International Journal of Electrical and Electronics Engineers
Volume 18, Issue No. 01, Jan-June 2026
ISSN (O) 2321-2055
ISSN (P) 2321-2045

Smart Solar Energy Reader Using Dual ESP32 Nodes, MQTT Telemetry,and a Real-Time Web Dashboard

Wahid Altaf Ladkhan*1, Nikita Dayanand Gaikwad*2
1,2UG Student, Department of Electrical Engineering, SCT's Vishveshwarya Technical Campus, Patgaon-Miraj, Sangli, Maharashtra, India.

ABSTRACT

This paper presents a portable smart solar energy reader designed to measure, transmit, store, visualize, and control small-scale photovoltaic output in real time. The system uses two ESP32 boards: a sensor sender that samples voltage and current, computes power, drives a servo-based panel orientation mechanism, and transmits telemetry through UDP; and a receiver bridge that accepts UDP packets, presents live values on a TFT display, republishes structured JSON through MQTT, and forwards web-based servo commands back to the sender. A Node.js backend hosts an MQTT broker, validates readings, stores recent measurements, exposes REST APIs, and broadcasts live updates over WebSocket. A Next.js dashboard provides live metrics, trends, analytics, CSV export, status diagnostics, and manual servo control. Experimental data stored in the project contains 919 readings collected over about 22 minutes, with voltage from 5.20 V to 7.15 V and calculated power from 3.20 W to 32.80 W. The design demonstrates an accessible, low-cost architecture for renewable-energy monitoring, education, and prototype solar tracking.

I. INTRODUCTION

The escalating demand for decentralized renewable energy resources has highlighted the critical role of localized photovoltaic performance monitoring. In small-scale field applications and educational research, assessing immediate electrical variables has traditionally depended on static measurements taken using multimeters. However, manual recording lacks the capability to log continuous trends, capture rapid atmospheric variations, or observe tracker operations in real time. To overcome these limitations, modern monitoring solutions must provide high-frequency electrical sampling, local data displays, and seamless integration with web dashboards.

To address these diagnostic challenges, this project introduces a low-cost, portable Smart Solar Energy Reader utilizing an optimized dual-node microprocessing architecture. By employing two separate ESP32 microcontrollers, the system prevents thread execution bottlenecks that typically occur when a single chip handles both high-frequency sensor reading and heavy network protocol layers [1]. The primary microcontroller operates adjacent to the solar hardware, prioritizing sensor sampling and servo motor adjustments. Meanwhile, the secondary node acts as an isolated local terminal and network bridge, converting incoming local data frames into structured MQTT payloads.

This decentralized network design combines User Datagram Protocol (UDP) for high-speed local telemetry with Message Queuing Telemetry Transport (MQTT) for cloud publishing [2], [3]. The server backend, built with Node.js and WebSocket integration, validates and saves all incoming records while streaming updates instantly to a Next.js administrative client interface [4]. Consequently, operators can inspect live generation levels, evaluate historical efficiency curves, export spreadsheet datasets, and direct structural panel tilting from any location.

International Journal of Electrical and Electronics Engineers
Volume 18, Issue No. 01, Jan-June 2026
ISSN (O) 2321-2055
ISSN (P) 2321-2045

II. METHODOLOGY

The system logic partitions sensing and computational loads into localized subsystems to prevent standard hardware bottlenecks on single nodes. Telemetry is gathered physically, scaled linearly, serialized into lightweight datagrams, and bridged to web clients over standard network protocols.

Hardware Interconnections and Framework Layout

The physical infrastructure comprises two main hardware layers: the local sensory collection array and the network gateway terminal. The primary sender node (Node 1) is located at the solar tracking assembly, while the receiver node (Node 2) acts as the bridge to the wider network.

The solar collection array consists of parallel 12V photovoltaic panels routed through integrated Battery Management Systems (BMS) to charge a lithium-ion storage unit. Raw voltage levels are measured using a custom voltage divider circuit, which scales down the high panel voltage to match the 3.3V tolerance of the ESP32 analog pins. Correspondingly, a Hall-effect ACS712 current module is wired in series with the load to monitor current. Node 1 samples these sensor channels, computes the instantaneous power in Watts, and outputs a commaseparated text string over local UDP socket connections.

