ESP32 and MIT App Inventor-Based VRLA Battery Telemetry Monitoring System for Preventive Maintenance

Hafizuddin Umar Siregar, Adi Chandranata,*, Efrizon , Nabilatul Khamsyah MB

Department of Electrical Engineering, Politeknik Negeri Padang, Padang, 25164, Indonesia

Corresponding author: adichandra@pnp.ac.id

***Abstract*— Manual monitoring of Valve Regulated Lead Acid (VRLA) batteries in industrial electrical systems is considered inefficient and risks delayed fault detection. This study aims to design an Internet of Things (IoT)-based telemetry monitoring system to automate the process. The system comprises a PZEM-016 sensor for reading battery voltage, an ESP32 microcontroller as the data processor, Firebase as the cloud database, and an Android application built with MIT App Inventor as the user interface. Voltage data is sent in real-time from the sensor to the ESP32 via Modbus RTU communication through an RS485 to TTL converter, then forwarded to Firebase via Wi-Fi, and finally displayed on the application. Test results show a sensor reading accuracy of 99.60% with an average error of only 0.40%. The system successfully transformed the monitoring method from a manual inspection taking approximately 10 minutes per unit to real-time automatic monitoring. Therefore, the designed system is proven effective for remote monitoring, supports preventive maintenance strategies, and enhances the operational reliability of VRLA batteries in industrial environments.**

***Keywords*— Firebase, Internet of Things, MIT App Inventor, Preventive Maintenance, Real-time Monitoring, Valve Regulated Lead Acid Battery**

Manuscript received 19 Jan. 2026; revised 25 Apr. 2026; accepted 31 May. 2026. Date of publication 30 June. 2026.

I. INTRODUCTION

Valve Regulated Lead Acid (VRLA) batteries are critical backup power sources in industrial power distribution systems, such as those used at PT Krakatau Chandra Energi (KCE) [1]. The reliability of the electrical system heavily depends on the condition of these batteries. However, field monitoring is still frequently conducted manually by technicians by measuring the voltage of each battery cell one by one. This method is considered inefficient because it takes approximately 10 minutes per unit, potentially leading to delayed fault detection and exhausting human resources. Therefore, an automated monitoring system is required to provide real-time battery condition information that is accessible remotely.

Several previous studies have developed Internet of Things (IoT)-based monitoring systems [2][3]. These studies designed wireless battery monitoring systems based on ESP32 and Firebase, demonstrating the potential of cloud integration for monitoring. On the other hand, [4] utilized MIT App Inventor to build remote control interfaces. Nevertheless, the implementation of telemetry systems specifically integrating the PZEM-016 sensor with Modbus RTU (RS485) communication for reading industrial battery voltages, combined with the Firebase platform and MIT App Inventor interface in a single solution applied to real corporate needs, remains limited.

This research aims to fill this gap by designing and constructing a fully integrated telemetry system, ranging from accurate voltage data acquisition and real-time data transmission to a user-friendly visualization, tailored to the specific requirements of

PT KCE for monitoring batteries on ACDB/DCDB panels. Based on this background, the objectives of this study are: (1) to design and realize an Android-based VRLA battery telemetry monitoring system; (2) to display the measurement results of key parameters such as voltage and estimated battery health in real-time; and (3) to analyze the improvement in monitoring efficiency compared to previous manual methods

II. MATERIALS AND METHOD

This research follows a systematic engineering approach, beginning with system requirement analysis, hardware and software design, followed by implementation and testing. The system is designed to monitor the voltage of VRLA batteries used in the ACDB/DCDB panels at PT Krakatau Chandra Energi.

A. System Architecture

The telemetry system is divided into three main layers: the perception layer (sensors), the network layer (ESP32 and Firebase), and the application layer (Android App). The overall system workflow is illustrated in the block diagram in Fig. 1.

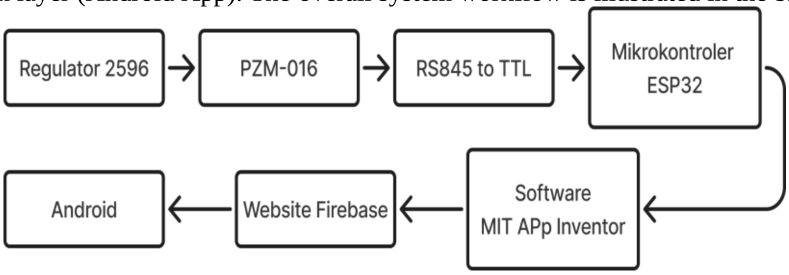

FIGURE. 1. Block diagram of the VRLA battery monitoring system

The PZEM-016 sensor reads the battery voltage data. Since the sensor uses the Modbus RTU protocol, an RS485 to TTL converter is utilized to enable communication with the ESP32 microcontroller [5]. The ESP32 processes the data and transmits it to the Firebase Realtime Database via a Wi-Fi connection.

