IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 2, June 2025
Impact Factor: 7.67

Decentralised Energy Consumption Monitoring using Painless Mesh and ESP-NOW Protocol

Janhavi Pujare[1], Shreya Sirsale[2], Dhananjay Gunjal[3], Santosh Lavate[4]
Student, Department of Electronics & Telecommunication[1-3]
Professor, Department of Electronics & Telecommunication[4]
AISSMS College of Engineering, Pune, India
janhavipujare@gmail.com, shreyasirsale@gmail.com,
dhananjay.gunjal07@gmail.com, santoshlavate@gmail.com

Abstract: *India is one of the world's largest consumers of electricity. With the growing population and rapid growth in industrialisation, the energy demand is brimming. However, beneath this impressive progress lies a concerning truth - India is not just consuming energy but also wasting it at an alarming rate. Much of the electricity wastage in India is not due to a lack of technology but rather to inefficient usage and behavioural habits, such as leaving appliances on standby, forgetting to turn off lights, or operating energy-intensive devices unnecessarily. In industries, machinery can drastically increase energy consumption, making it essential to monitor specific machines, floors, or zones to identify inefficiencies. By determining the difference between the expected energy consumption, a system is designed to consume and its actual consumption, our project provides a versatile energy monitoring solution. It enables real-time tracking, detects wasteful practices, and optimizes energy usage, helping industries and other applications reduce overall consumption effectively.*

Keywords: Energy monitoring, ESP NOW, Painless Mesh, Energy Consumption, ESP-32, RS-485

I. INTRODUCTION

The 21st century marks a transformative era of industrial revolution with the advent of Industry 4.0, characterized by the integration of automation, artificial intelligence, and smart technologies across various sectors. This rapid technological advancement has reshaped industrial processes, enhancing efficiency, connectivity, and data-driven decision-making. However, as industries increasingly rely on automated systems, high-power computing, and large-scale manufacturing, the demand for energy has surged, posing significant challenges in terms of energy efficiency, sustainability, and cost management.

Traditional energy monitoring systems, predominantly centralized and reliant on wired networks and cloud-based processing, present inherent limitations such as high infrastructure costs, latency issues, data congestion, and a single point of failure. These inefficiencies hinder real-time power management and contribute to excessive energy consumption, leading to increased operational costs and environmental impact. Moreover, centralized architectures often lack the flexibility to provide localized insights into energy consumption patterns, making it difficult for industries to implement effective energy optimization strategies.

To address these challenges, this paper presents a decentralized energy monitoring system leveraging ESP-NOW and Painless Mesh technologies, designed to provide real-time, cost-effective, and scalable energy tracking in industrial and commercial infrastructures. ESP-NOW, an ultra-low-latency, peer-to-peer wireless communication protocol developed by Espressif, enables direct device-to-device data transfer without reliance on internet connectivity, significantly reducing power consumption and network dependency. Complementing this, Painless Mesh creates a self-healing, multi-node communication network, ensuring reliable and scalable data transmission across distributed monitoring nodes without a centralized server.

The proposed system is implemented in a multi-floor industrial setting, where ESP32-based energy monitoring nodes gather real-time consumption data and transmit it using RS485 communication and ESP-NOW networking. This

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 2, June 2025
Impact Factor: 7.67

decentralized approach provides granular insights into localized energy usage, enabling industries to identify inefficiencies, optimize load distribution, and implement predictive energy-saving strategies. By eliminating reliance on extensive wiring, internet connectivity, or cloud-based processing, the system offers a cost-effective, adaptable, and robust alternative to traditional solutions, making it particularly beneficial for large-scale facilities.

Furthermore, the collected data undergoes statistical analysis to generate periodic reports, which assist industries in making informed decisions on energy conservation and regulatory compliance. This contributes not only to operational cost reduction but also to lowering carbon emissions, aligning with global sustainability initiatives and stringent energy efficiency regulations

II. LITERATURE REVIEW

The integration of Internet of Things (IoT) technologies into energy monitoring systems has gained significant attention due to its potential to enhance energy efficiency, reduce wastage, and provide real-time data analytics. Several studies have explored various IoT-based methodologies for energy monitoring, employing microcontrollers, wireless sensor networks (WSNs), cloud-based platforms, and mobile applications. This section reviews recent advancements in the field, highlighting key contributions from existing research.

El-Khozondar and Nassar (2024) developed a low-cost IoT-based energy monitoring system using an ESP32 microcontroller to collect real-time energy consumption data. Their system was specifically designed for areas facing energy shortages, such as Gaza, where consumers rely on private generators. The collected energy data is analysed and transmitted via WhatsApp using the Blynk platform, allowing users to track their consumption efficiently. This study demonstrates the feasibility of a cost-effective, real-time energy monitoring system using IoT.[1]

Aisyah Mohd et al. (2023) proposed an IoT-based energy monitoring system to minimize energy waste and costs. Their system integrates sensors, measuring devices, and IoT modules using a Wireless Sensor Network (WSN). Data collected from voltage and current sensors is uploaded to the ThingSpeak cloud using an ESP8266 Wi-Fi module, enabling real-time monitoring via the ThingView mobile app. This research emphasizes the role of cloud platforms in energy data visualization and their potential to assist users in managing electricity consumption effectively.[2]

Similarly, Raghu Yogesh et al. (2023) developed an IoT-based energy monitoring framework incorporating WSN technology. Their system employs an ESP8266 Wi-Fi module to collect energy parameters and transmit them to the ThingSpeak cloud, where real-time data is displayed on the ThingView mobile app. This study underscores the effectiveness of cloud-based energy management solutions in reducing energy wastage through real-time monitoring.[3]

