Spreadsheet-Native IoT Smart Energy Meter Using ESP8266 and PZEM-004T: Architecture and Functional Demonstration

Louis Chai Chang Luk, and Jamaah Suud
Department of Electrical Engineering, Politeknik Kuching Sarawak, Kuching, Sarawak, Malaysia

Abstract

The purpose of this research paper is to explain the functioning of the cost-effective and modular Internet of Things (IoT) type energy monitoring prototype of the system that uses in its operations a PZEM-004T AC measuring device, ESP8266 NodeMCU, the local Wi-Fi system, Google Apps Script, and Google Sheets. The structure is divided into two parts with sensing and edge processing functioning separately from visualization in the browser and cloud logging. The system showed the process of getting all required information about active power, summed up energy usage amounts, rejected all false readings, created time-registered spreadsheets, and collected all the data following the created system. The cost of electronics described in the paper is RM85, with the exclusion of its safety components and cases. One of the investigations showed that there was a summation error; the summation resulted in RM0.186 instead of RM0.040 as shown in the last reading. The results of the summation indicate that it is better to display the last aggregated reading of the system. The working prototype is aimed to show that it is possible to create educational tools with its help.

Keywords: *Internet of Things, smart meter, ESP8266, PZEM-004T, Google Sheets, energy monitoring, low-cost instrumentation*

1. Introduction

Electricity monitoring is moving away from methodical meter readings to connected systems providing information on consumption in near-real time. Modern smart meter development includes cloud connectivity, demand-side management, visualisation capabilities, power-quality analysis, and edge processing [1-7]. These features have the capacity to enhance energy awareness and operation decision-making while simultaneously increasing hardware cost, software complexity, and security requirements.

The inexpensive prototypes that use microcontrollers of the ESP kind are appealing for an educational need, home experimentation, and small business demonstrations since they have both Wi-Fi connectivity and embedded processing in a compact design. The systems that have been documented in publications have relied on custom mobile applications, Blynk, MQTT, Node-RED, Grafana, and customized cloud dashboards [3,5-10]. Although these solutions are functional, they may also create challenges for installation when a user only needs a clear record of time, energy, power, and estimated expenses.

This research evaluates an alternative that is reliant upon a spreadsheet. The PZEM-004T module measures the AC value of electrical readings and the ESP8266 provides edge computing and creates a

* Corresponding author.
E-mail address: jamaah@poliku.edu.my

Manuscript History:
Received 6 August, 2026, Revised 29 September, 2026, Accepted 29 October, 2026, Published 31 October, 2026
Copyright © 2026 UNIMAS Publisher. This is an open access article under the CC BY-NC-SA 4.0 license.
https://doi.org/10.33736/jese.13967.2026

e-ISSN: 2838-0010

local webpage while Google Apps Script directly inputs into Google Sheets. The spreadsheet operates as the cloud layer for data, providing filtering, graphs, sharing, and downloading without needing an externally operating database or server for dashboard processing.

This study documents a reproducible, spreadsheet-native sensing-to-cloud architecture, demonstrates the available functional evidence, and audits the energy-to-cost calculation. Its intended application is single-phase residential or small-load educational demonstration, not certified revenue metering.

The significant contributions of this research paper are listed below:

1. The development of a low cost and reproducible framework for the realization of methods for AC sensing, data processing, visualization, and cloud data logging;

2. Demonstration of timestamped spreadsheet logging and automatic worksheet-to-worksheet archiving;

3. The audit of billing logic aimed at making a distinction between the cumulative snapshot cost, and the cost per interval; and

4. The formation of the validity pathway which includes such aspects as the accuracy of measurements, reliability of communication, cybersecurity, scalability, and mains safety.

2. Related Work

The research into energy monitoring using IoT covers device-level meters, domestic systems, energy management systems for buildings, and industrial edge devices. Device-level energy monitoring was investigated by Muralidhara et al. [9], while Karthick et al. [8] reported integration of monitoring, fire regulation, supply quality, and demand management functions for commercial buildings. Several authors investigated smart energy management systems using real-time load profiles and cloud-based demand-side management [6,7]. More recent studies show that secure communication, general compatibility, validation at the site, harmonics, and edge analytics [1-5] are becoming a focus.

Table 1 compares the current system with representative practices of different systems. The new design does not try to compete with advanced systems in terms of calibration, quality of energy measurement, or potential for industrial implementation. The key factor of the presented design is using a generic spreadsheet as a base for the cloud database, cutting software deployment costs and enabling users with no special training to access their data right away.

According to Table 1, the systems are different in terms of the sensing hardware, types of data interfaces, and areas of application. Prior studies have covered device-level monitoring of consumption, demand management in commercial buildings, protected dashboards, and the analysis of power quality. The current ESP8266-PZEM-004T prototype, on the other hand, emphasizes on using local browser interface and on log data in Google Sheets. This comparison establishes the study's practical distinction in accessible spreadsheet-based data management; it does not establish superior accuracy, reliability, or security.

Table 1. Comparison of representative IoT energy-monitoring systems

Study	Platform and sensing	Interface or data layer	Main distinction
Muralidhara et al. [9]	Device-level smart meter	IoT/cloud interface	Device-level consumption monitoring
Karthick et al. [8]	Embedded commercial-building meter	IoT building-management interface	Power quality, control, and demand-side management
Saleem et al. [6,7]	Smart meters with IoT middleware	Cloud smart-energy platform	Deployment and demand-side management
El-Khozondar et al. [3]	ESP32-based monitoring node	Blynk and messaging interface	Low-cost consumer feedback
Khan et al. [2]	Universal low-cost acquisition hardware	Secure, configurable dashboard	Accessibility and on-site validation
Mirani et al. [4]	Industrial wired and wireless meters	Edge database and cloud	Industrial analytics and reporting
Garcés et al. [1]	Custom smart meter and signal processing	IoT dashboard	Calibration, harmonics, and frequency
Present study	ESP8266 and PZEM-004T	Local web page and Google Sheets	Spreadsheet-native logging and functional demonstration

The gap identified in this study has practical implications instead of being purely theoretical or conceptual. A reference design that is transparent, inexpensive, and based on spreadsheets can assist in conducting classes and prototyping procedures only when limitations are understood and when functional demonstration is not considered equivalent to validation for billing purposes.

3. Methodology

3.1 Study design and requirements

This design-and-functional-demonstration study targeted single-phase residential and small-load educational use. Requirements were affordable core electronics, acquisition of active power and cumulative energy, browser and mobile access, timestamped cloud records, and indicative energy-cost calculations. Functional operation was assessed using the assembled prototype and observed spreadsheet workflow. No claim of certified utility revenue metering is made.

3.2 System architecture

Figure 1 shows the modular architecture. The PZEM-004T-100A and external current transformer sense the single-phase AC load. The ESP8266 receives readings over the sensor serial interface, checks their validity, calculates indicative cost, and serves a local browser interface. Valid records are submitted to a Google Apps Script endpoint for server-side timestamping and insertion into Google Sheets. The local and cloud interfaces support different access paths without requiring a dedicated dashboard server.

As illustrated in Figure 1, the architecture follows two complementary data paths after measurement acquisition. The PZEM-004T sends electrical measurements to the ESP8266 edge node through serial

communication. The edge node processes the readings and makes them available to a phone or laptop through a local Wi-Fi browser interface. In parallel, it submits validated measurements to Google Apps Script, which timestamps and appends records in Google Sheets for subsequent remote access, review, and analysis. The diagram exhibits the operation of the data rather than providing details of the wiring diagrams.

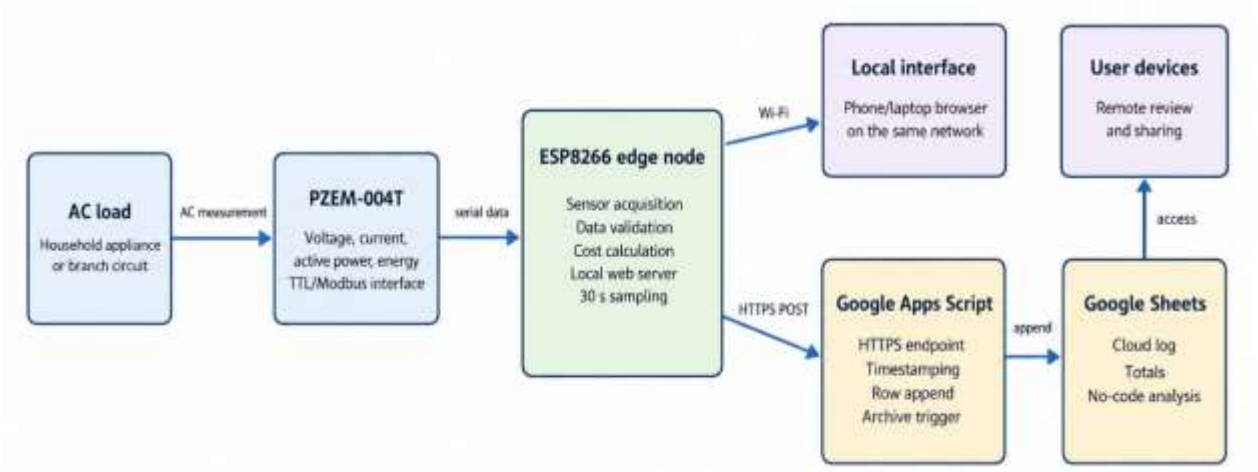


Figure 1. Architecture of the spreadsheet-native smart energy meter.

3.3 Hardware and electrical interface

The ESP8266 has the function of providing connectivity to the Internet through integrated circuits and serial interfaces that are suitable for small devices related to IoT. The PZEM-004T can determine the parameters of alternating current with help of a TTL (Transistor-Transistor Logic) / Modbus connection. The specifications produced by the producer are designed to define the module but the results received as a result of experiments are used for calculating of the functioning of the created device.

Table 2 gives the purpose and aspects of the use of each of the used components. The ESP8266 provides operating and connection functions, whereas the PZEM-004T-100A and split core current sensor give electrical components. The isolated USB supply is responsible for powering the controller and the housing and safety components are needed for the construction. The last column makes the distinction between the known functions of the prototype and aspects of the device the documentary requires extra proof of before its introduction.

Table 2. Hardware components and implementation roles

Component	Function	Implementation detail	Publication limitation
ESP8266 NodeMCU	Edge processing and Wi-Fi	Arduino-compatible firmware, serial communication, and local web server	Credentials and device secrets must not be embedded in public source code
PZEM-004T-100A	AC measurement	Active power and cumulative energy through TTL serial communication	Manufacturer accuracy does not establish system-level accuracy
Split-core current transformer	Current sensing	Clamped around the live conductor	Orientation and installation can affect measurement quality
Isolated USB supply	Low-voltage ESP8266 supply	Phone charger or isolated adapter	A certified isolated supply is required for deployment
Enclosure and protection components	Mechanical and electrical protection	Prototype enclosure, connectors, and protection device	Final construction requires touch-safe terminals, strain relief, and appropriate overcurrent protection

3.4. Firmware and local interface

The ESP8266 sets up the sensor, Wi-Fi connection, and the local web server, making measurements approximately every 30 seconds. Sensor readings that are inaccurate or incorrect (including NaN) are ignored before they can be visualized or uploaded. If any measurements are unsuccessful, the acquisition-error branch will try to make the measurement again during the next cycle. A failed cloud measurement does not render the local measurement invalid, but the implementation described does not store unsent data. Control flow is provided in Algorithm 1, while the persistent-retry feature runs separately.

Algorithm 1. Sensor validation and cloud data logging

Input: PZEM-004T readings; configured demonstration rate r (RM/kWh)
Output: Local display and timestamped Google Sheets records

```
Initialize ESP8266, PZEM-004T, Wi-Fi, and local web server.
REPEAT approximately every 30 s:
    Read active power P and cumulative energy E.
    IF P or E is NaN or otherwise invalid THEN
        Report sensor acquisition error; skip display and upload.
    ELSE
        Calculate indicative cumulative cost C = r × E.
        Update the local browser display with P, E, and C.
        Submit valid measurements to Google Apps Script via HTTP POST.
        IF cloud submission succeeds THEN
            Server timestamps and appends the record to Google Sheets;
            execute the configured worksheet archival routine.
        ELSE retain the local display; retry on a subsequent cycle.
        END IF
```

15 END IF
16 END REPEAT

Note: Persistent offline storage, sequence-number acknowledgements, and authenticated HTTPS are proposed improvements, not features verified in this prototype.

The firmware tested made use of a secure client, but server-certificate verification was disabled. This means that encrypted transport does not imply the transport is authenticated. The firmware needs to configure the Wi-Fi connection before actual field deployment, verify the server certificate or some trust anchor, and authenticate cloud submissions. These are all suggestions and not necessarily features of the implementation being reported.

3.5 Cloud logging and worksheet archival

Google Apps Script receives values, applies a timestamp on the server side, and appends row to the run active worksheet and documented archival process to create an archive worksheet, and to copy the existing rows and delete the currently active data including its header. This is worksheet archival, not CSV file export. Because copy-and-clear operations can conflict with concurrent submissions, production data retention should instead use append-only raw records with time-based partitions or separately generated archival exports.

3.6 Cost-estimation logic

For a fixed demonstration rate, the cost associated with a cumulative-energy reading is

$$C_n = rE_n \tag{1}$$

where C_n is the cumulative cost estimate at sample n, E_n is cumulative active energy in kWh, and r is the configured rate in RM/kWh. Because E_n is cumulative, C_n is also cumulative. The values C_1, C_2, ..., C_n are successive snapshots of the same accumulating quantity and must not be summed.

When row-wise summation is required, the energy assigned to sample n must be the interval increment

$$\Delta E_n = E_n - E_{n-1} \tag{2}$$

and the accumulated cost over N intervals is

$$Ctotal = r \sum^{n}_{n=1} \Delta E_n = r(E_n - E_0) \tag{3}$$

The fixed RM0.20/kWh coefficient is retained solely to audit the original demonstration. A proposed tariff-aware implementation should store the tariff identifier, effective dates, currency, and bracket thresholds, then apply the tariff schedule valid for the billing period. A tiered tariff cannot be represented accurately by multiplying all consumption by a single demonstration rate. The proposed tariff engine has not been implemented or tested in the reported prototype [14].

3.7 Functional verification protocol

Functional verification was based on observable end-to-end behaviours: successful sensor acquisition, local browser display, HTTP submission, timestamped spreadsheet insertion, mobile access, and worksheet archival. Evidence consisted of the implemented firmware and Apps Script workflow, the assembled prototype, and spreadsheet-interface records. The protocol did not include a calibrated reference meter, packet-sequence analysis, controlled network interruptions, penetration testing, or long-duration thermal and electrical-safety assessment.

Figure 2 summarizes the operational sequence implemented by the firmware and cloud-logging workflow. After power-on and initialization of the PZEM-004T and Wi-Fi, the controller acquires active-power and cumulative-energy readings. Invalid sensor values follow an error branch and are retried, whereas valid values are used for indicative cost calculation and the local webpage. The controller then tries sending the data to the cloud endpoint. When submission is successful, Google Apps Script timestamps the submission as well as appends it to the record and carries out the implemented archival process but in case of unsuccessful submission there will be no visible result and the screen will remain unchanged until the next cycle.

3.8 Electrical safety, security, and data responsibility

There is a danger of being connected to the mains highs voltages via the PZEM module. The assembly and testing must be done by qualified force that uses isolation, insulated terminals, limit switches, flame-retardant enclosure, strain relief and proper electrical creepage and clearance. The device cannot be promoted as a certified electricity meter. Fine-grained energy data can disclose the occupancy patterns; access control, authentication, encryption and retention policy must be employed accordingly[15].

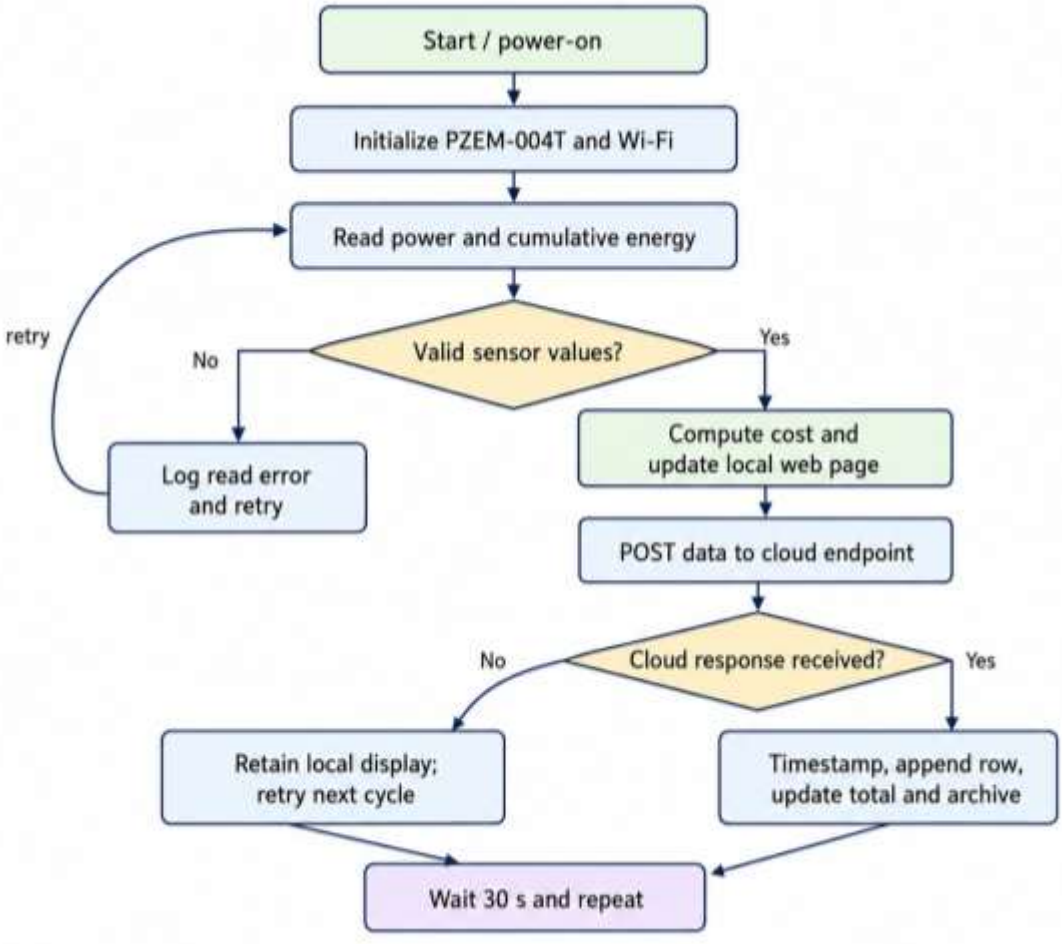


Figure 2. Firmware and cloud-logging workflow.

4. Results

4.1 End-to-end functional operation

The prototype achieved the desired data transmission path from AC sensing to local and online formats. The laboratory setup included the AC load, PZEM-004T, NodeMCU ESP8266, split-core current transformer, voltage-isolated energy supply unit and the safety component. The spreadsheet feature allowed for viewing the records, with the time of entry, including the active power, total energy used and estimated cost.

The data confirms the sensor system from start to finish including timestamps and online inputs. According to the firmware description, sensor acquisition takes 30 seconds; however, the times from the mobile log show that the records do not appear every five minutes. Thus it cannot be determined how the difference happened in terms of data downsampling, averaging or missing records in the transmission process.

In Figure 3, the theoretical aspects of the prototype design are illustrated. Figure 3(a) shows the lab setup in terms of AC load connection, split-core current transformer, PZEM-004T module, ESP8266 NodeMCU, low-voltage supply system, and protection device that is labeled. Figure 3(b) depicts the mobile version of Google Sheets with rows represented in terms of dates, times, active power, cumulative energy, and indicative cost. Although the displayed revenue confirms the transmission of information to mobile devices, the timestamps appear with an interval of around five minutes, which raises doubts about the period of transmission between the sensor and the cloud, which is said to be 30 seconds. The cumulative cost indicated in this example will be discussed separately in Section 4.2.

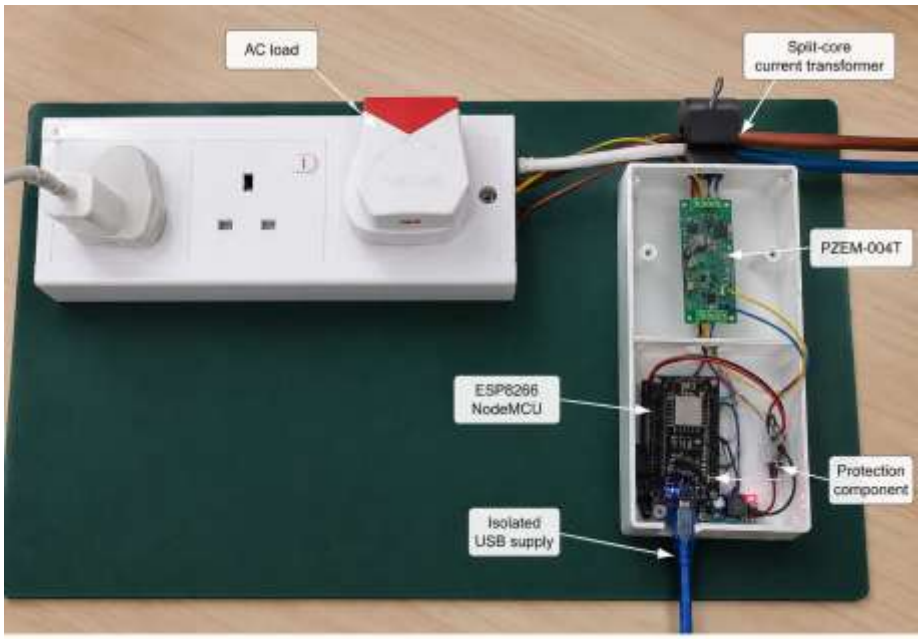
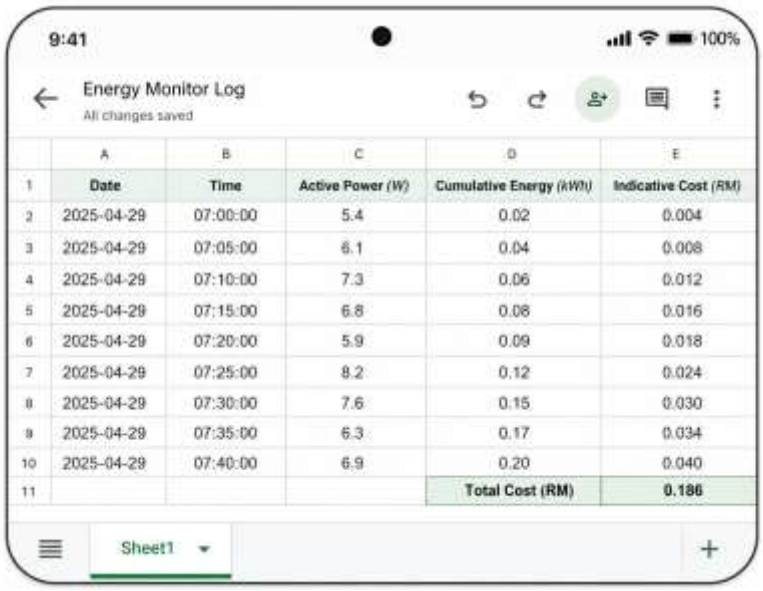


Figure 3a: Laboratory Prototype



	A	B	C	D	E
1	Date	Time	Active Power (W)	Cumulative Energy (kWh)	Indicative Cost (RM)
2	2025-04-29	07:00:00	5.4	0.02	0.004
3	2025-04-29	07:05:00	6.1	0.04	0.008
4	2025-04-29	07:10:00	7.3	0.06	0.012
5	2025-04-29	07:15:00	6.8	0.08	0.016
6	2025-04-29	07:20:00	5.9	0.09	0.018
7	2025-04-29	07:25:00	8.2	0.12	0.024
8	2025-04-29	07:30:00	7.6	0.15	0.030
9	2025-04-29	07:35:00	6.3	0.17	0.034
10	2025-04-29	07:40:00	6.9	0.20	0.040
11				Total Cost (RM)	0.186

Figure 3b: Spreadsheets mobile log

Figure 3. Functional demonstration of the prototype: (a) annotated laboratory assembly; (b) spreadsheet mobile interface.

Table 3 provides an overview of the functional checks that were carried out and the limits of their interpretation. Sensor acquisition, browser access from local computer, data uploaded to the cloud, and providing access via mobile app work successfully as expected, however, indicative billing had partially succeeded as the total calculation of the cumulative cost in the spreadsheet went wrong. Long-duration stability was not evaluated. Accordingly, the reported pass outcomes establish functional operation only, not calibrated metrological performance or quantified network reliability.

Table 3. Functional verification outcomes

Verification item	Observed evidence	Outcome	Interpretive limit
PZEM acquisition	Firmware reads active power and cumulative energy and rejects invalid NaN values	Pass	No calibrated reference-meter dataset
Local web access	Live readings are presented through the local browser interface	Pass	Refresh time and latency were not recorded
Cloud upload	HTTP POST and Apps Script append timestamped spreadsheet rows	Pass	No packet-loss or outage test
Mobile accessibility	The spreadsheet log is visible on a mobile device	Pass	No formal usability study
Worksheet archival	Rows are copied to a timestamped worksheet when the threshold is reached	Pass	The operation is not a CSV export; the 10-row threshold is too small for field use

Verification item	Observed evidence	Outcome	Interpretive limit
Indicative billing	Each row multiplies cumulative energy by a fixed rate	Partial	The row-wise total shown in the spreadsheet uses incorrect aggregation
Long-duration stability	No continuous test record was retained	Not evaluated	Required before field deployment

4.2 Billing-logic audit

Figure 3(b) displays a final cumulative energy reading of 0.20 kWh and an indicative cost of RM0.040 at the demonstration rate of RM0.20/kWh. Its displayed RM0.186 total sums successive cumulative-cost snapshots and thus repeatedly counts earlier consumption. The overstatement is RM0.146, equivalent to 365% of the correct reset-based cumulative estimate. This audit identifies a correction required in the spreadsheet; the supplied evidence does not verify that a corrected formula has been deployed.

Table 4 quantifies the billing-aggregation error visible in Figure 3(b). Using the illustrative RM0.20/kWh rate and final cumulative energy of 0.20 kWh, the cumulative cost since a zero-energy reset is RM0.040. Summing successive cumulative-cost entries instead gives RM0.186, an excess of RM0.146 or 365% relative to RM0.040. Between the initial reading (0.02 kWh) and final reading (0.20 kWh), the charge incurred is RM0.036 and is computed over various accounting limits.

Table 4. Audit of the spreadsheet cost calculation shown in Figure 3(b)

Quantity	Value	Interpretation
Configured demonstration rate	RM0.20/kWh	Derived from the displayed energy and cost pairs
First displayed cumulative energy	0.02 kWh	Starting snapshot in the visible table
Final displayed cumulative energy	0.20 kWh	Latest cumulative snapshot
Correct cumulative cost from a zero-energy reset	RM0.040	0.20 kWh × RM0.20/kWh
Correct cost across the visible interval	RM0.036	(0.20 − 0.02) kWh × RM0.20/kWh
Displayed sum of snapshot-cost cells	RM0.186	Incorrect because cumulative snapshots were summed
Excess relative to the reset-based cumulative cost	RM0.146 (365%)	The displayed total is 4.65 times the correct reset-based value

To adjust the spreadsheet, it must point to the most recent cumulative-cost data cell instead of summing up the entire cumulative-cost column. Alternatively, every cost incurred can be calculated as $r \times (E_n - E_{n-1})$, which indeed may have more complex scenarios behind it. In the case of the visible last reading, the cumulative estimate is RM0.040 and the cost of the range from the first to last displayed reading is RM0.036. A new screenshot or an updated sheet shall be provided to show the implementation of the updated formula

4.3 Measurement accuracy and communication reliability

The study documentation available provides no reference analyzer readings that would allow for bias assessments. As a result, it is impossible to determine any aspects such as system bias, percentage errors, repeatability, confidence intervals, mean absolute error, and root mean square error. The functional test of the sensor allows for data acquisition but comparison of reference meters with respect to the loads and operational conditions still needs to be performed as a part of calibration verification.

The lack of continuous transmission logs with unique identifiers and timestamps indicates that the following parameters remain unknown: successful delivery, duplication, missing transmissions, delay, disconnection frequency, and reconnecting time. A possible future experiment is proposed to conduct a 24 hour experiment with increasing sequence numbers assigned, using LittleFS to preserve not sent messages during outages, and retrying to send data with acknowledgment.

The evidences and following assessments suitable for the validation areas have been presented in Table 5. Voltage, current, power, and energy must be measured through calibrate comparisons. The measurements are carried out several times in order to estimate the repeatability of the calculations. The proposal has been made for the service record of at least 24 hours in order to evaluate the service delivery and to test interruption of the system. In addition to this, the authentication and testing process is called to prove the correctness of the results and the safety from the electric point of view.

Table 5. Status of quantitative validation and required follow-up tests

Performance area	Available evidence	Status	Required test
Voltage, current, and active-power accuracy	No paired reference measurements	Not evaluated	Calibrated analyser comparison at multiple load levels and load types
Cumulative-energy accuracy	No long-duration reference dataset	Not evaluated	Timed energy test against a traceable reference
Repeatability	No repeated observations retained	Not evaluated	Repeated trials with mean, standard deviation, and 95% confidence interval
Delivery-success rate	Timestamped rows observed but no sequence numbers	Not evaluated	At least 24 h logging with unique record identifiers
Latency and reconnection	No device and server timestamp pairs	Not evaluated	Controlled Wi-Fi and internet interruption tests
Cybersecurity	Certificate checking disabled and no authenticated payload	Not evaluated	TLS validation, authentication, replay-resistance, and access-control testing
Electrical safety	Functional enclosure shown	Not certified	Touch-safety, insulation, thermal, overcurrent, and enclosure assessment

4.4 Core prototype cost

Supposed safety BOM includes: isolated lifted USB supply; appropriate rated fuse along with fuse holder; safety terminal connections; insulated coverings; strain relief and cable glands; safety wiring; extra optional parts which depend on the results of electrical designing. The final cost of the prototype equals RM85 + costs of the parts of safety BOM. Hence, no total cost can be provided without receiving the quotations.

The cost of RM85 includes RM60 for NodeMCU development kit, breadboard, wires, and connectors as well as RM25 for PZEM-004T-100A. The mentioned cost refers only to EEs total, i.e., considers only expenses on electronic devices, not on the deployment-ready device price. The total price of the extra BOM should include the cost of isolated power supply, fuse, terminals, insulating case, wiring, and other devices which may be required by the final electrical design.

As shown in Table 6, the reported core electronics cost of RM85 can be broken down into RM60 for NodeMCU development kits and related prototyping accessories, and RM25 for the PZEM-004T-100A device. However, the subtotal does not include the costs of certified isolated power supply, protective devices, suitable terminals, housing, wiring, etc., so RM85 should be seen as the reported cost of the prototype only but not the total cost of a safe installation.

Table 6. Documented core bill of materials

Item	Cost (RM)	Scope note
NodeMCU development set, breadboard, jumper wires, and connectors	60.00	Documented prototype components
PZEM-004T-100A module	25.00	Documented sensing module
Total core prototype cost	85.00	Excludes deployment and certification components

5. Discussion

5.1 Contribution and intended use

Recent research carried out in the field of smart meters has also been focused on the topics such as calibration, parameters of power quality, secure communication, edge analytics and physical implementation [1,2,4,5]. What sets the current paper apart is the narrower framework that provides a minimalistic architecture able to generate ready-to-inspect information with the previously known devices. The level of transparency achieved is very important in engineering education, fast prototyping or energy awareness research where the aim is to track the path of the data rather than the application of the qualified smart meter.

5.2 Billing model and tariff relevance

Data from the spreadsheet audit shows that cumulative snapshots should be differentiated from interval consumption. The row sum of RM0.186 needs to be checked as it is inaccurate. The last cumulative snapshot produced a value of RM0.040 for the used RM0.20 per kWh rate. It is possible to improve the implementation by replacing the SUM-of-snapshots formula with the last reading reference or the interval-delta formula. The test should be conducted using known testing sequences. It

is needed to use effective-dated, versioned brackets of consumption, instead of assuming fixed rates for subsequent tariff support [14].

5.3 Measurement and communication validity

The current contribution is a functional data pipeline instead of a standardized measuring equipment. The specifications of the modules do not determine the accuracy of the entire system since it will depend on positioning of the current transformers, wiring, firmware, sampling, and the test load.

To ensure the reliability of the network, every reading should be accompanied by a device ID, sequence number, and acquisition time stamp. In case of upload failure, the firmware should supplement the request by a small amount of data and store the record in a small queue until the device reboots and reconnects to the network. Acknowledge sequence numbers sent to the cloud and reject any duplicated numbers.

Flash write endurance, queue capacity, and recovery behavior must be tested before using this system in practice.

5.4 Security, privacy, and safety

Figure 4 presents the proposed design of electrical safety features and isolation for the ESP8266-PZEM-004T smart energy monitoring system. This design includes a protective fuse fitted on the AC phase conductor to ensure overcurrent protection. Moreover, metal oxide varistor (MOV) can be provided to help in dealing with voltage surges. In addition, a current transformer (CT) is outfitted on the live conductor to allow measuring load current without a direct series connection. After collecting all electrical parameters such as voltage, current, active power, and total energy, the PZEM-004T passes the acquired data to ESP8266 through serial communication. To ensure electrical safety, the proposed design emphasizes the need for proper isolation measurements, which can be safely operated without direct contact with mains-connected circuitry. Moreover, a certified isolated 5 V DC power supply is used to power the ESP8266; the mains-powered components are properly shielded and placed to avoid unintentional contact with high voltage. This arrangement focuses on providing electrical protection and isolation in the circuit; however, this drawing depicts the theoretical model rather than the practical wiring diagram and needs to be verified for components characteristics, isolation values and safety level.

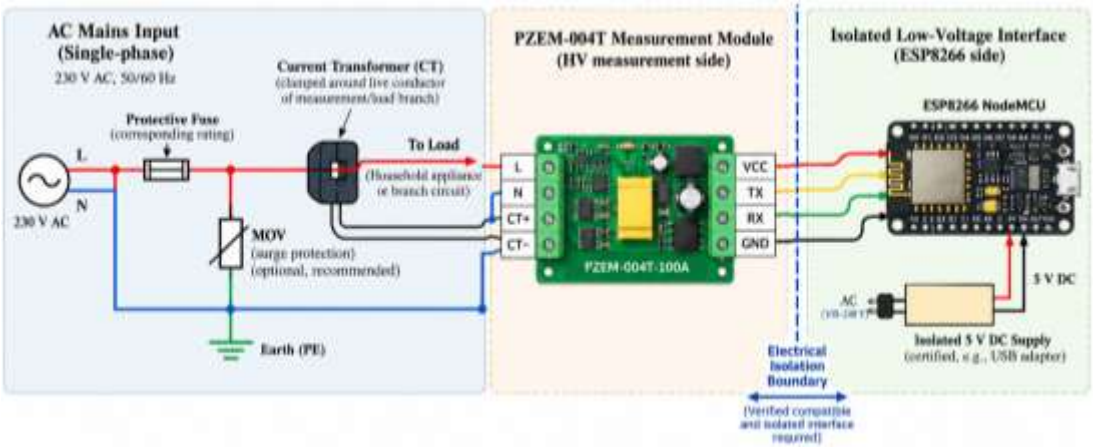


Figure 4: Proposed safe integration of a PZEM-004T energy monitoring module with an ESP8266 NodeMCU using an isolated low-voltage interface.

A security design that is oriented towards deployment should replace Wi-Fi credentials embedded inside with dynamic provisioning. An example of dynamic provision is WiFiManager or some other equivalent secure configuration procedure. The ESP8266 WiFiClientSecure library is used to compare the HTTPS server certificate against a trusted anchor, instead of employing an insecure client mode. Each POST must be authenticated, for instance, via a token which is checked with a controlled intermediary or through the authenticated solicitation workflow of Apps Script. The custom Authorization header cannot be assumed to work in the same manner with each Google Apps Script web app deployment. Secret keys need to be stored in a secure manner, and be rotated, and spreadsheet permissions kept limited. These modes are proposed and not tested in real life.

Since the sensing interface is linked to mains voltage, any practical design must be supported by a revised electrical schematic as well as effective line-side overcurrent protection, touch-safe terminal barriers, insulated wiring, strain relief, approved isolated low-voltage supply, and suitable enclosure materials. Fuse configuration, MOV protection, and any additional galvanic isolation have to be selected and tested according to the actual module topology and standards. There should be no claim that the optocoupler provides mains isolation unless the complete isolation barrier, distance, and characteristics have been designed and tested.

5.5 Scalability and data management

The shown limit of 10 rows in the spreadsheet illustrates archiving using automation, but it is not a plausible option for consistent registering of information. If 10 records are created every 30 seconds, it accounts for around 5 minutes of information gathering. Running copy-and-clear operations on a spreadsheet may hinder current typing and make audits more difficult. A possible way of production could be adding records to the unchanged raw data spreadsheet, storing it by date or month and getting CSV files out of finished folders, not deleting any records. A system based on locks or single-writer queues is required for protection from double or missing records.

6. Conclusion

This research shows an affordable IoT energy-monitoring system that uses a PZEM-004T-100A, ESP8266 NodeMCU, local interface, Google Scripts, and Google Sheets to perform its operations. Besides, this paper describes RM85 core electronic units, which can be successfully used for educational purposes and for the development of early prototypes. Presenting the results of the research, the authors provide sufficient evidence that supports their conclusions about the sensor acquisition, cloud data processing, mobile applications use, and sheet storage. The cost audit indicates the way improving the system by changing the method of cumulative costs calculation, which means that the aggregate costs should always be shown from the last energy snapshot rather than as a sum of costs from all previous snapshots. The problem is proposed, but it still needs independent verification. Implementing the suggested course of actions in the future research allows getting from the functional idea to the practical implementation of the power-monitoring system.

The software audit found that summing cost values derived from cumulative energy produces repeated counting. The total must instead use the latest cumulative cost or the sum of interval-energy costs. The available evidence does not support billing-grade accuracy, quantified communication reliability, cybersecurity readiness, or certified mains safety. Completing calibrated reference tests, 24 h sequence-based communication trials, secure transport and authentication, corrected billing aggregation, and deployment-grade electrical construction would provide the basis for a future field-validated system.

Acknowledgements

The author would like to thank the Department of Electrical Engineering, Politeknik Kuching Sarawak for providing opportunities and facilities to support this project.

Declarations

Author contributions. Louis Chai Chang Luk: conceptualization, hardware development, software, investigation, and original report preparation. Jamaah binti Suud: supervision, methodology guidance, project administration, and manuscript review.

Conflicts of interest

We declare no conflict regarding the publication of the study.

References

[1] H. O. Garcés, J. Godoy, G. Riffo, N. F. Sepúlveda, E. Espinosa, and M. A. Ahmed, "Development of an IoT-enabled smart electricity meter for real-time energy monitoring and efficiency," *Electronics*, vol. 14, no. 6, art. 1173, 2025. https://doi.org/10.3390/electronics14061173

[2] H. R. Khan, M. Kazmi, Lubaba, M. H. B. Khalid, U. Alam, K. Arshad, K. Assaleh, and S. A. Qazi, "A low-cost energy monitoring system with universal compatibility and real-time visualization for enhanced accessibility and power savings," *Sustainability*, vol. 16, no. 10, art. 4137, 2024. https://doi.org/10.3390/su16104137

[3] H. J. El-Khozondar et al., "A smart energy monitoring system using ESP32 microcontroller," *e-Prime - Advances in Electrical Engineering, Electronics and Energy*, vol. 9, art. 100666, 2024. https://doi.org/10.1016/j.prime.2024.100666

[4] A. A. Mirani, A. Awasthi, N. O'Mahony, and J. Walsh, "Industrial IoT-based energy monitoring system: Using data processing at edge," *IoT*, vol. 5, no. 4, pp. 608-633, 2024. https://doi.org/10.3390/iot5040027

[5] O. Muñoz et al., "Development of an IoT smart energy meter with power quality features for a smart grid architecture," *Sustainable Computing: Informatics and Systems*, vol. 43, art. 100990, 2024. https://doi.org/10.1016/j.suscom.2024.100990

[6] M. U. Saleem, M. Shakir, M. R. Usman, M. H. T. Bajwa, N. Shabbir, P. Shams Ghahfarokhi, and K. Daniel, "Integrating smart energy management system with Internet of Things and cloud computing for efficient demand side management in smart grids," *Energies*, vol. 16, no. 12, art. 4835, 2023. https://doi.org/10.3390/en16124835

[7] M. U. Saleem, M. R. Usman, M. A. Usman, and C. Politis, "Design, deployment and performance evaluation of an IoT based smart energy management system for demand side management in smart grid," *IEEE Access*, vol. 10, pp. 15261-15278, 2022. https://doi.org/10.1109/ACCESS.2022.3147484

[8] T. Karthick, S. C. Raja, J. D. Nesamalar, and K. Chandrasekaran, "Design of IoT based smart compact energy meter for monitoring and controlling the usage of energy and power quality issues with demand side management for a commercial building," *Sustainable Energy, Grids and Networks*, vol. 26, art. 100454, 2021. https://doi.org/10.1016/j.segan.2021.100454

[9] S. Muralidhara, N. Hegde, and P. M. Rekha, "An internet of things-based smart energy meter for monitoring device-level consumption of energy," *Computers & Electrical Engineering*, vol. 87, art. 106772, 2020. https://doi.org/10.1016/j.compeleceng.2020.106772

[10] M. D. Mudaliar and N. Sivakumar, "IoT based real time energy monitoring system using Raspberry Pi," *Internet of Things*, vol. 12, art. 100292, 2020. https://doi.org/10.1016/j.iot.2020.100292

[11] S. Sayed, T. Hussain, A. Gastli, and M. Benammar, "Design and realization of an open-source and modular smart meter," *Energy Science & Engineering*, vol. 7, no. 4, pp. 1405-1422, 2019. https://doi.org/10.1002/ese3.361

[12] Peacefair, PZEM-004T V3.0 AC Communication Module User Manual, 2019.

[13] Espressif Systems, ESP8266EX Datasheet, version 7.1. https://documentation.espressif.com/0a-esp8266ex_datasheet_en.pdf

[14] Sarawak Energy, "Electricity tariffs: Tariff D - Domestic," accessed 4 August 2026. https://www.sarawakenergy.com/customers/tariffs

[15] R. P. Díaz Redondo, A. Fernández-Vilas, and G. Fernández dos Reis, "Security aspects in smart meters: Analysis and prevention," *Sensors*, vol. 20, no. 14, art. 3977, 2020. https://doi.org/10.3390/s20143977