B. Hardware Components and Specifications

The microcontroller unit in this system is an ESP32 development board, which functions as the central processing unit. It features a dual-core 160MHz processor, integrated Wi-Fi (802.11 b/g/n), and multiple GPIO interfaces [6]. Operating at a 3.3V logic level, the ESP32 is connected to the RS485 converter through its UART2 pins, specifically GPIO16 (RX) and GPIO17 (TX) [7]. For voltage sensing, two PZEM-016 modules are employed, each capable of measuring voltages within a range of 80–260V and communicating via the Modbus RTU protocol [8]. Communication between the sensors and microcontroller is facilitated by a MAX485-based RS485-to-TTL converter, which enables reliable differential serial data transmission [9]. Power management is handled by an LM2596 buck converter that regulates the input voltage to a stable 5V DC for sensor operation [10]. The main system is powered by a separate 5V/10A switching power supply [11], while circuit protection is provided by a Schneider Electric EZ9F54110 miniature circuit breaker (MCB) rated at 10A, 1P [12].

C. Software Implementation

Firmware development was carried out using the Arduino IDE with the ESP32 board package [13]. The program implements the Modbus RTU protocol through the ModbusMaster library, enabling communication with two PZEM-016 sensors configured with slave IDs 1 and 2 [14]. The Modbus RTU communication operates at 9600 baud with a configuration of 8 data bits, no parity, and 2 stop bits (8N2) [15]. For cloud integration, voltage data is sent to a Firebase Realtime Database, where it is stored in a JSON structure that includes sensor readings and system status indicators [16]. The user interface is an Android application named "Voltage & Health Monitoring," developed using the MIT App Inventor 2 platform. This application provides real-time data display, classifies battery health, offers graphical visualization, and includes manual control functions [17][18][19][20].

D. Testing and Calibration Method

To ensure data accuracy, the PZEM-016 sensor underwent a calibration process using a Fluke 117 Multimeter as a standard reference. Testing was conducted by varying the input voltage using an LM2596 DC-DC buck converter to simulate battery voltage levels ranging from 1.8V to 2.5V. The accuracy is calculated using the following error percentage formula.1.

$$Error(\%) = \frac{Measured\ Value - \ Reference\ Value}{Reference\ Value} \times 100\%$$

III. RESULTS AND DISCUSSION

This section presents the comprehensive results of the VRLA battery telemetry system implementation, including hardware communication tests, cloud integration, and user interface performance.

A. Hardware Implementation and Sensor Accuracy

The initial testing focused on the data acquisition layer. The communication between the PZEM-016 sensor and the ESP32 was facilitated by an RS485 to TTL converter. The activity of this communication link was verified through the LED indicators on the converter module, which signals successful data polling via the Modbus RTU protocol.

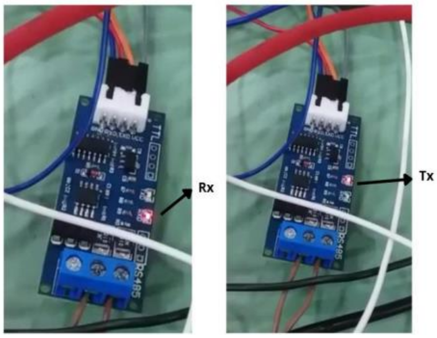

FIGURE. 2. RS485 to TTL communication activity

Furthermore, the accuracy of the PZEM-016 sensor was validated using a Fluke 117 Multimeter as a reference. The calibration results, ranging from 1.8V to 2.5V, are detailed in Table I.

TABLE I
Voltage Measurement and Accuracy Analysis

No	Reference Voltage	Sensor 1	Sensor 2	Error (%)
1	1.80V	1.80V	1.79V	0.28
2	1.85V	1.84V	1.86V	0.54
3	1.90V	1.91V	1.90V	0.26
4	1.95V	1.94V	1.95V	0.26
5	2.00V	2.01V	2.01V	0.50
6	2.05V	2.04V	2.05V	0.25
7	2.10V	2.11V	2.09V	0.48
8	2.15V	2.16V	2.14V	0.47
9	2.20V	2.21V	2.22V	0.68
10	2.25V	2.24V	2.25V	0.22
11	2.30V	2.29V	2.30V	0.22
12	2.35V	2.36V	2.35V	0.22
13	2.40V	2.41V	2.40V	0.21
14	2.45V	2.46V	2.44V	0.41
15	2.50V	2.50V	2.49V	0.20

The data show an average error of 0.40% (99.60% accuracy), confirming the system's reliability for precision monitoring in industrial environments.

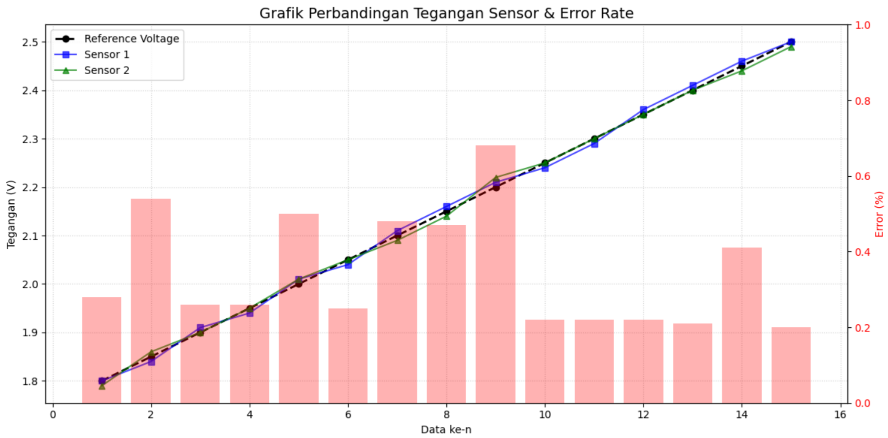

FIGURE. 3 Sensor Voltage Calibration and Error Distribution Plot

The experimental results conducted across 15 voltage reference points (ranging from 1.80V to 2.50V) demonstrate high precision in the monitoring system. Data analysis reveals that the PZEM-016 sensor readings yield an average error of only 0.35%, with the maximum error observed at 0.68% (at 2.20V) and the minimum error reaching 0.20% (at 2.50V). The consistency between Sensor 1 and Sensor 2 compared to the reference voltage confirms that the Modbus RTU protocol integration on the ESP32 effectively minimizes noise and data degradation. Consequently, these results validate the system's reliability for industrial applications requiring stringent accuracy in VRLA battery monitoring to support robust preventive maintenance strategies.

B. Data Integration and Monitoring Interface

The telemetry data processed by the ESP32 was transmitted via Wi-Fi to the Firebase Realtime Database. The database successfully stored and updated the voltage values in a JSON structure, ensuring that the information is ready for retrieval by the mobile application.

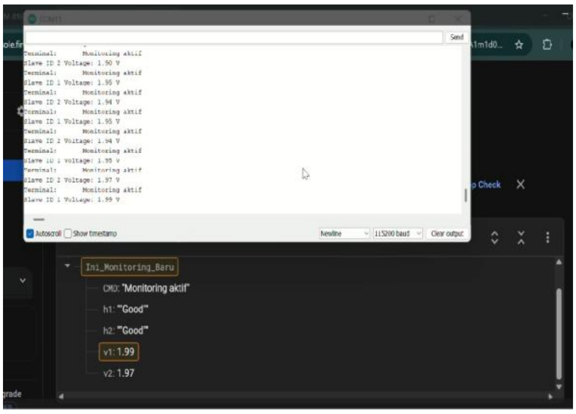

FIGURE. 4 Real-time monitoring interface on Android application

The final layer of the system is the Android-based monitoring application developed using MIT App Inventor. The application provides a visual representation of the battery health status based on predefined thresholds: Normal (>2.1V), Warning (1.9V - 2.1V), and Critical (<1.9V).

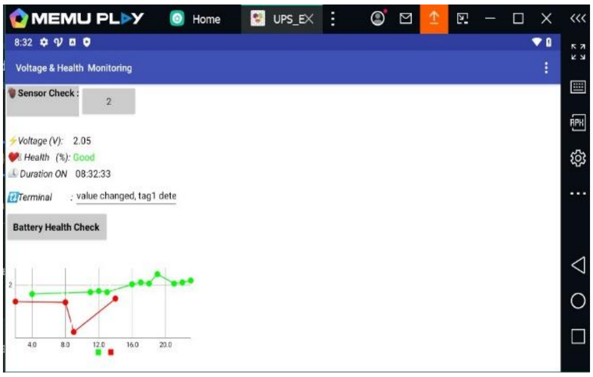

FIGURE. 5 Android application monitoring display

Field tests at PT Krakatau Chandra Energi proved that this automated system provides instantaneous updates, effectively replacing the time-consuming manual measurement process of approximately 10 minutes per unit.

IV. Conclusion

The IoT-based VRLA battery telemetry monitoring system using ESP32 and MIT App Inventor has been successfully developed and implemented. The research results demonstrate that the system achieves high measurement precision with an average error of only 0.40%, or an accuracy rate of 99.60%. The integration of the PZEM-016 sensor via Modbus RTU communication ensures stable data acquisition, while Firebase and the Android application provide an effective real-time remote monitoring interface.

This system significantly improves operational efficiency by transforming the manual inspection process which previously took approximately 10 minutes per unit—into a continuous, automated solution. This enables early detection of battery degradation and supports more effective preventive maintenance strategies at PT Krakatau Chandra Energi. For future development, it is recommended to integrate an automated alarm notification system to alert technicians immediately when a critical threshold is reached and to implement local data logging to ensure data continuity during network interruptions.

Acknowledgment

The authors would like to express their gratitude to the Department of Electrical Engineering at Politeknik Negeri Padang for providing the facilities and support necessary to conduct this research. Special thanks are also extended to PT Krakatau Chandra Energi for granting permission for field testing and for providing the technical data required during the development of this telemetry system. This work would not have been possible without the valuable contributions and guidance from the technical staff and colleagues who supported this project from inception to completion.

References

[1.] G. Alfaseno dan Suryono, “Rancang Bangun Sistem Telemetri Multi-Channel untuk Monitoring Suhu dan pH Air Mengunakan Jaringan Wifi,” *Youngster Physics Journal*, vol. 4, no. 4, pp. 257–264, 2015.
[2.] H. Hilmansyah, et al., “Sistem Monitoring Baterai Nirkabel Rancang Bangun Berbasis ESP32,” *Prosiding SNITT POLTEKBA*, vol. 4, pp. 194–199, 2020.
[3.] R. Susanti et al., "Identifikasi Warna Adonan Kerupuk Labusiam Menggunakan Sensor Warna Berbasis IoT," *Elektron: Jurnal Ilmiah*, pp. 102-109, 2024.
[4.] H. Kusumah dan R. A. Pradana, “Penerapan Trainer Interfacing Mikrokontroler dan Internet of Things Berbasis ESP32 pada Mata Kuliah Interfacing,” *Jurnal Penerapan Teknologi*, vol. 2, no. 1, pp. 45-52, 2019.
[5.] M. K. A. Rosa, R. S. Rinaldi, dan R. Illahi, "Rancang Bangun Prototype Mesin Pemotong Rumput Kendali Jarak Jauh Menggunakan Aplikasi Android," *Jurnal Teknik Elektro Universitas Bengkulu*, vol. 1, no. 1, 2020.
[6.] Peacefair, "PZEM-014/016 AC Communication Module Datasheet," *Peacefair Electronic Technology*, China, 2018.
[7.] S. R. Nayak, "Performance Analysis of ESP32 Microcontroller for IoT Applications," *International Journal of Computer Applications*, 2020.
[8.] Espressif Systems, "ESP32 Series Datasheet v3.9," 2023.
[9.] Peacefair, "PZEM-016 AC Communication Module User Manual," 2018.
[10.] Maxim Integrated, "MAX485: Low-Power, Slew-Rate-Limited Transceivers for RS-485," Datasheet, 2014.
[11.] Texas Instruments, "LM2596 SIMPLE SWITCHER Power Converter 150-kHz 3A Step-Down Voltage Regulator," Datasheet, 2016.
[12.] Mean Well, "LRS-50-5: 50W Single Output Switching Power Supply," Datasheet, 2015.

[13.] Schneider Electric, "Easy9 Miniature Circuit Breaker (MCB) EZ9F Datasheet," 2021.
[14.] M. Banzi and M. Shiloh, *Getting Started with Arduino: The Open Source Electronics Prototyping Platform*, O'Reilly Media, 2014.
[15.] Modbus Organization, "Modbus Application Protocol Specification V1.1b3," 2012.
[16.] K. Shvets, "Real-time Data Processing with Google Firebase," *Journal of Software Engineering and Applications*, 2018.
[17.] Wolber, *App Inventor 2: Create Your Own Android Apps*, O'Reilly Media, 2014.
[18.] Al-Rahamneh, I. Izco, A. Serrano-Hernandez, and J. Faulin, "Machine Learning-Based State-of-Health Estimation of Battery Management Systems Using Experimental and Simulation Data," Electronics, vol. 12, no. 8, p. 1894, Apr. 2023, doi: 10.3390/s23136113.
[19.] E. Petkovski, I. Marri, L. Cristaldi, and M. Faifer, "State of Health Estimation Procedure for Lithium-Ion Batteries Using Partial Discharge Data and Support Vector Regression," Sensors, vol. 23, no. 13, p. 6113, Jun. 2023, doi: 10.3390/electronics12081894.
[20.] Y.-H. Sun, H.-L. Jou, and J.-C. Wu, "Auxiliary diagnosis method for lead–acid battery health based on sample entropy," Journal of Energy Storage, vol. 73, Part A, p. 108999, Dec. 2023, doi: 10.1016/j.est.2023.108999.