Design of an MPPT Monitoring System for a Solar Power Plant Based on the Internet of Things Using ESP32 and ThingSpeak

Pujianto[1], M. Zaky Zaim Muhtadi[2*], Miranda Odilia Lambiombir[3]

[1,2,3]Program Studi Teknik Instrumentasi Kilang, Politeknik Energi dan Mineral Akamigas, Indonesia
[*] Corresponding author:
Email: mzakyzm@gmail.com

Abstract.
This study designed and implemented an Internet of Things (IoT)-based telemetry system to monitor electrical parameters in a solar power plant (SPP) installation equipped with a Maximum Power Point Tracking (MPPT) controller. The system was developed using an ESP32 microcontroller connected to an EPever MPPT controller via RS485-based Modbus RTU communication, which then transmits data in real-time over a WiFi network to the ThingSpeak cloud platform for storage, visualization, and remote monitoring. The research method used was descriptive-experimental, covering hardware and software design, communication testing, and operational data acquisition. The observed parameters included input voltage, input current, output voltage, output current, discharge voltage, and discharge current. Test results over the time range of 10:10–17:00—comprising seven sample points with non-uniform intervals—showed a maximum input voltage of 18.35 V with a current of 0.66 A at 13:00, reflecting optimal irradiation conditions, a maximum output voltage of 12.27 V, and relatively stable discharge parameters within a voltage range of 12.64–14.45 V and a current range of 0.05–0.19 A, serving as an indicator of the MPPT controller's power regulation effectiveness. Analysis of the ThingSpeak graph confirmed that the input voltage pattern followed the daily irradiance curve, the dynamic response of the output voltage to irradiance fluctuations, and the identification of transient anomalies requiring further confirmation with an irradiance sensor. These findings indicate that the ESP32- and ThingSpeak-based telemetry system is suitable for use as an effective and economical PV system monitoring solution, with recommendations for further research including continuous data acquisition at fixed intervals, the addition of a pyranometer, and comparative validation with reference measurement instruments.
Keywords: *Internet of Things (IoT), ESP32, MPPT, PV system and ThingSpeak.*

I. INTRODUCTION

The exponential growth in global electricity demand and the limitations of fossil fuel energy sources have driven the adoption of renewable energy technologies, particularly solar power generation systems (SPGS) or photovoltaic (PV) systems. SPGS offer a sustainable solution to meet energy demand while reducing environmental impacts such as greenhouse gas emissions and climate change. In an effort to maximize the energy output of PV systems, the Maximum Power Point Tracking (MPPT) algorithm plays a central role, and various studies indicate that MPPT techniques continue to advance, ranging from conventional approaches to those based on artificial intelligence [1], [2], [3].

Although widely implemented across various sectors, existing PV systems still face limitations in performance optimization due to conventional monitoring systems that typically operate locally, thereby restricting access to real-time data and remote management capabilities. One of the main bottlenecks in PV systems is suboptimal system design and the lack of effective power monitoring mechanisms to determine the amount of power generated and consumed. ESP32- and IoT-based monitoring systems have proven capable of addressing these limitations by providing remote access to voltage, current, temperature, and power parameters simultaneously.

Conventional monitoring systems generally can only perform local observations with limited range, thereby reducing operational efficiency and responsiveness to changes in system conditions. Various studies indicate that IoT-based monitoring systems developed with low-cost components can acquire and transmit electrical parameter data in real-time to a cloud platform, while also providing estimates of PV system performance. Advances in IoT technology have opened opportunities to develop monitoring systems accessible online, enabling operators to perform remote monitoring and diagnostics via web-based applications or dashboards.

A Data Acquisition System (DAS) is a critical component of IoT monitoring infrastructure as it serves as the interface between the analog domain (such as sensors and transducers) and the digital domain (data processing and storage units). Telemetry, as a technique for measuring parameters based on remote data transmission (whether via wired or wireless means) offers a solution to simplify the process of continuous measurement, monitoring, and data collection. A telemetry system for a solar power plant requires key components such as sensors to detect electrical parameters, microcontroller-based transmitters, wireless transmission channels, and a data reception unit integrated with a cloud computing system.

