Real-Time Monitoring for a Building-Integrated Photovoltaic System based on the Internet of Things and a Web Application

Atef Ftirich
Laboratory of Computing for Industrial Systems (LISI), INSAT, University of Carthage, Tunisia
aatef.ftirich@gmail.com

Bechir Bouaziz
Laboratory of Computing for Industrial Systems (LISI), INSAT, University of Carthage, Tunisia
bechir.bouaziz@isimg.tn (corresponding author)

Faouzi Bacha
Laboratory of Computing for Industrial Systems (LISI), INSAT, University of Carthage, Tunisia
faouzi.bacha@esstt.run.tn

Received: 17 April 2024 | Revised: 20 May 2024 | Accepted: 5 June 2024

 DOI: https://doi.org/10.48084/etasr.7531

ABSTRACT

Building-Integrated Photovoltaic (BIPV) systems have become the most attractive clean solution for generating sustainable energy in building structures. Thus, the challenge of improving their efficiency is of extreme importance. The deployment of remote monitoring systems based on the Internet of Things (IoT) presents an opportunity to reduce the overall costs associated with BIPV systems. However, the performance of these monitoring systems varies depending on different parameters and environmental conditions. This paper presents a low-cost IoT-based prototype for monitoring of a solar photovoltaic panel connected to a battery. The current, voltage, and temperature values of the panel and battery, along with environmental parameters such as humidity, temperature, and solar irradiance were measured using low-cost relevant sensors. The data were sent and stored in PostgreSQL database servers through the Raspberry Pi4 Wi-Fi microcontroller board. Real-time visualization of data was facilitated through Web monitoring interfaces. The intelligent monitoring system was implemented in a facility located in Gabes, Tunisia.

Keywords-Internet of Things; low-cost prototype; real-time monitoring; Building Integrated Photovoltaic (BIPV) system

I. INTRODUCTION

The Internet of Things (IoT) is a set of smart devices implanted with software, sensors, and other technologies that allow them to communicate and collect data. The connectivity of these devices allows them to effortlessly interact with each other, opening new possibilities for creative applications in a variety of industries [1]. In the context of the energy sector, IoT plays a pivotal role in enhancing the efficiency and functionality of Photovoltaic (PV) systems. PV systems are designed to harness sunlight and convert it into electrical energy through solar panels. Hence, integrating IoT with these systems introduces a new level of intelligence and control, transforming traditional solar energy setups into smart and adaptive solutions. The use of IoT in the monitoring of PV systems is on the rise, as it allows the employment of sensors to collect data on the operational aspects and effectiveness of solar panels, inverters, and other components within a solar PV system. Subsequently, the collected data are subjected to analysis, allowing optimization of system performance and detection of potential issues or faults [2]. In addition, the integration of IoT facilitates improved energy management and grid integration. Smart inverters and controllers, connected through IoT, can dynamically adjust energy output according to demand and grid conditions. This flexibility contributes to grid stability and supports the overall transition to a more sustainable and efficient energy infrastructure [3, 4].

Overall, the convergence of IoT and PV systems opens new possibilities for the energy sector. By harnessing the power of connectivity and data analytics, smarter, more resilient, and more efficient solar energy solutions can be developed, which can play a major role in the larger context of sustainable and intelligent infrastructure [5].

This paper aims to explore the implementation of IoT for monitoring PV systems by exploiting the connectivity of the sensors to the Raspberry Pi for data acquisition and logging to a PostgreSQL database. Additionally, Symfony and Angular were employed for the IoT presentation layer. This comprehensive approach demonstrates the potential to create a sophisticated, data-driven, and user-friendly application that contributes to the advancement of efficient and intelligent PV systems.

II. PROTOTYPE OF THE INTELLIGENT MONITORING SYSTEM

A. *Smart Energy Application*

In [6] the components involved in a smart energy application of IoT are described. This is very useful in fault detection and providing a monitoring and maintenance facility to the user. Technologies used in such systems are LoRa, Z-Wave, SigFox, etc, [7, 8]. These systems are highly sustainable, providing greater safety for users, higher efficiency and level of optimization. In this work, the concept of smart energy was integrated into the realm of Building Integrated PV (BIPV) automation. The PV panel plays a crucial role in this energy management system, as it not only harnesses solar energy, but also contributes to the overall energy efficiency of the smart building system.