Yahya et al. (2023) introduced a real-time IoT-based Energy Management System (EMS) composed of four functional modules: sensing, control, display, and switching. Their system integrates current transformers and voltage sensors with Arduino Nano and ESP32 microcontrollers, enabling local processing and cloud-based real-time energy tracking. Additionally, a web-based application was developed to retrieve, display, and manage energy data from the cloud, ensuring seamless user interaction. This research highlights the advantages of multi-controller IoT architectures in energy monitoring.[4]

B. Sujatha et al. (2020) addressed the inefficiencies of manual meter reading by proposing an automated IoT-based energy monitoring system. Their system employs Arduino microcontrollers to measure voltage, current, power, and energy consumption parameters, reducing human intervention and errors. The study demonstrates the efficacy of IoT in automating energy monitoring, thereby improving data accuracy and energy efficiency.[5]

Gan et al. (2018) developed an IoT-based industrial energy monitoring platform utilizing MQTT protocol and LoRa communication networks. Their system transmits energy data to a MySQL-based cloud server, while an Apache-driven web dashboard allows users to interactively analyse energy consumption. This study highlights the importance of robust communication protocols such as MQTT and LoRa in ensuring secure and efficient data transmission for large-scale energy monitoring applications.[6]

The reviewed literature indicates that IoT-enabled energy monitoring systems have evolved significantly, with a focus on real-time data acquisition, cloud storage, mobile-based visualization, and automation. The ESP32 and ESP8266

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 2, June 2025
Impact Factor: 7.67

microcontrollers, along with WSN-based communication, have proven effective for energy monitoring applications. Additionally, cloud platforms like ThingSpeak and MQTT-based communication protocols have been widely adopted for real-time data accessibility. Future research can explore AI-driven predictive energy management, enhanced security for IoT energy systems, and scalable architectures for industrial energy monitoring.

III. METHODOLOGY

The decentralized energy monitoring system is designed to measure and analyse energy consumption using a smart energy meter. The acquired data undergoes analog-to-digital conversion via RS-485 and is processed by an ESP32 microcontroller on each floor. The system is implemented across three floors, where each floor's ESP32 module transmits its energy data to the level below, cumulatively integrating the data from all preceding floors. At the lowest floor, an ESP32 module forwards the aggregated dataset to an MQTT server, enabling real-time monitoring through a user interface.

This approach ensures efficient data aggregation, allowing comprehensive energy analysis, identification of inefficiencies, and optimization strategies. The project is structured into three core components:

3.1 Hardware Implementation
3.2 Software Implementation
3.3 Connectivity

3.1 Hardware Implementation:

This section describes the systematic approach adopted for the hardware implementation of the proposed system. In the design of an electronic system, key considerations include cost-effectiveness, reduced circuit complexity, minimal power consumption, and overall system reliability. The choice of components was made to ensure these requirements were met while preserving the intended operational functionality. Figure 1 is the illustration of block diagram of the system.

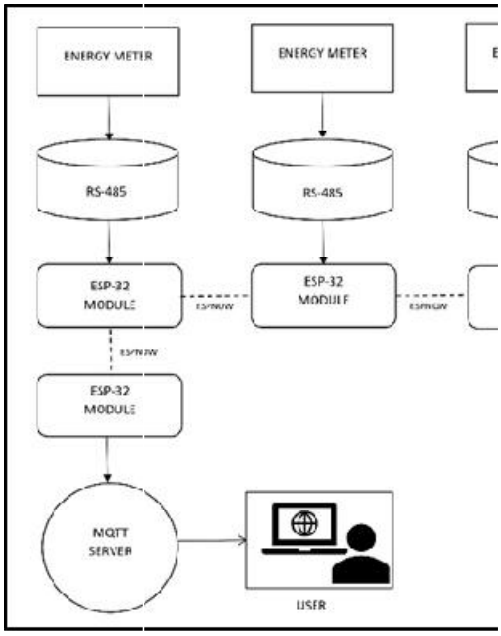

Fig. 1. Block diagram of System

The block diagram illustrates a decentralized energy consumption monitoring system that integrates energy meters, RS-485 communication, ESP-32 modules, and an MQTT server to facilitate real-time data acquisition, transmission, and analysis. The system is designed to function without traditional wired networking, relying on ESP-NOW for efficient wireless communication.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 2, June 2025
Impact Factor: 7.67

A. Data Acquisition Layer

The data acquisition layer serves as the foundation of the system, responsible for collecting real-time energy consumption data. This process begins with energy meters installed at different sections of the building, each tasked with measuring parameters such as voltage, current, power factor, and overall power consumption. Since energy meters typically operate with serial communication, RS-485 modules are used to facilitate data transmission. RS-485 is chosen due to its high noise immunity and long-distance communication capabilities, making it ideal for industrial and multi-floor environments. The collected data is then forwarded to ESP-32 microcontrollers, which act as intermediaries, processing and formatting the data for further transmission. The ESP-32 modules ensure that the data remains structured and ready for wireless communication, reducing latency and improving efficiency.