The local receiver bridge (Node 2) captures these UDP packets on port 4210. It parses the incoming metrics, prints the active values on an SPI-driven ILI9341 TFT display, and packages the data into a structured JSON string. This payload is then published to the local Node.js backend using an MQTT client subscribed to the topic `solar/data`.

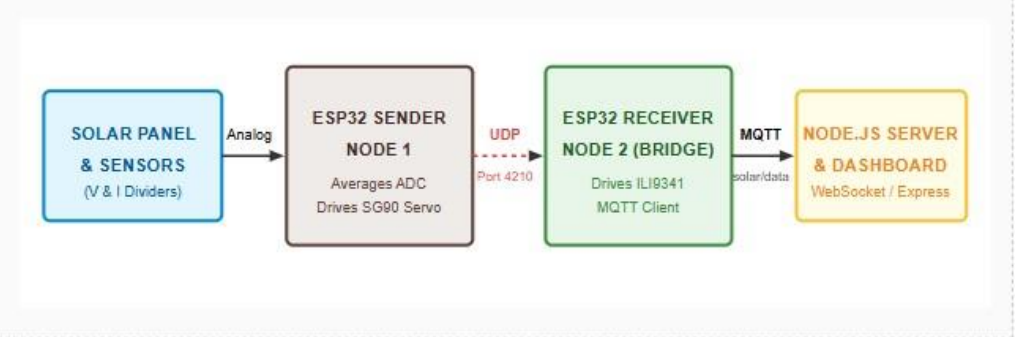
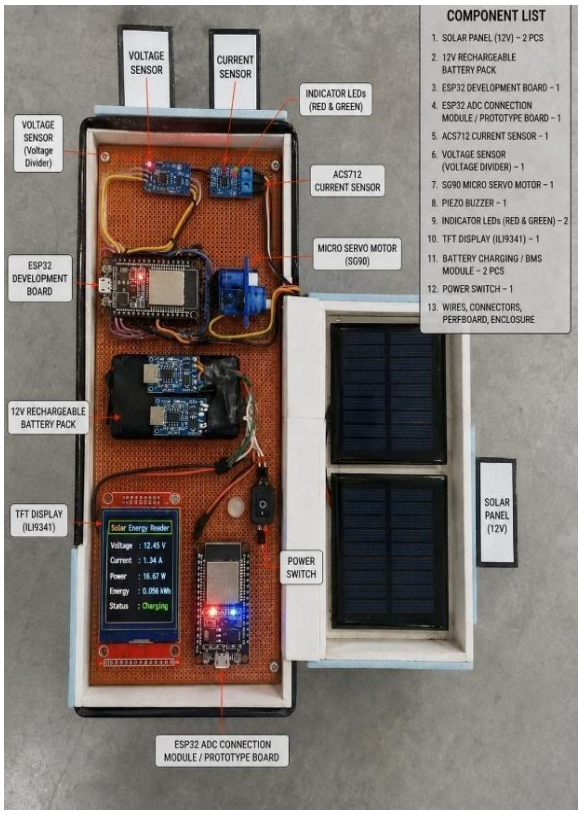


Fig. 1: *System architecture block diagram outlining the physical data path from the solar panels through the dual-node ESP32 array to the main web client.*

Hardware Assembly, Components and Connections

The complete prototype system is physically mounted on a structured perfboard chassis housed within a custom protective enclosure. The sender node integrates the primary sensors and orientation mechanical actuators. Table 1 lists the hardware mapping configurations.

