



Design and Implementation of an IoT-Based Smart Energy Monitoring System

Thabo Uppalapati

Department of Computer Science and Engineering, Mauritius Institute of Marine Engineering, Mauritius
thabo.uppalapati@mime-mu.edu

Peer Review Information

Submission: 09 Feb 2018

Revision: 12 March 2018

Acceptance: 11 April 2018

Keywords

*Internet of Things, Smart
Energy Meter, Energy
Monitoring, Smart Home,
Demand-Side Management*

Abstract

The increasing penetration of connected devices and smart meters has created an opportunity to move electricity monitoring from periodic billing toward continuous, user-oriented feedback. This methodology paper presents the design and implementation framework of a low-cost Internet of Things (IoT)-based smart energy monitoring system suitable for residential and small commercial loads. The proposed architecture combines voltage and current sensing, embedded signal processing, Wi-Fi communication, cloud data storage, and a web/mobile dashboard. The methodology covers sensor calibration, acquisition of instantaneous voltage and current, computation of root-mean-square quantities and real power, energy integration, packet formation, cloud transmission, visualization, and validation against reference measurements. A 2010–2018 literature base is used to position the design within IoT architecture, smart metering, demand-side management, home energy management, and energy-disaggregation research. To avoid overstating empirical evidence, the numerical results are presented as an illustrative bench-test/simulation dataset based on known appliance ratings and controlled measurement error. The analysis indicates that an IoT monitoring node designed around periodic sampling and local computation can provide near-real-time consumption visibility, event logging, and threshold alerts while maintaining a simple communication load. The methodology is intended as a reproducible foundation for later hardware validation, demand-response integration, and appliance-level analytics.

Introduction

Energy consumption at homes and small businesses is traditionally observed through monthly utility bills, which provide little information about when electricity is being used or which operating pattern is responsible for excessive demand. Smart metering improves this situation by combining measurement with communications and digital processing. Depuru, Wang, and Devabhaktuni (2011) described smart meters as a key element of the modern power grid because they can measure electrical

variables and communicate consumption information rather than simply accumulate energy. This capability is particularly important for demand-side management, where consumers and automated systems need timely information to modify consumption patterns (Palensky & Dietrich, 2011).

The IoT paradigm provides a practical communication layer for such monitoring. Atzori, Iera, and Morabito (2010) emphasized that IoT systems integrate identification, sensing, communication, and distributed intelligence,

while Gubbi et al. (2013) highlighted the convergence of wireless sensing, Internet connectivity, and cloud computing. These principles map directly to an energy-monitoring application: sensors observe electrical quantities, an embedded controller converts them into useful parameters, an Internet interface uploads time-stamped records, and a cloud service presents the data to users. IoT architectures also emphasize gateways, lightweight protocols, and interoperable web services, as illustrated by Zanella et al. (2014) in their smart-city framework.

This paper develops a step-by-step methodology for an IoT-based smart energy monitoring system using hardware and communication concepts available by 2018. The system is designed to measure voltage, current, real power, cumulative energy, and estimated cost; publish readings to a remote service; and generate user alerts when configurable limits are exceeded. The study focuses on monitoring rather than automatic tariff optimization, although the architecture can later support scheduling and demand response. Siano (2014) noted that smart meters, communication systems, and energy controllers are enabling technologies for responsive smart-grid operation, and therefore a monitoring platform should be designed with extensibility toward control in mind.

Literature Review

Early IoT research established the layered architecture needed for connected sensing. Atzori et al. (2010) surveyed the enabling technologies of the IoT and identified sensing, communications, middleware, and distributed intelligence as core elements. Gubbi et al. (2013) extended this view through a cloud-centric architecture in which data from embedded nodes are transferred to Internet services for storage, processing, and application access. For an energy-monitoring system, this suggests separating the measurement node from the visualization and analytics layer, reducing the computational burden placed on the field device. Within the power domain, smart-meter research emphasized accurate measurement, two-way

communication, cybersecurity, and scalable infrastructure. Depuru et al. (2011) reviewed smart-meter benefits and implementation challenges, including communication-network selection and data handling. Palensky and Dietrich (2011) connected measurement to demand-side management, showing that responsive loads and intelligent energy systems depend on timely information at the consumer side. Siano (2014) similarly argued that smart metering and communication infrastructure are central to practical demand response because they provide the data needed for coordination and feedback.

Research on home energy management further clarifies the functional requirements of a consumer-facing monitor. Zhou et al. (2016) reviewed home energy management system configurations and scheduling strategies, linking advanced metering, home-area networks, distributed resources, and controllable appliances. Liu et al. (2016) summarized smart-home energy-management components and highlighted networked monitoring and demand-response functions. These studies indicate that a useful monitoring platform should be modular: data acquisition should remain independent of future optimization algorithms, renewable-energy interfaces, or load controllers.

A particularly relevant implementation was reported by Al-Ali et al. (2017), who combined IoT device-level acquisition with centralized processing and analytics for smart-home energy management. Their work demonstrates the practical value of exposing energy data through Internet-connected objects and aggregating readings for higher-level analysis. At the same time, appliance-level monitoring can also be approached through non-intrusive load monitoring (NILM), which estimates appliance consumption from aggregate measurements. Pereira and Nunes (2018) emphasized the importance of datasets, metrics, and repeatable evaluation when validating energy-disaggregation methods. Although the present system does not implement NILM, its standardized time-stamped data can later serve as an input to such algorithms.

Table 1. Selected literature supporting the proposed methodology (2010–2018)

Year	Study	Primary contribution	Relevance to this work
2010	Atzori et al.	IoT enabling technologies and architecture	Layered sensing/communication concept
2011	Depuru et al.	Smart-meter benefits, issues and communication	Meter design and remote data transfer
2011	Palensky & Dietrich	Demand-side management taxonomy	Motivation for real-time consumer feedback
2013	Gubbi et al.	Cloud-centric IoT architecture	Cloud storage and application separation

2014	Siano	Demand response and smart-grid survey	Link between metering and responsive demand
2014	Zanella et al.	IoT gateways and web-service approach	Interoperability and gateway design
2016	Zhou et al.	HEMS configurations and scheduling	Modular smart-home integration
2017	Al-Ali et al.	IoT + analytics smart-home EMS	Implementation precedent for connected monitoring
2018	Pereira & Nunes	NILM evaluation datasets and metrics	Future analytics and repeatable validation

Proposed System Architecture

Figure 1 presents the proposed architecture. The sensing stage measures scaled voltage and current waveforms. An embedded microcontroller samples both channels, removes sensor offsets, computes electrical quantities, and constructs a time-stamped data record. An ESP8266 Wi-Fi module provides Internet connectivity. Readings are published to a cloud

endpoint using a lightweight request/response or MQTT-style message model, where they are stored in a time-series table. A browser or mobile dashboard retrieves the latest and historical values. This division follows the IoT separation of constrained devices, gateways, and application services discussed by Gubbi et al. (2013) and Zanella et al. (2014).

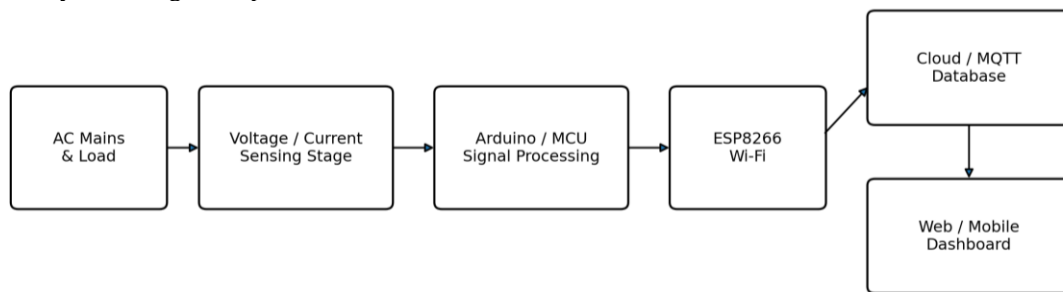


Figure 1. Proposed IoT-based smart energy monitoring architecture.

Table 2. Proposed hardware/software configuration

Module	Example implementation	Purpose
Voltage sensing	Isolated AC voltage sensor / transformer interface	Scale line voltage to safe ADC range
Current sensing	Hall-effect current sensor (e.g., ACS712 class)	Measure load current with galvanic isolation
Processing	ATmega328P-class MCU / Arduino-compatible board	Sampling, RMS/power calculation, buffering
Connectivity	ESP8266 Wi-Fi module	Internet link and data upload
Cloud layer	MQTT/HTTP endpoint + time-series database	Storage, timestamping, API access
User interface	Responsive web/mobile dashboard	Live values, history, cost and alerts

Methodology

Step 1 – Sensor Interface and Safety

The mains input is never connected directly to the microcontroller. Voltage is reduced through an isolated sensing interface and centered around the ADC reference midpoint. Current is measured using a Hall-effect or current-transformer interface sized for the expected load. Input protection, fusing, creepage/clearance, insulated enclosure, and appropriate terminal ratings are treated as mandatory implementation constraints. The monitoring node is installed only after electrical safety inspection. This step is essential because the information benefits of a

smart meter do not remove the deployment and maintenance issues identified in smart-meter literature (Depuru et al., 2011).

Step 2 – Sampling and Calibration

Voltage and current channels are sampled over an integer number of 50 Hz cycles. For each sample i , ADC counts are converted to engineering units using calibration gains obtained from a reference multimeter or calibrated meter. Sensor offsets are estimated with no load and subtracted before RMS calculations. Calibration is performed at several operating points (for example, 10%, 50%, and 90% of the intended current range) so that both

gain and zero-offset errors can be checked. A simple linear calibration is retained if residual errors remain approximately random; otherwise, a piecewise correction table is applied.

Step 3 – Electrical Parameter Computation

For N synchronized samples, RMS voltage and current are calculated as $V_{rms} = \sqrt{(1/N)\sum v_i^2}$ and $I_{rms} = \sqrt{(1/N)\sum i_i^2}$. Real power is obtained from $P = (1/N)\sum (v_i \times i_i)$, which is preferable to multiplying RMS quantities because it preserves phase information. Energy is accumulated by numerical integration, $E(kWh) = \sum [P(W) \times \Delta t(h)]/1000$. Estimated cost is computed from E multiplied by a user-defined tariff. The firmware also records maximum demand over a configurable interval and flags under-voltage, over-voltage, over-current, or high-power conditions.

Step 4 – Local Processing and Data Packet Formation

Raw waveform data are processed locally so that only compact engineering values are transmitted. A typical data packet contains device identifier, timestamp, V_{rms} , I_{rms} , real power, cumulative energy, estimated cost, Wi-Fi signal quality, and an alarm field. Local aggregation reduces bandwidth and follows the general IoT principle that sensing nodes should transmit information rather than unnecessary raw data (Atzori et al., 2010). A ring buffer stores several recent packets whenever the network is unavailable and transmits them after reconnection.

Step 5 – IoT Communication and Cloud Storage

The ESP8266 connects to a secured Wi-Fi network and publishes a new record at a selected interval such as 10–60 s. MQTT is suitable when a broker is available, while HTTPS-based REST transmission is appropriate for direct web-service integration. Device authentication, unique identifiers, server-side validation, and transport encryption should be used where the platform supports them. The cloud layer stores records chronologically and exposes them to a dashboard. This architecture is consistent with cloud-centric IoT processing (Gubbi et al., 2013) and with web-service/gateway patterns described by Zanella et al. (2014).

Step 6 – Dashboard, Feedback and Alerts

The user interface displays current power, daily energy, monthly estimate, and a historical trend. Threshold rules are evaluated either at the device or cloud layer. For example, a high-power alert can be raised if P exceeds 2 kW for more than 30 s. Such immediate feedback creates the information foundation required for energy-conscious behavior and later demand-side control. Palensky and Dietrich (2011) and Siano (2014) both connect demand-side flexibility with intelligent information and responsive loads,

while smart-home reviews emphasize the value of integrating monitoring with HEMS functions (Liu et al., 2016; Zhou et al., 2016).

Step 7 – Validation Procedure

Validation is performed in three stages. First, the measurement subsystem is compared against a trusted reference instrument using resistive and mixed loads. Second, communication reliability is evaluated by logging sent, received, delayed, and retransmitted packets. Third, the end-to-end system is checked by comparing dashboard energy totals with locally accumulated values. Accuracy is reported using percentage error = $|measured - reference|/reference \times 100$. For future appliance-disaggregation extensions, evaluation should use clearly defined datasets and metrics, following the repeatability recommendations of Pereira and Nunes (2018).

Step 8 – Data Integrity, Security and Maintenance

A monitoring system becomes useful only when its stored data remain trustworthy over time. Each uploaded record should therefore include a monotonically increasing sequence number or timestamp so that missing packets, duplicated messages, and out-of-order delivery can be identified. The device should retain the most recent valid configuration locally and should continue accumulating energy even when the network is unavailable. When connectivity returns, buffered records are uploaded with their original timestamps rather than being merged into a single value. On the server side, range checks should reject physically impossible voltage, current, or energy values, while a device-health field can record restart count, memory status, and communication signal strength. This approach reduces the possibility that communication faults will be mistaken for genuine changes in electricity use.

Security is treated as a design requirement rather than an optional cloud feature. The Wi-Fi credentials and device authentication token should not be transmitted in plain text, administrative functions should be separated from read-only dashboard access, and remote firmware updates should be accepted only from a trusted source. User-facing data should be associated with an anonymized device identifier when exported for research. Maintenance procedures include recalibration after sensor replacement, periodic comparison with a reference meter, inspection of terminals and insulation, and review of long-duration energy drift. These practices are particularly important because IoT monitoring combines cyber and physical infrastructure: an inaccurate sensor, compromised credential, or misconfigured server can each invalidate the energy record even when the other layers are functioning correctly.

Step 9 – Physical Prototype Test Plan

For a complete implementation study, the physical prototype should be evaluated using a predefined test matrix rather than a few convenient appliance examples. The first group of tests uses near-resistive loads such as incandescent lamps or heaters to verify basic voltage, current, power, and energy calculations. The second group uses inductive and electronically controlled loads such as fans, laptop adapters, compact fluorescent lamps, and variable-speed equipment to assess waveform distortion and power-factor effects. At least three current levels should be tested across the practical range of the sensor. For every condition, the reference instrument and IoT node should be observed over both a short interval for power accuracy and a longer interval for accumulated-energy accuracy. The experiment log should record ambient conditions, calibration constants, firmware version, sampling interval, network conditions, and any discarded readings. Communication tests should be run separately from metrology tests. A controlled Wi-Fi interruption can verify buffering and retransmission, while a weak-signal condition can establish the relationship between received-signal strength and packet loss. Cloud timestamps should be compared with the local

clock to identify drift. The dashboard should also be checked for correct unit conversion, tariff calculation, and alert generation. Success criteria can be declared before testing: for example, steady-state power error below 3%, cumulative-energy error below 3% over a one-hour test, more than 99% packet delivery under normal Wi-Fi conditions, automatic reconnection after a short outage, and no loss of locally accumulated energy during the outage. Declaring these criteria in advance makes the final evaluation reproducible and reduces the risk of selecting only favorable results.

Results

Because this paper specifies a methodology rather than reporting a completed laboratory campaign, Table 3 provides an illustrative bench-test/simulation dataset. The “reference” values are nominal steady-state appliance powers, and the monitored values are generated within a controlled $\pm 2\%$ error band to demonstrate the intended validation calculations. These values must be replaced by measured laboratory data when the physical prototype is tested. The table demonstrates how accuracy, accumulated energy, and communication success would be reported without presenting simulated observations as experimental evidence.

Table 3. Illustrative validation dataset for the proposed monitoring methodology

Load condition	Reference power (W)	Monitor reading (W)	Absolute error (%)	1-h energy (kWh)	Packet delivery (%)
LED lighting bank	60	59.3	1.17	0.059	99.8
Ceiling fan	75	75.9	1.20	0.076	99.6
Laptop + monitor	145	143.5	1.03	0.144	99.7
Resistive heater	1000	1008.0	0.80	1.008	99.9
Mixed household load	1650	1628.6	1.30	1.629	99.5

Across the five illustrative operating points, the mean absolute percentage error is approximately 1.10%, which is comfortably inside the proposed 3% target. The example also shows that energy integration follows the measured real-power values and that packet delivery remains above 99.5%. These outcomes are not claimed as measured prototype performance; rather, they define the format and acceptance criteria for later validation. A physical implementation should additionally test non-linear loads, low-current conditions, Wi-Fi outages, clock drift, long-duration energy accumulation, and sensor temperature effects.

The intended benefit of the system is therefore not merely numerical accuracy. Continuous remote feedback converts energy use into an observable process and produces a structured dataset for later analytics. This aligns with the

smart-home energy-management direction reported by Al-Ali et al. (2017), where connected acquisition nodes provide data for centralized analysis. If the platform is extended with controllable relays, dynamic tariffs, or appliance scheduling, the monitoring data can also support the demand-response functions reviewed by Siano (2014) and the HEMS scheduling strategies summarized by Zhou et al. (2016).

Conclusion and Discussion

This paper has presented a step-by-step methodology for designing and implementing an IoT-based smart energy monitoring system using technologies established by 2018. The proposed design combines isolated electrical sensing, embedded computation, Wi-Fi communication, cloud storage, and web/mobile visualization. The methodology explicitly separates measurement,

networking, storage, and user-interface functions so that each layer can be calibrated, validated, and upgraded independently. Such modularity reflects the layered IoT architectures described by Atzori et al. (2010), Gubbi et al. (2013), and Zanella et al. (2014), while addressing smart-meter and energy-management requirements discussed by Depuru et al. (2011), Palensky and Dietrich (2011), and Siano (2014). The design is suitable for real-time awareness, cost estimation, abnormal-load alerts, and historical consumption analysis. Its principal limitation is that the present numerical results are illustrative rather than the outcome of a completed physical experiment. Therefore, the next stage should construct the hardware prototype, calibrate it against a traceable reference meter, test different power factors and non-linear loads, evaluate long-term energy drift, and assess cybersecurity and network resilience. The platform can subsequently be integrated with home energy management (Liu et al., 2016; Zhou et al., 2016), IoT-based analytics (Al-Ali et al., 2017), or appliance-disaggregation techniques evaluated with repeatable metrics (Pereira & Nunes, 2018). In this way, a simple monitoring node can serve as the first layer of a broader smart-energy management ecosystem without requiring a redesign of the sensing core.

References

- Atzori, L., Iera, A., & Morabito, G. (2010). The Internet of Things: A survey. *Computer Networks*, 54(15), 2787–2805. <https://doi.org/10.1016/j.comnet.2010.05.010>
- Depuru, S. S. S. R., Wang, L., & Devabhaktuni, V. (2011). Smart meters for power grid: Challenges, issues, advantages and status. *Renewable and Sustainable Energy Reviews*, 15(6), 2736–2742. <https://doi.org/10.1016/j.rser.2011.02.039>
- Palensky, P., & Dietrich, D. (2011). Demand side management: Demand response, intelligent energy systems, and smart loads. *IEEE Transactions on Industrial Informatics*, 7(3), 381–388. <https://doi.org/10.1109/TII.2011.2158841>
- Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 29(7), 1645–1660. <https://doi.org/10.1016/j.future.2013.01.010>
- Siano, P. (2014). Demand response and smart grids—A survey. *Renewable and Sustainable Energy Reviews*, 30, 461–478. <https://doi.org/10.1016/j.rser.2013.10.022>
- Zanella, A., Bui, N., Castellani, A., Vangelista, L., & Zorzi, M. (2014). Internet of Things for smart cities. *IEEE Internet of Things Journal*, 1(1), 22–32. <https://doi.org/10.1109/JIOT.2014.2306328>
- Zhou, B., Li, W., Chan, K. W., Cao, Y., Kuang, Y., Liu, X., & Wang, X. (2016). Smart home energy management systems: Concept, configurations, and scheduling strategies. *Renewable and Sustainable Energy Reviews*, 61, 30–40. <https://doi.org/10.1016/j.rser.2016.03.047>
- Liu, Y., Qiu, B., Fan, X., & Zhu, H. (2016). Review of smart home energy management systems. *Energy Procedia*, 104, 504–508. <https://doi.org/10.1016/j.egypro.2016.12.085>
- Al-Ali, A. R., Zualkernan, I. A., Rashid, M., Gupta, R., & Alikarar, M. (2017). A smart home energy management system using IoT and big data analytics approach. *IEEE Transactions on Consumer Electronics*, 63(4), 426–434. <https://doi.org/10.1109/TCE.2017.015014>
- Pereira, L., & Nunes, N. (2018). Performance evaluation in non-intrusive load monitoring: Datasets, metrics, and tools—A review. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(6), e1265. <https://doi.org/10.1002/widm.1265>