B. *System Architecture*

The architecture of the developed Intelligent Monitoring System (IMS) is provided in Figure 1, which illustrates a scenario where the sensors were connected to a Raspberry Pi using a Wi-Fi communication protocol. This setup allows for real-time data storage in the database. A Web application was developed to monitor the data and manage the load.

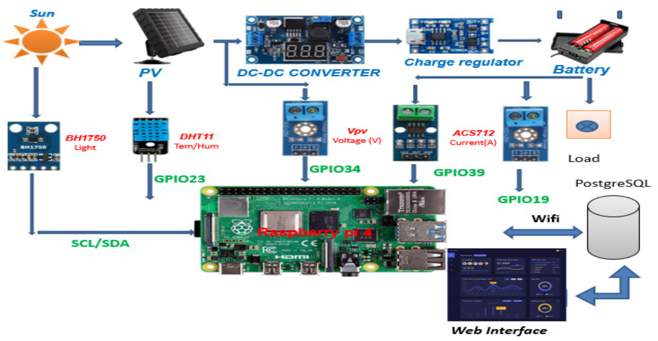


Fig. 1. Bloc diagram and components of the proposed BIPV monitoring system.

C. *Hardware Overview*

Raspberry Pi, is designed as a cost-effective platform and its primary objective is to enhance global accessibility to computer science education. Raspberry Pi is a versatile device that supports various versions of the free GNU/Linux operating system, including Debian and Ubuntu, alongside compatible software, and can accommodate Microsoft's Windows 10 IoT Core. Equipped with General Purpose Input/Output (GPIO) pins, Raspberry Pi facilitates the control of electronic components, making it an ideal choice for physical computing and exploration of IoT. Its affordability, modularity, and open design make it a popular hardware utilized in diverse fields such as weather monitoring. Furthermore, the incorporation of HDMI and USB standards further enhances its appeal in computer, electronic, and hobbyist communities.

The flowchart of this work is shown in Figure 2 and begins with the initialization of the Raspberry Pi and its Wi-Fi module. Then is checked if the initialization is successful. If not, an error message is sent. If successful, it proceeds to establish an internet connection, followed by a connection to a database. After successfully connecting, it reads analog pin values and takes input from the PV panel before uploading the data to the Internet and storing them. This flowchart can be used as a guide for setting up Raspberry Pi to monitor and store data from a PV panel.

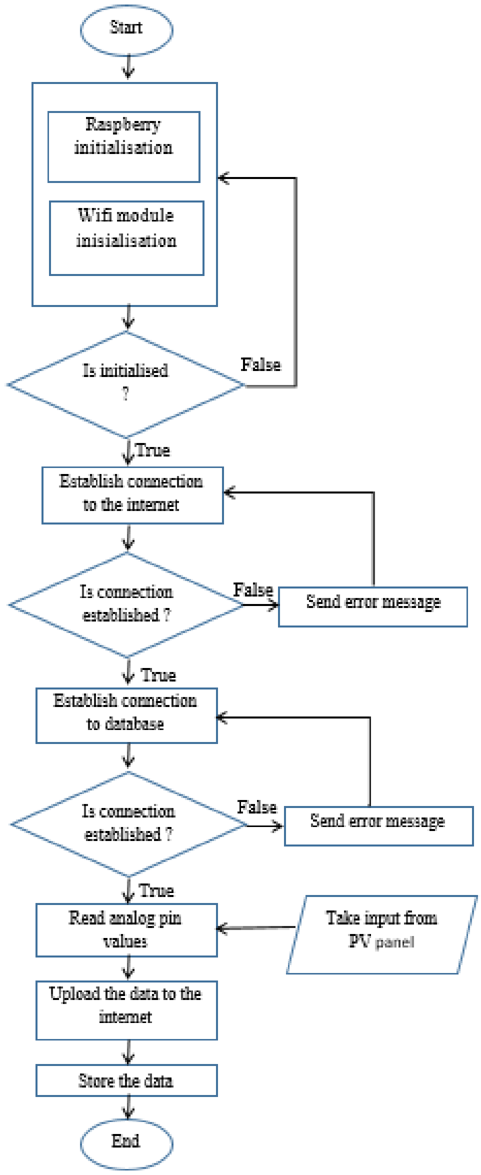


Fig. 2. Flowchart of Raspberry Pi initialization and data acquisition.

