Infact: Computer International Journal (ICIJ)
Vol.9, No.02, September 2025
p-ISSN : 2527-8363, e-ISSN: 2829-0259
DOI:

Design and Development of an IoT-Based Monitoring System for Solar Phone Charging Stations

Anita Yuan[1*], Rivaldo Saputra[2], Rutnaria S.S. Sakoikoi[3]
[1]Department of Physics, Faculty of Science and Computer, Universitas Kristen Immanuel
[2]Department of Informatics, Faculty of Science and Computer, Universitas Kristen Immanuel
[3]International Master Program in Engineering, National Changhua University of Education.
[1,2]Jln. Ukrim km 11.1, Purwomartani, Kalasan, Sleman, D.I Yogyakarta, 55571
[3]No.1, Jinde Rd., Changhua City, Changhua County 50007, Taiwan (R.O.C.)

E-mail: yuan@ukrimuniversity.ac.id

Submitted: 09/10/2025 , Revision: 09/16/2025, Accepted : 09/17/2025

Abstract

Solar energy, harnessed through photovoltaic (PV) technology, offers a promising alternative energy source, yet its efficiency and stability require reliable real-time monitoring. This study developed an IoT-based monitoring system for a Solar Phone Charging Station using Arduino Mega 2560 as the central processor, NodeMCU ESP8266 for data transmission, and sensors including a 0–25 V voltage sensor, ACS712 current sensor, and DHT11 temperature-humidity sensor. The system successfully recorded and transmitted data in real time under internet connectivity. However, anomalies were detected, such as 4% of voltage data spiking up to 80 V, exceeding the expected 22 V limit, and instances where zero voltage was recorded while current remained at 22%. Furthermore, comparisons with Wunderground data revealed deviations of 18% for temperature and 45% for humidity. These results emphasize the need for improved calibration to enhance accuracy and system reliability.

Keywords:
Internet of Things;
Monitoring;
Photovoltaic;
Arduino Mega;
ThingSpeak.

***Corresponding Author: Anita Yuan**
E-mail: yuan@ukrimuniversity.ac.id

1. Introduction

Solar panels, or photovoltaics, are capable of converting solar energy into electricity, offering a clean and sustainable alternative to fossil-based energy sources. The use of solar panels as an energy source for various applications continues to increase, including for charging electric vehicles. In this context, photovoltaic-powered charging stations represent a highly relevant solution, especially in today's energy transition era [1].

However, one of the main challenges in implementing photovoltaic technology is the need to monitor and manage the power generated and consumed. Effective monitoring enables better energy management, the identification of potential issues, and ensures that the system operates optimally. With the advancement of Internet of Things (IoT) technology, monitoring can now be performed in real time and with greater accuracy. IoT allows the integration of various sensors and devices that can monitor and control the power output of photovoltaic-based charging stations. By utilizing IoT technology, the collected data can be analyzed to provide useful insights for both users and operators of the charging station [2][3][4].

The implementation of IoT-based monitoring in Solar Home Systems (SHS) has the potential to significantly enhance system performance by enabling early detection of technical issues, facilitating remote supervision, and optimizing maintenance efficiency. Evidence from SHS installations in Sumba demonstrates that, in the absence of systematic monitoring, only 58% of the systems remained operational while the rest failed to function. By employing IoT-based solutions, technicians are able to track the real-time status of batteries, photovoltaic panels, and loads, allowing for rapid response to system malfunctions [5].

Previous studies have been conducted to measure and control household electricity usage by employing sensors to monitor current, voltage, power, and energy [6][7][8]. However, most of these studies focused primarily on monitoring household energy consumption, without direct relevance to renewable energy systems.

Another study by Sari & Away [9] successfully designed a device to monitor the power factor of household appliances used in an online PV On-Grid system. Nevertheless, the observed parameters were limited to electrical characteristics of household appliances and did not account for environmental aspects such as temperature and humidity, which significantly influence solar panel electricity production.

Therefore, this research aims to design and implement the development of an IoT-based monitoring system for a solar phone charging station, to develop a real-time accessible data visualization platform for monitoring solar panel performance, and to analyze the obtained data in order to provide recommendations for system optimization and maintenance.