B. Wireless Communication Layer

The wireless communication layer is responsible for transmitting the processed energy consumption data across different nodes without relying on traditional Wi-Fi or wired connections. This is achieved using ESP-NOW, a low-power, low-latency wireless communication protocol supported by ESP-32 microcontrollers. ESP-NOW enables direct peer-to-peer communication between multiple ESP-32 nodes, forming a hierarchical data flow. In this setup, ESP-32 modules located on higher floors transmit their data to modules on lower floors until the information reaches a designated gateway ESP-32 module. This decentralized transmission approach ensures robustness by eliminating single points of failure, reducing dependency on internet connectivity, and minimizing infrastructure costs. By leveraging ESP-NOW, the system achieves efficient data transfer with minimal energy consumption, making it suitable for large-scale deployments.

C. Data Processing and User Interaction layer

The data processing and user interaction layer plays a crucial role in aggregating, storing, and visualizing the collected data. At this stage, the gateway ESP-32 module forwards the energy consumption data to an MQTT server. MQTT, a lightweight messaging protocol, is used to ensure real-time communication between devices and the central database. The server organizes and stores the received data, making it accessible for further analysis.

The front-end of the system was designed and implemented using HTML, CSS, and React. HTML and CSS were used to structure and style the user interface, while React—a widely used JavaScript library—was employed for creating dynamic and responsive UI components. This technology stack provided flexibility, scalability, and performance efficiency in building a real-time web interface for monitoring energy consumption data fetched from the MQTT server.

3.1.1. Smart Energy Meter:

To monitor electrical parameters on each floor of the building, the system incorporates the Selec EM2M-1P-C-100A, a precision-engineered single-phase energy meter designed for high-accuracy measurements and seamless data communication. This energy meter is capable of capturing voltage, current, active/reactive/apparent power, and energy, each monitored node. This model provides accurate RMS measurement. It supports RS-485 Modbus RTU communication, which is used in this system to interface with the ESP32 microcontroller for robust, long-distance, and noise-immune data transmission. The meter operates as a self-powered device with direct connection capability up to 100A. It includes an LCD backlit display for real-time parameter visualization and supports two pulse outputs for energy integration or relay-based applications.

TABLE I: Energy meter specifications

Parameters	Ratings
Phase Type	Single Phase
Voltage Measurement	11-300 V AC
Current Measurement	0-100 A
Power Consumption	Active, Reactive, Apparent

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 2, June 2025
Impact Factor: 7.67

Frequency Range	45-65 Hz
Pulse Output	2 Outputs
Power Consumption	8 VA max

3.1.2. ESP-32 Microcontroller:

The ESP32-WROOM-32 is a powerful and versatile Wi-Fi and Bluetooth microcontroller module developed by Espressif Systems. It is based on the ESP32-D0WDQ6 chip and integrates a dual-core processor with low power consumption, high processing capabilities, and multiple I/O interfaces, making it ideal for various IoT and embedded system applications. The module supports both classic Bluetooth and Bluetooth Low Energy (BLE) standards, enabling flexible wireless communication in a wide range of environments. With integrated flash memory, programmable GPIOs, and hardware acceleration for cryptographic algorithms, the ESP32-WROOM-32 ensures secure and efficient performance.

In the system, the ESP32-WROOM-32 modules serve as the primary processing and communication nodes on each floor. Each ESP32 module is interfaced with an energy meter via RS-485 protocol to acquire real-time power consumption data. These modules are configured to transmit the collected data wirelessly using the ESP-NOW protocol, forming a mesh network for hierarchical data transfer. The lowest floor's ESP32 module acts as a gateway, pushing the aggregated data to an MQTT server for storage and visualization.

TABLE II: ESP-32 SPECIFICATIONS

Parameter	**Specification**
Operating Voltage	3.0V to 3.6V
Operating Frequency	Up to 240 MHz
Flash Memory	4 MB (external SPI flash)
SRAM	520 KB
Wireless Connectivity	Wi-Fi 802.11 b/g/n, Bluetooth v4.2 BR/EDR and BLE
Interface Options	SPI, I2C, UART, ADC, DAC, PWM, GPIO
Operating Temperature	-40°C to 85°C
Antenna	PCB onboard antenna

3.1.3. RS-485:

The RS-485 module, integrated into the system for serial communication, is based on the MAX485 IC, a low-power transceiver that enables differential signalling. This module ensures reliable and noise-resistant data transmission over long distances, which is essential for industrial-grade communication. Operating on a 5V DC supply, it facilitates bidirectional half-duplex communication and can support multiple nodes on the same bus. Its differential signalling approach minimizes electromagnetic interference and enhances signal integrity. The RS-485 module is interfaced with the ESP32 microcontroller, where the receiver output (RO) pin is connected to the ESP32's RX pin and the driver input (DI) to the TX pin, allowing seamless UART-level communication. This robust design ensures minimal data loss even in electrically noisy environments, making the RS-485 module a reliable choice for the proposed decentralized energy monitoring system.

TABLE III: RS-485 SPECIFICATIONS

Parameters	**Ratings**
Phase Type	Single-phase
Voltage Measurement	5V DC
Current Measurement	~5–10 mA
Power Consumption	Minimal
Data Rate	Up to 250 kbps
Output	UART level
Temperature	-40°C to +85°C

DOI: 10.48175/IJARSCT-27473

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 2, June 2025
Impact Factor: 7.67

3.2 Software Implementation:

The software stack implemented in this system has been carefully chosen to support the core requirements of real-time energy data acquisition, decentralized communication, data persistence, and user-oriented visualization. The integration of embedded programming tools, messaging protocols, database management systems, and frontend-backend development frameworks provides a robust and scalable architecture. Each software component plays a distinct role in ensuring seamless interaction between hardware nodes and the user interface.