In this study, various sensors were used to measure values that required evaluation, including PV panel voltage, battery voltage, battery current, as well as ambient air humidity, temperature, and solar irradiance values. The ACS712-20A current sensor was employed for measuring the current of the battery. The DH11 humidity and temperature sensor was used to measure temperature and humidity. The BH1750 sensor was used for precise measurement of ambient brightness. In direct sunlight, a panel generates between 12 and 15 V of voltage. The TP4056 charge controller module was used for charging rechargeable lithium batteries Li-ion 18650 / 6800 mAh / 3.7 V using the Constant-Current/Constant-Voltage (CC/CV) charging method. For safe charging of the lithium batteries, the module also provides the necessary protection required by lithium batteries.

III. WEB PAGE DEVELOPMENT

The aim of this study is to design and implement a real-time monitoring system for a BIPV system based on the IoT technique and a Web application. The architecture of the approach relies on two robust frameworks: Angular and Symfony. The latter, serving as a PHP framework, facilitates the development of a RESTful Application Programming Interface (API) and seamless communication with a PostgreSQL database, and leverages the API Platform bundle to effortlessly generate documentation and implement CRUD (Create, Read, Update, and Delete) operations for entities. On the front-end, Angular was used, which is a Type Script framework that enables efficient consumption of the API and crafts a dynamic user interface. Throughout the development process, both frameworks ensured the creation of a code base that is not only functional, but also clean and easily maintainable.

A. System Component Overview

A sequence diagram is illustrated in Figure 3 and shows four key components: PV Panel, Raspberry, Database, and Interface. The PV panel represents the source responsible for the generation of electricity through sunlight absorption. It undergoes initialization along with the Raspberry Pi. In the event of an unsuccessful initialization, denoted "False," the PV panel abstains from connecting with the database. The Raspberry component stands for a microcontroller device tasked with controlling and managing the PV panel. It establishes a connection with the database unless the connection process yields a "False" response. The Database is the repository for storing and managing data received from the PV panel and facilitates the display of relevant information on the interface. The Interface serves as a user interface where data from the database are displayed. Two distinct user roles, administrator and user, can access and interact with the displayed information.

B. Communication between Raspberry Pi 4 and the Database

The integration of the Raspberry Pi 4 with its software environment is an indispensable and foundational step that emphasizes compatibility and ensures effective communication between the Raspberry Pi 4 and the designated PostgreSQL database using Wi-Fi protocols. To provide clarity and insight into the communication process, a visual aid in the form of a block diagram was used, as shown in Figure 4. This diagram serves as a guide elucidating the interaction between the Raspberry Pi 4 and the PostgreSQL database over the Wi-Fi connection. Such visualization assists in understanding the complete flow of communication [9]. With the connections successfully in place, the focus was shifted towards the PostgreSQL environment on the Raspberry Pi 4, a diligent implementation of the prototype and the corresponding code was carried out, and the configured interface, along with the established Wi-Fi connection, was leveraged to ensure a cohesive and functional project.

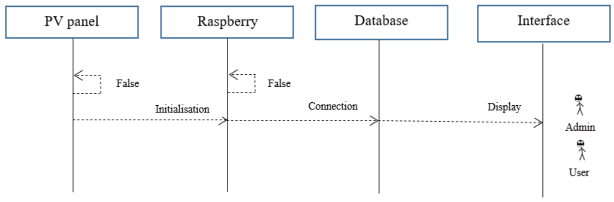


Fig. 3. Sequence diagram of key components: PV panel, Raspberry, Database, and Interface.

In essence, the proposed approach involves specific steps, starting with the configuration of the Raspberry Pi 4 interface and software environment, followed by the establishment of a reliable Wi-Fi connection to the PostgreSQL database. This method not only ensures an efficient wireless communication process, but also tailors the project to meet specific requirements, thereby enhancing flexibility and convenience in data transfer.

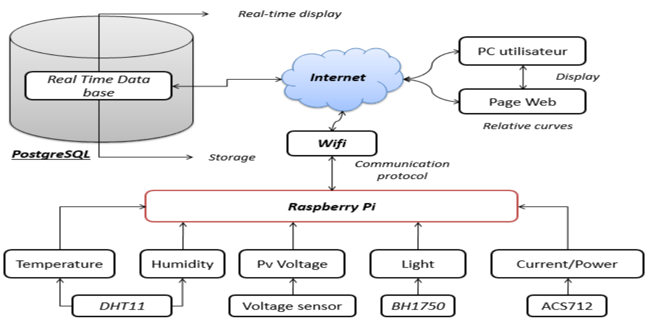


Fig. 4. Block diagram of communication between Raspberry Pi 4 and PostgreSQL database over Wi-Fi.

C. Efficient Data Flow

In modern Web development, establishing a robust and efficient communication pipeline between different components is crucial to a smooth user experience. In this research, the power of PostgreSQL was used to facilitate communication between a Raspberry Pi 4 and a Symfony 6 application running on a PC. The Raspberry Pi 4 serves as a local PostgreSQL host, allowing secure data storage and retrieval, while Symfony 6 orchestrates the interaction between the Web application and the database. This work delves into the details of this communication setup, exploring the configuration of PostgreSQL on the Raspberry Pi 4, the utilization of localhost for local interactions, and the incorporation of the Raspberry Pi's IP address for remote connections, as shown in Figure 5. By navigating through the integration process, the objective was to create a reliable and efficient data flow that harmonizes the capabilities of these diverse components [10].

Fig. 5. Communication pipeline: integration of Symfony 6 and Raspberry Pi 4 with PostgreSQL for efficient data flow in Web development.

D. Real-Time Communication with the Mercure Protocol

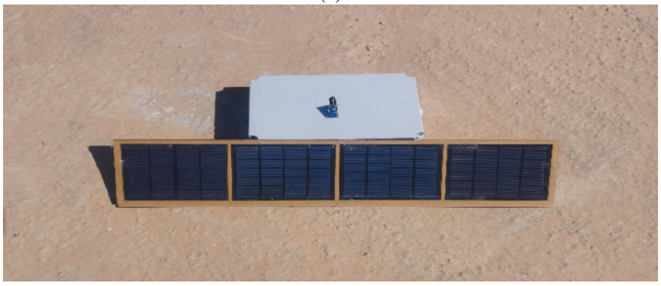
The demand for real-time data interaction in Web development has emerged as a key element in creating compelling user experiences. In this work, Mercure, Symfony, and Angular formed a resilient framework for seamless real-time communication. Mercure is a protocol designed to effortlessly push data updates to Web browsers and various HTTP clients. The Symfony framework assumes the role of an orchestrator on the server-side, managing communication between the database and the front-end. This strategic integration ensured a harmonious flow of information, which improved the overall efficiency of the application [11]. On the client-side, Angular takes the reins and relies on its reactive features to empower a responsive and fluid user interface. The merger of Mercure, Symfony, and Angular simultaneously fosters real-time communication and lays the foundation for a robust and engaging Web experience. An illustration of real-time data communication in a modern Web development setup can be found in [12]. It features three main components: a Symfony application, a Mercure Hub, and various client-side platforms including Web browsers, mobile applications, and IoT devices. The Symfony application sends data to the Mercure Hub using POST requests. The Mercure Hub then uses Server-Sent Events (SSE) to push updates in real-time to the connected client-side platforms. This flow diagram effectively visualizes the data flow and interactions in the system, highlighting the role of each component in the process [13]. In the proposed prototype setup, as shown in Figure 6, the Raspberry Pi 4 board within the BIPV data measurement, recording and monitoring apparatus should initially establish a connection to a wireless network. Once connected, data from the sensors are sequentially measured, sent, and then saved to the PostgreSQL real-time database. The PostgreSQL cloud system was used to create the Web application to monitor real-time data, maintain session and record information, make new announcements, and create control units. The BIPV data sent to the cloud system can be tracked, analyzed, and visualized in a Web application by end-users.

