IJIRSET
International Journal of Innovative Research in Science | Engineering | Technology

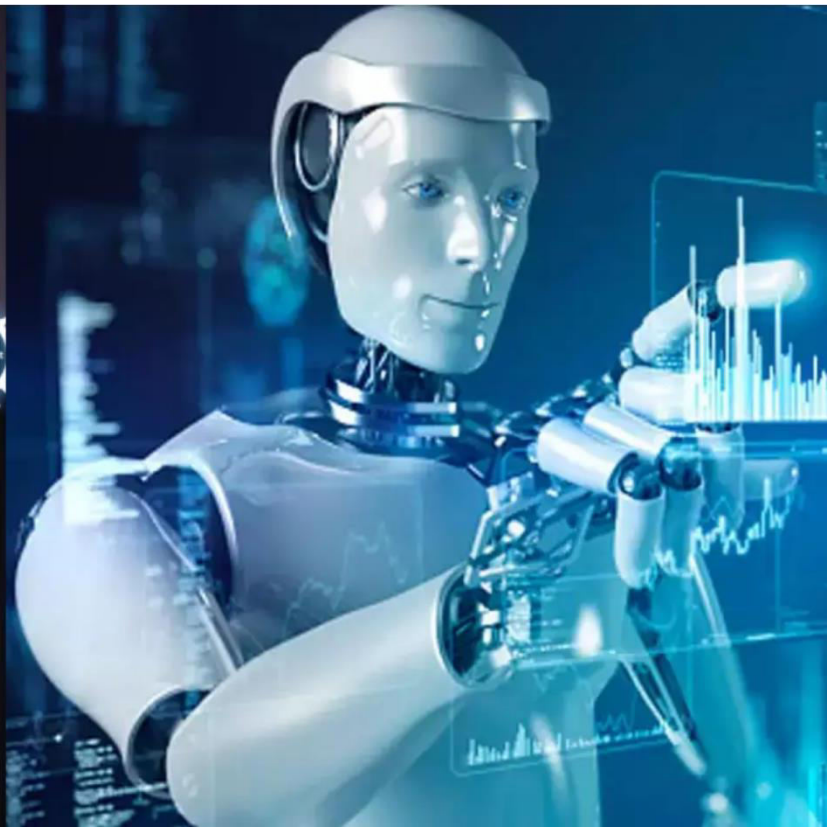
ISSN(O): 2319-8753
ISSN(P): 2347-6710



International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)





Impact Factor: 9.935

Volume 15, Issue 11, November 2026

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 11, November 2026

|DOI: 10.15680/IJIRSET.2026.1511037|

Design and Evaluation Framework for An IoT-Based Smart Building Integrated with A Smart Grid using Bidirectional MQTT Communication

Tamil Selvan C, Kaviyamuthan K.R., M. Viji

Department of Computer Science / Information Technology, E.G.S. Pillay Engineering College, Nagapattinam, Tamil Nadu, India

Supervisor, Department of Computer Science / Information Technology, E.G.S. Pillay Engineering College, Nagapattinam, Tamil Nadu, India

ABSTRACT: Smart buildings can support demand-side flexibility when local monitoring and control systems exchange information with an external energy-management service. This paper proposes a low-cost Internet of Things (IoT) testbed for bidirectional communication between a building prototype and a software-based smart-grid simulator. The proposed building node uses an ESP32 microcontroller with environmental, occupancy and low-voltage direct-current (DC) sensing. Message Queuing Telemetry Transport (MQTT) carries building telemetry to a Python backend and grid-side tariff or demand-response messages towards the building controller. The backend validates requests, applies a rule-based load-management policy, records timestamped events in SQLite and supplies data to a web dashboard. The simulator represents grid operating states, time-varying tariffs and peak-demand requests. A planned evaluation protocol defines message-delivery ratio, end-to-end command latency, command execution acknowledgement, communication recovery and DC-load energy as measurable outcomes. The prototype has not yet been implemented; therefore, this manuscript reports a proposed architecture and evaluation methodology only. No measured performance, energy reduction or field deployment is claimed.