3.2.1 Arduino IDE:

All ESP32 modules in the network are programmed using the Arduino IDE with C language. The firmware is responsible for continuous data acquisition from the energy meters through RS-485 communication. Each ESP32 node is programmed to format, verify, and transmit the acquired data using the ESP-NOW protocol. The ESP-NOW communication is implemented to operate on a hierarchical model, enabling nodes on higher floors to relay data to the next level until it reaches the gateway node on the ground floor. This approach eliminates the need for traditional Wi-Fi or wired connections and is optimized for low latency and minimal power consumption. The embedded firmware also handles error-checking routines and device-to-device acknowledgments to ensure data integrity during wireless transmission.

3.2.2 MQTT Server:

The gateway ESP32 node, designated as the communication endpoint, receives aggregated data from upper-level nodes and publishes it to the MQTT server. MQTT serves as the protocol layer responsible for bridging the hardware and software environments. The ESP32 uses a publish-subscribe mechanism to push time stamped energy data under designated topics. The MQTT server listens to these topics, receives incoming messages in real time, and passes them to backend services. This ensures asynchronous message flow and fault-tolerant communication between physical nodes and storage systems. The use of MQTT also facilitates system expansion by allowing additional nodes to publish to new topics without reconfiguring the entire server logic.

3.2.3 Visual Studio Code (VS Code):

Backend scripting and logic integration are carried out using Python within the Visual Studio Code environment. The application code developed in VS Code is responsible for establishing a persistent connection to the MQTT broker, subscribing to the relevant energy data topics, and processing the received data packets. This processed data is then prepared for database insertion by applying format validation, timestamp conversion, and floor-wise classification. Additionally, VS Code is used to implement the business logic that bridges MQTT data with the PostgreSQL database and prepares JSON-formatted responses for frontend queries. Exception handling, logging, and retry mechanisms are also incorporated to ensure fault tolerance and data consistency.

3.2.4 pgAdmin4 (PostgreSQL):

The PostgreSQL database, managed via the pgAdmin4 interface, acts as the primary data storage unit. Structured tables are designed to store data entries with fields such as floor number, node ID, timestamp, voltage, current, power, and frequency. These records are optimized for fast querying and support aggregated statistical views required for real-time dashboards and historical analyses. PostgreSQL also supports SQL-based triggers for data validation and anomaly detection. The database schema is normalized to reduce redundancy and supports future schema migration to accommodate additional system parameters or new hardware modules.

3.2.5 HTML (Hyper Text Markup Language):

The structure of the user interface was developed using HTML, which served as the foundational markup language for the web-based frontend. HTML enabled the creation of a clean, organized layout for displaying real-time energy consumption data. It facilitated the integration of key elements such as data tables, live MQTT readings, floor-level

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 2, June 2025
Impact Factor: 7.67

energy displays, and user authentication fields. By providing a semantic structure, HTML ensured that the interface was accessible, efficient, and aligned with modern web development standards.

3.2.6 CSS (Cascading Style Sheets):

CSS was used for styling the user interface elements defined by HTML. It allowed for the customization of fonts, colors, backgrounds, spacing, and overall visual presentation of the monitoring dashboard. With responsive design techniques, CSS ensured that the UI rendered correctly across various screen sizes and devices, such as desktops, laptops, and mobile phones. The use of CSS significantly enhanced the readability and aesthetic appeal of the interface, contributing to a user-friendly experience for monitoring energy data in real time.

3.2.7 React (JavaScript Library):

To add dynamic behavior and real-time responsiveness to the UI, React was used as the primary JavaScript library. React enabled the development of modular, reusable components that could handle live updates from the MQTT server without needing to reload the webpage. This allowed for seamless visualization of changing energy values and user interactions. React's virtual DOM and efficient state management made it ideal for building a scalable and performant frontend suitable for an energy monitoring system across multiple floors.

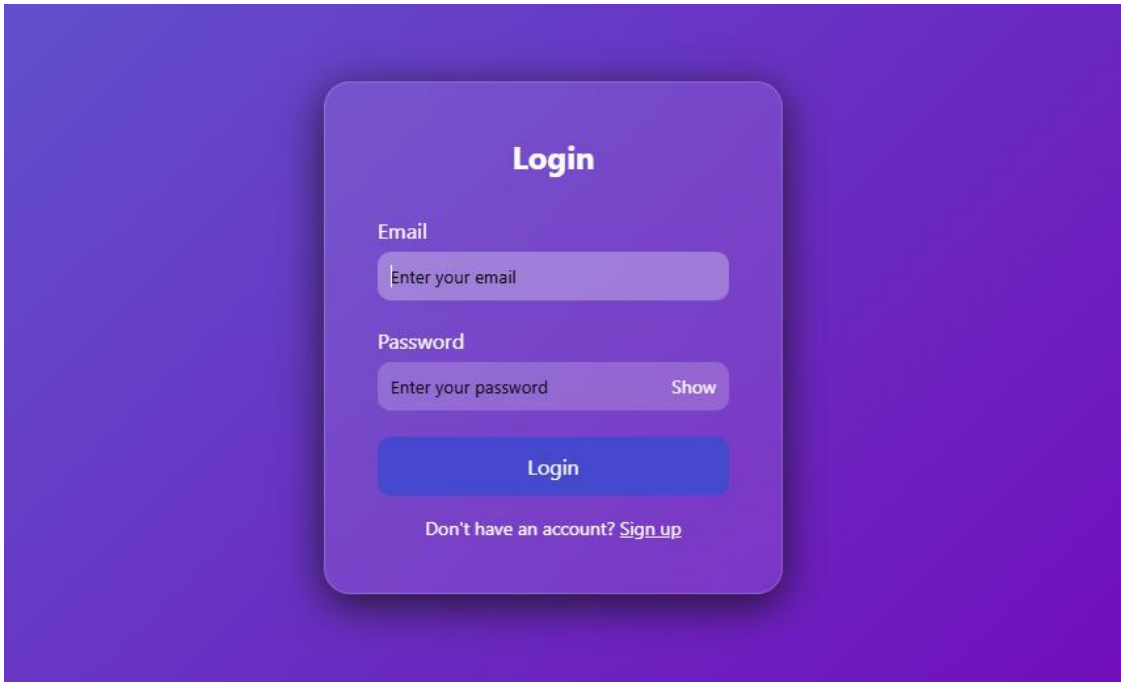

Figure 2: Login Page Image

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 2, June 2025
Impact Factor: 7.67