IV. IOT PLATFORM

A. Data Localization

The proposed intelligent monitoring system was operated by a 24-hour in depth-test at the study location in Gabes, Tunisia. The testing involved the meticulous examination of various parameters to ensure the system's performance under diverse conditions. This allowed the gathering and processing of all relevant data, presented in a user-friendly interface on the Web. The innovative methodology not only provided real-time insights into the functionality of the PV system but also showcased the potential for efficient data localization and integration in solar energy projects.

(a)



(b)

Fig. 6. (a) Experimental component board, (b) experimental setup.

B. User Interface

A comprehensive view of the project Web interface is provided in Figure 7. This user-friendly and intuitive platform allows users to monitor various environmental parameters, including temperature, humidity, power, voltage, light, and battery levels. The data are presented through visually appealing and informative graphical representations that are updated in real time for precise analytics. Furthermore, an integrated map is included in the platform that offers geographical data, which improves the user's situational awareness.

C. Case Study and Results

The proposed intelligent monitoring system was installed in the study area of Gabes, Tunisia. This location exhibits an interesting interaction between sunlight and temperature, particularly in the month of February, during which the presence of the sun is predominant for an average of 7.5 hours per day, and the maximum sunshine occurs in the afternoon. These climactic fluctuations were captured by the DHT11 sensor. DHT11 is a commonly used temperature and humidity sensor. The sensor can measure the temperature from 0 - 50 °C and the humidity from 20 - 95% with an accuracy of ±2°C and ±5%, respectively. In Figure.8 is shown the display of the temperature and humidity values sent by the IoT and the measuring instrument running in real-time.