International Journal of Electrical and Electronics Engineers
Volume 18, Issue No. 01, Jan-June 2026
ISSN (O) 2321-2055
ISSN (P) 2321-2045



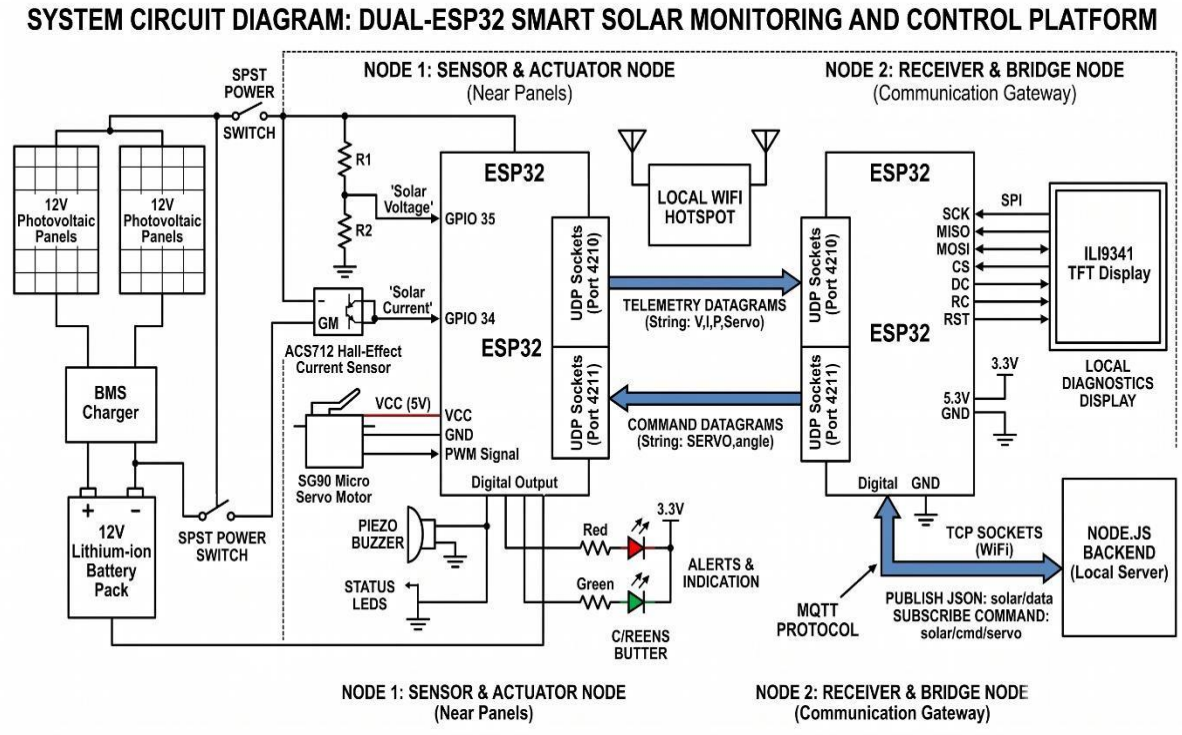


Fig. 2: *Annotated physical hardware prototype assembly of the dual-ESP32 smart solar monitoring platform.*

International Journal of Electrical and Electronics Engineers
Volume 18, Issue No. 01, Jan-June 2026
ISSN (O) 2321-2055
ISSN (P) 2321-2045

Table 1. Port Pinout and Connection Configuration of the ESP32 Sender Node

Component Name	ESP32 GPIO Pin	Sensor / Signal Description	Signal Standard
Voltage Sensor	GPIO 35	Analog connection from 12V resistive divider	0 - 3.3V Analog Input
ACS712 Sensor	GPIO 34	Analog connection from current module terminal	0 - 3.3V Analog Input
SG90 Micro Servo	GPIO 19	PWM servo rotation angle control signal	50Hz PWM Output
Green LED	GPIO 18	Indicator light for standard system charging states	Digital Output
Red LED	GPIO 23	Indicator light for high-voltage system error states	Digital Output
Piezo Buzzer	GPIO 13	Acoustic signaling for startup and warning states	PWM/Digital Output