Low-cost IoT-based monitoring approaches have proven viable across various scales of PV system installations, including off-grid systems in remote locations, where the acquisition of electrical and environmental parameter data can be performed accurately and in real-time via a web-based interface [4], [5]. This research develops an IoT-based telemetry system utilizing the ESP32 microcontroller for monitoring parameters in photovoltaic MPPT installations. The ESP32 was selected due to its compatibility with data acquisition systems, real-time data processing capabilities, and integrated wireless communication functionality that enables data transmission to the cloud platform [6], [7], [8].

The proposed system integrates measurements of DC voltage, output current, and power generated by the MPPT controller, thereby facilitating remote diagnostics and comprehensive performance analysis of the PV system. Furthermore, the use of the Modbus RTU protocol in the MPPT controller offers the advantage of reliable and structured acquisition of electrical parameter data within the energy monitoring system. Thus, the integration of the ESP32, RS485/Modbus RTU communication, and the cloud platform constitutes a relevant approach for building a simple, cost-effective, and easily implementable PV system monitoring system.

This study aims to design and implement an ESP32-based IoT telemetry system capable of continuously reading and recording MPPT electrical parameters, as well as transmitting data in real time to a cloud platform for remote access and analysis. The practical benefits of this research include improved operational efficiency of PV power plants through real-time monitoring, reduced implementation costs by utilizing cost-effective microcontrollers and sensors, and the opening of opportunities for the development of smart energy systems that can be managed remotely. The academic benefits include contributing to the development of PV installation monitoring technology toward sustainable energy infrastructure that can support the national energy transition.

II. METHODS

This study employs a quantitative approach with a descriptive-experimental design to design, implement, and evaluate an Internet of Things (IoT)-based telemetry system on a solar power plant (SPP) installation equipped with a Maximum Power Point Tracking (MPPT) controller. The research was conducted in an electrical engineering laboratory on a controlled-scale SPP installation over a three-month period, covering the system design phase, hardware and software integration, communication testing, data acquisition, and analysis of results. This approach was chosen because it focuses not only on parameter observation but also on data retrieval from the developed system under real operational conditions. The developed system is designed to integrate with a commercial MPPT controller (EPever) via the Modbus RTU communication protocol, then wirelessly transmit data to a cloud platform for remote analysis and visualization. The use of an IoT-based monitoring system in solar power plants is relevant for improving real-time data accessibility, operational efficiency, and remote diagnostic capabilities in modern solar energy systems.

Research Type and Design

This study is a quantitative study with a descriptive-experimental design. This design was chosen because the study aims to develop a telemetry system that simultaneously tests the system's ability to read, record, and transmit real-time electrical parameter data from the MPPT controller. The research stages are divided into four main phases, namely: (1) system design and planning, (2) hardware and software implementation, (3) functional testing and data communication validation, and (4) analysis and interpretation of measurement data.

This design enables a comprehensive evaluation of reading consistency, Modbus RTU communication stability, the success of WiFi-based data transmission, and the reliability of data storage on the cloud platform. This type of monitoring model has been widely recommended in the development of smart energy systems because it supports flexible and sustainable monitoring of photovoltaic systems, including through an open-source approach based on HTTP and TCP/IP protocols that has proven to be economical and efficient.

Research Object and Location

The research subject is a PV system with an EPever MPPT controller as the power control center. The system consists of a 20 Wp polycrystalline solar panel, a 12 V/7 Ah storage battery, a 20 A MPPT controller, an ESP32 microcontroller, a TTL-to-RS485 module, and a lamp load. The research was conducted in an electrical engineering laboratory with a controlled testing environment to minimize the influence of external variables, such as installation fluctuations, network interference, and measurement instability. The laboratory environment was selected so that the testing process could be replicated under relatively uniform conditions.

