is sufficient to meet the operation requirement, minimizing the energy consumption. The control logic that determines the artificial light status, S_{light}, is given by:

$$S_{light} = \begin{cases} 0(\text{OFF}), & L_{natural} \geq 10{,}000\text{lux} \\ 1(\text{ON}), & L_{natural} \leq 500\text{lux} \\ S_{light}(t-1), & 500 < L_{natural} < 10{,}000 \end{cases}$$

(6)

By defining the deactivation threshold as 10,000 lux, a level that is representative of bright daylight, and the activation threshold as 500 lux, a level that is defined by the international standard for task-oriented office lighting, the mathematical model ensures that $L_{artificial}$ is only activated when $L_{natural}$ is insufficient. This directly results in the energy savings observed during peak daylight hours.

Firmware Logic and Cloud Integration

Data acquisition from sensor data is carried out by using a process of time multiplexing analog-to-digital conversion sampling. The data is then transmitted periodically to the Arduino IoT cloud using secure Wi-Fi communication. In order to maintain the integrity of the data, the sensor variables are made read-only on the cloud interface, with the relay states made read-write to allow for the possibility of remote override. In the case of network loss, the system will run an asynchronous reconnection algorithm to allow the local automation logic to run regardless of the internet connectivity.

Sensor Calibration Procedures

Calibration was conducted to ensure measurement accuracy and system reliability. The voltage sensor was calibrated using a reference multimeter by adjusting scaling factors until measured RMS values matched actual mains voltage. Current calibration was performed using known resistive loads and verified with a clamp meter. Environmental sensors were validated through controlled testing to confirm correct threshold-triggered responses. Measurement accuracy was assessed using percentage error analysis:

$$Error(\%) = \frac{|x_{Measured} - x_{Reference}|}{x_{Reference}} X\ 100\% \quad (7)$$

Table 2: Sensor Accuracy

Sensor Module	Parameter	Reference Instrument	Error (%)
ZMPT101B	Voltage (V)	Digital Multimeter	0.77%
ACS712	Current (A)	AC Clamp Meter	9.00%
BH1750	Light (Lux)	Digital Lux Meter	4.00%
DHT11	Humidity (%)	Ref. Psychrometer	8.33%
Relay 1 & 2	Actuator	D5, D6	Load Control

noise in the signals as well as to ensure the safety of the end-user. This was housed in a specially designed and engineered enclosure with ventilation to facilitate standard AC outlets for plug-and-play connections with appliances, lamp holders for lighting tests, and LEDs for live status updates for Wi-Fi and relay status.

Firmware Deployment and Data Pipeline

The system firmware is the intelligence layer that manages a trio of concurrent data streams:

Local Processing Loop: The NodeMCU is continuously polling the environmental sensors and running the ADC multiplexing logic. The Automation triggers (Hysteresis and Daylight harvesting) run entirely on the edge, ensuring system functionality even when there is no internet.

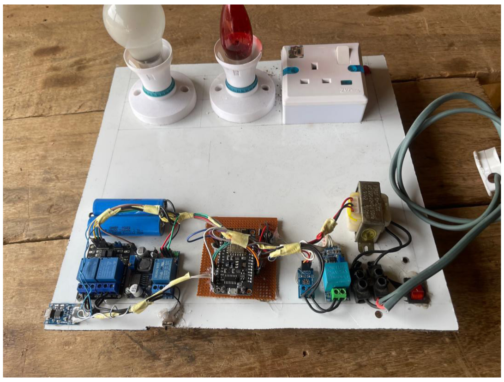

Figure 3: Assembled Prototype

Cloud Synchronization: Through MQTT-based communication, the system sends the telemetry data to the Arduino IoT Cloud at 1-second intervals, thus providing the low-latency requirement that the “Real Time Current Response” analysis demands.

Command Arbitration: The code incorporates a priority-based system in which user-sent remote commands from the mobile interface override decisions made by the sensor, thus ensuring that the user has total control at all times.

System Verification and Functional Testing

A wide variety of functional verification activities were performed after integration, such as:

Stability Testing: The system has been tested under continuous load to check stability and network persistence for 72 hours.