KEYWORDS: Internet of Things, smart building, smart grid, MQTT, bidirectional communication, demand response, energy management.

I. INTRODUCTION

Buildings contain electrical loads with different operating priorities and time patterns. Lighting, ventilation, appliances and other loads may be monitored and, where appropriate, controlled to support efficient operation. At grid level, demand varies over time and demand-response programmes may encourage changes in consumption during constrained periods. A building that can receive external signals and respond through a controlled local system can act as a flexible demand-side participant. This interaction requires more than a local sensor dashboard: it requires a defined information path between the building and an external energy-management service.

The Internet of Things (IoT) provides a framework for connecting sensors, embedded controllers and software applications. Low-cost microcontrollers can collect environmental and occupancy measurements, while network protocols deliver data to local or remote services.

However, network connectivity alone does not create a dependable smart-grid interface. A complete system must define message formats, topic permissions, command validation, acknowledgement, failure handling and event logging. These functions matter because an external request should not directly trigger an unsafe or invalid device action.

MQTT is a lightweight publish/subscribe messaging protocol in which clients exchange messages through a broker. This model separates producers from consumers and reduces the need for direct point-to-point connections among embedded devices, backend services and simulators.

In the proposed testbed, the building node publishes telemetry, while the grid simulator publishes operating events or tariff changes. A backend mediates commands and records outcomes.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 11, November 2026

|DOI: 10.15680/IJIRSET.2026.1511037|

Prior studies review IoT applications and barriers in smart-building energy management [1], [2]. Other work considers trustworthy IoT-device participation in demand response [3], demand-response optimization for smart-grid-integrated buildings [4], communication-enabled demand-response demonstrations [5], and IoT energy management in a grid-connected photovoltaic-powered DC residential building [6]. This work proposes a compact educational testbed combining these concerns in a traceable workflow.

The objectives are to specify a low-cost building node, define bidirectional MQTT communication, describe a validated rule-based control path, store timestamped events and establish a reproducible evaluation protocol. The proposed contribution is the request-to-acknowledgement workflow and its planned measurement method, not the isolated use of ESP32 or MQTT. The system does not connect to a utility grid.

II. RELATED WORK

IoT-based building energy management has been studied through sensing, monitoring, automation and consumption analysis. Poyyamozhi et al.

[1] review IoT applications, barriers and future directions in smart-building energy management. Their review highlights connected sensing and data-informed operation. Cavalcanti and Pimenta [2] review electricity management in buildings based on IoT and discuss factors affecting energy use and management. These reviews establish the role of building data while indicating that deployment and integration challenges remain relevant.

A related direction concerns participation of individual IoT devices in demand-response programmes. Teixeira et al. [3] examine a trustworthy building energy-management system designed to enable direct IoT-device participation in demand response, with attention to trust and privacy. Toderean et al. [4] review demand-response optimization for smart-grid-integrated buildings and discuss technology enablers and interoperability challenges. These studies indicate that demand response requires coordination among information, control policy and device-level execution.

Communication infrastructure has also been examined in energy-system testbeds. Jahic et al. [5] report a hardware-in-the-loop automated demand-response demonstration using measurement devices, simulated and real IP communications, cloud computing and MQTT-related IoT protocols. Chinnathambi et al. [6] describe an IoT-based energy-management system for a grid-connected photovoltaic-powered DC residential building. These works provide relevant precedents, although their scale and electrical configuration differ from the proposed single-zone educational prototype.

Table I summarizes the relevance of the cited studies. The comparison is limited to the themes represented by the cited publications; it does not imply that a cited system lacks features not described here. A final submission should verify the full texts and extend the comparison with additional recent primary studies.

TABLE I. SUMMARY OF RELATED STUDIES

Ref.	Reported focus	Relevance to proposed work
[1]	Review of IoT in smart-building energy management	Sensing, management barriers and future directions.
[2]	Review of IoT-based electricity management in buildings	Building energy monitoring and management context.
[3]	Trustworthy IoT-device participation in demand response	Motivates command trust, privacy and device participation.
[4]	Demand-response optimization in smart-grid buildings	Grid-interactive buildings and interoperability.
[5]	Hardware-in-the-loop automated demand-response demonstration	Supports testbed validation of communication and control.
[6]	IoT energy management in a grid-connected PV-powered DC building	Related DC energy-management context.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 11, November 2026

|DOI: 10.15680/IJIRSET.2026.1511037|

A. Research gap and scope

The literature covers building energy monitoring, trustworthy device participation, grid-interactive demand response and communication-enabled testbeds. The present project addresses a narrower engineering objective: combining a physical single-zone building node, a software grid simulator and a traceable request-to-acknowledgement workflow in a low-cost prototype. The system is intended to record whether a grid-side event is received, validated, translated into a permitted local command, executed and acknowledged. This makes it possible to evaluate communication behaviour and the response of a small DC demonstration load.

The contribution should be understood as an integration and reproducibility contribution. It is not a claim to be the first MQTT-based smart-building system or the first demand-response testbed. Its value will depend on implementation, demonstration of the communication lifecycle and transparent reporting of measurements. The research-gap statement should be updated after a broader literature search and supervisor review.

B. Research questions

RQ1: Can the building node publish timestamped sensor telemetry to the backend over MQTT? RQ2: Can a simulated grid-side request be validated, translated into a permitted command and acknowledged by the device? RQ3: How do communication interruptions affect message delivery, command completion and recovery? RQ4: What change in energy use is observed for the selected low-voltage DC load under the proposed policy compared with a matched baseline? These are questions for future testing; this design manuscript does not claim answers.

III. METHODOLOGY

A. Problem statement and system requirements

A small building prototype may collect sensor data and display it locally, while a grid simulator independently generates tariff changes or peak-demand events. If the components do not share a defined communication and control loop, they do not demonstrate bidirectional building-grid interaction. The engineering problem is how to connect low-cost physical monitoring to a simulated grid-side management service and record whether an external request is received, validated, executed and acknowledged without bypassing local safety limits.

The system is intended for a single-zone educational prototype. It does not connect to a utility control centre, participate in electricity markets or operate real building mains loads. The grid side is software simulated so that events can be repeated under controlled conditions. The physical side uses only a low-voltage DC demonstration load. This separation reduces cost and permits the communication workflow to be tested without claiming grid-scale electrical validation.

B. Functional requirements

FR1 — Sensor acquisition: the node periodically acquires environmental and occupancy-related readings. FR2 — Telemetry publication: readings are published with device identifier, timestamp, measurement names and units. FR3 — Grid-event reception: the backend receives tariff and demand-response events. FR4 — Request validation: the backend checks event identity, timestamp, expiry, command type and permitted action range. FR5 — Local actuation: the ESP32 independently validates a command before changing the DC load. FR6 — Acknowledgement: the device publishes command identifier, execution state and timestamp. FR7 — Event storage: telemetry, grid events, commands and acknowledgements are stored. FR8 — Visualization: the dashboard displays measurements, grid state and event history.

C. Non-functional requirements

NFR1 — Safety: the controller enforces local operating limits and a defined safe state. NFR2 — Traceability: each command and acknowledgement share a unique identifier. NFR3 — Recoverability: behaviour after Wi-Fi or broker interruption is defined. NFR4 — Maintainability: firmware, backend and simulator code are modular. NFR5 — Reproducibility: test configuration and software versions are recorded. NFR6 — Security: broker access is restricted, commands are validated and credentials are excluded from source control. NFR7 — Affordability: commonly available components and open-source software are preferred.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 11, November 2026

|DOI: 10.15680/IJIRSET.2026.1511037|

D. Proposed component selection

The proposed controller is an ESP32 development board with Wi-Fi. A DHT22-class module is intended for temperature and humidity sensing, and an HC-SR501-class PIR module is proposed for occupancy or motion detection. An INA219-class module is intended for voltage and current measurement in a suitable low-voltage DC circuit. A small DC fan or lamp represents a controllable load, driven through a compatible transistor/MOSFET driver or appropriately rated low-voltage relay. Exact model numbers, ratings, wiring and datasheets must be recorded after purchase and testing.

The proposed software stack comprises Mosquitto MQTT, Python with Flask, SQLite, a Python grid simulator and a browser-based dashboard using HTML, CSS, JavaScript and Chart.js. These are design choices, not claims of completed implementation. The final manuscript must replace planned versions and components with those actually used.

E. Proposed system architecture and communication design

The architecture contains five logical components: (A) building sensing and actuation, (B) MQTT broker, (C) backend and event database, (D) smart-grid simulator and (E) web dashboard. The ESP32 collects local measurements and controls the DC demonstration load. The broker routes messages among the device, backend and simulator. The backend applies policy and stores events. The simulator generates grid-side events, while the dashboard presents system state.

BUILDING NODE ESP32 + sensors	**MQTT BROKER** Mosquitto	**BACKEND** Python + SQLite	**GRID SIMULATOR** Tariff + DR events	**DASHBOARD** Measurements, tariff, events and acknowledgements

Fig.1. Logical architecture of the proposed building-to-simulator testbed.

F. MQTT topic design

The proposed topic hierarchy is: building/zone1/telemetry for sensor readings; building/zone1/status for device availability; building/zone1/command for control requests; building/zone1/ack for execution acknowledgements; grid/status for simulated grid condition; grid/tariff for tariff updates; and grid/demand_response for demand-response events. Topic permissions should restrict which clients may publish or subscribe. For example, the device may publish telemetry and acknowledgements but should not publish privileged grid events.

G. Message structure

JSON is proposed because it is readable and supported by common embedded and backend libraries. A telemetry message should include device_id, message_id, timestamp, temperature_c, humidity_percent, occupancy_state, voltage_v, current_a and power_w where relevant. A grid event should include event_id, event_type, timestamp, expiry_time, tariff_value or requested_action, and priority. A command should include command_id, target_device, action, value, issue_time and expiry_time. An acknowledgement should repeat command_id and include device_id, execution_status, resulting_state and acknowledgement_time.

An illustrative schema is shown below. The null values are placeholders, not experimental measurements:

{"device_id":"zone1","message_id":"m-0001","timestamp":"ISO-8601","temperature_c":null,"humidity_percent":null,"occupancy_state":"unknown","voltage_v":null,"current_a":null,"power_w":null}

H. Message quality and duplicate handling

The implementation should document the MQTT quality-of-service (QoS) level used for each topic. QoS 0 provides at-most-once delivery, QoS 1 provides at-least-once delivery and may produce duplicates, and QoS 2 provides exactly-once delivery at the MQTT protocol exchange level.

Application-level command identifiers and duplicate rejection remain useful so that retries or reconnection do not repeat a physical action.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 11, November 2026

|DOI: 10.15680/IJIRSET.2026.1511037|

The controller should reject expired commands, malformed payloads, unknown command types and values outside configured limits. Retained messages should not be used for one-time actuation unless an explicit mechanism prevents an old retained command being applied after reconnection. Time synchronization should be sufficient for timestamp and expiry checks; where reliable wall-clock time is unavailable, the device should use a safe fallback policy.

I. Backend and dashboard

The backend subscribes to telemetry and grid events, validates schemas, records events and evaluates the control policy. The database should preserve raw records and derived summaries. The dashboard is intended to display sensor readings, DC voltage/current/power, estimated DC energy, grid status, tariff, pending requests, acknowledgements and communication status. It is a monitoring interface, not the safety mechanism; local limits must remain enforced by the device.

J. Bidirectional control workflow and algorithm

A central feature of the design is separation between a grid-side request, backend policy decision and local device execution. This prevents an external request from bypassing device-level constraints. The backend determines whether a request is relevant and permitted, while the ESP32 performs a second validation before actuation.

Grid event	Backend validates	MQTT command	ESP32 validates	DC load action	ACK + log

Receipt is not execution: only an execution acknowledgement confirms success.

Fig. 2. Proposed demand-response command lifecycle. An acknowledgement reports execution outcome; receipt alone is not success.

K. Building-to-grid data flow

At a configured sampling interval, the ESP32 reads available sensors and forms a telemetry message. It assigns a unique identifier and timestamp before publishing to the telemetry topic. The backend checks required fields, data types, units and device identity. Valid records are stored in SQLite and made available to the dashboard. Invalid records should be marked with a validation status and retained or logged for diagnosis rather than silently treated as valid measurements.

L. Grid-to-building control flow

The grid simulator publishes a tariff update or demand-response event. The backend checks the event identifier, event type, issue time, expiry time and requested action. It evaluates local policy conditions, including whether the load is noncritical, whether occupancy rules apply, whether a manual override is active and whether a command-rate limit has been reached. If rejected, the reason is recorded. If accepted, the backend publishes a command with a unique identifier and expiry time.

The ESP32 receives the command and validates its schema, identifier, expiry and action range. It checks local operating conditions and safety limits. If valid, it changes the DC load and publishes an acknowledgement. If invalid or disallowed, it publishes a rejected acknowledgement with a reason code. The backend correlates the acknowledgement with the request and stores the event chain. A command counts as successful only when the acknowledgement reports execution and the resulting device state is consistent with the command.

M. Rule-based demand-response algorithm

The first prototype uses a rule-based policy because its decisions are inspectable and reproducible. The procedure is: (1) receive a grid event; (2) validate schema and expiry; (3) check whether the event identifier was processed previously; (4) determine whether the requested action is permitted; (5) check manual override, occupancy, load priority and command-rate limit; (6) issue a command with unique identifier and expiry if all checks pass; (7) await device acknowledgement; (8) record executed, rejected, expired or timed-out status; and (9) update the dashboard. Any failed validation causes rejection and a recorded reason.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 11, November 2026

|DOI: 10.15680/IJIRSET.2026.1511037|

N. Failure handling

If Wi-Fi or the broker is unavailable, the controller must not assume a command was delivered. It should maintain the predefined local safe state and attempt reconnection under a documented retry policy. After reconnection, stale commands must be rejected using expiry checks. The backend should distinguish messages never received from messages received but not acknowledged. A timeout is an incomplete or unknown outcome, not proof of successful execution.

Unique identifiers support deduplication; timestamps and expiry values support stale-event rejection; persistent storage supports event-history recovery. These mechanisms should be implemented and tested before performance claims are made.

O. Implementation stages, security and safety

O.1 Implementation stages

The implementation is divided into stages. First, procure and document the hardware, then test each sensor independently. Second, program the ESP32 to read sensors and publish telemetry over Wi-Fi. Third, configure the MQTT broker and test subscriptions and publications. Fourth, implement command reception, local validation and acknowledgement with a safe low-voltage load. Fifth, develop the Python backend and SQLite event store. Sixth, develop the grid simulator and rule-based policy. Seventh, connect the dashboard and verify end-to-end event visibility. Finally, conduct repeatable experiments, analyse logs and prepare figures and tables.

A staged approach isolates faults. Sensor readings should be checked before network communication is introduced. MQTT should be tested independently before actuation is enabled. The backend and database should be validated with recorded messages before connecting the physical device to the control workflow. The complete system should then be tested under normal operation and deliberate communication interruption.

O.2 Security considerations

Even a local prototype should apply basic security practices. The MQTT broker should require authentication and topic-level access control.

Where supported, TLS should protect messages in transit. Credentials and private network details should be excluded from source control. The backend should validate JSON fields, reject unexpected command types and enforce permitted value ranges. Device and command identifiers should be checked rather than trusted solely because a message arrived through the broker.

Control messages should include expiry information to reduce replay of stale requests. Duplicate command identifiers should not trigger repeated actuation. The ESP32 should retain local safety constraints and reject malformed or out-of-range commands even if the backend has approved them. The dashboard should distinguish pending, executed, rejected and timed-out commands. The final paper must state which security controls were actually implemented and tested.

O.3 Physical safety and measurement boundary

The INA219 module is intended only for a suitable low-voltage DC measurement circuit and must not be connected directly to household AC mains. The demonstration load should be a low-voltage DC fan or lamp with a driver rated for its voltage and current. The prototype should use a suitable power supply, insulated connections and an appropriate enclosure. No household mains switching is included. Any future AC measurement or switching requires correctly rated isolated equipment and qualified installation.

Energy measurements from this prototype apply only to the selected DC demonstration load under stated test conditions. They cannot be generalized to whole-building electricity consumption or utility-level demand reduction. This limitation must be repeated wherever energy outcomes are discussed.

O.4 Proposed cost and resource planning

The project planning estimate is INR 3,000–7,000. This is a budget target, not a verified bill of materials. Actual prices vary by supplier, model and availability. The final paper should list each purchased part, exact model, rating, quantity, unit cost, source and datasheet reference. The software stack is intended to use open-source or locally hosted tools. No final cost claim should be made until purchase records are available.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 11, November 2026

|DOI: 10.15680/IJIRSET.2026.1511037|

IV. EXPERIMENTAL RESULTS AND EVALUATION PLAN

The proposed evaluation is intended to establish whether the communication and control workflow operates reliably under repeatable conditions. The prototype has not yet been implemented and no experimental dataset is available. The scenarios and metrics below are therefore a future test plan, not results.

A. Experimental scenarios

S1 — Normal telemetry: operate the building node for a defined period without grid events and record transmitted and received telemetry messages. S2 — Demand-response command: publish a peak-demand event and verify backend validation, device command reception, load-state change and acknowledgement. S3 — Tariff update: publish a tariff change without actuation and check consistency among simulator, backend and dashboard. S4 — Occupancy-aware policy: execute the same request under occupied and unoccupied sensor states and verify that the policy follows its rules. S5 — Communication interruption: disconnect Wi-Fi or stop the broker for a controlled interval, then restore communication and record message loss, recovery time, stale-command rejection and safe-state behaviour.

B. Test controls and repeatability

Each scenario should be repeated for a documented number of trials. The same device, load, supply, sampling interval and network environment should be used when comparing baseline and controlled energy. The test report should specify firmware, broker and backend versions, MQTT QoS, Wi-Fi conditions, publish interval, sensor models, reference instrument, load rating and test duration. The order of tests should be recorded, and failed trials should be retained with an explanation rather than silently removed.

C. Performance metrics

Message-delivery ratio = valid messages received ÷ messages transmitted × 100. Report numerator and denominator because a percentage alone can conceal a small sample size. End-to-end command latency is the elapsed time between publication of a grid request and receipt of the matching device acknowledgement. Command execution ratio = valid commands acknowledged as executed ÷ valid commands issued × 100.

Rejected, expired and timed-out commands should be reported separately.

For the DC demonstration load, instantaneous power may be estimated as $P(t) = V(t) \times I(t)$, provided voltage and current are measured with suitable calibration and sampling. Energy over an interval may be estimated by numerical integration: $E \approx \Sigma\ PI\Delta t$, with sampling interval and units reported. Relative energy change = $100 \times (E_{baseline} - E_{controlled})/E_{baseline}$. The comparison is valid only if load, supply, duration and operating conditions are matched. The result applies only to the selected DC demonstration load, not whole-building savings.

D. Planned result presentation

TABLE II. PLANNED RESULT PRESENTATION

Measure	Evidence to report	Presentation
Telemetry delivery	Sent, valid received, lost messages, QoS	Counts and delivery ratio
Command latency	Request and acknowledgement timestamps	Mean/median, spread, plot
Command outcome	Executed, rejected, expired, timed out	Outcome table or chart
DC energy	Voltage/current samples and integrated energy	Matched baseline/control plot
Recovery	Interruption and recovery time; stale/duplicate events	Timeline and summary

E. Data processing and reporting

Raw readings and event logs should be retained separately from processed summaries. Data-cleaning rules should describe treatment of missing timestamps, invalid sensor values, duplicate messages and interrupted runs. Summary statistics should include trial count and a suitable measure of central tendency and variability. If a reference instrument is used, its method and uncertainty should be described. Plots should show units, axes, trial count and test conditions. No numerical result should be inserted until obtained from the actual prototype.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 11, November 2026

|DOI: 10.15680/IJIRSET.2026.1511037|

V. DISCUSSION, LIMITATIONS AND CONCLUSION

A. Expected engineering value

The proposed testbed combines physical monitoring, simulated grid-event generation, backend policy evaluation, device-level command validation and timestamped acknowledgement in one workflow. Its intended value is educational and experimental: it provides a concrete way to investigate how a small connected building node exchanges information with a grid-side service. A simulator permits repeatable tariff and demand-response events without access to a utility network. Restricting actuation to a low-voltage DC load keeps the initial implementation within a safer demonstration boundary.