The ESP32 was used as the primary unit for data acquisition and wireless communication due to its real-time processing capabilities, WiFi integration, and compatibility with various sensors and industrial communication protocols. Meanwhile, the EPever MPPT controller is used as the primary data source because it supports Modbus RTU communication, which is commonly implemented in small- to medium-scale PV monitoring systems [6],[14],[16]. The selection of MPPT technology in a PV system is a critical aspect that directly affects the efficiency of energy extraction from solar panels.

Research Equipment, Materials, and Instruments

The devices and instruments used in this study are presented inTable 1.

Table 1. Hardware Component Specifications of the Telemetry System

Component	**Specifications**	**Function in the System**
Solar Panel	20Wp Polycrystalline, Voc: 22V, Isc: 1.4 A	Primary energy source; input voltage/current parameters are monitored
MPPT Controller	EPever 20A, 12V/24V DC, Modbus RTU communication	Optimal power tracking, data transmission to ESP32 via RS485
ESP32 microcontroller	Dual-core 240MHz, 520KB RAM, WiFi 802.11b/g/n	Data acquisition unit, processor, cloud communication gateway
TTL to RS485 module	MAX485-based converter, operating voltage 3–30 Vdc	Communication interface between the ESP32 UART and the RS485 bus
Storage battery	Lead-acid/VLRA, 12V/7Ah nominal capacity	Energy storage, voltage/discharge current monitoring

Research support instruments include a digital multimeter for measurement verification, Arduino IDE software for ESP32 firmware programming, the ThingSpeak cloud platform for data visualization, and a web application on a PC as the user interface.

Telemetry System Design

The telemetry system is designed to integrate data readings from the MPPT controller with real-time data transmission to a cloud server via a WiFi network, as shown in Fig.1. The main communication path utilizes the Modbus RTU protocol via a TTL-to-RS485 converter that connects the ESP32 to the MPPT controller, where the ESP32 reads electrical data registers from the controller and processes observed parameters (such as voltage and current on the input, output, and discharge sides). The selection of the Modbus RTU protocol is based on its ability to accurately acquire electrical parameter data (including voltage, current, power, and frequency) in a one-master-multi-slave architecture on a bus network in real-time [10], [11].

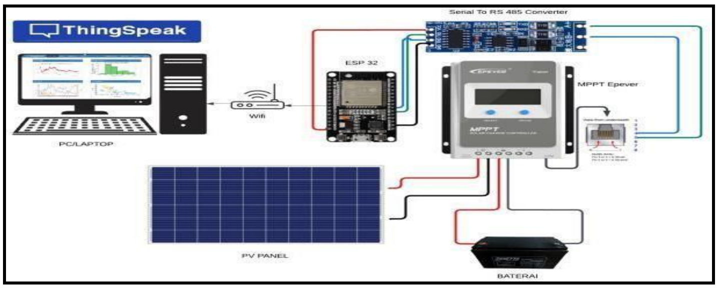


Fig.1. Block Diagram of the ESP32-Based MPPT Telemetry System

Technically, the ESP32's Serial pins are used for Modbus RTU communication. Meanwhile, analog sensor readings are taken via the ESP32's ADC pins. This integration enables the reading of key electrical parameters, such as panel input voltage, input current, battery voltage, charging current, and load discharge current, and then transmits them to a cloud-based monitoring dashboard. This system design aligns with modern data acquisition and telemetry concepts that emphasize measurement efficiency, long-distance communication, and continuous monitoring.

Research Stages

This research procedure was carried out through systematic stages, beginning with the design of the telemetry system's hardware and software architecture, including the selection of the ESP32, MPPT controller, sensors, RS485 module, and the ThingSpeak cloud platform. This real-time monitoring approach, which integrates an IoT module as a bridge between the PV system hardware and the cloud platform, has proven capable of displaying panel, battery, load, and environmental parameter data in an integrated manner on a single visualization dashboard.

The next step is assembling all the hardware into a single monitoring panel system, accompanied by firmware development using the Arduino IDE based on Modbus and Wi-Fi libraries for data processing and transmission to the cloud. Once the devices are ready, connectivity testing is performed to verify Modbus RTU communication and Wi-Fi network connectivity. The system is then operated under operational conditions for real-time acquisition of electrical and environmental data via ThingSpeak. In the final stage, data consistency analysis is performed by observing the alignment of reading patterns with the theoretically known operational characteristics of the photovoltaic system, such as the relationship between radiation intensity and the voltage and current of the solar panels.