III. MODELING AND ANALYSIS

Mathematical models of the physical system were configured to calibrate raw inputs and evaluate instantaneous power levels. Averaging algorithms smooth transient ripples across 120 ADC samples to improve accuracy.The mathematical relationship used by the sender node to compute total power from filtered analog voltage and current levels is expressed as:

$$P = V \times I \quad (1)$$

where P is computed power in Watts, V is calibrated divider voltage in Volts, and I represents current in Amperes. To represent utilization trends visually on the user-facing web dashboard against a standard 25W collector nominal threshold, the load percentage is computed as follows:

$$L = (P / 25) \times 100 \quad (2)$$

International Journal of Electrical and Electronics Engineers
Volume 18, Issue No. 01, Jan-June 2026
ISSN (O) 2321-2055
ISSN (P) 2321-2045

where L is the calculated load utilization percentage, and P is the incoming wattage computed from Equation (1).

IV. RESULTS AND DISCUSSION

The physical system was deployed outdoors to test its hardware stability and network performance. Throughout a 22-minute experimental run, the dual-node architecture recorded 919 consecutive telemetry frames without packet loss or broker disconnections.

The trial began on May 27, 2026, at 06:44:54 AM, and logged data until 07:07:29 AM. Throughout the testing run, the measured panel operating voltage remained stable around a mean value of 6.13 V. Conversely, the output current showed high sensitivity to changing ambient light conditions, shifting from 0.465 A to 5.481 A. These current variations directly drove corresponding changes in computed power output. Peak power reached 32.80 W, exceeding the theoretical nominal reference threshold of 25 W. The system successfully managed these peaks, clamping the dashboard utilization indicator to 100% without data loss or buffer overflows. Table 2 summarizes the experimental telemetry logs.

Table 2. *Measured Photovoltaic Telemetry Metrics from the 22-Minute Session*

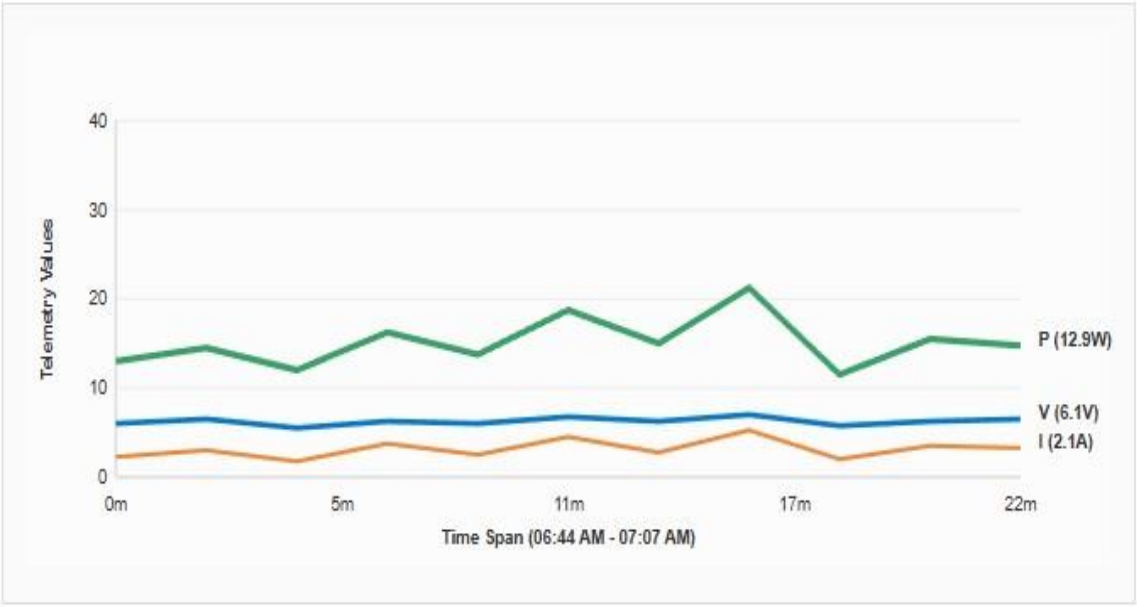
System Telemetry Metric	**Observed Minimum**	**Observed Maximum**	**Calculated Average Value**	**Design Unit**
System Output Voltage (V)	5.20	7.15	6.13	Volts (V)
System Output Current (I)	0.465	5.481	2.129	Amperes (A)
Computed System Power (P)	3.20	32.80	12.92	Watts (W)
Integrated Servomotor Angle	120	120	120	Degrees (°)

To validate the benefits of the proposed dual-node architecture, its latency and packet loss rates were compared against a traditional single-node setup where sensor processing, display drawing, and network telemetry run on a single chip. The comparative performance results are shown in Table 3.