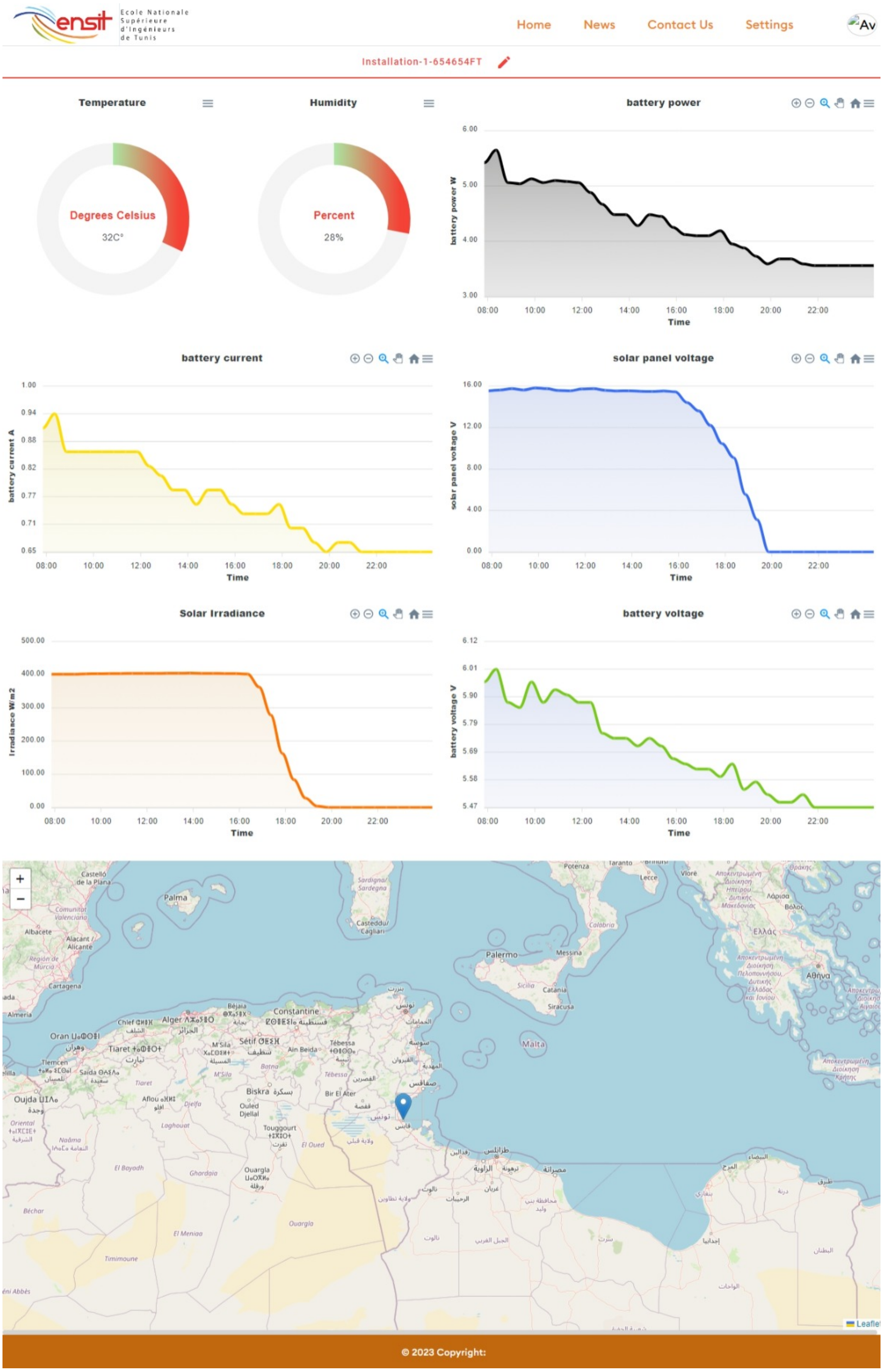


Fig. 7. User interface navigation.

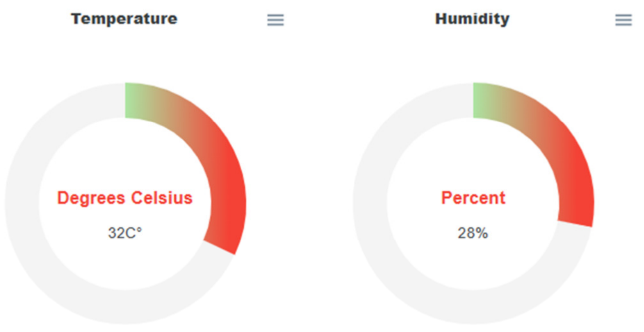


Fig. 8. Temperature and humidity values in Gabes, Tunisia.

The ACS712 sensor measures the current passing through a load. Sensor data, depicted in Figure 9, reveal the proportionality of power-load with current flow and the consequent variations in power as the current fluctuates. This provides a comprehensive understanding of load power dynamics.

Rigorous testing shows that the solar panel can generate a maximum voltage between 12 and 15 V. This voltage is crucial for charging and storing energy in batteries, as depicted in Figure 10, which illustrates the respective voltage curves and shows the efficiency of the solar panels in energy production. The study indicates the capacity of a panel to generate between 12 and 15 V. The system included a DC/DC converter and a voltage sensor to measure and adjust the output voltage. The TP4056 charge controller, connected in series with lithium 18650 li-ion rechargeable batteries, indicated the charging status via LED indicators, as shown in Figure 11.

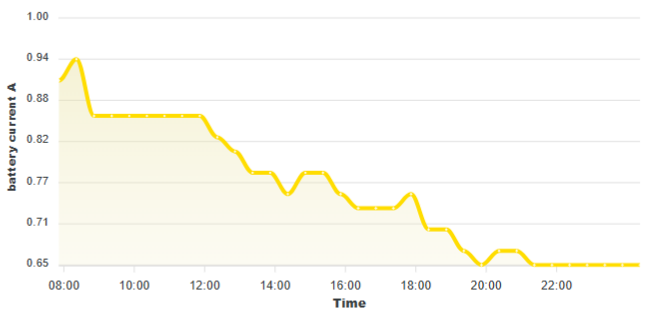


Fig. 9. Load current.

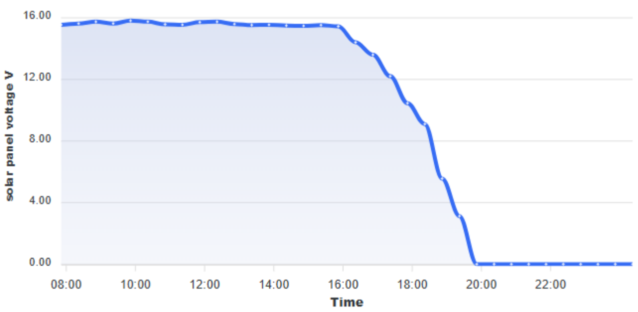


Fig. 10. Solar panel voltage.

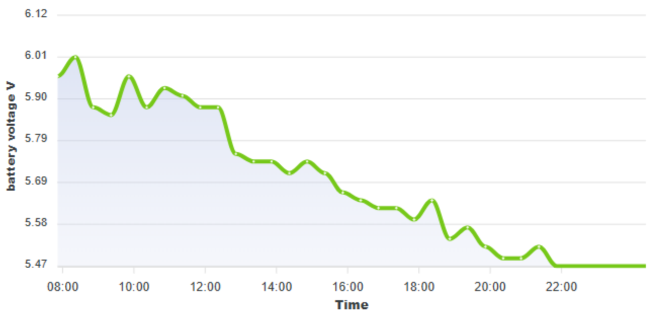


Fig. 11. Battery voltage varaitions.

In this proposed intelligent monitoring system, the position of the current sensor depends on the load. The reference voltage, which is set to calculate the power of this load, allows the observation of power variations with changes in the load. Hence, the dynamic nature of power consumption can be observed in response to load variations, as shown in Figure 12.

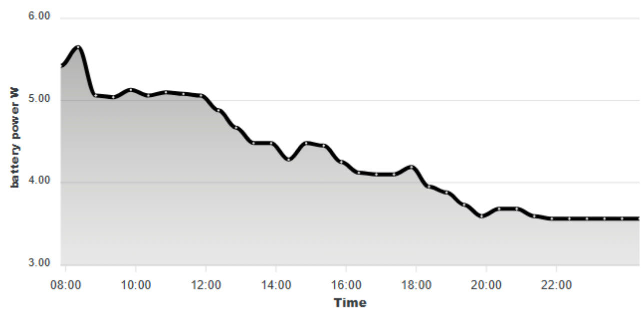


Fig. 12. Load power variations.

The BH1750 is an ambient light intensity sensor with high resolution (about 1 to 65535 Lux) and output in Lux units. The flux value that was received by the BH1750 light sensor was converted to irradiance. All data were converted to W/m^2 by multiplying with 0.079. The graph of solar radiation over time is shown in Figure 13. The obtained data revealed valuable information on sunlight intensity trends, especially during peak sunlight hours. This information serves as a testament to the capability of the BH1750 sensor to provide precise measurements, making it a reliable tool for calculating solar irradiance and understanding the irradiation variations at the location of the BIPV system.

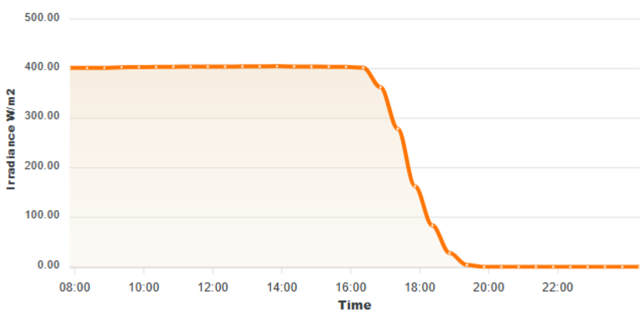


Fig. 13. Measurements of sunlight intensity using the BH1750 sensor.

V. CONCLUSIONS

The study focuses on the development of an intelligent monitoring system based on the Internet of Things (IoT) technique and the Web application for the real-time monitoring of a Building Integrated Photovoltaic System (BIPV). The self-consumption BIPV system is described in detail, as well as its context in a smart energy system. The main objective of the proposed method is to provide a smart energy monitoring solution using simple software and low-cost hardware. Using the Raspberry Pi 4, PostgreSQL, and Symfony in the backend, coupled with Angular in the front-end, a robust and efficient framework was established. The real-time monitoring facilitated by Mercure added a layer of sophistication that ensured instant updates and insights. The IoT module was based on a Raspberry Pi 4 and low-cost sensors that collected metrological and energy data from BIPV in real-time. The results demonstrate that the proposed monitoring system can be a promising solution and a pathway for intelligent remote monitoring in real-time of a BIPV system.