III. RESULT AND DISCUSSION

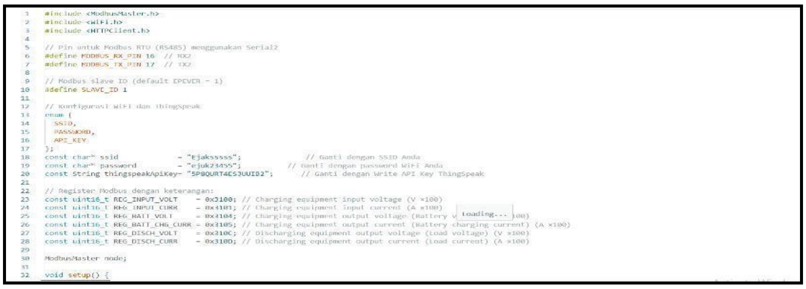
This research resulted in an ESP32-based Maximum Power Point Tracking (MPPT) monitoring system capable of acquiring, processing, and transmitting electrical parameter data in real-time to the ThingSpeak cloud platform. The research results are presented in two main subsections: the system implementation results (hardware and software) and the analysis of operational measurement data.

System Implementation

The hardware implementation resulted in a monitoring system integrated into a single compact control panel, consisting of an EPever MPPT controller, an ESP32 microcontroller, a TTL-to-RS485 module, and a buck converter to regulate the ESP32's supply voltage. The MPPT controller manages the power flow from the solar panel to the battery, while the ESP32 serves as the central hub for data acquisition, signal processing, and information transmission to the cloud platform. The components are organized within the panel enclosure, with systematic cable routing and a cooling fan added at the top to maintain thermal stability during continuous operation. The physical implementation of the system, including the PV panel and panel box along with the connections between components, is shown in Fig. 2.

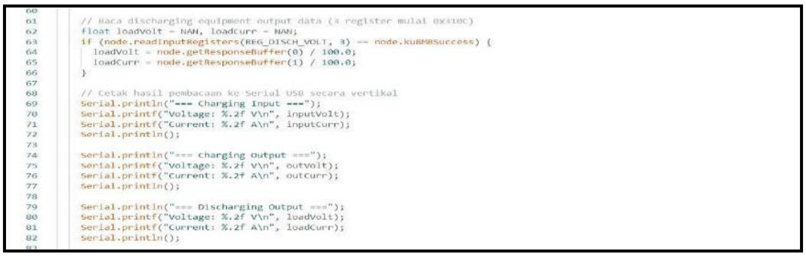
Fig. 2. Design results: (a) side view, (b) panel box assembly

On the software side, the ESP32 firmware was developed using the Arduino IDE, utilizing the Modbus RTU and WiFi libraries. The software workflow includes serial communication initialization, WiFi connection, reading Modbus registers from the MPPT controller, data processing, and periodic data transmission to ThingSpeak via the HTTP protocol using a pre-configured API key. The system reads six main parameters: voltage and current on the panel's input side, battery charging output, and load discharge. Visualizations of the firmware logic structure are shown at Fig. 3 and Fig. 4, which demonstrate the integration of Modbus RTU reading, WiFi connection management, and cloud data transmission within a single repetitive work cycle. The successful Modbus RTU communication and data transmission via WiFi confirm that the system has met the basic requirements as a functional IoT telemetry platform.



```
#include <ModbusMaster.h>
#include <WiFi.h>
#include <HTTPClient.h>

// Pin untuk Modbus RTU (RS485) menggunakan Serial2
#define MODBUS_RX_PIN 16  // RX2
#define MODBUS_TX_PIN 17  // TX2

// Modbus slave ID (default EPEVER = 1)
#define SLAVE_ID 1

// Konfigurasi WiFi dan ThingSpeak
enum {
  SSID,
  PASSWORD,
  API_KEY
};
const char* ssid              = "Ejaksssss";            // Ganti dengan SSID Anda
const char* password          = "ejuk23455";        // Ganti dengan password WiFi Anda
const String thingspeakApiKey= "5P8QURT4ES3UUIB2";      // Ganti dengan Write API Key ThingSpeak

// Register Modbus dengan keterangan:
const uint16_t REG_INPUT_VOLT    = 0x3100; // Charging equipment input voltage (V ×100)
const uint16_t REG_INPUT_CURR    = 0x3101; // Charging equipment input current (A ×100)
const uint16_t REG_BATT_VOLT     = 0x3104; // Charging equipment output voltage (Battery v Loading... 100)
const uint16_t REG_BATT_CHG_CURR = 0x3105; // Charging equipment output current (Battery charging current) (A ×100)
const uint16_t REG_DISCH_VOLT    = 0x310C; // Discharging equipment output voltage (load voltage) (V ×100)
const uint16_t REG_DISCH_CURR    = 0x310D; // Discharging equipment output current (load current) (A ×100)

ModbusMaster node;

void setup() {
```

Fig. 3. Software Design Program



```
    // Baca discharging equipment output data (3 register mulai 0x310C)
    float loadVolt = NAN, loadCurr = NAN;
    if (node.readInputRegisters(REG_DISCH_VOLT, 3) == node.ku8MBSuccess) {
      loadVolt = node.getResponseBuffer(0) / 100.0;
      loadCurr = node.getResponseBuffer(1) / 100.0;
    }

    // Cetak hasil pembacaan ke Serial USB secara vertikal
    Serial.println("=== Charging Input ===");
    Serial.printf("Voltage: %.2f V\n", inputVolt);
    Serial.printf("Current: %.2f A\n", inputCurr);
    Serial.println();

    Serial.println("=== Charging Output ===");
    Serial.printf("Voltage: %.2f V\n", outVolt);
    Serial.printf("Current: %.2f A\n", outCurr);
    Serial.println();

    Serial.println("=== Discharging Output ===");
    Serial.printf("Voltage: %.2f V\n", loadVolt);
    Serial.printf("Current: %.2f A\n", loadCurr);
    Serial.println();
```

Fig. 4. Advanced Program

Analysis of Operational Measurement Data

Data collection was conducted between 10:10 AM and 5:00 PM to evaluate the monitoring system's performance under real-world operational conditions. The observed parameters included input voltage, input current, output voltage, output current, discharge voltage, and discharge current, as summarized in Table 2. Data were recorded in real-time via the ThingSpeak platform and analyzed based on three aspects: daily input voltage profiles, output voltage response to irradiation conditions, and discharge parameter stability.

Table 2. Real-time system measurement data for the time period 10:10 – 17:00

No	Time	Input Voltage	Input Current	Output Voltage	Output Current	Discharge Voltage	Discharge Current
1	10:10 AM	15.21 V	0.51 A	7.87 V	0.02 A	14.07 V	0.16 A
2	11:10 AM	14.02 V	0.25 A	3.64 V	0.20 A	12.99 V	0.19 A
3	1:00 PM	18.35 V	0.66 A	12.27 V	0.20 A	14.45 V	0.17 A
4	2:00 PM	13.90 V	0.15 A	2.19 V	0.10 A	12.93 V	0.08 A
5	3:00 PM	13.69 V	0.7 A	11.02 V	0.10 A	13.02 V	0.09 A
6	4:00 PM	14.03 V	0.23 A	0.12 V	0.10 A	12.64 V	0.05 A
7	5:00 PM	13.48 V	0.00 A	0.12 V	0.10 A	12.64 V	0.07 A

Referring to the data in the Table 2, the solar panel input voltage shows a fluctuating pattern that generally follows the daily solar irradiance curve. The highest value was recorded at 1:00 PM at 18.35 V with an input current of 0.66 A, indicating that solar radiation intensity was at its optimal level during that period. This is confirmed by the ThingSpeak graph atFig. 5, which shows the input voltage fluctuating within the range of 18–20 V around 12:50–13:00, with most measurement points concentrated in the 19–20 V range, while lower values (15–16 V) only appear at the beginning and end of that period. The momentary fluctuations visible in the graph reflect variations in light intensity due to changing environmental conditions, such as cloud movement or shifts in the angle of incidence of sunlight.

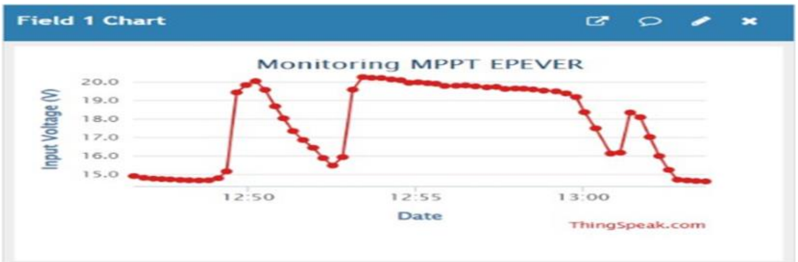


Fig. 5. Input Voltage Graph (Field 1) at 1:00 PM

In contrast, the graph at Fig. 6 shows that in the period leading up to 5:00 PM, the input voltage tended to stabilize in the range of 13.5 V with very little variation, except for a single significant spike to around 15.5 V at approximately 4:55 PM before dropping back down. This spike was likely caused by a transient irradiance reflection or a transient disturbance in the sensor reading; however, given the lack of irradiance sensor (pyranometer) data in this study, further experimental confirmation is strongly recommended. In general, this flat pattern contrasts sharply with the fluctuating profile during the day and indicates that solar radiation intensity has decreased significantly, causing the panel to operate near open-circuit voltage without a significant current contribution, as confirmed by input current data reaching 0.00 A at 5:00 PM. This condition aligns with the operational characteristics of photovoltaic panels, where energy output is highly dependent on the angle of incidence of sunlight and irradiance intensity [1], [2].

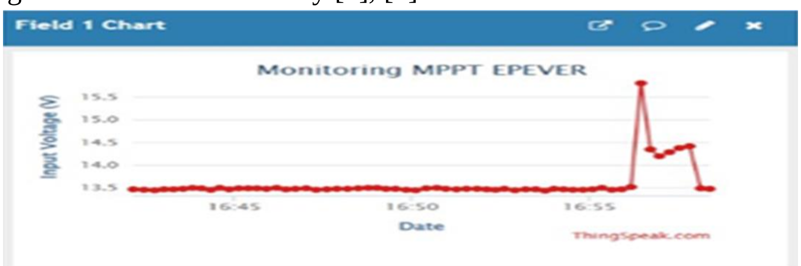


Fig. 6. Input Voltage Graph (Field 1) at 5:00 PM

Referring to the output voltage column in Table 2, the voltage on the battery charging side shows a sensitive response to changes in irradiation conditions throughout the day. The ThingSpeak graph atFig. 7 shows that between 12:50 PM and 1:00 PM, the output voltage exhibits a brief spike to the 10–15 V range amidst periods predominantly below 10 V, with the highest peak value recorded in the table at 12.27 V as it approaches 1:00 PM. This pattern reflects the response of the MPPT algorithm, which actively adjusts the operating point to maximize power transfer from the panel to the battery when energy is available under optimal conditions.

Conversely, the graph Fig. 8 shows that between 3:15 PM and 3:45 PM, the output voltage dropped drastically to consistently near 0–5 V, indicating that the energy available for transfer from the panel to the battery was severely limited, effectively halting the charging process during that period. This condition aligns with the data at 4:00 PM, which recorded an output voltage of only 0.12 V. This decline pattern is a normal response of a PV system with MPPT control, where the controller automatically adjusts the operating point based on available energy conditions [10], [11].

Additionally, there are two anomalies worth noting. First, at 11:10 AM, the output voltage was recorded at only 3.64 V even though the input voltage was still 14.02 V; this condition was likely caused by the battery nearing full capacity, prompting the MPPT controller to automatically reduce the charging voltage to prevent overcharging. Second, at 14:00, the output voltage dropped sharply to 2.19 V, which can be explained by the possibility of transient cloud shading or the MPPT algorithm's response to a sudden load change. Both explanations are inferential, given that this study was not equipped with an irradiance sensor (pyranometer) or simultaneous battery status recording; therefore, experimental confirmation through further testing is highly recommended.

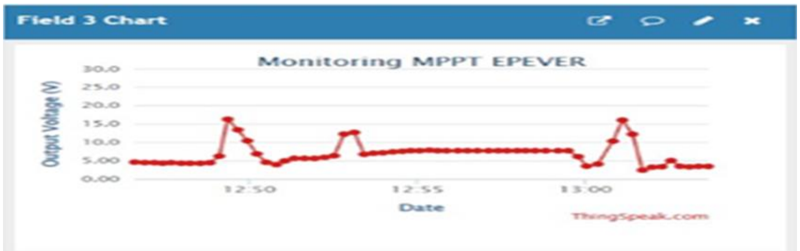


Fig. 7 Output Voltage Graph (Field 3) around 1:00 PM

Fig. 8. Output Voltage Graph (Field 3) between 3:15 PM and 3:45 PM

Unlike the previous two parameters, the data in Table 2 in the discharge voltage and current columns show a much more stable profile throughout the entire observation period. The discharge voltage is in the range of 12.64–14.45 V, while the discharge current is in the range of 0.05–0.19 A. This stability is visually confirmed in Fig. 9, where the discharge voltage graph is observed to remain consistently flat in the 12–14 V range between 15:15 and 15:45, even though the input and output voltages experienced significant variations during the same period. This condition indicates that the MPPT controller successfully performed its power regulation function effectively, namely maintaining a steady power supply to the load even when the primary energy source was not in optimal condition. In a PV system, the ability to maintain discharge-side stability is a key indicator that the power storage and control system is operating reliably [11], [12].

Fig. 9. Discharge Voltage Graph (Field 5) at 3:00 PM

Overall, the data patterns obtained from Table 2 are consistent with the operational characteristics of photovoltaic systems reported in the literature, where input voltage and current follow the daily irradiance curve while battery discharge voltage tends to remain stable thanks to the MPPT controller's regulation function. The use of ThingSpeak as a storage and visualization platform provides significant added value because its historical data storage capability supports long-term performance analysis, while field-based graphical visualization has proven to intuitively represent the dynamics of electrical parameters and is easy to interpret, as shown in Fig. 5 through Fig. 9. This approach opens up opportunities for further development, such as for early fault detection and integration with more adaptive IoT-based energy management systems [12],[13], [14]. As a limitation, the number of data points in this study (seven points over a seven-hour span with non-uniform intervals) limits the depth of temporal analysis; therefore, in future research, continuous data collection at fixed intervals of every 15 or 30 minutes is highly recommended to enable a more comprehensive analysis of the system's daily patterns. In addition, a quantitative accuracy evaluation through direct comparison between the system's readings and those of a reference measuring instrument (multimeter or standardized data logger) should also be conducted as a comparative validation before the system is implemented on a field scale.

IV. CONCLUSION

This study successfully designed and implemented an Internet of Things (IoT)-based telemetry system using an ESP32 microcontroller for monitoring electrical parameters in a solar power plant (SPP) installation with an EPever MPPT controller, where data is acquired via RS485-based Modbus RTU communication and transmitted in real-time to the ThingSpeak cloud platform for storage, visualization, and remote monitoring. Operational testing during the 10:10–17:00 timeframe showed that the system was able to record a maximum input voltage of 18.35 V with a current of 0.66 A at 13:00, reflecting optimal irradiation conditions, a maximum output voltage of 12.27 V, as well as relatively stable discharge parameters within a voltage range of 12.64–14.45 V and a current range of 0.05–0.19 A, indicating the effectiveness of the MPPT controller's power regulation function. Analysis of the ThingSpeak graph confirms that the input voltage pattern follows the daily irradiance curve, the output voltage responds dynamically to irradiance fluctuations, and there are transient anomalies at some measurement points that are inferential in nature and require further confirmation with an irradiance sensor (pyranometer).

Thus, the developed system has proven viable as an effective and economical PV monitoring solution; however, limitations such as the small number of data points with non-uniform intervals and the lack of comparative validation with reference measurement instruments are recommended as topics for further research, along with the addition of supporting sensors and continuous data acquisition at fixed intervals of 15–30 minutes for a more comprehensive analysis of daily patterns.

REFERENCES

[1] O. Boubaker, "MPPT techniques for photovoltaic systems: a systematic review in current trends and recent advances in artificial intelligence," Discover Energy, vol. 3, no. 1, p. 9, 2023.

[2] M. L. Katche, A. B. Makokha, S. O. Zachary, and M. S. Adaramola, "A comprehensive review of maximum power point tracking (mppt) techniques used in solar pv systems," Energies, vol. 16, no. 5, p. 2206, 2023.

[3] N. Karami, N. Moubayed, and R. Outbib, "General review and classification of different MPPT Techniques," Renewable and sustainable energy Reviews, vol. 68, pp. 1–18, 2017.

[4] N. J. Beltrán Castañón, F. Chura Acero, J. Ramos Cutipa, O. Chayña Velásquez, H. Shuta Lloclla, and E. Cruz Ticona, "A Low-Cost Wireless Monitoring System for Photovoltaic Systems: Performance Analysis and Potential Application in Direct-Current Nanogrids," Energies, vol. 18, no. 9, p. 2279, 2025.

[5] A. Hamied, A. Mellit, M. Benghanem, and S. Boubaker, "IoT-based low-cost photovoltaic monitoring for a greenhouse farm in an arid region," Energies, vol. 16, no. 9, p. 3860, 2023.

[6] J. S. Tutuko, A. Prawoto, D. Ratnaningsih, A. E. Kristiyono, and S. Dai Robbi, "Air Compressor Design and Construction With Esp32 Based Microcontroller Monitoring System," ***International Journal of Science and Environment (IJSE)***, vol. 6, no. 1, pp. 1134–1140, 2026.

[7] D. Hercog, T. Lerher, M. Truntič, and O. Težak, "Design and implementation of ESP32-based IoT devices," Sensors, vol. 23, no. 15, p. 6739, 2023.

[8] T. Lavenia and F. R. Saputri, "Blynk app-based plant monitoring system design," ***International Journal of Science and Environment (IJSE)***, vol. 3, no. 4, pp. 145–150, 2023.

[9] M. Ula and A. Rahmadani, "Rancang Bangun Maximum Power Point Tracking pada Panel Surya dengan Metode Incremental Conductance Menggunakan Zeta Konverter," Techné: ***Jurnal Ilmiah Elektroteknik***a, vol. 22, no. 1, pp. 1–20, 2023.

[10] C. Urrea and C. Morales, "Enhancing Modbus-RTU communications for smart metering in building energy management systems," Security and Communication Networks, vol. 2019, no. 1, p. 7010717, 2019.

[11] W. Khalid, M. Jamil, A. A. Khan, and Q. Awais, "Open-source internet of things-based supervisory control and data acquisition system for photovoltaic monitoring and control using HTTP and TCP/IP protocols," Energies, vol. 17, no. 16, p. 4083, 2024.

[12] N. Rouibah et al., "Smart monitoring of photovoltaic energy systems: An IoT-based prototype approach," Scientific African, p. e02973, 2025.

[13] E. F. Orumwense and K. Abo-Al-Ez, "Internet of Things for smart energy systems: A review on its applications, challenges and future trends," AIMS Electronics and Electrical Engineering, vol. 7, no. 1, pp. 50–74, 2023, doi: 10.3934/electreng.2023004.

[14] S. Ansari, A. Ayob, M. S. H. Lipu, M. H. M. Saad, and A. Hussain, "A review of monitoring technologies for solar PV systems using data processing modules and transmission protocols: Progress, challenges and prospects," Sustainability, vol. 13, no. 15, p. 8120, 2021.