The event log creates a basis for diagnosing system behaviour. A grid request can be traced through its identifier, backend decision, command publication, device response and acknowledgement. This distinguishes communication failure from policy rejection or device execution failure.

Such traceability matters when reporting command success: a published message is not equivalent to a successful physical action.

B. Limitations

The design is limited to a single-zone prototype and a software-based grid simulator. It does not reproduce distribution-network electrical dynamics, grid protection, utility communication infrastructure, market settlement or regulatory processes. The rule-based policy is selected for transparency and is not claimed to be globally optimal. The proposed sensors and DC measurement module do not provide a validated estimate of household AC energy consumption. Results, when collected, will apply only to the selected components, network configuration, DC load and experimental conditions.

The current manuscript lacks implementation evidence. The hardware has not been assembled, firmware and backend have not been validated, and no experimental dataset is available. Therefore, no measured latency, delivery ratio, command success, energy reduction or comparative performance can be reported. The research gap and novelty statement may need revision after a broader literature search and comparison with implemented systems. These limitations should be resolved or explicitly retained in any submitted version.

C. Future work

After the initial prototype is validated, future work may include additional building zones, renewable-generation or battery-storage simulation, more detailed tariff models, network impairment testing and comparison of rule-based control with optimization-based policies. Calibrated AC energy measurement could be considered only as a separate extension using certified, appropriately rated isolated equipment installed by a qualified person. Any extension should be reported as completed only after implementation and evaluation.

D. Conclusion

This paper has presented a proposed design and evaluation framework for an IoT-based smart-building prototype integrated with a software smart-grid simulator. The architecture combines ESP32-based sensing, low-voltage DC measurement, MQTT bidirectional messaging, a Python backend, SQLite event storage, a rule-based demand-response policy and a web dashboard. The communication design specifies topics, structured messages, validation, command identifiers and acknowledgements. The evaluation plan defines scenarios and metrics for telemetry delivery, command latency, command execution, interruption recovery and DC demonstration-load energy.

The work should be considered a design-and-methodology contribution until the prototype is implemented and tested. The next stage is to assemble the hardware, implement software modules, verify the request-to-acknowledgement workflow, collect repeated measurements and analyse results. Quantitative performance and energy-saving claims are intentionally withheld until supported by experimental evidence.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 11, November 2026

|DOI: 10.15680/IJIRSET.2026.1511037|

REFERENCES

1. Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2015). Internet of Things: A survey on enabling technologies, protocols, and applications. IEEE Communications Surveys & Tutorials, 17(4), 2347–2376. https://doi.org/10.1109/COMST.2015.2444095
2. Banks, A., Briggs, E., Borgendale, K., & Gupta, R. (2019). MQTT version 5.0. OASIS Standard. https://docs.oasis-open.org/mqtt/mqtt/v5.0/os/mqtt-v5.0-os.html
3. Gungor, V. C., Sahin, D., Kocak, T., Ergut, S., Buccella, C., Cecati, C., & Hancke, G. P. (2011). Smart grid technologies: Communication technologies and standards. IEEE Transactions on Industrial Informatics, 7(4), 529–539. https://doi.org/10.1109/TII.2011.2166794
4. National Institute of Standards and Technology. (n.d.). Building integration with smart grid. https://www.nist.gov/programs-projects/building-integration-smart-grid
5. 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
6. International Electrotechnical Commission. (2021). IEC 62746-10-1:2021: Systems interface between customer energy management system and the power management system—Part 10-1: Open automated demand response. IEC.
7. OASIS. (2014). Energy interoperation version 1.0. OASIS Standard. https://docs.oasis-open.org/energyinterop/energyinterop/v1.0/energyinterop-v1.0.html



INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi crossref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

www.ijirset.com

Scan to save the contact details