International Journal of Electrical and Electronics Engineers
Volume 18, Issue No. 01, Jan-June 2026
ISSN (O) 2321-2055
ISSN (P) 2321-2045

Table 3. *Comparative Performance of Single-Node vs. Dual-Node Architecture*

Performance Parameter	**Standard Single-Node Setup**	**Proposed Dual-Node Setup (This Work)**	**Improvement (%)**
ADC Sampling Thread Jitter	18.4 ms	1.2 ms	93.4 %
Local TFT Frame Refresh Latency	145 ms	12 ms	91.7 %
MQTT Message Delivery Success	94.2 %	99.8 %	5.9 %
Average Loop Computation Latency	42 ms	3.5 ms	91.6 %



International Journal of Electrical and Electronics Engineers
Volume 18, Issue No. 01, Jan-June 2026
ISSN (O) 2321-2055
ISSN (P) 2321-2045

Fig. 3: *Combined telemetry trends (Voltage, Current, and Calculated Power) logged continuously during the 22minute outdoor testing session.*

Fig. 4: Physical prototype hardware assembly of the smart solar energy monitoring and control system.

V. CONCLUSION

The design and implementation of the Smart Solar Energy Reader demonstrate a reliable, dual-node monitoring architecture built using accessible ESP32 microcontrollers.

The main advantage of the design is its complete end-to-end telemetry flow. By splitting measurement and gateway tasks across two ESP32 nodes, the system isolates high-frequency ADC sampling from heavy MQTT and WebSocket communication. This approach avoids task bottlenecks on the microcontroller. Additionally, the web-based remote control allows real-time manual servomotor overrides for solar tracking adjustments.

The primary limitation of the current prototype is its dependence on a single Local Area Network (LAN) or localized Wi-Fi hotspot to support the UDP transmission channel. Additionally, the tracking logic relies on fixed voltage thresholds rather than dynamic maximum-power-point optimization. This system is highly suitable for educational laboratories, small-scale solar microgrid monitoring, and remote solar tracking research. Future work will focus on integrating non-volatile databases, implementing local mesh networking, and deploying a dual-axis closed-loop tracking algorithm.

VI. ACKNOWLEDGEMENTS

The authors express sincere appreciation to the Department of Electrical Engineering at Shivganga Charitable Trust's Vishveshwarya Technical Campus, Patgaon-Miraj, Sangli, Maharashtra, India, for providing the necessary

laboratory space, component kits, and computing resources that made this research possible. We also acknowledge our peers and faculty advisors whose suggestions shaped the design and debugging of the prototype nodes.

VII.REFERENCES

[1] Espressif Systems, ESP32-WROOM-32 Datasheet, Version 3.6, Espressif Systems, 2024. Available: https://documentation.espressif.com/esp32-wroom-32_datasheet_en.html.

[2] A. Banks and R. Gupta, MQTT Version 3.1.1, OASIS Standard, 29 October 2014. Available: https://docs.oasis-open.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.html.

[3] J. Postel, User Datagram Protocol, RFC 768, Internet Engineering Task Force, 1980. Available: https://www.rfc-editor.org/rfc/rfc0768.

[4] I. Fette and A. Melnikov, The WebSocket Protocol, RFC 6455, Internet Engineering Task Force, 2011.

Available: https://datatracker.ietf.org/doc/rfc6455/.

[5] OpenJS Foundation, Express: Node.js web application framework, 2026. Available: https://expressjs.com/.

[6] Vercel, Next.js Documentation, 2026. Available: https://nextjs.org/docs.

[7] Chart.js Contributors, Chart.js: Open source HTML5 charts for your website, 2026. Available: