IRAQI Academic Scientific Journals

University of Mosul - Iraq College of Engineering AREJ Al-Rafidain Engineering Journal 1993

Al-Rafidain Engineering Journal (AREJ) Vol. 30, No. 2, September 2025, pp. 61-75

Privileging and Prioritizing Processes of Sustainable Energy Resources for Residential Loads to Energize Green Cities

Basman M. Hasan Alhafidh*
bm.alhafidh@uomosul.edu.iq

Rabee M. Hagem*
rabeehagem@uomosul.edu.iq

Mustafa Qassab*
mustafa.qassab@uomosul.edu.iq

Mohammed B. Mahmood**
mohammed.basil@ntu.edu.iq

* Computer Engineering Department, College of Engineering, University of Mosul, Mosul, Iraq
** Cyber Security and Cloud Computing Techniques Engineering Department, Technical Engineering College for Computer and AI, Northern Technical University, Mosul, Iraq

Received: March 9th, 2025 Received in revised form: June 28th, 2025 Accepted: July 27th, 2025

ABSTRACT

The protection and management of residential solar panel energy systems in smart cities that depend on the Internet of Things (IoT) are vital. Controlling residential energy consumption manually is hard. In addition, significant challenges should be addressed, including facilitating communication between devices, conducting real-time data analysis, and ensuring security. In other words, constraints associated with interoperability, real-time data analysis, usability, scalability, and cost-effectiveness should be overcome in any reliable management system design. In this paper, a smart energy management system based on IoT to improve the efficiency and safety of residential solar panels is presented. The suggested approach focuses on prioritizing load shedding, which plays a crucial role in promoting sustainable energy practices in smart cities. The system is designed using Particle Photon, which includes a microcontroller with Wi-Fi and current transformers (CTs) to dynamically manage energy usage, helping minimize wastage and extend the life of limited power sources such as solar panels or local generators. The energy consumption for the proposed system is calculated assuming a 3576 mAh battery capacity, and the battery lifetime is 1000 hours. OPNET 14.5 Network Simulator is used to simulate the system. The Blynk platform is suggested for publishing data to the cloud for further analysis. Furthermore, recommendations concerning data privacy and security issues are proposed. As a result, stakeholders can smoothly monitor and control their energy consumption through a user-friendly mobile interface. The research shows that integrating IoT technology enhances energy management and supports the development of greener, more sustainable urban environments.

Keywords:

Embedded Wireless Communication; Smart Energy Management System; Smart Cities; Solar Panel; load-shedding.

https://rengj.uomosul.edu.iq/
Email: alrafidain_engjournal3@uomosul.edu.iq

1. INTRODUCTION

The tremendous increase in electronic devices and modern technologies has led to massive consumption of energy resources. The great demand for building the Internet of Things (IoT) network implies the use of a huge number of connected sensors, actuators, and devices that represent the basic elements for building smart things. In general, IoT is a system of connected devices, persons, or animals with unique IDs that can be directed remotely through the Internet. IoT aims to make everything smart, such as smart devices, smart grid,…etc.

In other words, the Internet has transformed from a network of computers into a network of objects or smart things [1-3]. As a result, we can notice the increased demand for energy to build a huge and clean network, ending with what we can call Green Cities. The critical direction of building green cities leads researchers to depend on renewable energy resources instead of non-renewable ones. The former represents a promising solution to face the energy crisis, in addition to eliminate many issues related to the use of fossil fuels, such as environmental pollution [4]. Many renewable energy sources that green power factories produce exist in our world, such as hydropower, geothermal, solar, wind, and marine energies [5]. Fig. 1 shows how we can build two different societies using renewable energy resources and non-renewable energy resources [6].

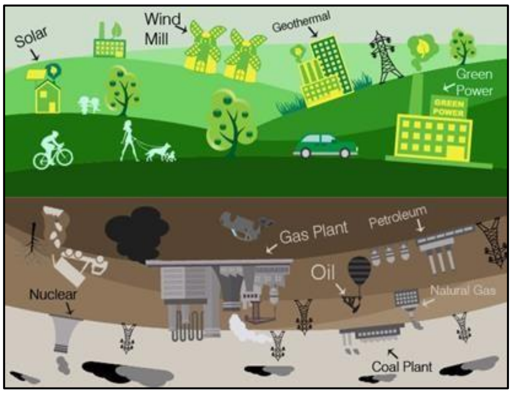

Fig. 1 Societies using renewable energy resources and non-renewable energy resources in a city [6]

Although renewable energy is a promising solution for the increasing demand for green resources such as solar panel energy or wind energy [7], it still has some limitations that need to be addressed in this paper.

In this paper, we focus on protecting and managing residential solar panel energy systems depending on the Internet of Things (IoT). One way to achieve this is by applying priority and privileged principles toward different kinds of residential home appliances that present a variety of loads on the renewable sources. In particular, solar panels are widely accessible and are increasingly being used. In addition, solar panels need to be managed carefully to increase their lifespan [8]. This paper investigates the priority of load shedding based on several decisions. The decisions are generated by analyzing the collected data from the devices connected to the solar system. As an extension, the idea of extending to a large number of solar panels distributed nationwide will definitely protect the solar panel system from collapsing, saving energy for a long lifespan, detecting faults, and providing reliability and protection of limited power resources from power outages.

The remainder of this paper is organized as follows: Section 2 discusses the previous works. Sections 3 and 4 present the research challenges and the proposed system, respectively, while the research methodology is presented in Section 5. The results are demonstrated in Section 6, and our conclusions are given in Section 7.

2. LITERATURE REVIEW

Monitoring systems are vital in different applications [9-11]; therefore, many proposed research studies use various strategies to build a home power management system that targets the efficient use of limited energy resources with different uses of technologies and methods as follows:

A system for monitoring and control using Power Line Communication (PLC) technology was proposed by Chia-Hung in 2008. According to researchers, home power consumption tends to grow in proportion to the increase in large-sized electric home appliances. An embedded system without additional Wiring has been developed for home power management using PLC technology [8].

An investigation of an off-grid photovoltaic system instead of local generators that provide electricity for 12 hours per day was mentioned in [12]. The investigation involves designing and simulating an off-grid PV system for one villa at the Avro City residential compound in Duhok. The energy consumption and sizing of the system have been computed, together with the economic evaluation of the system. The results show that the system works at 48 V direct with unused energy of 17.6%, energy supplied to the user is 63.9%, system losses are 6.5%, and collection losses are 12%. In addition, the results reveal that the performance ratio of the system is about 64% and the solar fraction is 0.96 in the case of using a single-phase 5 kW inverter, as was done in this work.

A. Zipperer, R. Roche, L. Earle, and P. Bauleo 2013 discussed the state of the art in electricity management and various enabling technologies in smart homes that focus on consumer behavior concerning energy usage [13]. Zeeshan Abbas and Wonyong Yoon in 2015 presented a survey on the issues of energy conservation and then solutions when using diverse wireless radio access technologies for the connectivity of IoT, such as IEEE 802.11ah or Bluetooth LE and other technologies [14].

Using IoT, Junyon Kim in 2016 constructed the HEMS (Home Energy Management System) model to achieve an active and systematic system for energy management [15]. In a report presented by Maria C. at the University of Catalonia in 2016, various solutions to make homes more efficient have been compared to managing electrical power consumption in homes using Internet of Things techniques [16]. Energy-hungry IoT devices are the challenge faced by the usage of those devices, which was overviewed by Rushan A., Saman Z., Munam A., and Abdul W. in 2017. Besides, the green IoT and strategies used to reduce energy consumption were discussed in [17]. Nowadays, the research area in IoT energy management is becoming very wide, each one has its technology that tries to facilitate green energies in an efficient and adequate way of use.n 2019, Princy S, Varun K, Aiswarya C, Sujith A., and Sithara K. proposed an IoT system in Arduino monitoring and showed the renewable energy usage, which used solar energy to manage the device switching [18]. Five months earlier, Bruno M., João C., and Nuno C. 2019 optimized the IoT platform energy consumption using Lora BEMS [19].

A comparison analysis between the suggested system and well-known works [20-22] in the field of smart energy solutions in response to the body of research on energy management is shown in Table 1. There is still a lack of systems that concurrently handle real-time prioritization, secure communication, and scalable deployment, even though solar energy integration, IoT-based monitoring, and energy optimization techniques have been the subject of numerous studies. We compare the architectural design and performance features of our system to three current and exemplary works in order to demonstrate the novelty of our methodology. This analysis illustrates how our embedded energy modelling, security-aware IoT framework, and hybridscheduling approach set us apart from other conceptual frameworks and real-world applications in the literature.

Table 1: A comparison between different up-to-date review works with the proposed system

Feature	Your Paper	Paper [20]	Paper [21]	Paper [22]
Scope	Residential IoT energy mgmt	School PV system design	Residential energy conservation via HEMS	Practical IoT energy system
Methodology	Hybrid scheduling, OPNET, Blynk	PVsyst, real site data	Load prioritization, thermostat control, and solar integration	ESP8266, cloud, MQTT
IoT Integration	Full-stack with cloud/mobile	None	Basic app and sensor-based control, user coordination	Moderate with practical tools
Load Prioritization	Yes (hybrid algorithm)	No	Yes – essential vs non-essential	Possibly, not detailed
Security Analysis	Detailed threats & countermeasures	Not discussed	Brief mention of data security needs	Minimal
Energy Modeling & Simulation	Yes (formulas, runtime)	Yes (PVsyst)	Estimates for loads, solar PV output, thermostat impact	Basic
Architecture Type	Networked Embedded System with IoT, cloud, and MCU (Photon)	Conventional Off-grid Solar Design	Conceptual HEMS model with utility and user integration	Prototype IoT System (Arduino/ESP8266)
Scheduling Algorithms	Hybrid model + comparison of static vs. dynamic scheduling	Not applicable	Time-of-use load shifting with user coordination	Not implemented explicitly
Scalability Architecture	Designed for scalability via dynamic plug-n-play priorities	Fixed system for one building	Scalable via coordinated user networks, but with cost barriers	Limited scalability
Response Time / Real-Time	Real-time scheduling & analysis system	Not applicable	App-driven semi-real-time interaction	Limited real-time capabilities

3. RESEARCH CHALLENGES

The research challenges that we identified in previous works are related to many points that should be addressed by researchers. It is essential to overcome those limitations in the current and future works by researchers.

- Interoperability: Common energy monitoring and distribution systems often use different kinds and brands of home appliances, making it challenging to share data between systems. The proposed system overcomes the interoperability challenge by securely transferring data through a middleware device by prioritizing smooth data flow and creating a unified structure for all system nodes.
- Real-time data analysis: Common energy monitoring and distribution systems generate large volumes of data in real-time. In the proposed system, analyzing the data and prioritizing the working nodes in real time are mandatory to ensure the system's affordability. By swiftly identifying and responding to energy source and node requirements, the system coordinates with the requirements for effective real-time data analysis tools.
- Data privacy and security: Common energy monitoring and distribution systems collect energy consumption data that might be sensitive in some cases or applications. There is a need to ensure that the networks between the nodes are secure enough to assure the privacy and integrity of the data. The implementation of a new security model in the proposed system directly targets the issues in data privacy, integrity, and security.
- Usability: Common energy monitoring and distribution systems are only effective if stakeholders enroll and use different loads. There is a need to use the proposed paradigm dynamically to assess the simplicity and mobility of use by individual users to meet the needs of their intended users.
- Cost-effectiveness: Common energy

monitoring and distribution systems can be expensive to implement and maintain. By analyzing network protocols and configurations to identify efficient and available choices, the system coordinates to provide a cost-effective solution suitable for resource-limited settings.

- Scalability: Most common energy monitoring and distribution systems are designed for a specific number of users or a small geographical area. The proposed system can be adapted to be used efficiently by any number of users that the customer needs. Using the proposed priority-based smart wall plugs makes the system more adaptable to varying user demands.

4. THE PROPOSED SYSTEM DESIGN

4.1 Networked Embedded System (NES) design

The smart energy management system using a microcontroller unit (MCU) at its core is illustrated in Fig. 2. The MCU connects to a sustainable energy source, such as solar panels, and regulates the load shedding process. The other parts contain sensors, actuators, and end devices for monitoring and control processes.

- Sensors: Collect data from the environment (e.g., temperature, humidity) and send it to the MCU.
- Actuators: Devices like fans, lights, and appliances that the MCU controls to optimize energy use.
- Monitoring: The system is monitored through cloud services and interfaces, allowing for remote management and data analysis.

The design presents an IoT-based setup for efficient energy management in a smart home environment through the use of specific scheduling strategies as discussed in sub-section 4.2

4.2 Scheduling strategies

The term scheduling in its general form was introduced in [23]: "Scheduling is the allocation, subject to constraints, or resources to objects being placed in space-time, in such a way as to minimize the total cost of some set of the resources used". Thus, adopting an efficient priority algorithm that meets the needs of dealing with power constraints and yet allows the system to operate with high performance is a must. Here are some scheduling algorithms that can be considered [24, 25]:

- User-Defined Priority Settings: Users can select device priorities based on their needs using a Graphical User Interface (GUI). This priority list is created manually and stays fixed unless the user modifies it.
- Demand-Response Optimization: Manages the operation of devices based on available power capacity and real-time demand. This is done by continuously monitoring devices and their power demands and usage patterns to set priorities dynamically. For instance, some devices will be operated during peak power demand periods if they have high critical functions or higher power demand.

A quick comparison of the two aforementioned scheduling algorithms is given in Table 2. By comparing the two algorithms, we recommend using the demand-response optimization algorithm for the proposed system due to its advantages over the user-defined settings. As a matter of fact, after going deep into studying the workflow of the proposed system, a hybrid scheduling algorithm was used as discussed later in section 5.2. This choice comes primarily to help in optimizing power management, especially during power constraints and shortages, and to ensure that critical and essential devices are provided with power when needed.

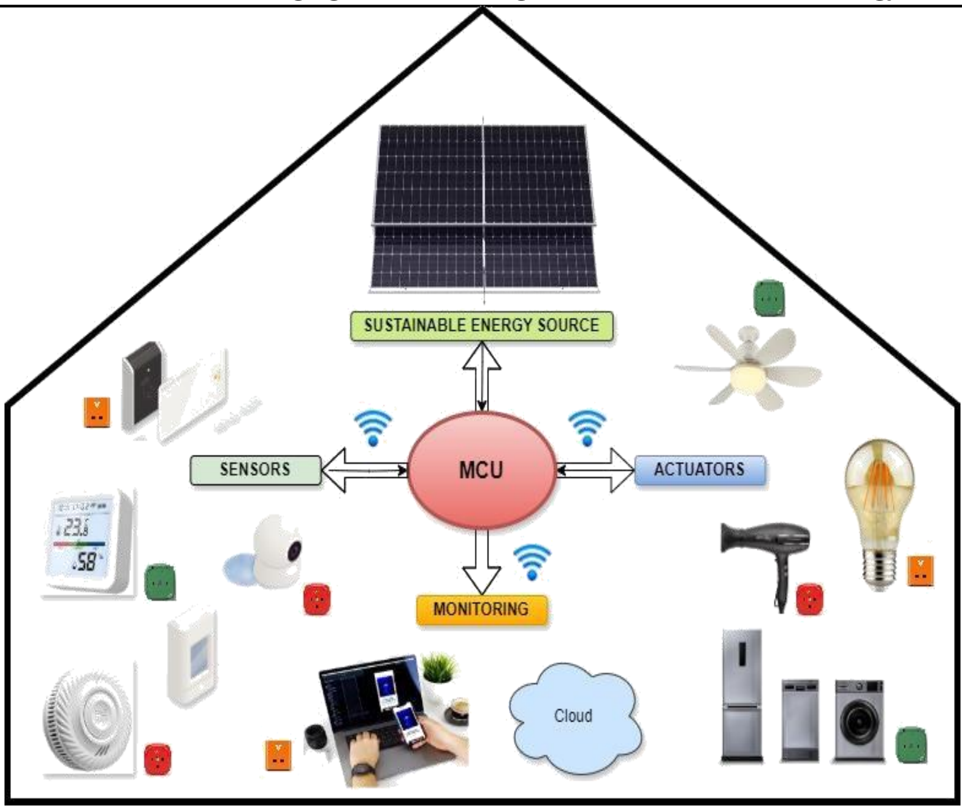

Fig. 2 The smart energy management system

Table 2: A comparison between two different scheduling algorithms.

Algorithm Metric	User-Defined Priority Settings	Demand-Response Optimization
User Control	Direct control to adjust device priorities.	Dynamic allocation according to device functionality and power demand.
Complexity	Relatively simple to understand and implement.	More complicated and collects device patterns to make decisions.
Priority	Static priority creates a specific sequence of operating connected devices.	Dynamic priority is set by multiple parameters that are being monitored.
Automation	Cannot adapt to changes or events.	More flexible and responsive to changes or events.
Scalability	Challenging to scale up due to manual configuration of priorities.	Easier to handle scalability issues as it works dynamically.

The following sub-section considers the energy consumption, an essential part of any IoT system. It examines how different parts affect total power use, especially in settings with limited resources. Sensor nodes, data processing units, and communication modules get particular attention. Optimizing system performance and battery life requires an understanding of these patterns.

4.3 Energy consumption

Understanding energy consumption for embedded systems in microcontroller applications, including the Photon and sensors in the proposed work, is crucial for optimizing performance and prolonging battery life. It evaluates power requirements across different operational modes such as active execution, sleep states, and transmission/reception tasks. By quantifying energy needs through these modes, developers can make informed decisions to balance functionality with efficient power management strategies, ensuring reliable operation and extended battery longevity. Generally, any microcontroller operates in one of these four modes:

- Active Mode: Includes data processing, code execution, peripheral control, and performing various tasks.
- Sleep Mode: Involves reducing power consumption by keeping only essential peripherals active while running the main program; the clock speed of the microcontroller is reduced to save power.
- Transmission Mode: Data is transmitted using communication interfaces like UART, Bluetooth, and Wi-Fi. This mode usually

consumes more power than other modes.

- Reception Mode: The microcontroller listens for incoming data to be received.

By considering these four modes, the consumed power can be calculated as:

$$p_{avg} = \frac{(p_{active}\times t_{active})+(p_{sleep}\times t_{sleep})+(p_{tx}\times t_{tx})+(p_{rx}\times t_{rx})}{t_{total}} \ldots\ldots (1)$$

The total operation time could be found using:

$$t_{total} = t_{active} + t_{sleep} + t_{tx} + t_{rx} \ldots\ldots (2)$$

where,
P_{avg}: Average power consumption.
P_{active}: Power consumption in active mode.
P_{sleep}: Power consumption in sleep mode.P_{tx}: Power consumption in transmission mode.
P_{rx}: Power consumption in reception mode.
t_{active}: Time spent in active mode.
t_{sleep}: Time spent in sleep mode.
t_{tx}: Time spent in transmission mode.
t_{rx}: Time spent in reception mode.
total: Total operational time.

The total Energy Consumption (E_{total}) can be easily calculated by:

$$E_{total} = P_{avg} \times t_{total} \ldots\ldots\ldots\ldots (3)$$

Or

$$E_{total} = (P_{active} \times t_{active}) + (P_{sleep} \times t_{sle}) + (P_{tx} \times t_{tx}) + (P_{rx} \times t_{rx}) \ldots\ldots\text{Eq } (4)$$

In general, battery capacity is measured in milliamp-hours (mAh), while energy is either given in joules (J) or watt-hours (Wh). To calculate battery capacity (in mAh) using energy (in joules), the battery voltage (V Battery) should be known:

$$C_{battery\,(mAh)} = \frac{E_{total}\times 1000}{V_{battery}} \ldots\ldots\ldots\ldots (5)$$

Hence, battery life could be estimated by

$$T = \frac{C_{battery}\times V_{battery}}{E_{total}} \ldots\ldots\ldots\ldots (6)$$

where,

T: Battery life in hours.

where,
$C_{battery\,(mAh)}$: Battery capacity in milliamp-hours.
E_{total}: Total energy consumption.
$V_{battery}$: Battery voltage.

Energy consumption calculation helps in estimating the required battery capacity and, at the design stage of the system, ensures reliable performance while maintaining the functionality and sustainability of the devices used.

4.4 Security concerns

To attain a certain level of security in an Intelligent Power Management System (IPMS), several aspects should be considered in the proposed system, such as secure boot, data ciphering, data integrity, and authentication to prevent or at least detect any cyberattack. This could be achieved by implementing effective strategies to reduce possible threats and provide an acceptable level of IoT system security. Possible threats are explained in the following; see Table 3 for more details.

- Unauthorized Access: Stolen credentials and weak passwords may lead to unauthorized access to data.
- Misconfiguration: Keeping default passwords or misconfiguring some settings could expose the system to vulnerabilities like data breaches and unauthorized access.
- Attacks on Server: This could lead to disrupting services, which is often done by Distributed Denial of Service (DDoS) attacks and unauthorized attempts to sign into the server.
- Network Threats: Disrupting network communication could result in data leaks or even service interruptions; network spoofing and packet sniffing are ways to pose such threats.
- Application Layer Threats: Accessing and manipulating data could be done by taking advantage of vulnerabilities within the application itself. Techniques like cross-site scripting (XSS) and SQL injection are used to serve such purposes.
- Lack of Network Monitoring: A belated response to a network breach could prolong the access time for the attackers and cause serious harm to the network.
- Compromising IoT Devices: Gaining control of the device remotely by exploiting vulnerabilities could lead to data leakage or launching further attacks.
- Physical Security: Attackers may access the device's location and tamper with it, allowing them to steal data or alter some of its functionality.

Table 3: Possible security threats and countermeasures for the proposed system.

Threat Name	Threat Type	Countermeasure	Justification
Unauthorized Access	Credential Theft	Credential Theft	Credential Theft
	Applying two-factor authentication (2FA).	Applying two-factor authentication (2FA).	Applying two-factor authentication (2FA).
Misconfiguration	Default Settings	Altering vulnerable settings and default passwords.	Factory settings and passwords are widely known; therefore, they should be changed.
	Open Ports	Scanning for and closing unneeded open ports.	Limiting the attack surface by detecting and closing unneeded ports.
Attacks on Server	DDoS Attacks	Implementing mitigation solutions and rate limiting for DDoS.	Minimizing and managing excessive network traffic by adopting such methods.
Network Threats	Sniffing	Applying Wi-Fi encryption and VPNs.	Disallowing the capture and reading of data.
	Man-in-the-Middle Attack	Adopting data encryption mechanisms such as SSL/TLS encryption.	Ensuring that data is received intact.
Application Layer Threats	Injection Attacks	Assure and check for any user input.	Allowing only correct inputs prevents malicious code injection.
Lack of Network Monitoring	Unnoticed Intrusions	Using Intrusion Detection System (IDS) and Intrusion Prevention System (IPS) solutions.	Preventing and detecting various suspicious activities by continuous monitoring allows for quick responses.
Compromising IoT Devices	Malware	Adopting anti-malware and antivirus techniques.	Detecting and removing harmful codes and programs.
	Firmware Weaknesses	Updating device firmware regularly and applying up-to-date patches	Preventing attackers from exploiting certain security vulnerabilities.
Physical Security	Tampering	Adopting Hardware Security Modules (HSMs) and secure boot processes.	HSMs protect cryptographic keys and data, ensuring that devices run only trusted firmware using secure boot.
	Device Theft	Keeping devices in a well-locked enclosures.	Denying the intruders from physical access to devices.

Generally, considering security recommendations will improve data protection and user credentials. Some good practices include maintaining up-to-date protective software, monitoring and reacting to any suspicious events, and applying multi-layer authentication approaches. Adopting such practices leads to a trustworthy environment for IoT systems. The suggested security practices are detailed in section 5.3.

5. RESEARCH METHODOLOGY

The systematic approach and processes used to conduct our research were divided into five sub-sections. It begins with the embedded hardware layout selection, which presents the required hardware components to build the system. The system's workflow is presented in the 5.2 subsection, while security performance, which discusses the possible threats and suggested solutions, is covered in the 5.3 subsection. Analyzing network performance via the simulation process is advocated in Subsection 5.4. The system performance assessment regarding battery life and security is arranged in the 5.5 sub-section.

5.1 The embedded hardware layout selection

In this section, we explain the layout of the energy management hardware system to protect the solar panel system from overload current consumption and system collapse. Table 4 explains the hardware requirements.

The proposed embedded hardware system is shown in Fig. 3. The appliances in this figure are classified into three classes: Low Priority (L.P.), Medium Priority (M.P.), and High Priority (H.P.). The brain of the suggested system is the Photon (MCU), which is used in the proposed system rather than Arduino for the reasons shown in Table 5. The Photon has a powerful ARM Cortex M3 microcontroller with a Broadcom Wi-Fi chip in a tiny thumbnail-sized module. Our choice of Photon board was made based on its unique compatibility with our cloud services and the simplicity of its integration within our prototype setup; however, the MKR1000 and similar boards may serve the same purpose. The process starts with reading the value of the current sensor through the analog pins. Then, the Photon software program reads the incoming values from all CTs and triggers a control signal for the relay to turn

ON or OFF the device. A voltage divider of 10K ohms and 100K ohms resistors is needed to decrease the voltage to 5V, as the Photon input works on 5V.

In addition, the architecture has three wired devices connected to the Current Transducer (CT) and Relay (R), and multiple wireless connected nodes attached to home appliances with different predefined priority levels.

Table 4. Hardware requirements.

Components	Specifications
Microcontroller	Particle Photon
Interface App.	Blynk App
Current transducer	SCT-013-030 (30Amp)
Resisters	10k ohm and 100k ohm

As shown in the figure, the red color node represents one of the highest priority devices, while the green node color represents one of the lowest priority devices. The CT in each node feeds the current readings to the Photon microcontroller after converting the analogue current reading to a voltage between (0 to 3.3 V), which would be converted to a digital value in the ADC of Photon. After processing the current values, the Photon decides which device can be turned ON and which has to wait in the controlled queue system. The Photon uses the value of the CT to determine the energy needed by each device to control the turning ON or OFF of the required device through a relay.

Table 5: A comparison between Particle Photon and Arduino microcontroller.

Feature	Particle Photon	Arduino Boards
Wi-Fi	Built-in Wi-Fi with TCP/IP support	Needs external Wi-Fi modules (e.g., ESP8266)
Cloud & OTA	Native cloud integration and OTA updates	Requires third-party services and extra setup
Size & Power	Small and power-efficient; ideal for portable projects	Larger size, higher power usage
Performance	120 MHz ARM CPU, 128 KB RAM, 1 MB Flash	16 MHz CPU, 2 KB RAM, 32 KB Flash (e.g., Uno)
Security	Built-in TLS, encryption, and device authentication	No built-in security; needs custom development
Developer Tools	IoT-focused IDE, CLI, mobile app, and API support	General-purpose tools; less support for connected systems

The proposed system prioritizes the devices, preventing them from operating simultaneously to protect the solar panel system from overcurrent, where the devices run sequentially according to precedence. Also, if the power level is low, the lower-priority devices will automatically be turned off while the higher-priority ones will continue to operate. Moreover, if the power level is too low, then all devices will be turned off, and an indication signal will be sent to the user. This leads to the protection of the solar system and efficient distribution of energy between devices.

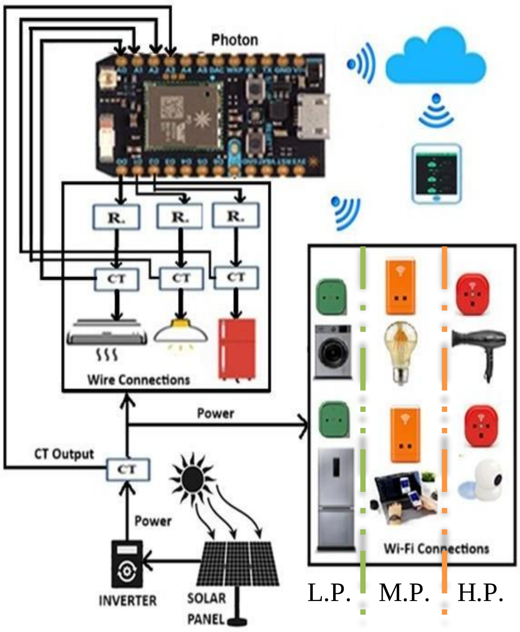

Fig. 3 The proposed embedded hardware system.

Figure 4.a and 4.b show the hardware setup of the proposed system. The solar panel system is connected to the current transducer sensor to read the instant current value that is being supplied to certain devices. The Photon senses the current and voltage value through analog pins. By facilitating these values, the Photon program calculates the instant power and energy used.

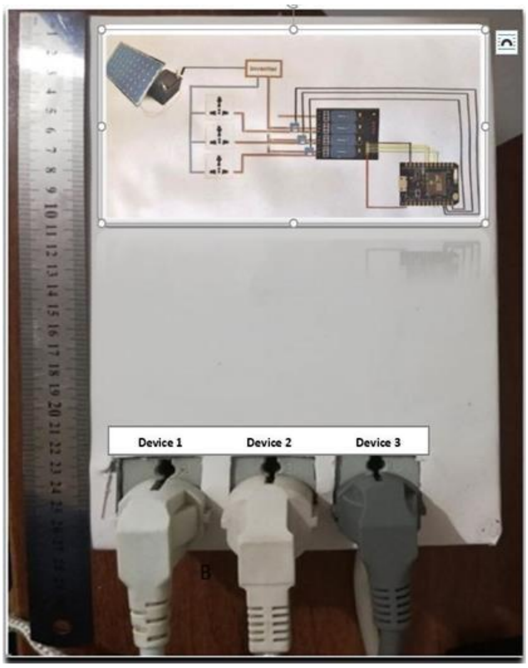

Fig. 4.a Hardware configuration setup - Top Layer

Fig. 4.b Hardware configuration setup – bottom layer

As a matter of fact, Blynk is a platform that is compatible with iOS and Android apps. The authors use the app to control photons (nodes) over the Internet; therefore, photon data, device status, and available power are uploaded to the cloud through the Blynk application. This paper uses Blynk as the interface between the proposed system and the client, as shown in Fig. 5.

Additionally, Blynk can remotely control Photons to turn the connected devices ON or OFF through the Blynk software application.

Fig 5. Monitoring GUI page on cell phone using Blynk app

5.2 The workflow of the proposed work

As shown in Fig. 6, the workflow of the proposed work represents a control algorithm for managing essential devices based on the value of the main current transducer (CT) and a maximum limit (Max) of Amps for each home. It consists of three main phases: Normal Operation, Transient, and Interrupted Operation phases. The detailed steps are as follows:

1. The Particle Photon (MCU) reads the PV main current transducer value. This value represents the total amount of current produced by the PV system.
2. The MCU checks if the value of the current transducer is below the maximum allowable limit (CT < Max?). If the CT value is less than Max. value, then the MCU goes into a loop that tries to ensure that all predefined essential devices (Cooler, Refrigerator, TV, etc) are operational. If not all essential devices are ON, then the MCU tries to activate the next Highest Priority Essential Device (HPED), indicating a need to ensure all critical devices are operated.
3. On the other hand, if the CT current value is greater than the maximum limit value, one of the non-essential devices with High Priority (Red Node Color as shown previously in Fig.3) has been turned ON. In this case, the MCU stores the value of the current transducer in the C1 variable. After that, the MCU turns off the next lowest priority essential device to reduce the total load current and manage resource usage effectively. After adjusting the devices, it rereads the value of the current transducer and stores its value in C2 for further processing. The DevC (Device Current) differs between the C1 and C2 readings. DevC represents the need for the lowest priority device (LPD) to be turned off. This value will help power up the last turned-off LPD in case extra amps are still available.
4. Then the work continues to check if the (CT <MAX) condition to turn off the next lowest priority essential device (Green Node) in case the CT current value is more than the full load limit. When the condition is met, step five will be implemented.

This step checks if (Max - C2) > DevC. This decision point evaluates whether the difference between the maximum load limit of PV and the current transducer value (C2), which presents the available Amps, is greater than the device's current (DevC). So, if the condition is met, then the MCU powers up the next HPED, since the difference is enough to supply the last, to ensure an efficient resource utilization. Else, it reads back again the CT value, stores it in the C2 variable, and checks for any changes.

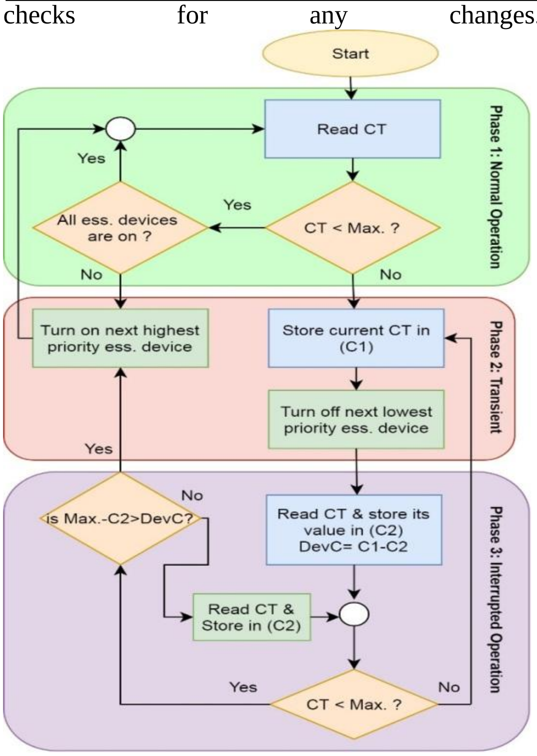

Fig. 6 Workflow of proposed system.

The workflow is designed for dynamic management of device operations based on real-time usage metrics. It aims to prevent exceeding a maximum limit (8 Amps as an example), ensuring efficient resource utilization. In addition, it emphasizes prioritizing essential devices, which suggests that not all devices have the same importance. This prioritization is critical in scenarios where resources are limited. The continuous reading of the value of the current transducer as a feedback mechanism creates a feedback loop that allows the system to adapt to changes in usage, maintaining balance within acceptable limits. Finally, it helps ensure that essential operations continue without exceeding predetermined limits, fostering efficient resource management in home for daily life applications.

Table 6: Possible security threats and countermeasures for the proposed system.

Threat Name	Threat Type	Countermeasure	Justification
Unauthorized Access	Applying two-factor authentication (2FA).	Applying two-factor authentication (2FA).	Applying two-factor authentication (2FA).
Network Threats	Man-in-the-Middle Attack	AES-256 encryption using Crypto library	Encrypts data in transit to prevent interception or tampering.
	Sniffing	AES-256 encryption	Ensures data confidentiality, preventing readable data capture.
Application Layer Threats	Data Integrity	SHA-256 HMAC authentication	Verifies data integrity and authenticity to ensure message was not altered.
Misconfiguration	Default Settings	Altering vulnerable settings and default passwords.	Factory settings and passwords are widely known thereby they should be changed.
	Open Ports	Scanning for and closing unneeded open ports.	Limiting the attack surface by detecting and closing unneeded ports.
Attacks on Server	DDoS Attacks	Implementing mitigation solutions and rate limiting for DDoS.	Minimizing and managing excessive network traffic by adopting such methods.
Lack of Network Monitoring	Unnoticed Intrusions	Using Intrusion Detection System (IDS) and Intrusion Prevention System (IPS) solutions.	Preventing and detecting various suspicious activities by continuous monitoring allows for quick responses.
Compromising IoT Devices	Malware	Adopting anti-malware and antivirus techniques.	Detecting and removing harmful codes and programs.
	Firmware Weaknesses	Updating device firmware regularly and applying up-to-date patches	Preventing attackers from exploiting certain security vulnerabilities.
Physical Security	Tampering	Adopting Hardware Security Modules (HSMs) and secure boot processes.	HSMs protect cryptographic keys and data, also assuring that devices run only trusted firmware by using secure boot.
	Device Theft	Keeping devices in well-locked enclosures.	Denying the intruders from physical access to devices.

5.3 Security Issues

Security is one of the key aspects that any networked system should focus on. In the followings a brief security model is presented.

5.3.1 Proposed security model and analysis

The suggested system should consider several factors, including secure boot, data ciphering, data integrity, and authentication to stop or at least identify any attacks, to achieve a given level of security. This might be accomplished by implementing practical methods to lessen potential risks and offer a respectable degree of security for IoT systems. The following describes potential hazards and countermeasures; refer to Table 6 for further information.

5.3.2 How to implement the suggested IoT security

Codes could be written and applied to the Particle Photon node to implement the aforementioned security model. Writing, compiling, and flashing the code to the node is done using the Particle development tool.

1) The process can be done on your smartphone by downloading the Particle application, creating an account, and connecting a Photon node to the Wi-Fi network. One way to program the node is through the Particle Web IDE. This can be done by accessing build.particle.io to use the online IDE editor that allows you to write, compile, and flash any code to the Photon node using the browser on your phone. The code is written in Wiring, which is a programming language similar to C/C++.2) To apply some security measures like encryption, AES-256 is suggested. This could be done using 'crypto' library, then uploading a code to the node, similar to this code:

#include "Crypto.h"

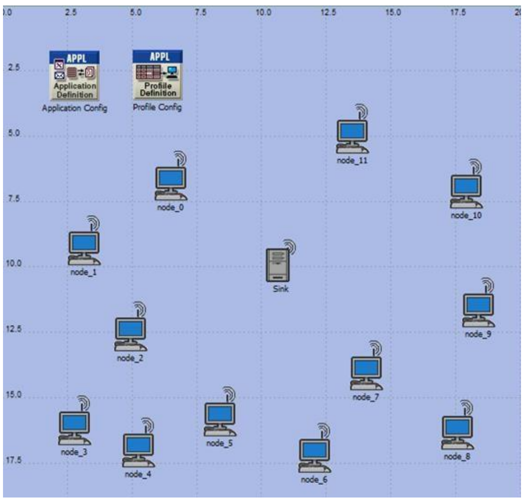

Fig.7 The proposed system network's simulation.

byte key[32] = {….} // 256-bit key (32 bytes)
byte iv[16] = {….} // Initialization Vector (16 bytes)
void setup() {
byte plaintext[16] = "Hello, Particle!"; byte ciphertext[16];
AES aes Encryptor (key, iv, AES::AES_MODE_256, AES::CIPHER_ENCRYPT);
Aes Encryptor. processBlock(plaintext, ciphertext);

1) SHA-256 HMAC could be implemented in the proposed system to implement authentication and integrity features. This is also done using the 'crypto' library by applying code similar to: #include "Crypto. h"

void setup() {
Serial. begin (9600); wait For (Serial.is Connected, 10000);
byte key[32] = {….} // Key for HMAC (32 bytes for SHA-256)
const char* message = "Hello, Particle!";
// Message to hash
byte hmac Result[32]; // Buffer to hold the HMAC output (32 bytes for SHA-256)
HMAC hmac ((uint8_t*)key, sizeof (key)); // Initialize with the key

y applying these security features, messages can be exchanged in the proposed system with minimal security risk as they are encrypted, originate, and remain unaltered.

5.4 Assessment of network performance

The authors utilized the simulation model using OPNET to simulate all the required settings and observe the results, as detailed in sections 5.4.1 and 5.4.2.

5.4.1 Simulation settings

The network diagram is created using the OPNET 14.5 Network Simulator to simulate the proposed system, as shown in Figure 7. As can be seen, the sink node collects data from socket nodes that are distributed randomly. Thus, it is better to place it in the middle of the network to ensure better signal reception from all socket nodes. The simulation settings are shown in Table 7.

The scenario assumes that there are 12 smart socket nodes and a single sink node. Additionally, the readings are sent from each socket node to the sink every 10 ms with a data size of 4 bytes. Meanwhile, the sink node is assumed to send a control packet to switch on/off a certain socket node in case a higher priority device needs to be operated. It is assumed, as a worst-case scenario, that the sink node sends, on average, one control packet every minute. Four different wireless protocols are considered, one for each case.

Table 7: Simulation settings

Parameter	Value
Simulation time	1 hour
Number of nodes	12 socket nodes and one sink node
Network area span	10 meters x 10 meters (100 m2)
Distance between nodes	Unequal
WLAN protocols and data rate	802.11b (5.5Mbps) 802.11b (11Mbps) 802.11g (54Mbps) 802.11a (54Mbps)
Nodes status	Running (with or without the device being plugged)

5.4.2 Simulation results

After running the simulator, using a different WLAN protocol for each run, the main network performance parameters were measured, including latency, throughput, traffic sent/received, packet loss, and jitter. Results show an acceptable value of packet loss for all different scenarios that could be ignored, less than 2%. In addition, the number of network packets was just above 1200 packets/second, resulting in a traffic of about 39,000 bps of actual data being exchanged. The jitter value ranged from 0.03 µsec to 0.7 µsec, depending on the WLAN protocol used, and could be ignored. The throughput was the same for all cases, at around 300 kbps for the network. However, latency was directly affected by the network protocol and data rate, as illustrated in Fig. 8.

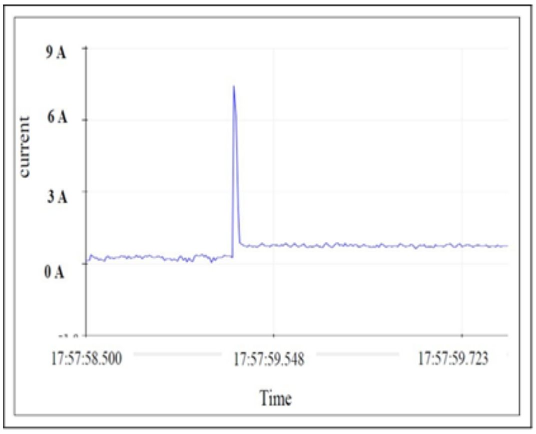

Fig. 8 WLAN protocol and data rate against network latency.

As can be seen in Fig. 8, the latency is affected by the data rate of the WLAN, and it is inversely proportional to it. This means that acting in a closer to real-time manner requires a higher network data rate.

5.5 System performance analysis

For the suggested system, its performance could be measured by considering the following assumptions in Table 8:

Table 8: Performance evaluation assumptions.

Parameter	Value
Power in active mode	100 mA
Power in sleep mode	1 mA
Power in transmission mode	200 mA
Power in reception mode	80 mA
Time spent in active mode	60 seconds
Time spent in sleep mode	58 Minutes
Time spent in transmission	30 seconds
Time spent in reception mode	30 seconds
Battery Voltage	5V

According to the above values for a 1-hour operational time, the following results are obtained when considering the aforementioned equations (Eq.1 to Eq.6); see Table 9.

Table 9: Performance evaluation results.

Parameter	Value
Average power consumption	4.97 mW
Total energy consumption	17.88 Joules
Battery capacity	3576 mAh
Battery life	1000 hours (41.67 days)

The above mathematical results indicate the efficient lifetime of the system, even in the case of no battery charging. For the proposed system, the batteries are charged once the plug is inserted into the socket. This is a sustainable scenario as the solar panels are being used.

6. RESULTS AND DISCUSSION

The proposed work illustrates the Solar Energy Monitoring System results as an example of a limited energy resource. Fig. 9 shows the required current for the first device to start up and the current in the steady state. It is clear from the figure that the starting current measurement is about six times higher than the steady-state current. Furthermore, we can clearly note that there is a ripple in the curve. The last part represents the start of operation, where the device needs a higher current to start up until it returns to the steady state.

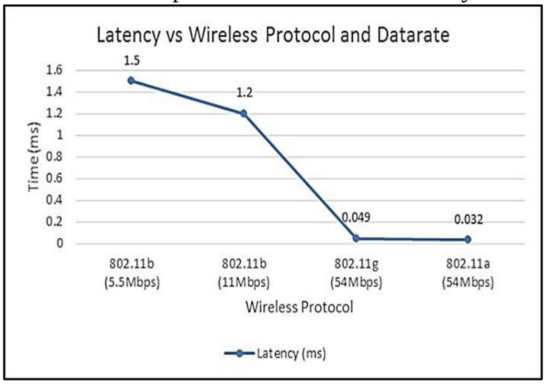

Fig. 9 The current response for the first device.

Figure 10 shows the current requirement for the second device to start up and the current requirement in the steady state. Also, we can note a ripple in the curve. This ripple represents the start of operation when the device needs a high current and then returns to the steady state.

When comparing the previous two figures, we notice that the first device needs 6.71 A at the beginning of the operation, while the second device needs 5.26 A at the beginning. So, if these devices work simultaneously, they will overload the solar panel inverter (10 Amps); therefore, the proposed system protects the solar panel system from entering this undesired state.