2. Methods

The research method employed in the development of the IoT-based Solar Phone Charging Station Monitoring System follows an experimental approach consisting of several stages. The study begins with a literature review to obtain theoretical foundations and relevant references related to photovoltaic (PV) technology, monitoring systems, and the Internet of Things (IoT). This is followed by the system design stage, which involves preparing the technical design for both hardware and software. The next stage is implementation, where the design is realized into a functional prototype. The prototype is then subjected to a series of tests to ensure its performance and functionality in accordance with the specifications. The test results are subsequently analyzed to evaluate the effectiveness of the system. Finally, the entire process and research findings are documented systematically in the form of a research report.

2.1. Photovoltaic (PV)

Photovoltaic (PV) is a technology that converts solar energy into electrical energy through the photovoltaic effect. A PV system typically consists of a number of solar cells integrated into a solar module, which can be connected in series or parallel to generate electrical power [10].

One of the implementations of photovoltaic technology at Universitas Kristen Immanuel (UKRIM) is the Solar Phone Charging Station. This station is designed to provide solar-based electrical energy for charging electronic devices. It is equipped with main components including a 120 Wp photovoltaic module, charge controller, 33 Ah storage battery, 300 W inverter, AC and DC charging ports, and two 10 W LED lamps, as illustrated in Figure 1.

Figure 1. Solar Phone Charging Station UKRIM

2.2. Sensor and Microcontroller

In the current monitoring system of an IoT-based photovoltaic charger station, the use of microcontrollers and sensors is essential for collecting data related to system performance. The sensors employed, such as the voltage sensor, current sensor, and temperature sensor, play a crucial role in ensuring the optimal operation of the system.

2.2.1. Voltage Sensor 0 – 25V

The 0–25V Voltage Sensor, as shown in Figure 2 below, is a sensor module designed to measure DC voltage within the range of 0 to 25 volts.

Figure 2. Voltage Sensor 0-25V

2.2.2. Acs712 Current Sensor

The ACS712 current sensor, as illustrated in Figure 3 below, is a Hall-effect–based device designed to measure both AC and DC currents with high accuracy. This sensor is capable of measuring currents up to 30 amperes and provides a voltage output proportional to the measured current value, thereby facilitating integration with microcontrollers such as Arduino. The ACS712 is equipped with galvanic isolation, which ensures safety in high-current measurements without posing a risk of damage to electronic components [11].

Figure 3. Acs712 Current Sensor

2.2.3. DHT11 Temperature Sensor

The DHT11, as shown in Figure 4, is a digital temperature and humidity sensor used to measure air temperature within the range of 0 to 50 °C with an accuracy of ±2 °C, and relative humidity between 20% and 90% with an accuracy of ±5%. This sensor employs digital technology to provide easily readable output data, making it suitable for applications requiring environmental monitoring, such as room temperature control systems or Internet of Things (IoT) applications [12].

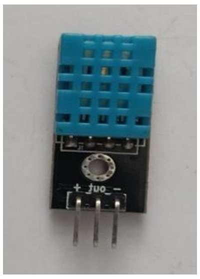

Figure 4. DHT11 Temperature Sensor

2.2.4. Arduino Mega 2560

The Arduino Mega 2560, as shown in Figure 5 below, is a microcontroller board based on the ATmega2560, designed for projects that require numerous inputs and outputs. With 54 digital I/O pins, 16 analog inputs, and 4 serial ports (UART), the Arduino Mega 2560 is capable of handling a large number of sensors and actuators simultaneously, making it ideal for complex applications such as robotics, home automation, and Internet of Things (IoT) projects. The board is also equipped with a USB connection for programming and can operate both online and offline. Another advantage lies in its larger memory capacity compared to other Arduino models, which allows developers to run more complex programs and store greater amounts of data [1][13].

Figure 5. Arduino Mega 2560

2.2.5. NodeMCU Esp8266

The NodeMCU ESP8266, as shown in Figure 6 below, is an open-source development platform that integrates the ESP8266 Wi-Fi module with firmware based on the Lua programming language. Designed for Internet of Things (IoT) applications, NodeMCU enables users to easily connect devices to the internet and control them through a network. With its compact size and efficient data processing capability, the NodeMCU is equipped with multiple I/O pins that can be used to interface with sensors and actuators. Its main advantages are ease of programming and connectivity, making it highly popular among developers and hobbyists for creating IoT-based projects [14].

Figure 6. NodeMCU Esp8266

2.2.6. ThingSpeak Software

ThingSpeak is a cloud-based platform used to collect, store, and analyze data from IoT devices, as illustrated in Figure 2.8. According to the study conducted by Badawi & Harahap (2023), ThingSpeak functions as an interface for displaying real-time voltage and temperature data transmitted from an Arduino-based monitoring system. By utilizing ThingSpeak, information obtained from distribution substations can be accessed and monitored directly via the internet, thereby facilitating management in decision-making processes related to the efficiency and reliability of electrical systems [6][15].

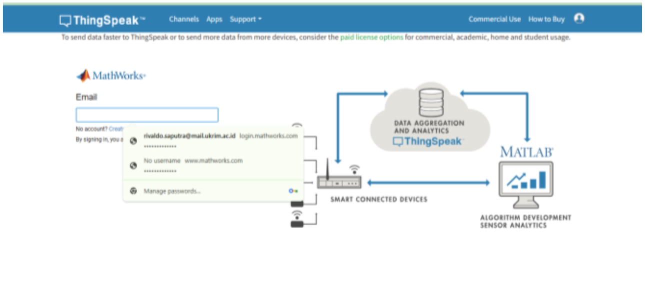

Figure 7. ThingSpeak

2.3. System Design

2.3.1. Hardware Architecture

The hardware architecture of the photovoltaic solar phone charging station monitoring system, as shown in Figure 8 below, is designed to read, process, and transmit data from the sensors to the Internet of Things (IoT) cloud platform, ThingSpeak, for real-time visualization.

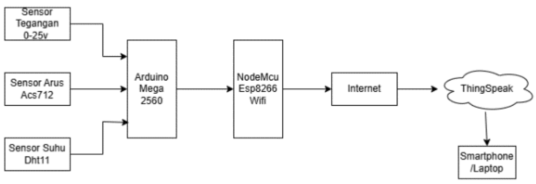

Figure 8. Hardware Architecture

In this system, the input section consists of three main sensors, namely the voltage sensor (0–25V), the ACS712 current sensor, and the DHT11 temperature and humidity sensor. The voltage sensor functions to monitor the voltage level from power sources such as solar panels or batteries, producing an analog signal that is transmitted to the analog pins of the Arduino Mega. The ACS712 current sensor is used to detect the amount of current flowing in the system when a load is applied, with its output in the form of an analog signal. Meanwhile, the DHT11 sensor measures the surrounding environmental temperature and humidity, providing digital data that is sent to the digital pins of the Arduino.

All data obtained from the sensors are processed by the Arduino Mega 2560. This microcontroller was selected due to its capability of supporting multiple input/output pins and managing several sensors simultaneously. The Arduino reads both analog and digital signals from the sensors, converts them into digital format, and organizes them into a structured string. The data are then transmitted via TTL serial communication to the NodeMCU ESP8266 for further transmission over the internet.

The NodeMCU ESP8266 functions as the wireless communication module in the system. It receives data from the Arduino via serial communication, processes the data, and transmits them to the ThingSpeak platform through a Wi-Fi connection. ThingSpeak was chosen as the platform due to its capability to store data and display them in real-time graphical form, as well as its accessibility from various devices such as smartphones and laptops [15].

2.3.2. Hardware Connection Circuit

Figure 9 illustrates the connection diagram between the sensors, Arduino Mega 2560, level shifter, and NodeMCU ESP8266. This diagram represents the flow of data, starting from sensor readings to the transmission of data to the ThingSpeak platform. Communication between the microcontrollers is carried out using TTL serial communication, assisted by a level shifter to adjust the logic voltage levels between the Arduino (5V) and the NodeMCU (3.3V).

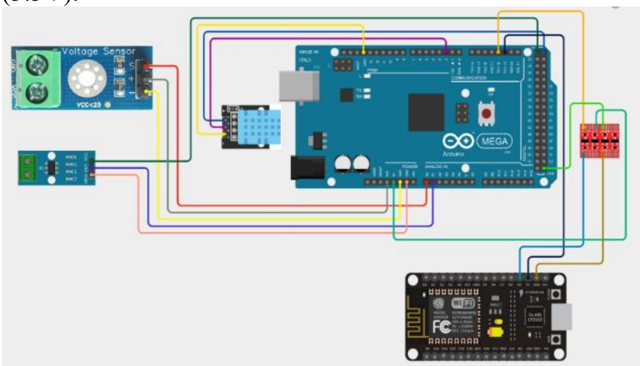

Figure 9. Hardware Connection

Table 1 below describes the connections of each component within the system:

Table 1. Component Connections in the System

Component	Pin Connection	Description
Voltage Sensor	VCC → 5V, GND → GND, OUT → A0	Measures DC voltage (0–25V)
ACS712 Current Sensor	VCC → 5V, GND → GND, OUT → A1	Measures electrical current
DHT11 Sensor	VCC → 5V, GND → GND, DATA → D2	Measures temperature & humidity
Arduino Mega	TX1 (18) ↔ RX Level Shifter, RX1 (19) ↔ TX Level Shifter	Sends data to NodeMCU via serial communication
Level Shifter	TX ↔ RX (5V ↔ 3.3V logic bridge)	Adjusts voltage levels between Arduino and NodeMCU
NodeMCU ESP8266	TX → RX Arduino, RX → TX Arduino	Transmits data to the internet (ThingSpeak)

2.3.3. Software Architecture

The software design describes the system workflow and the operational process of the device, as illustrated in the flowchart in Figure 10. The development environment used is Arduino IDE version 2.3.6, with the Arduino AVR Boards and NodeMCU ESP8266 boards installed. Additionally, the system employs the Adafruit Unified Sensor library, the ACS712 library, and the DHT Sensor library. To establish communication with ThingSpeak, the ThingSpeak library is utilized.

2.3.4. Integration of NodeMCU with ThingSpeak

To transmit data from the NodeMCU ESP8266 to the ThingSpeak platform, an HTTP connection based on the GET method is used. Each time the NodeMCU receives data from the Arduino via serial communication, the data is parsed into variables such as voltage, current, temperature, and humidity. The NodeMCU then constructs a URL containing the API Key and the data to be sent to ThingSpeak. An example of the URL structure is as follows: *https://api.thingspeak.com/update?api_key=WRITE_API_KEY&field1=tegangan&field2=arus&field3=suhu&field4=kelembapan*.

The URL is transmitted using the WiFiClient or HTTPClient functions from the Arduino library. Once successfully transmitted, ThingSpeak automatically stores and visualizes the data in graphical form for each field.

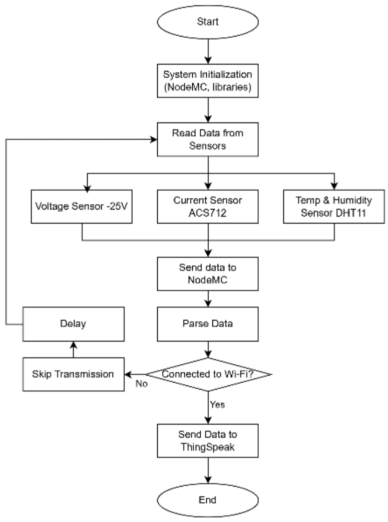

Figure 10. Software Design Flowchart

The ThingSpeak channel consists of several components:

- Channel Name: the identifier representing the project.
- Field 1 – Voltage (V)
- Field 2 – Current (A)
- Field 3 – Temperature (°C)
- Field 4 – Humidity (%)
- Write API Key: the authentication key used by the NodeMCU to send data.

ThingSpeak provides real-time graphical visualization for each field. These graphs can be customized in terms of title, color, time axis, and other parameters, and can be adjusted according to the data update interval. Data transmitted at a few-second intervals are displayed in the form of time-series charts, enabling users to monitor system conditions and trends directly from anywhere.

2.3.5. Monitoring System Architecture

The monitoring system architecture shown in Figure 11 is designed to observe the performance of the solar phone charging station in real-time. The system utilizes several sensors to measure key parameters such as voltage, current, temperature, and humidity. The measurement data are transmitted to the Arduino Mega 2560, which serves as the central processing unit. The Arduino processes the data and forwards them to the NodeMCU ESP8266, a wireless communication module that enables internet connectivity.

The transmitted data are then stored on the ThingSpeak server and database, an Internet of Things (IoT) platform managed by MathWorks, the developer of MATLAB. Through this server, the data can be accessed, visualized, and analyzed using devices such as smartphones or laptops.

With this architecture, the system's condition can be monitored periodically to assess component performance and support decision-making for preventive actions in case of anomalies. Thus, the system allows for the optimization of solar energy utilization while ensuring the smooth operation of the charging station.

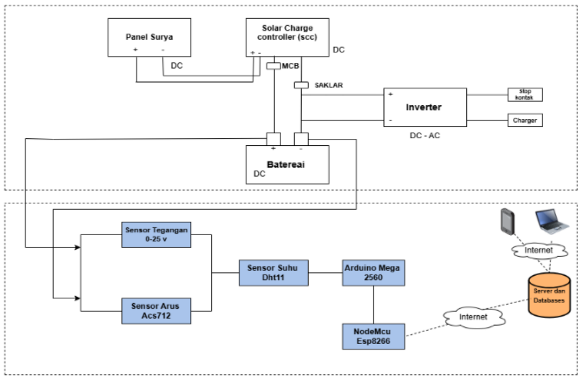

Figure 11. Monitoring System Architecture

2.4. Monitoring System Implementation

The system consists of several main components, namely the Arduino Mega 2560 as the central data processor, the NodeMCU ESP8266 as the WiFi-based communication module, and several sensors such as the 0–25V voltage sensor, ACS712 current sensor, and DHT11 temperature and humidity sensor. Data from the sensors is read by the Arduino Mega and sent serially to the NodeMCU, which then uploads the data to the IoT platform ThingSpeak.

The Arduino Mega was chosen because it has a sufficient number of pins to handle multiple sensor inputs simultaneously, as well as sufficient memory to store programs and manage sensor data. The NodeMCU ESP8266 is used due to its ability to connect to a WiFi network and its ease of communication with IoT servers. Communication between devices is carried out via TTL serial communication with the help of a level shifter to avoid voltage logic differences between the Arduino (5V) and the NodeMCU (3.3V).

This system is designed to send data to ThingSpeak every 5 seconds. Testing was conducted from noon to afternoon in an open area, assuming the solar panel received optimal sunlight. The transmitted data includes the voltage value from the solar panel battery, load current, and the surrounding temperature and humidity of the charging station. All data is sent in string format and parsed again on the NodeMCU side before being transmitted to the designated channel on ThingSpeak.

2.5. **System Testing**

The purpose of system testing is to evaluate the performance and reliability of the IoT-based monitoring system for the solar phone charging station that has been developed. The testing covers the process of reading data from the sensors (voltage, current, temperature, and humidity), transmitting data from the Arduino Mega to the NodeMCU via TTL serial communication, and uploading data directly to the ThingSpeak platform through a Wi-Fi connection.

The trials were conducted in an open area during the daytime until late afternoon to simulate real operating conditions of the system. The test data was analyzed to assess the system's performance stability, sensor accuracy, energy usage, as well as the reliability of communication and data visualization on the IoT platform.

3. **Results and Discussion**

The collected data was analyzed and compared with the system's expected performance, providing a comprehensive overview of the effectiveness and reliability of the developed system.

3.1. **Monitoring Results**

The monitored parameters include Voltage, Current, Temperature, and Humidity. These data were periodically collected using sensors connected to the microcontroller and transmitted to a cloud-based monitoring platform such as ThingSpeak.

The data recording process conducted by the IoT-based solar phone charging station monitoring system revealed discrepancies between the programmed data logging interval and the actual recording time captured on the platform. In the system's programming, the data transmission interval was set to every 5 seconds. However, based on the recorded data on the ThingSpeak platform, the time intervals between data points varied and were not always exactly 5 seconds.

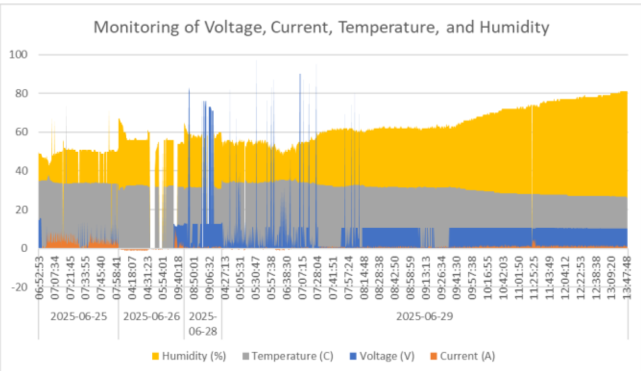

Figure 12. Monitoring Results Graph

The monitoring results from June 25–29, 2025, as shown in Figure 11, present the recorded voltage and current, temperature, and humidity over time.

On the X-axis, the graph displays the data logging time (in hh:mm:ss format), while the Y-axis on the left represents the values of voltage (V), current (A), temperature (°C), and humidity (%). From the graph, it can be observed that the voltage values (blue line) tend to fluctuate, with several sharp drops approaching zero volts and occasional spikes exceeding 22 volts. These fluctuations indicate instability in the data recording system. Furthermore, on June 27, 2025, no data was recorded at all.

3.1.1. Monitoring Results of Voltage and Current Data

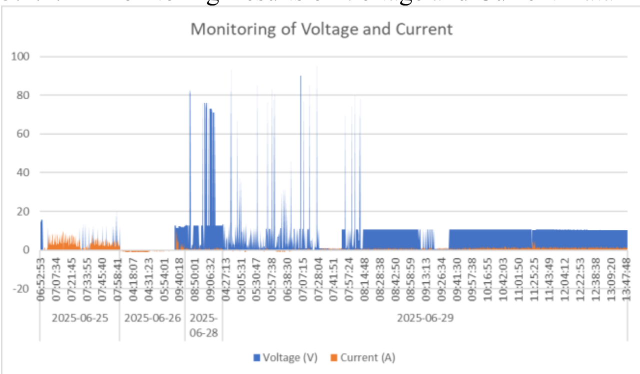

Figure 13. Voltage and Current Monitoring Results Graph

Based on the monitoring results of voltage and current shown in Figure 13, a fairly significant fluctuation pattern was observed. The voltage exhibited considerable variation, particularly between June 26 and 28, with recorded peak values exceeding 80 V. These values do not reflect the actual operating condition, as around 4% of the data was recorded abnormally, producing maximum voltages that should have been limited to 22 V. Therefore, the voltage spikes appearing in the graph can be interpreted as data anomalies or monitoring instrument errors, either due to sensor reading disturbances or logging inaccuracies. After June 28, the voltage values tended to stabilize at around 20 V, which is more consistent with actual operational conditions.

Meanwhile, the measured current was relatively small compared to the voltage, with most values remaining below 5 A. Current only appeared during specific periods, generally coinciding with voltage increases, indicating load activity during those times. After June 28, the measured current occurred less frequently, suggesting that the system's supplied load decreased or that the system was in standby mode. In addition, several data anomalies were found, such as conditions where voltage was recorded as zero while current readings reached up to 22%.

There were also several periods where both voltage and current approached zero, which were most likely caused by non-operational conditions. Overall, these measurement results indicate that although the graph displays anomalies in voltage data, the actual system operated at a relatively stable voltage not exceeding 22 V, with low and varying current depending on the supplied load.

3.1.2. Monitoring Results of Temperature and Humidity.

Based on the monitoring results of temperature and humidity during the period of June 25–29, 2025, a relatively stable pattern with certain tendencies was observed. The temperature (gray line) ranged between 28–35 °C and did not show any extreme changes throughout the observation period. This indicates that the environmental conditions around the system were within a reasonable range and were not significantly affected by weather fluctuations.

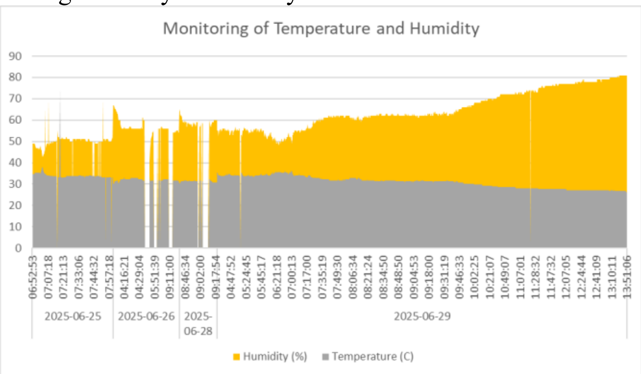

Figure 14. Temperature and Humidity Monitoring Results Graph

Meanwhile, humidity (yellow line) showed an upward trend from around 50% to more than 80% at the end of the recording period. This increase is consistent with atmospheric conditions that likely shifted from relatively dry to more humid in the following days.

A comparison of the recorded results with secondary data sources was conducted using data from WunderGround [16][17][18]. From this comparison, a discrepancy of 18% was found in temperature readings and 45% in humidity readings. The relatively large difference, particularly for the humidity parameter, indicates potential inaccuracies in the instrument or disturbances in the data logging system. Therefore, further calibration or the use of higher-accuracy sensors is required to ensure that the monitoring results more accurately represent the actual environmental conditions.

4. Conclusions

The monitoring system was successfully designed and implemented using the Arduino Mega 2560 as the central data processor, the NodeMCU ESP8266 for transmitting data to the internet, and sensors such as the 0–25V voltage sensor, the ACS712 current sensor, and the DHT11 temperature and humidity sensor.

The monitoring results revealed significant voltage fluctuations, with peak values exceeding 80 V during the period of June 26–28. However, these values do not represent the actual operating conditions, as approximately 4% of the data were identified as anomalies caused by sensor disturbances or logging errors, given that the maximum voltage should have been limited to 22 V. Following June 28, the voltage stabilized at around 20 V, while the current remained relatively low, with most values below 5 A, and was only observed during periods of load activity. Several anomalies were also detected, including conditions where the voltage was recorded as zero while the current reached up to 22%, as well as periods where both voltage and current approached zero, indicating non-operational states of the system. In addition, the humidity parameter exhibited an upward trend, increasing from approximately 50% to over 80% at the end of the monitoring period. A comparison with secondary data from WunderGround indicated discrepancies of 18% in temperature and 45% in humidity, underscoring the necessity for further instrument calibration or the deployment of higher-accuracy sensors to ensure more valid representation of actual environmental conditions.

Overall, this IoT-based solar phone charging station monitoring system has functioned quite well in capturing and transmitting data, provided that internet connectivity is available. However, under certain conditions, the sensors failed to perform readings, resulting in data values of 0, NaN, or unrealistic measurements. The disruption observed on June 27 indicates that the system's performance is highly dependent on the stability of internet connectivity and the reliability of the power source, particularly the battery of the solar phone charging station, which can only operate optimally under sufficient sunlight exposure. In addition, the relatively high error rate in current, temperature, and humidity readings suggests the need for improved system accuracy through better sensor calibration. Furthermore, to enhance the accuracy of temperature and voltage measurements, comparisons with reference data from locations with higher precision can be applied.

Acknowledgments

The author would like to express sincere gratitude to the Institute for Research and Community Service (LPPM) of Universitas Kristen Immanuel (UKRIM) for the support and financial assistance provided, which made this research possible.

References

[1] I. Rachman, D. Mukhsin Al Hamdani, and Y. Setia Permadi, "Rancang Bangun Sistem Monitoring Daya Listrik Cerdas Berbasis Internet of Things Menggunakan Mikrokontroler Arduino Mega Pro 2560 Dan ESP8266-01," *Ranah Res. J. Multidiscip. Res. Dev.*, vol. 5, no. 4, pp. 329–334, 2023, doi: 10.38035/rrj.v5i4.784.

[2] S. S. Berutu and Y.-C. Chen, "IoT Power Monitor System Development and Using Deep Learning Technology for Disturbances Classification," Asia University, 2021.

[3] S. Nor, "Penerapan Internet Of Things (Iot) Sebagai

Pengendali Peralatan Listrik Dan Pemantau Daya Listrik Berbasis Web," *J. EEICT (Electric, Electron. Instrumentation, Control. Telecommun.*, vol. 2, no. 2, pp. 22–28, 2019, doi: 10.31602/eeict.v2i2.4431.

[4] T. S. B. Tedi, "Integrasi Internet of Things dan Teknologi Komputer: Tantangan dan Prospek dalam Era Transformasi Digital," *J. Portal Data*, vol. 3, no. 10, pp. 2023–2024, 2023, [Online]. Available: http://portaldata.org/index.php/portaldata/article/download/491/475

[5] A. Yuan, D. Pianka, and E. Setyowati, "Evaluating The Implementation of Solar Home Systems (SHS) In Sumba – East Indonesia," *CSID J. Infrastruct. Dev.*, vol. 5, no. 1, p. 113, 2022, doi: 10.32783/csid-jid.v5i1.259.