REFERENCES

[1] S. Zafar, G. Miraj, R. Baloch, D. Murtaza, and K. Arshad, "An IoT Based Real-Time Environmental Monitoring System Using Arduino and Cloud Service," *Engineering, Technology & Applied Science Research*, vol. 8, no. 4, pp. 3238–3242, Aug. 2018, https://doi.org/10.48084/etasr.2144.

[2] L. Boulemzaoud, S. Latreche, and M. Khemliche, "Design and Implementation of a Long Range Wireless Data Acquisition System for Photovoltaic Installation based on LoRa Technology," *Engineering, Technology & Applied Science Research*, vol. 12, no. 2, pp. 8473–8481, Apr. 2022, https://doi.org/10.48084/etasr.4859.

[3] 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, Jan. 2023, Art. no. 3860, https://doi.org/10.3390/en16093860.

[4] S. M. A. Ghaly and M. A. Kadampur, "Scalable Incident Reporting Framework: A Sensor and IoT Research," *Engineering, Technology & Applied Science Research*, vol. 13, no. 3, pp. 10748–10753, Jun. 2023, https://doi.org/10.48084/etasr.5820.

[5] Z. Liu *et al.*, "A review of data-driven smart building-integrated photovoltaic systems: Challenges and objectives," *Energy*, vol. 263, Part E, Jan. 2023, Art. no. 126082, https://doi.org/10.1016/j.energy.2022.126082.

[6] D. C. Nath *et al.*, "Internet of Things integrated with solar energy applications: a state-of-the-art review," *Environment, Development and Sustainability*, Oct. 2023, https://doi.org/10.1007/s10668-023-03691-2.

[7] D. D. Prasanna Rani, D. Suresh, P. Rao Kapula, C. H. Mohammad Akram, N. Hemalatha, and P. Kumar Soni, "IoT based smart solar energy monitoring systems," *Materials Today: Proceedings*, vol. 80, Part 3, pp. 3540–3545, Jan. 2023, https://doi.org/10.1016/j.matpr.2021.07.293.

[8] A. Ftirich, B. Bouaziz, and F. Bacha, "Design and Implementation of Real-Time Monitoring System for Building-integrated Photovoltaic System," in *2023 IEEE International Conference on Advanced Systems and Emergent Technologies (IC_ASET)*, Hammamet, Tunisia, Apr. 2023, pp. 1–6, https://doi.org/10.1109/IC_ASET58101.2023.10151167.

[9] B. E. Demir, "A New Low-Cost Internet of Things-Based Monitoring System Design for Stand-Alone Solar Photovoltaic Plant and Power Estimation," *Applied Sciences*, vol. 13, no. 24, Jan. 2023, Art. no. 13072, https://doi.org/10.3390/app132413072.

[10] C. K. Rao, S. K. Sahoo, and F. F. Yanine, "An IoT-based intelligent smart energy monitoring system for solar PV power generation," *Energy Harvesting and Systems*, vol. 11, no. 1, Jan. 2024, Art. no. 20230015, https://doi.org/10.1515/ehs-2023-0015.

[11] M. A. A. Radia, M. K. E. Nimr, and A. S. Atlam, "IoT-based wireless data acquisition and control system for photovoltaic module performance analysis," *e-Prime - Advances in Electrical Engineering, Electronics and Energy*, vol. 6, Dec. 2023, Art. no. 100348, https://doi.org/10.1016/j.prime.2023.100348.

[12] "Pushing Data to Clients Using the Mercure Protocol (Symfony Docs).", Symfony, https://symfony.com/doc/current/mercure.html.

[13] Y. Cheddadi, H. Cheddadi, F. Cheddadi, F. Errahimi, and N. Es-sbai, "Design and implementation of an intelligent low-cost IoT solution for energy monitoring of photovoltaic stations," *SN Applied Sciences*, vol. 2, no. 7, Jun. 2020, Art. no. 1165, https://doi.org/10.1007/s42452-020-2997-4.