LAUTECH Journal of Engineering and Technology 19 (4) 2025: 48-56
10.36108/laujet/5202.91.0450

Development of an Android-Based Energy Meter Reading with Load Control Monitoring

[1]Okomba N. S., [2]Sobowale A. A., [3]Esan A. O., [4]Omodunbi B. A., [5]Onyebolise I. M. and [6]Awoyemi T.

[1,2,3,4,5,6]*Department of Computer Engineering, Federal University Oye-Ekiti, Ekiti State, Nigeria*

Article Info

Article history:

Received: July 03, 2025
Revised: Aug. 16, 2025
Accepted: Aug. 27, 2025

Keywords:

Android, Display, Microcontroller, Visualization, Technology

Corresponding Author:
nnamdi.okomba@fuoye.edu.ng

ABSTRACT

In today's power systems, managing energy efficiently depends on accurate measurement and effective control. While traditional meters do well in basic readings, they often require manual checks, lack real-time tracking, and cannot control loads remotely, making it harder for users to act quickly and manage consumption effectively. This work introduces an Android-based energy meter reading system with built-in load control, combining the reliability of traditional measurement with modern features like automatic meter reading, live data display, and remote load switching through a simple smartphone app. The setup uses voltage and current sensors, a relay module, and an LCD display connected to an ESP32 microcontroller, with IoT communication enabling instant access and smooth user interaction. Testing showed that the system records energy use with high accuracy, manages loads effectively, and provides stable real-time monitoring. By solving common limitations of conventional meters, it offers better efficiency, user convenience, and potential cost savings, supporting a smarter and more sustainable approach to energy use.

INTRODUCTION

Energy metering is essential in modern society for managing energy consumption, billing, and resource allocation. With the global demand for energy on the rise, accurate and efficient metering systems are crucial for fair distribution, energy conservation, and informed decision-making. However, traditional meter reading systems face significant challenges, including manual processes prone to errors and a lack of advanced features for real-time monitoring. This project addresses these limitations by developing an Android-based system for automated meter reading and load control, leveraging the widespread use of smartphones and advancements in IoT technology (Amin *et al.*, 2021).

Conventional energy meter reading methods, primarily reliant on manual processes and electromechanical meters, face numerous inefficiencies and constraints (Zhang *et al.,* 2023). These include human reading errors, delays in data collection, the need for on-site visits, susceptibility to tampering, lack of real-time monitoring, and difficulties in detecting abnormal consumption or faults quickly. Intelligent lighting control systems for energy savings in buildings using artificial neural networks can assist in limiting power waste (Okomba *et al*., 2024). Electromechanical meters, in particular, are limited by their moving parts, which wear out over time, their inability to integrate with modern communication technologies, and the fact that they only provide cumulative readings without offering any insight into usage patterns (Kumar *et al.,* 2020). Such limitations make it harder to manage energy effectively and place unnecessary economic strain on both service providers and consumers.

Android-based energy meter reading with load control integrates Android smartphones or tablets with smart energy meters equipped with communication modules like Wi-Fi, Bluetooth, or Zigbee (Munthe, 2023). Through a dedicated Android application, users can remotely access and monitor their energy consumption data in real-time, receive notifications about abnormal usage patterns, and adjust their energy usage accordingly (Singh *et al.*, 2020). The load control features enable users to manage their electricity consumption actively by remotely controlling appliances, adjusting thermostat settings, or scheduling energy-intensive tasks during off-peak hours. This empowers consumers to reduce their electricity bills while supporting grid stability and reliability. By leveraging the ubiquity and versatility of Android-based mobile devices, this innovative approach addresses the limitations of conventional systems and provides consumers with greater visibility and control over their energy usage (Rahman *et al.*, 2022). It also lays the foundation for future advancements in smart grid technologies, paving the way for a more sustainable and resilient energy ecosystem. The adoption of Android-based energy meter reading with load control holds promise for revolutionizing energy consumption monitoring. Chamim *et al.* (2023) presented the development of an Internet of Things (IoT)-based system designed for controlling and monitoring household electrical energy usage. The research includes the creation of a dedicated Android application that is integral to the system, serving as the primary interface for users. This application provides real-time data on the energy consumption of various household appliances, allowing users to view detailed consumption statistics, control appliances remotely, and receive alerts about excessive energy use or potential faults. The study details the system's design, including the integration of sensors for real-time data collection and the communication protocols used to transmit this data to the application. The research highlights the advantages of employing mobile applications for energy management, noting improvements in user convenience, real-time monitoring capabilities, and remote control, all of which contribute to better energy efficiency and cost savings in residential settings.

Sahani (2023) developed a system that monitors residential electricity consumption using an ACS712 current sensor and an Android-based application. The study demonstrates the system's effectiveness by showing how Android applications may help with real-time monitoring and management of household energy usage. The paper describes the technical implementation of the ACS712 sensor for accurate electric current measurement, as well as its integration into an Android application that provides users with real-time energy consumption statistics. The integration enables effective monitoring and management of household electricity, demonstrating the practical benefits of using mobile applications for energy management.

Islam *et al.* (2020) built a real-time energy meter monitoring system using the Internet of Things. They used an Arduino Uno and an optical sensor to detect the blinking pulses on the energy meter, then sent that data wirelessly to an Android app. This made tracking power usage quick and reduced human reading errors. A major strength of this design was its low cost and ease of use. However, it required a stable wireless network and might struggle in areas with poor connectivity. Ramalingam *et al.* (2020) developed a smart energy metering setup that connected an Android app to a NodeMCU module. The device converted the meter's analog readings into digital form and transferred them to the app. This made it convenient for users to monitor electricity from their phones. The strength here was real-time access to data and simple installation. Its main limitation was reliance on WiFi coverage, which could limit performance in rural or low signal areas.

Alam and Zeyad (2020) proposed a GSM-based smart meter billing system. They used an ATmega328p microcontroller with a SIM900 GSM module to automate meter reading and send billing updates to users via SMS. This helped notify users about high consumption to encourage energy saving. Its strength was the wide coverage of GSM networks, even in remote areas. A limitation was that it only sent text updates and could not offer detailed usage charts like mobile app solutions. Mohamed and Hanees (2018) created an automated energy meter for Sri Lanka using GSM and ZigBee technologies with microcontroller modules. It collected readings, sent theft alerts, and updated billing automatically without physical meter checks. Its main strength was the dual communication system, offering flexibility and backup in case one network failed. The limitation was the higher cost and complexity compared to single network systems.

MATERIALS AND METHODS

The architecture of the system design is shown in Figure 1, illustrating the integration of key components such as sensors, a microcontroller, a relay and the Android-based user interface

The block diagram represents a smart energy meter system with load control, where the ESP32 microcontroller serves as the central control unit. Voltage and current sensors measure the mains supply and send data to the microcontroller for processing, with results shown on an LCD. A relay, managed by the microcontroller, connects or disconnects the load, while a GSM module allows remote monitoring and control via mobile communication. During testing, the system successfully tracked daily energy use and enabled remote switching. For instance, a fan accidentally left running was turned off remotely, preventing unnecessary energy consumption. The trial confirmed the system's effectiveness in promoting better energy management and reducing waste. The system collects energy consumption data directly from the energy meter through an optical sensor placed over the meter's LED pulse output.

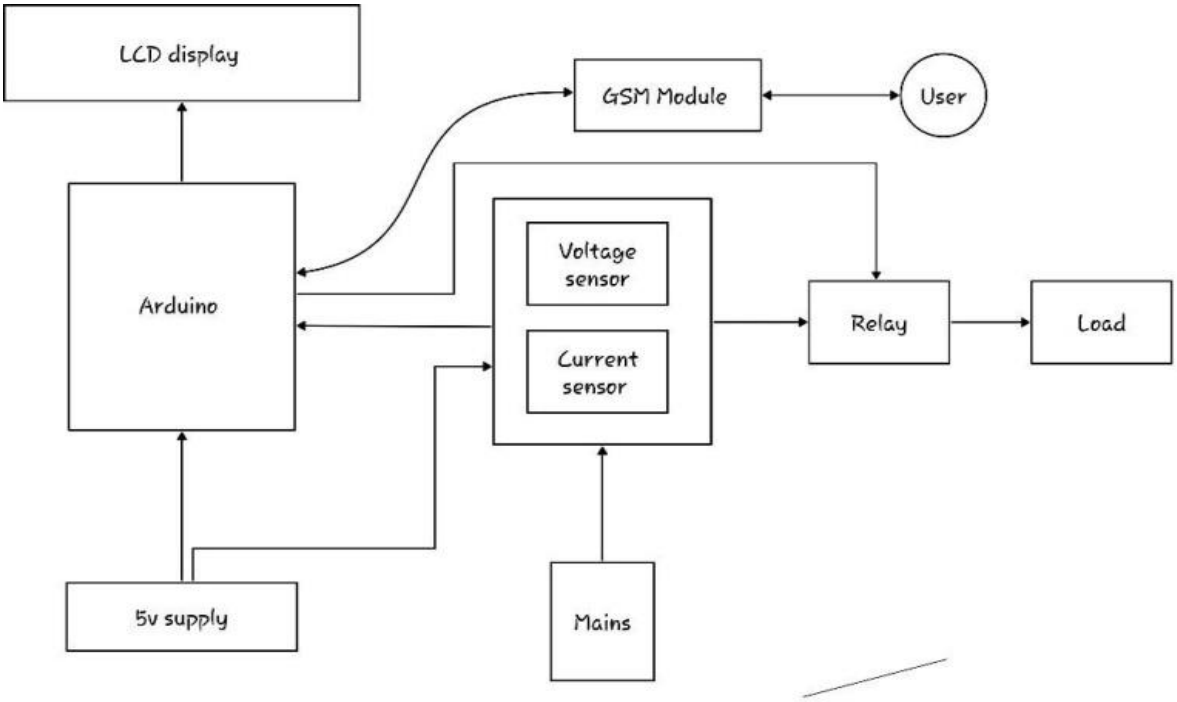

Figure 1: Architecture of the System

Each pulse corresponds to a fixed amount of energy consumed, as specified by the meter's calibration constant. These pulses are detected and processed by the microcontroller, which converts them into readable consumption values. The processed data is then sent via the chosen communication module to the mobile application in near real time. The mobile app stores and displays the data for analysis, ensuring accurate and continuous tracking without manual readings.

Hardware Implementation

The system combines an Android application with smart energy meters that use communication modules such as WiFi, Bluetooth, or Zigbee. At the core is the microcontroller, which handles data collection and communication. Voltage and current sensors measure energy use, while a relay module allows remote switching of connected appliances. A six-ampere power supply and a backup battery keep the system running even during power outages. The microcontroller acts as the brain of the setup. It reads data from a two-hundred-and-fifty-volt voltage sensor and a thirty-ampere current sensor, controls the relay and sends updates to the I2C LCD. The relay works as an electronic switch, making it possible to turn devices such as fans or lights on and off from a distance. The LCD shows real-time readings and is linked to the microcontroller through the I2C protocol.

This arrangement ensures accurate readings, reliable operation and real-time control while keeping costs affordable. The sensors supply precise data, which the microcontroller processes to calculate total energy use over time. The results can be sent to the Android application, where preset tariffs are applied to compute bills automatically. Users can track their consumption history, receive cost updates and remotely switch off appliances to avoid unnecessary expenses. The interconnection of the hardware components is shown in Figure 2

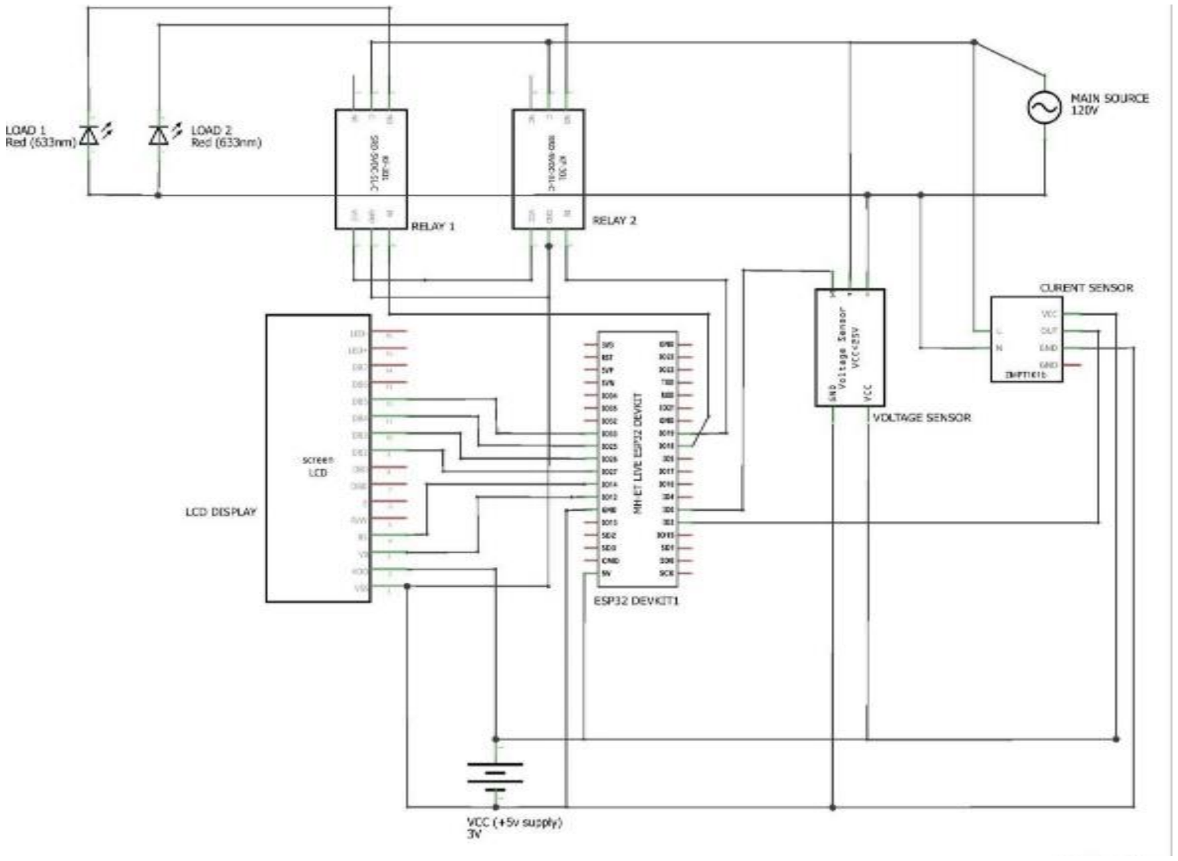

Figure 2: Circuit Diagram

Software Implementation

The software implementation for the smart energy meter system integrates multiple modules and platforms to deliver a comprehensive energy monitoring and control solution. The system leverages the Arduino IDE and C++ for initial hardware-software integration, while the Blynk IoT platform is employed for Android-based remote monitoring and control. The core software of the energy meter system, developed using the Arduino IDE and C++, is made up of four interconnected modules. The Sensor Module collects data from voltage and current sensors, while the Display Module presents the collected power usage information on an LCD. Furthermore, the Wi-Fi Module facilitates communication with the Blynk IoT application, allowing for remote monitoring and control. The Relay Module controls the power supply by executing human commands or automatic procedures to ensure efficient energy management.

Development Process

i. **Requirement Analysis:** The research began with a careful requirement analysis where the needs of the system were defined in terms of accuracy, reliability, and ease of use. Specific values for voltage and current measurement ranges were set, along with display formats for the LCD and the commands that would be sent through SMS.

ii. **Design Phase:** UML diagrams and flowcharts were created to map out the system architecture and behavior, while algorithms were developed to handle power calculations and SMS-based communication.

iii. **Implementation Phase:** The different modules were programmed using Arduino libraries, with external libraries such as Software-Serial and Liquid-Crystal used for GSM and LCD functions. Each section of the code was tested and debugged through the Arduino serial monitor, while a multimeter was used to verify the accuracy of readings from the sensors.

iv. **Integration Phase:** all modules were connected and tested together. Various loads, such as lights and fans, were attached to confirm that the system could measure consumption accurately, process data, and carry out remote control commands without error.

Android Integration Steps

i. **Setup and Configuration**: In this process, the Blynk IoT application was installed and configured for a new user. This involves creating a new account, starting a project within the app, selecting the ESP32 microcontroller as the hardware, and generating an authentication token, which is used to link the hardware to the Blynk Cloud. Widgets were then added and customized to suit the specific needs of the project.

ii. **Firmware Development:** The firmware for the ESP32 was developed using the Arduino IDE. This involved installing the Blynk library, writing the necessary code for sensor data acquisition, Wi-Fi communication, and linking with the Blynk Cloud. The authentication token from the app was embedded in the firmware to ensure secure communication between the ESP32 and the Blynk platform.

iii. **Integration and Testing**: During the integration and testing phase, the hardware components, including sensors, the ESP32 microcontroller, and a relay module, were linked to the Blynk IoT application. The sensors were wired to the ESP32, which was configured to communicate with the Blynk Cloud via the firmware's integrated authentication token.

After physically connecting the hardware and uploading the firmware, the Blynk IoT application was used to monitor and control the system. Each widget in the Blynk dashboard was configured to correspond with specific hardware functions, such as reading sensor data. Extensive testing was conducted to confirm that the widgets in the Blynk IoT application accurately reflected real-time data and provided reliable control over the system. Figure 3 shows the user interface of Blynk IoT. The voltage, current and power are displayed on the app homepage. It shows the consumed power by the appliances. The Blynk app acted as the main tool for checking and controlling the energy meter from anywhere. It simple dashboard displayed live readings in numbers, dials, and history charts, making the information clear and easy to follow. Users could get instant alerts if the system detected unusual power usage, allowing them to respond quickly. Blynk also stores past data in the cloud, making it possible to review and analyze energy usage over time for better decision-making. The ESP32 microcontroller connects to the Blynk Cloud via Wi-Fi, providing data from voltage and current sensors for real-time monitoring.

RESULTS AND DISCUSSION

The developed system successfully integrates Android technology with smart energy meters, providing real-time energy consumption data and load control functionalities. To measure the accuracy, the readings obtained from the smart energy meter were compared against reference values obtained using a calibrated digital multimeter under varying load conditions. This comparison allowed for validation of both the current and voltage measurements recorded by the system.

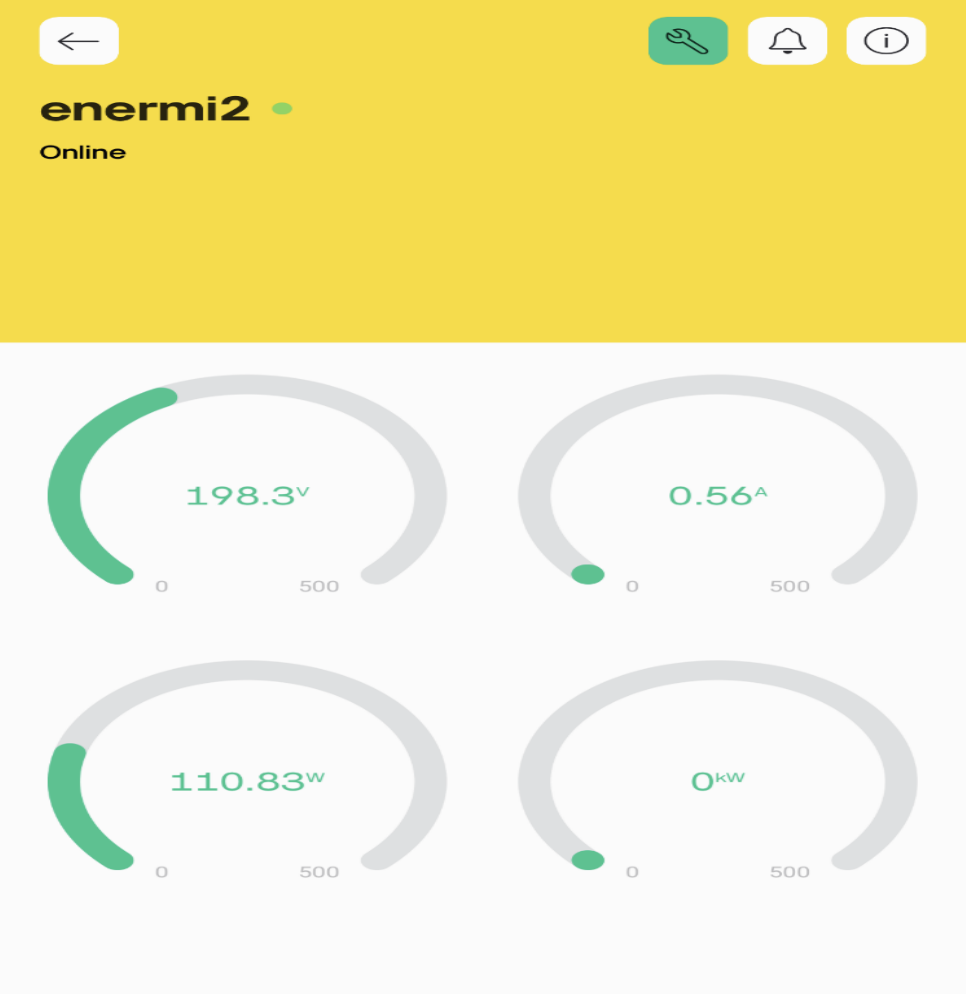

Figure 3: User Interface of the Blynk IoT application

The system's performance was further evaluated by analyzing the relationship between the current, output, and ADC readings, confirming the reliability and consistency of the measurements.

The data in Tables 1, 2 and 3 show a linear relationship between the current and the output voltage for days 1, 7 and 14, respectively. As the current increases, the output voltage increases linearly, ensuring accurate sensor readings. The ADC readings correspond proportionally to the output voltage, confirming the ESP32 microcontroller's accuracy in converting analog signals to digital data. The system's performance was further evaluated by looking at the relationship between the applied voltage (in volts), output voltage (in volts), and ADC readings.

The method used to calculate actual voltage and current values from sensor data was critical for the accuracy and reliability of the Android-based energy meter reading system with load control. To make the readings stable and accurate, the system first applied calibration factors taken from earlier measurements. This helped bring both the voltage and current readings in line with what a standard multimeter showed. The system then took several quick readings in a short time, added them together, and calculated the average. This approach reduced sudden spikes in the values. Extra mathematical filters were also applied to remove high-frequency noise from the data, which made the output steady and reliable across different tests.

Table 1: Day 1 results of the output

Current (A)	**Output Voltage (v)**	**ADC Reading (W)**
0	1.6500	2047.50
5	1.8678	2317.77
10	2.0856	2588.04
15	2.3034	2858.31
20	2.5212	3128.58
25	2.7390	3398.85
30	3.2868	4078.62

Table 2: Table showing day 7 results of the output

Current (A)	**Output Voltage (V)**	**ADC Reading (W)**
0	1.6600	2060.00
5	1.890	2340.20
10	2.0700	2560.50
15	2.3200	2890.10
20	2.5400	3155.40
25	2.7550	3405.00
30	3.300	4120.90

Table 3: Table showing day 14 results of the output

Current (A)	Output Voltage (V)	ADC Reading (W)
0	1.640	2025.30
5	1.8500	2298.60
10	2.0950	2515.00
15	2.2950	2822.50
20	2.5050	3110.10
25	2.7250	3360.40
30	3.2500	4050.00

The ability to calculate RMS values, real power, apparent power, and power factor provided a comprehensive and accurate representation of energy consumption, enhancing the system's effectiveness for real-time monitoring and load control. The system’s RMS values, real power, apparent power, and power factor were not only calculated but also verified using a calibrated digital power meter. Test results with both resistive and inductive loads showed close agreement with theoretical values, confirming the accuracy and reliability of these parameters. The results validate the system's design and implementation, demonstrating its potential for broader adoption in energy management applications.

The implementation and evaluation of the Android-based energy meter reading system with load control reveal its effectiveness and feasibility. The system successfully enabled remote energy monitoring and load control, with real-time data visualization through the Android application, enhancing user awareness and potential energy savings. Performance was consistent with project goals, demonstrating accurate energy measurements and responsive load control. Challenges such as sensor calibration and communication issues were effectively addressed through iterative testing. The system's reliability was confirmed by consistent data collection and precise measurements compared to traditional meters. Future improvements could focus on refining the user interface, incorporating predictive analytics, and exploring additional communication protocols. Compared to existing systems, this solution stands out with its user-friendly interface, real-time monitoring, and remote load control, making it a promising advancement in smart energy management.

CONCLUSION

This project successfully delivered a practical and dependable system for monitoring energy use in real time. The results show that it delivers accurate readings, runs efficiently, and operates reliably. It responds quickly, with almost no delay in updating data, which improves both the user experience and the trustworthiness of energy management. By applying the right calibration and data processing, the system minimizes noise and fluctuations, giving consistent and stable results. Overall, it offers a simple yet effective way to track consumption, improve billing accuracy, and encourage smarter use of electricity, with room for future upgrades and integration into larger smart grid systems.

REFERENCES

Alam, A., and Zeyad, M. (2020). Smart Cities and Buildings: GSM-Based Smart Electricity Energy Meter Billing System. 2019 IEEE International Conference on Power, Electrical, Electronics and Industrial Applications (PEEIACON)*, 7*, 142756–142764.

Amin, S., Rathore, S., and Paul, A. (2021). IoT-enabled smart energy meter for load monitoring and control. *Energy Reports, 7*, 2178–2185.

Chamim, M., Sutopo, W., and Supriyadi, S. (2023). An IoT-based energy management system with an Android application for real-time energy usage control. *International Journal of Electrical and Computer Engineering, 13*(1), 80–89.

Islam, H., Niazi, M., and Khan, A. (2021). Development of an Android-enabled energy meter for efficient energy monitoring. *International Journal of Smart Grid, 5*(1), 15–23.

Kumar, R., Singh, P., and Gupta, S. (2020). Design and analysis of an IoT-based smart energy meter with real-time load management. *Journal of Electrical Systems and Information Technology, 7*(4), 200–208.

Mohamed, M., and Hanees, A. (2018). Development of IoT-Based Smart Energy Meter Reading and Monitoring System. Proceedings of 8th International Symposium-2018, SEUSL, 366-373

Munthe, S. (2023). An Android application for real-time household electricity consumption monitoring using the ACS712 sensor. *Journal of Electrical Engineering, 11*(2), 45–55.

Okomba N. S., Esan, A. O., Omodunbi, B. A., Sobowale, A. A., Adanigbo, O. O., and Aromolaran, P. 2024. Intelligent Lighting Control Systems for Energy Savings in Hospital Buildings Using Artificial Neural Networks, FUDMA Journal of Sciences (FJS), 8(2), 390-398

Rahman, M., Chowdhury, T., and Akter, S. (2022). A low-cost smart metering system with mobile app integration for residential energy monitoring. *IEEE Internet of Things Journal, 9*(11), 8600–8609.

Ramalingam, Y., Wang, Z., and Sun, J. (2020). Remote control of household loads via IoT-based energy meters and Android interface. *Sustainable Energy Technologies and Assessments, 44*, Article 101042.

Sahani, M., Mishra, R., and Swain, S. (2023). Smart energy metering using IoT. In *2023 IEEE International Conference on Power Electronics, Drives and Energy Systems (PEDES)*. 1-6

Singh, K., Verma, D., and Jha, S. (2020). Real-time electricity consumption monitoring using Android applications and IoT. *Energy and Buildings, 224*, Article 110238.

Zhang, T., Liu, C., and Yuan, F. (2023). Implementation of a smart energy meter with remote load control via an Android application. *Energy Informatics, 6*, Article 98.