Trigger Validation: The verification ensures that the transition between the “No-Load” state and the “Bulb + Fan” state, as captured by the ACS712 sensor and updated in the dashboard, occurs within a latency period of 2.1 seconds.

Fail-Safe Check: Verification that the system re-enters a safe mode known as the ‘Reconnection Mode’ after Wi-Fi dropouts have occurred, without losing the energy information.

RESULTS AND DISCUSSION

In this section, the results obtained from the implementation of the proposed system will be presented, interpreted, and documented. The results will be evaluated on the basis of the accuracy of the sensors, the reliability of the automation, the latency of the communication, as well as the performance of the proposed system in saving energy.

Electrical Measurement Accuracy

The accuracy of the voltage and current sensing subsystem was validated by cross-checking the results with a digital multimeter that is known to be accurate. The results are shown in Table III.

Table 3: Sensor Accuracy Validation with Real-World Baseline

Tested Load Type	Ref. Voltage (V)	System Voltage (V)	Ref. Current (A)	System Current (A)
No Load	226.4	226.1	0.01	0.03
60 W Bulb	224.8	223.5	0.27	0.31
Standing Fan	223.5	221.8	0.45	0.52
Bulb + Fan	221.4	219.8	0.72	0.81

At no-load conditions, the system correctly identified a near-zero current baseline, with a measured value of 0.03 A, which accounts for the quiescent consumption of the system's internal power supply and the inherent noise floor of the ACS712 Hall-effect sensor. As resistive and inductive loads were introduced, including a 60 W incandescent bulb and a standing fan, the system demonstrated consistent tracking of load current with a maximum observed current deviation of 0.09 A.

Voltage measurements also exhibited high accuracy, with errors remaining below 2 V across all test cases. The results further capture the expected physical phenomenon of voltage sag, where the mains voltage dropped from 226.4 V at no-load to 221.4 V under combined loading conditions. This confirms that the ZMPT101B sensor reliably reflects real-world supply variations rather than idealized values.

The calibration linearity plots shown in Figs. 4 and 5 further validate sensor performance. Both voltage and current measurements exhibit a strong linear correlation with reference values, closely following the ideal y=xy = xy=x line. This confirms that the applied calibration constants effectively convert raw ADC samples into accurate RMS electrical quantities, as defined in Section III.

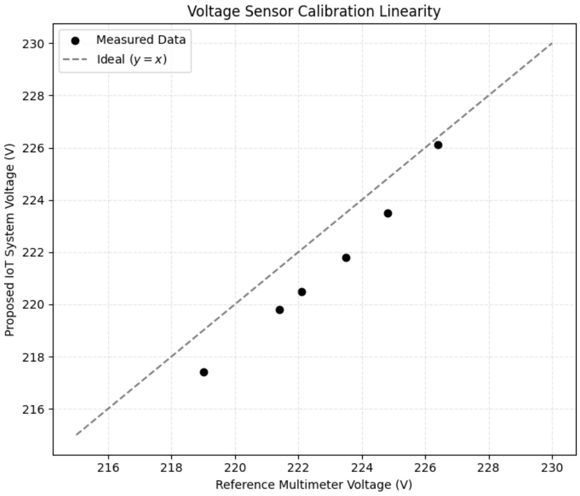

Figure 4: Voltage Sensor Calibration Linearity

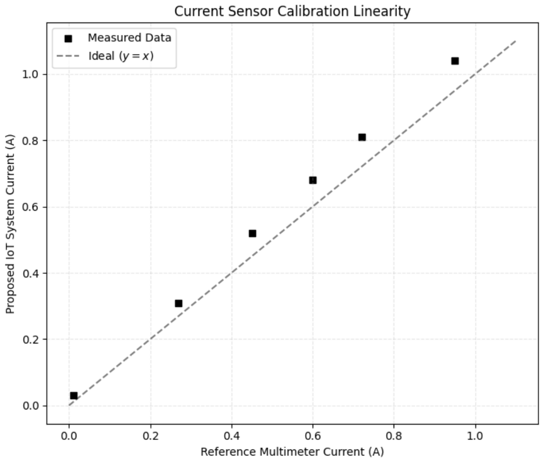

Figure 5: Current sensor Calibration Linearity (System vs. Reference)

Environmental Automation Performance

The effectiveness of the environmental automation logic was evaluated by testing system response to controlled variations in ambient light and humidity. The observed trigger points and response times are presented in Table IV.

Table 4: Automation Threshold Performance

Control Parameter	Trigger Point	Response Time (s)	System Action
Light (Low Level)	495 Lux	1.8 s	Relay 1: Light ON
Light (High Level)	10,020 Lux	1.5 s	Relay 1: Light OFF
Humidity (Upper)	61.2% RH	2.4 s	Relay 2: Fan ON
Humidity (Lower)	49.4% RH	2.1 s	Relay 2: Fan OFF

For lighting automation, the BH1750 light sensor was successful in turning the light relay on and off as the light level went above and below the set thresholds of 500 lux and 10,000 lux. This was found to be within 1% of the setpoint in the firmware.

Similarly, the humidity-controlled fan also employed a relay chatter-free control scheme based on the concept of a hysteresis-based model. The operation of the fan was triggered at an RH value of 61.2%, while the shutdown was triggered at an RH value of 49.4%. This is almost identical to the theoretical model presented in Section III. The time taken by the fans to react to the changes was in the order of 2.1 to 2.4 seconds, making the fans suitable for applications that require comfort-based environmental control.

Communication Latency and System Responsiveness

The communication efficiency of the IoT framework was evaluated by measuring local processing delays, cloud update latency, and remote actuation response times. The results are summarized in Table V.

Table 5: System Communication and Latency Metrics

Performance Metric	Measured Range (s)	Average Value (s)
Local Data Acquisition	0.5 – 1.2 s	0.8 s
Cloud Dashboard Update	2.0 – 3.5 s	2.8 s
Remote Relay Actuation	1.0 – 3.0 s	2.1 s
Wi-Fi Recovery/ Reconnection	25.0 – 40.0 s	30.0 s

Local data acquisition and processing took less than 1.2 seconds to execute, thus establishing that NodeMCU ESP8266 is capable of real-time sensing and computing. Moreover, updates to the cloud dashboards took an average of 2.8 seconds to execute, thus establishing that such systems based on WiFi have an expected level of performance.

Remote control of relays took an average of 2.1 seconds to execute, thus establishing that there is reliable two-way communication between the mobile interface and system. In terms of system error handling, re-establishment of connection took an average of 30 seconds to execute, thus establishing that there is reliable error-handling functionality. Most importantly, system automation continued to execute independently when there was a loss of connection.

Real-Time Monitoring and Dynamic Load Response

This is demonstrated in Fig. 6, which shows a graph of cumulative energy consumption over a period of 30 minutes. This graph clearly shows differences between the no-load, single-load, and combined-load working stages. As demonstrated in the graph, there is a very sharp slope indicating very little consumption of energy in the no-load working stage.

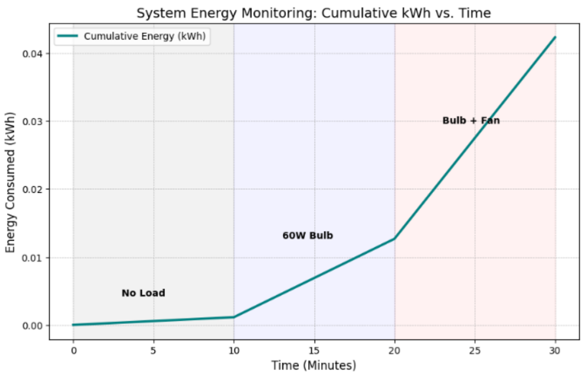

Figure 6: Cumulative Energy Consumption

Dynamic current behavior is again highlighted in Fig 7 below, where step changes in current are indicated according to changes in activated loads. The large changes in current at definite time intervals again emphasize the system's capacity to recognize activity at a level of detail per appliance, thus justifying and substantiating the efficiency of the sequential ADC sampling method.

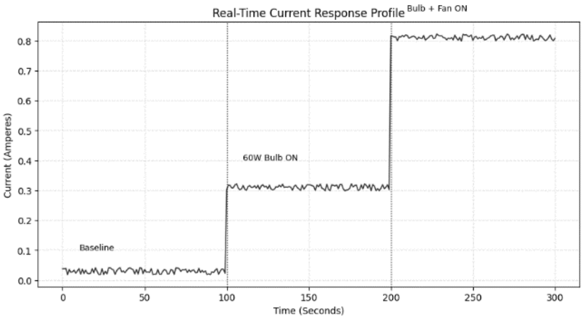

Figure 7: Real time current response profile

Comparative Energy Consumption Analysis

To assess the practical impact of the proposed system on energy conservation, a 72-hour field evaluation was conducted under varying environmental conditions. The comparative performance between conventional manual control and the proposed IoT-based automation is summarized in Table VI and illustrated in Fig. 8.

The total electrical energy consumption EEE (kWh) was computed by integrating appliance power ratings over their respective operating durations, expressed as:

Table 6: Comparative Performance Summary

Operational Day	Manual Mode (kWh)	IoT Mode (kWh)	Energy Saved (kWh)	Savings (%)
Day 1 (Sunny)	1.920	0.995	0.925	48.2%
Day 2 (Ext. Sun)	1.920	0.900	1.020	53.1%
Day 3 (Rainy)	1.920	1.280	0.640	33.3%
Average	1.920	1.058	0.862	44.9%

$E=(\sum(P_n\times t_n))/1000$

where P_n denotes the rated power of each appliance and t_n represents the corresponding operating time in hours.

Under manual operation, both appliances were assumed to operate continuously over a 12-hour daily window (08:00–20:00), resulting in a constant baseline energy consumption of:

$E_{manual}=((60\times12)+(100\times12))/1000=1.92$”kWh”

In contrast, in the IoT-controlled mode, there was a dynamic runtime adjustment of the appliances based on environmental sensing as well as user intervention. To exemplify this assertion, it is noted that during Day 1 under typical sunny weather conditions, there was a runtime of 3.25 hours for lighting and a cumulative runtime of 8 hours for the fan. The corresponding energy consumption of these appliances was calculated as:

$E_{auto,Day1}=((60\times3.25)+(100\times8))/1000=0.995$”kWh”

Similar computations were applied for Days 2 and 3, with the resulting values reported in Table VI.

The highest energy savings was recorded on Day 2 at 53.1%, a period characterized by long hours of daylight and moderate humidity. Conversely, the minimum saved energy was recorded on Day 3 at 33.3%, a period characterized by rainy conditions that required long hours of illumination and moderate humidity to operate the appliances. It should be noted that this reflects the intelligent control of the system, prioritizing user comfort and functional requirements over aggressive energy reduction when environmental conditions demand it.

Overall, the proposed IoT-based energy management system achieved an average energy reduction of 44.9% compared to manual operation. These results confirm the effectiveness of the system as a practical and intelligent solution for energy monitoring and automated load control in residential and small-office environments.

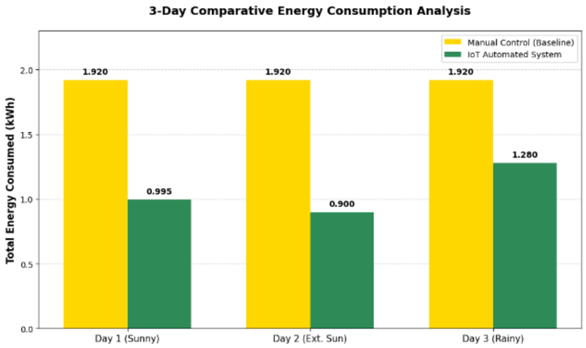

Figure 8: 3-day comparative energy consumption analysis

It is important to note that the reported average energy reduction of 44.9% is highly dependent on the specific environmental and operational conditions under which the evaluation was conducted. The observed savings are primarily influenced by the availability of natural daylight and ambient humidity levels, which directly affect the activation duration of the lighting and ventilation loads. Consequently, these results should be interpreted as scenario-specific rather than universally representative of all residential environments. Under conditions of reduced daylight availability or persistently high humidity, the achievable energy savings would be lower, as the system prioritizes user comfort and functional requirements over aggressive energy minimization. Therefore, the presented results demonstrate the potential of environment-aware automation to reduce energy consumption under favorable conditions, rather than guaranteeing fixed savings across all use cases.

Discussion Summary

Collectively, the experimental results show that the proposed IoT-based smart energy meter can offer accurate electrical measurements, reliable automation, responsive remote control, as well as significant energy savings. The integration of environmental sensing with real-time monitoring transforms the system from a passive meter into an intelligent, context-aware controller capable of supporting energy efficiency without compromising user comfort.

In addition to the quantitative results discussed above, the practical implementation of the system is illustrated through the mobile application interface and the fully developed hardware prototype. Figs. 9, 10, 11 and 12 present the Arduino IoT Cloud mobile dashboard used for real-time monitoring and remote load control, as well as the enclosed working prototype of the proposed system. These figures visually confirm the successful integration of sensing, control, cloud communication, and user interaction components into a functional and deployable energy management solution.

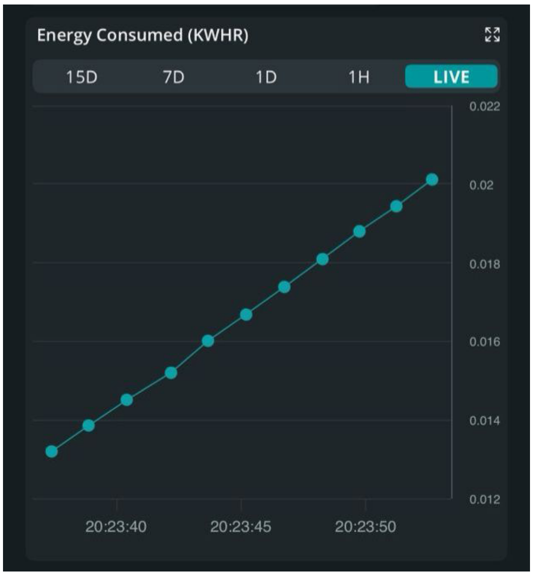

Figure 9: Energy Consumed Reading in Kwhr

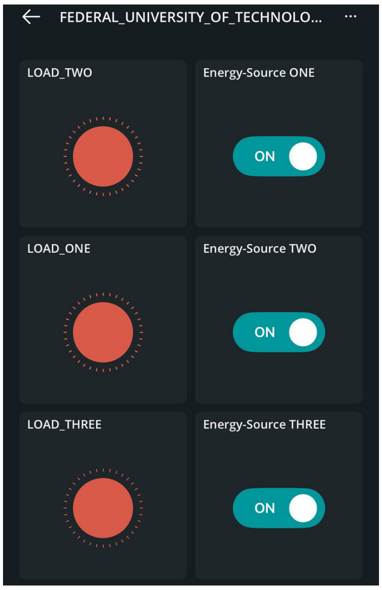

Figure 10: Dashboard with all 3 loads switched ON

CONCLUSION

The paper reports the design and development of an internet-of-things (IoT)-based smart energy meter to enable real-time measurement, monitoring, and control functions for electrical energy. The device has been able to measure the value of voltage, current, power, and

Figure 11: Working Prototype

Figure 12: Enclosed Project

energy consumption using the ACS712 and ZMPT101B sensors, with results being accurate and consistent with those provided by a standard multimeter. Environmental awareness was achieved through the integration of light intensity and humidity sensors to support basic automation. The wireless communication aspect is achieved through the NodeMCU ESP8266 chip as well as the Arduino IoT Cloud.

Real-time data visualization and control through the mobile app ensure the efficient and reliable operation of the connected loads. The relay-control functionality of the loads ensures reliable switching operation, and the auto-control functionality based on environmental factors ensures enhanced efficiencies. The whole system was combined and placed in a fully enclosed prototype setup to ensure performance under different loads. In conclusion, the developed system demonstrates the feasibility of the use of the IoT-based smart energy meters for the enhancement of awareness, remote control, and automatic management of energy.

REFERENCES

Alaudin, A. H. B., *et al.* (2018). Real-time residential energy monitoring device using Internet of Things. Proceedings of the IEEE 8th International Conference on Systems Engineering and Technology.

Aniedu, A. N., Inyiama, H. C., & Azubogu, A. C. O. (2017). Smart meters and advanced metering infrastructure: Panacea to Nigeria's energy billing and monitoring problem. *International Journal of Engineering Research and Development, 11*.

Anupama, S., & Mahadevaswamy, U. B. (2018). Design and development of a smart device for energy monitoring and control of domestic appliances: An Android application. I*nternational Journal of Image, Graphics and Signal Processing, 10*(1).

Bhatt, A., & Patoliya, J. (2016). Cost effective digitization of home appliances for home automation with low-power WiFi devices. Proceedings of the IEEE 2nd International Conference on Advances in Electrical, Electronics, Communication and Bio-Informatics.

Federal Energy Regulatory Commission. (2020). Assessment of demand response and advanced metering. https://www.ferc.gov

Gavhane, V. V., *et al.* (2021). IoT-based energy meter with smart monitoring of home appliances. Proceedings of the 6th International Conference on Convergence in Technology (I2CT).

Karthick, T., *et al.* (2021). Design of IoT-based smart compact energy meter for monitoring and controlling the usage of energy and power quality issues. Sustainable Energy, Grids and Networks, 26.

Khunchai, S., & Thongchaisuratkrul, C. (2019). Development of smart home system controlled by Android application. Proceedings of the 6th International Conference on Technical Education.

Komathi, C., *et al.* (2021). Smart energy metering for cost and power reduction in household applications. Proceedings of the 7th International Conference on Electrical Energy Systems (ICEES).

Kumar, L. A., *et al.* (2021). Design, power quality analysis, and implementation of smart energy meter using Internet of Things. Computers & Electrical Engineering, 93.

Kumar, S. (2014). Ubiquitous smart home system using Android application. *International Journal of Computer Networks and Communications, 6*(1).

Mahmood, H., Iqbal, H. M. Z., & Tahir, M. W. (2020). Automatic energy meter reading using smart energy meter. IEEE Conference Proceedings.

Muralidhara, S., Hegde, N., & R. P. M. (2020). An Internet of Things-based smart energy meter for monitoring device-level consumption of energy. Computers &

Electrical Engineering, 87.

Nigerian Electricity Regulatory Commission. (2023). Quarterly report on metering progress in Nigeria. https://nerc.gov.ng

Reddy, P. S. N., *et al.* (2016). An IoT based home automation using Android application. Proceedings of the International Conference on Signal Processing, Communication, Power and Embedded Systems.

Rupesh, M., & Selvan, D. (2021). Monitoring energy level of smart energy system for home appliances using Internet of Things. *International Journal of Engineering and Advanced Technology, 10*(4).

Samson, B. B., Rajalakshmi, P., & Praveen, P. (2019). IoT-based real-time energy monitoring system with load control and theft detection. *International Journal of Engineering and Advanced Technology, 8*(6S3).

Suryachakam, M., Kumar, P. P., & Technological, O. F. (2019). Domestic appliances controlled automation system with NodeMCU and Android application. *International Journal of Scientific Research in Engineering and Management, 10*(9).

Thakare, S., *et al.* (2016). Implementation of an energy monitoring and control device based on IoT. Proceedings of the IEEE Annual India Conference (INDICON).

Yaghmaee, M. H., & Hejazi, H. (2018). Design and implementation of an Internet of Things based smart energy metering. Proceedings of the IEEE International Conference on Smart Energy Grid Engineering.

Zakirulla, S., Thube, N. N., Sayyed, R., & Nagrare, V. (2020). Talking energy meter for blind and low-income homes using APR33A3 module. *Journal of Emerging Technologies and Innovative Research, 7*(5).