[6] M. F. Pela and R. Pramudita, "Sistem Monitoring Penggunaan Daya Listrik Berbasis Internet of Things Pada Rumah Dengan Menggunakan Aplikasi Blynk," *Infotech J. Technol. Inf.*, vol. 7, no. 1, pp. 47–54, 2021, doi: 10.37365/jti.v7i1.106.

[7] B. Prayitno and P. Palupiningsih, "Prototipe Sistem Monitoring Penggunaan Daya Listrik Peralatan Elektronik Rumah Tangga Berbasis Internet Of Things," *Petir*, vol. 12, no. 1, pp. 72–80, 2019, doi: 10.33322/petir.v12i1.333.

[8] B. A. Maslyawan, S. Nurcahyo, and A. Murtono, "Sistem Monitoring Konsumsi Daya Listrik Pada Kamar Kost Serta Estimasi Biaya Keluaran Berbasis IoT (Internet of Things)," *J. Elektron. dan Otomasi Ind.*, vol. 8, no. 2, p. 76, 2021, doi: 10.33795/elk.v8i2.278.

[9] N. Sari, Y. Away, and S. Suriadi, "Desain Perangkat Monitoring Faktor Daya Pada Sistem Pv on-Grid Berbasis Iot," *J. Komputer, Inf. Teknol. dan Elektro*, vol. 5, no. 3, pp. 25–32, 2020, doi: 10.24815/kitektro.v5i3.17797.

[10] V. Quasching, *Understanding Renewable Energy Systems*, 2nd ed. London: Earthscan London, 2016.

[11] T. Ratnasari and A. Senen, "Perancangan Prototipe Alat Ukur Arus Listrik Ac Dan Dc Berbasis Mikrokontroler Arduino Dengan Sensor Arus Acs-712 30 Ampere," *J. Sutet*, vol. 7, no. 2, pp. 28–33, 2017.

[12] F. Akbar and S. Sugeng, "Implementasi Sistem Monitoring Suhu dan Kelembapan Ruangan Penyimpanan Obat Berbasis Internet Of Things (IoT) di Puskesmas Kecamatan Taman Sari Jakarta Barat," *J. Sos. Teknol.*, vol. 1, no. 9, pp. 1021–1028, 2021, doi: 10.59188/jurnalsostech.v1i9.198.

[13] W. Winasis, A. W. W. Nugraha, I. Rosyadi, and F. S. T. Nugroho, "Desain Sistem Monitoring Sistem Photovoltaic Berbasis Internet of Things (IoT)," *J. Nas. Tek. Elektro dan Teknol. Inf.*, vol. 5, no. 4, pp. 328–333, 2016, doi: 10.22146/jnteti.v5i4.281.

[14] A. D. Pangestu, F. Ardianto, and B. Alfaresi, "Sistem Monitoring Beban Listrik Berbasis Arduino Nodemcu Esp8266," *J. Ampere*, vol. 4, no. 1, p. 187, 2019, doi: 10.31851/ampere.v4i1.2745.

[15] I. D. Syahwir, A. Samsi, V. Firmansyah, P. W. Yunanto, R. Maheswara, and N. P. Anggraeni, "Internet of Things of Electronic Seal Base on Gprs Short Message Service and Thingspeak," *Spektra J. Fis. dan Apl.*, vol. 4, no. 3, pp. 125–132, 2019, doi: 10.21009/spektra.043.03.

[16] WunderGround, "Weather Station 25-07-2025." Accessed: Sep. 06, 2025. [Online]. Available: https://www.wunderground.com/dashboard/pws/ISLEMA35/table/2025-06-25/2025-06-25/daily

[17] WunderGround, "Weather Station 26-07-2025." Accessed: Sep. 06, 2025. [Online]. Available: https://www.wunderground.com/dashboard/pws/ISLEMA35/table/2025-06-26/2025-06-26/daily

[18] WunderGround, "Weather Station 28-07-2025." Accessed: Sep. 06, 2025. [Online]. Available: https://www.wunderground.com/dashboard/pws/ISLEMA35/table/2025-06-28/2025-06-28/daily