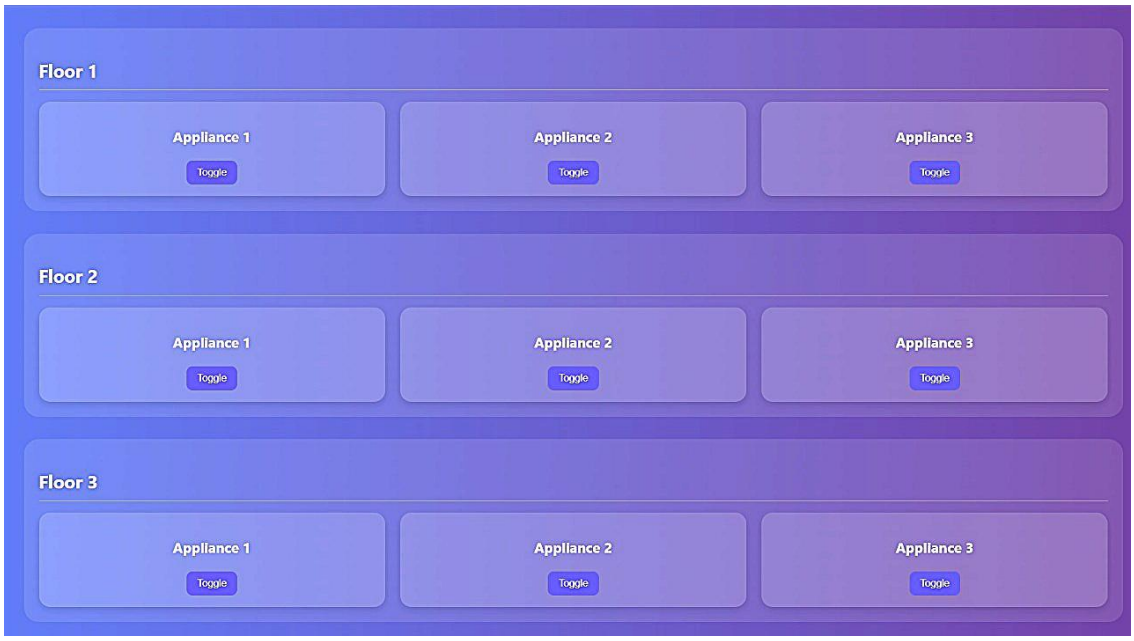

Figure 3: Selection Page

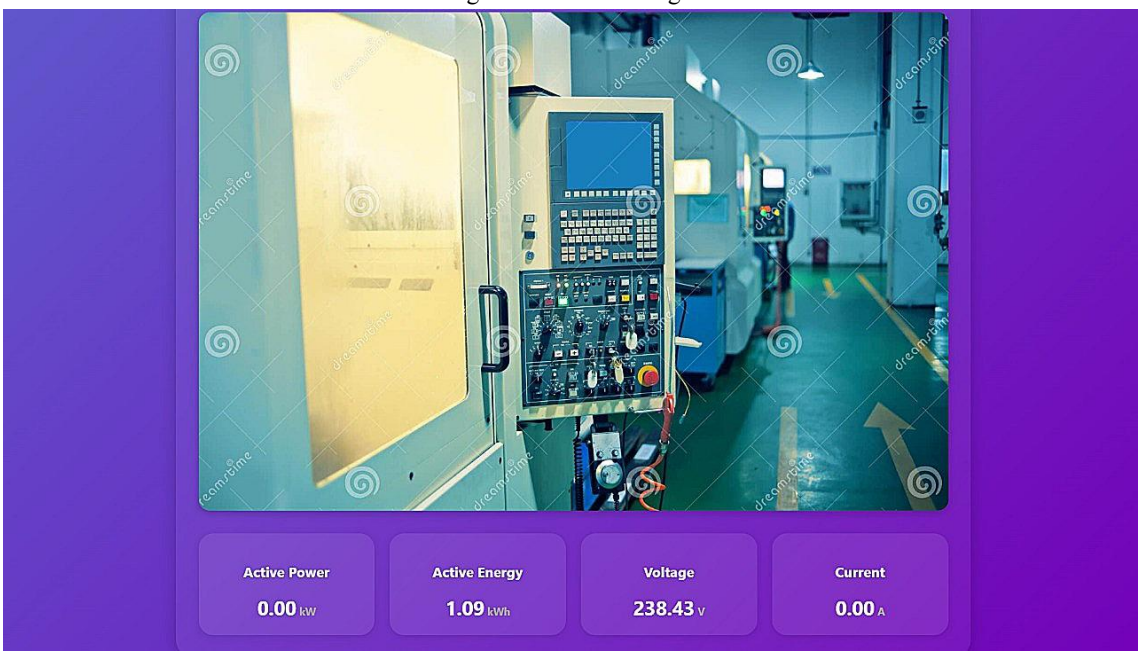

Figure 4: Data Output

4.3 Connectivity Protocols:

The proposed system employs a combination of communication protocols to ensure seamless data acquisition, reliable wireless transmission, and efficient data delivery to the cloud interface. These protocols play a vital role in maintaining synchronization between hardware components and facilitating low-latency, bi-directional communication across different layers of the system. The connectivity framework consists primarily of ESP-NOW, UART, and MQTT protocols, each chosen for its performance characteristics and compatibility with the system architecture.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 2, June 2025
Impact Factor: 7.67

4.3.1 ESP-NOW:

ESP-NOW is a low-power, peer-to-peer wireless communication protocol developed by Espressif Systems. It allows multiple ESP32 modules to exchange data directly without the need for an active Wi-Fi connection or external network infrastructure. This protocol is highly suitable for mesh-based communication due to its low overhead and reduced latency. In the present system, ESP-NOW is used to establish a hierarchical data relay between nodes situated on each floor. The energy consumption data is forwarded progressively from higher to lower floors, culminating at a central gateway node. This decentralized transmission structure reduces reliance on centralized routers, lowers power consumption, and enhances system robustness in multi-floor environments.

4.3.2 UART (Universal Asynchronous Receiver-Transmitter):

UART is a widely used serial communication protocol that facilitates direct, asynchronous data exchange between two devices. In this project, UART is employed to interface the smart energy meter with the ESP32 microcontroller using the RS-485 protocol. UART ensures reliable digital conversion and low-error data transmission from the meter to the processing unit. It provides a consistent communication link for the acquisition of voltage, current, power, and frequency parameters. This layer ensures that analog signals from the energy meter are accurately digitized and forwarded to the ESP32 without data distortion, forming the basis for real-time monitoring and control.

4.3.3 MQTT (Message Queuing Telemetry Transport):

MQTT is a lightweight, publish-subscribe messaging protocol designed for resource-constrained devices and low-bandwidth networks. It is employed in the final communication stage, where the ESP32 gateway module transmits aggregated energy data to a centralized server. The MQTT protocol enables real-time data synchronization between the ESP32 and the cloud database with minimal overhead. The publish-subscribe mechanism ensures scalability, allowing multiple devices to interact with the server efficiently. Once the data is received, it is parsed and stored in a PostgreSQL database. The MQTT protocol thereby acts as a critical bridge between the hardware and software subsystems, ensuring smooth end-to-end communication and supporting real-time energy analytics on the user interface.

IV. RESULTS AND DISCUSSIONS

The project aimed to implement a decentralized energy monitoring system using ESP-NOW and Painless Mesh communication protocols across three floors of a building. The data collected includes electrical parameters like Total Active Energy, Active Power, Voltage (Line to Neutral), and Current, which are read using an ESP32 microcontroller interfaced with a digital energy meter via the RS-485 Modbus protocol. The readings are extracted from specific Modbus register addresses as defined in the meter's datasheet.

MODBUS REGISTER ADDRESSES LIST

Readable parameters for Communication [Length (Register) : 2; Data Structure : Float (Swapped)]

Address	Hex Address	Parameter
30000	0x00	Total Active Energy
30002	0x02	Import Active Energy
30004	0x04	Export Active Energy
30006	0x06	Total Reactive Energy
30008	0x08	Import Reactive Energy
30010	0x0A	Export Reactive Energy
30012	0x0C	Apparent Energy
30014	0x0E	Active Power
30016	0x10	Reactive Power
30018	0x12	Apparent Power
30020	0x14	Voltage L-N
30022	0x16	Current
30024	0x18	Power Factor
30026	0x1A	Frequency
30028	0x1C	Max Demand Active Power
30030	0x1E	Max Demand Reactive Power
30032	0x20	Max Demand Apparent Power

Figure 5: MODBUS Register Address List

DOI: 10.48175/IJARSCT-27473

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 2, June 2025
Impact Factor: 7.67

The data includes:
u0: Total Active Energy (kWh)
u14: Active Power (kW)
u20: Voltage L-N (V)
u22: Current (A)
timestamp: Time of reading
The ESP32 reads these values, and depending on the floor, aggregates data from higher floors before sending it downwards via ESP-NOW to the node on the floor below. The ground floor node finally pushes this accumulated data to an MQTT server, which is then stored in a PostgreSQL database for further analysis.
The PostgreSQL database view shows consistent values recorded over time. For example, the Total Active Energy remains at 0.96 kWh over a series of timestamps. These are displayed in a time-series plot, which illustrates that the energy data remains stable over a short observation window, indicating either a low power load or an idle condition during the testing phase.
Voltage L-N values fluctuate between 250.66V and 251.32V, showing the voltage is stable and within an acceptable operating range.
Current values range from 0.09A to 0.12A, and Active Power stays mostly around 0.02 kW, confirming minimal power usage during this test.
Time stamped data is stored accurately in the database, allowing for future aggregation on a daily, weekly, or monthly basis.
Below is the sample data collected during testing, viewed in PostgreSQL database.

	id [PK] integer	u0 character varying (10)	u14 character varying (10)	u20 character varying (10)	u22 character varying (10)	timestamp timestamp without time zone
13	13	0.96	0.03	250.66	0.11	2025-02-20 01:37:14.781709
14	14	0.96	0.02	250.84	0.09	2025-02-20 01:37:20.106237
15	15	0.96	0.02	250.84	0.12	2025-02-20 01:37:25.472753
16	16	0.96	0.02	250.76	0.10	2025-02-20 01:37:30.755398
17	17	0.96	0.02	250.91	0.10	2025-02-20 01:37:36.007097
18	18	0.96	0.02	250.98	0.10	2025-02-20 01:37:41.777624
19	19	0.96	0.02	251.05	0.11	2025-02-20 01:37:47.548776
20	20	0.96	0.02	251.10	0.09	2025-02-20 01:37:52.812653
21	21	0.96	0.02	250.99	0.10	2025-02-20 01:37:58.198127
22	22	0.96	0.02	250.96	0.10	2025-02-20 01:38:03.593187
23	23	0.96	0.02	250.98	0.10	2025-02-20 01:38:08.704756
24	24	0.96	0.02	251.02	0.10	2025-02-20 01:38:14.07658
25	25	0.96	0.02	251.07	0.10	2025-02-20 01:38:19.394084
26	26	0.96	0.03	251.11	0.11	2025-02-20 01:38:24.664166
27	27	0.96	0.02	251.16	0.09	2025-02-20 01:38:29.947245
28	28	0.96	0.02	251.32	0.10	2025-02-20 01:38:35.363917
29	29	0.96	0.02	251.29	0.10	2025-02-20 01:38:40.795483
30	30	0.96	0.03	251.09	0.12	2025-02-20 01:38:45.909043
31	31	0.96	0.02	251.09	0.09	2025-02-20 01:38:51.240395
32	32	0.96	0.02	251.09	0.10	2025-02-20 01:38:56.565208

Figure 6: Database stored measurements

Consistent total energy consumption at 0.96 kWh
Voltage remained stable between 250.66V to 251.32V
Current values fluctuated slightly between 0.09A to 0.12A
Active Power remained low, indicating light load or idle operation

DOI: 10.48175/IJARSCT-27473

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 2, June 2025
Impact Factor: 7.67

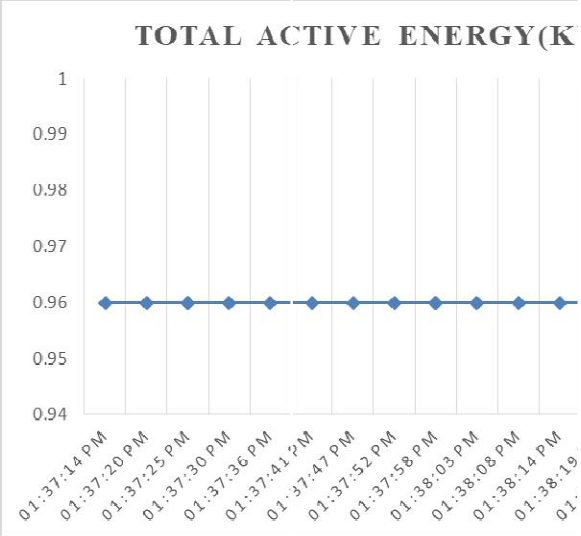

Figure 6: Total Active Energy (kWh) VS Time graph

As observed from the graph, the total active energy value remains constant at 0.96 kWh across the time interval. This is expected because the connected load was minimal or static during testing. Even though there were minor changes in current and voltage values, the total energy accumulated over short intervals stayed nearly the same.

This verified system establishes the groundwork for the development of the user interface, where energy usage can be visualized and managed more efficiently. The UI, designed in HTML and built using CSS and React, will allow filtering by time and floor, along with graphical analysis and optimization recommendations based on real-time data.

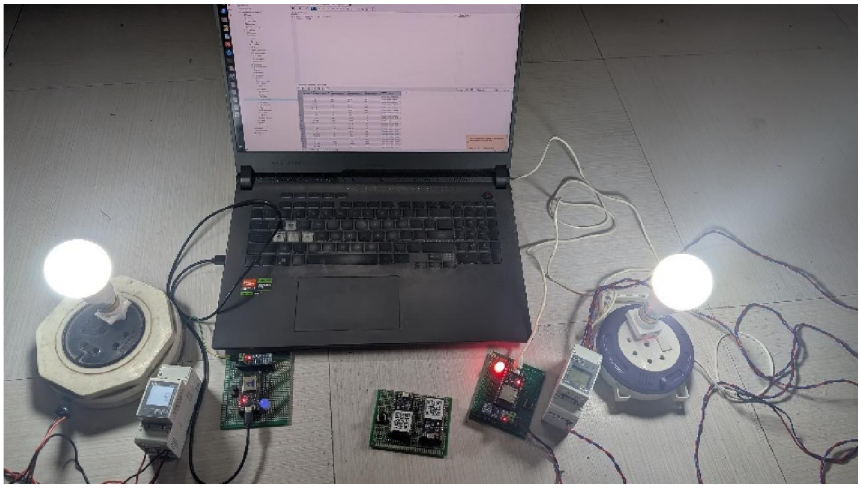

Figure 8: Working Setup

A multi-node testing environment, where two separate ESP32 nodes are connected to two individual energy meters and respective loads (light bulbs). A laptop displaying the data in PgAdmin4 confirms that the data received from the ESP32 nodes is correctly aggregated and forwarded to the PostgreSQL database. This setup simulates a multi-floor deployment, validating the communication between nodes over ESPNOW and the successful transfer of data to the central MQTT broker and storage system. The glowing bulbs confirm real-time load presence and the system's ability to detect and log power consumption changes accurately.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 2, June 2025
Impact Factor: 7.67

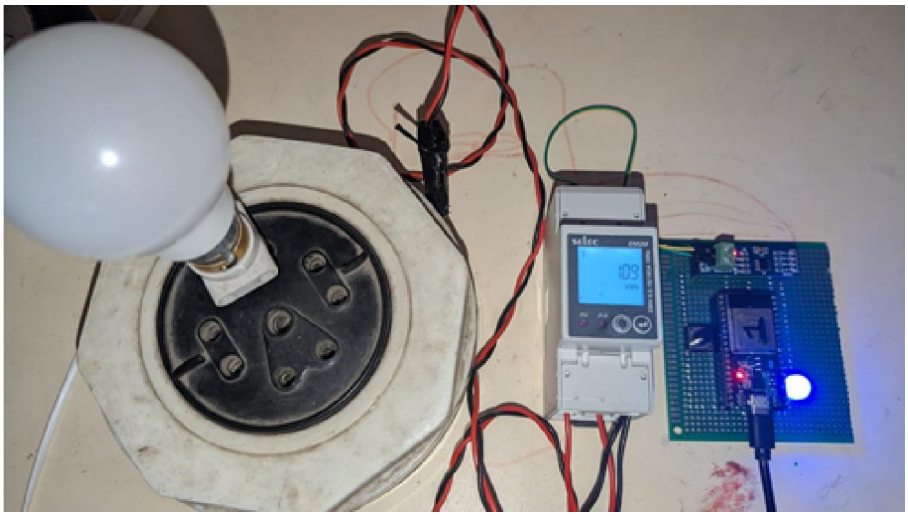

Figure 7: Energy consumption measured by the energy meter

The images included in the results section visually support the successful implementation and testing of the project setup. A working prototype of a single ESP32 node connected to the selected digital energy meter through the RS-485 interface. The meter displays the energy consumption in kWh while the ESP32, powered via USB, is transmitting the data. This demonstrates the proper functioning of the hardware interface between the energy meter and the ESP32.

This confirms that the data is successfully read from the digital energy meter using the RS-485 protocol, ensuring accurate acquisition of energy parameters. The communication between ESP32 nodes using the ESP-NOW protocol has proven to be reliable across all floors, enabling smooth data transfer without noticeable packet loss. The aggregation and forwarding of data from upper floors to the ground floor node worked as intended, demonstrating the effectiveness of the decentralized mesh approach. Finally, the MQTT publishing mechanism and integration with the PostgreSQL database were verified to be functioning properly, completing the full data pipeline from sensor acquisition to database storage.

V. CONCLUSION

The decentralized energy consumption monitoring system proposed in this study effectively integrates hardware and software components to enable real-time energy tracking in a multi-floor building. The implementation of ESP32 microcontrollers, RS-485 communication, and ESP-NOW wireless transmission ensures seamless data acquisition and aggregation from multiple floors. The collected energy data is further processed and stored using an MQTT server, facilitating real-time monitoring through a user-friendly interface developed using HTML, CSS and React.js.

This system not only provides accurate energy consumption metrics but also allows users to analyze historical trends and optimize power usage. The ability to filter data based on floors, specific machines, and timeframes enhances its applicability in industrial and commercial settings. The scalable architecture ensures that additional nodes and energy meters can be integrated without significant modifications. By leveraging statistical analysis and visualization techniques, the system aids in identifying inefficiencies and implementing energy-saving strategies, ultimately contributing to sustainable energy management.

REFERENCES

[1]. H. J. El-Khozondar and Y. F. Nassar, “A smart energy monitoring system using ESP32 microcontroller,” *e-Prime - Adv. Electr. Eng., Electron. Energy*, vol. 9, Article no. 100666, 2024.

[2]. A Mohd, N. Sazali, A. S. Jamaludin, and N. Misdan, “IoT-based energy monitoring system for energy conservation,” in *Proc. E3S Web Conf.*, vol. 437, Article no. 04006, IConGEET, 2023.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 2, June 2025
Impact Factor: 7.67

[3]. R. Yogesh, M. Keerthivasagan, and S. Vijayanand, "Energy monitoring system using IoT," *Int. J. Adv. Eng. Manag. (IJAEM)*, vol. 5, no. 6, pp. 264–267, Jun. 2023.

[4]. M. S. Yahya, B. Muhammad, M. A. Abubakar, U. I. Abdullahi, and Z. I. Musa, "Implementation of a real-time IoT-based energy management system," *J. Eng. Res. Rep.*, vol. 25, no. 10, pp. 19–29, 2023, Article no. JERR.107232.

[5]. B. Sujatha, N. Aamani, P. Harshini, and M. Vaishnavi, "Energy monitoring system using IoT technology," *Int. J. Creative Res. Thoughts (IJCRT)*, vol. 8, no. 4, Apr. 2020.

[6]. S. Gan, K. Li, Y. Wang, and C. Cameron, "IoT-based energy consumption monitoring platform for industrial processes," in *Proc. UKACC 12th Int. Conf. Control (CONTROL)*, Sheffield, U.K., Sep. 5–7, 2018, pp. 236–240.