As shown in Fig. 11, the current startup trajectory of three devices, which are controlled and monitored by the particle photon microcontroller, has been designed in our proposed system.

The Photon starts to turn on the devices sequentially, beginning with the first device, which has a higher priority, and then checks the current until reaching a steady state. If there is a problem, such as the device cannot reach the steady state automatically, then the Photon will cut off the power to that device. Consequently, after 4 seconds, the next device will start up, and so on.

Figure 12 shows the difference between the two cases. The first case does not use a current management controller, and all devices work simultaneously. The second case represents the proposed system of turning the devices ON sequentially. As shown in the figure, the starting current for all the devices in the case rises to a maximum point of 16A. Meanwhile, the maximum current is only 6.5A for the same devices when they work in a sequence programming order.

This facility is beneficial in solar systems because the solar system has startup power limitations. So, the proposed system protected the solar panel system from overload issues.

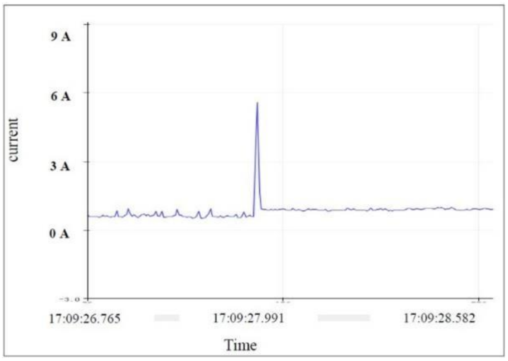

Fig. 10 The current response of the second device

Additionally, the proposed design uses a smart strategy to prioritize different loads in the system. By incorporating the presented smart switch using a Particle Photon microcontroller, we can better understand the various loads and their significance to the customers, effectively assigning importance to each load within the network. The last ensures that maintaining a reliable power supply without interruptions is a top priority for customers, helping to protect essential devices from damage and reduce maintenance costs.

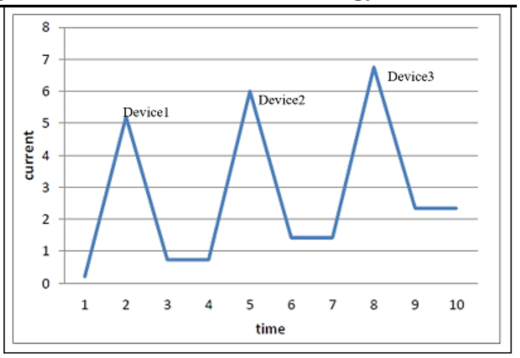

Fig. 11 Sequential starting current of three devices

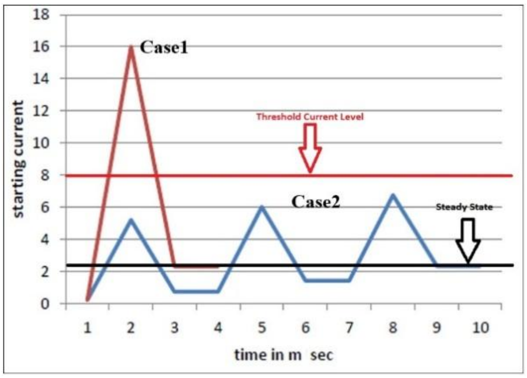

Fig. 12 Comparison between the Sequential and simultaneous starting current of three devices

Ultimately, this system facilitates the efficient utilization of limited energy sources by distributing privileges and prioritization among different kinds of loads in a household. This approach can be applied in residential construction and factories, hospitals, and government institutions as building blocks of smart, modern, and green cities.

7. CONCLUSIONS

In this paper, the authors presented a smart energy management system that leverages the Internet of Things (IoT) to optimize the use of limited residential power sources such as solar panels or local generators. Our approach focuses on privileging and prioritizing energy distribution to enhance the sustainability and efficiency of green city initiatives.

Many challenges in obtaining a smart and effective building system were overcome in this IoT-designed system, such as interoperability, real-time data analysis, data privacy, usability, cost-effectiveness, and scalability. By employing advanced scheduling strategies and implementing robust security measures, our system ensures reliable energy management and protection of solar panel infrastructure. Using a networked embedded system design allows for the dynamic allocation of energy resources, minimizing wastage and maximizing the lifespan of solar panels or local generators.

Integrating IoT capabilities provides users with a seamless and interactive interface for monitoring and controlling their energy usage, contributing to a more sustainable and efficient energy ecosystem. Our findings suggest that further research into expanding the scalability and cost-effectiveness can contribute significantly to developing smart and sustainable urban environments. Regarding energy storage, backup solutions, the user can add more backup batteries to the system and update some conditions to the algorithm to help in a comprehensive residential energy management strategy. The simulation results clearly measure system latency, packet loss, and jitter. However, the OPNET simulation can be expanded to include complex, realistic scenarios, such as burst traffic patterns, multi-hop communications, and larger networks (up to 500 household devices across 50 homes to evaluate scalability) for future research. Additionally, more security investigations, including actual deployment and evaluation, may be conducted. Finally, the proposed system successfully offers more flexible, scalable, energy-conserving, and money-saving solutions for smart cities.

REFERENCES

[1] C. Srinivasan, B. Rajesh, P. Saikalyan, K. Premsagar, and E. S. Yadav, "A review on the different types of Internet of Things (IoT)," *Journal of Advanced Research in Dynamical and Control Systems,* Vol. 11, Issue: 1, pp. 154-158, 2019.

[2] A. Y. Mohammed and R .M. Hagem, "Low cost portable system for converting mosul electrical substations to smart one's," *Al-Rafidain Engineering Journal (AREJ),* Vol. 26, Issue: 2, pp.323-339,2021.https://doi.org/10.33899/rengj.2021.130620.1115

[3] Q. Ibrahim and M. Qassab, "Theory, Concepts and Future of Self Organizing Networks (SON)," *Recent Advances in Computer Science and Communications (Formerly: Recent Patents on Computer Science),* Vol. 15, Issue: 7, pp.904-928, Sep.2022.https://doi.org/10.2174/2666255814666210201102326

[4] S. F. Hamid, A. N. B. Alsammak, and K. T. Atta, "Using Solar Photovoltaic Systems, Battery Energy Storage Systems, and Underfrequency Load-Shedding to Improve the Frequency Stability of Power Systems," *Al-Rafidain Engineering Journal (AREJ),* Vol. 28, Issue: 1, pp. 165-172, March 2023.https://doi.org/10.33899/rengj.2022.136061.1201

[5] A. Qazi, F. Hussain, N. A. Rahim, G. Hardaker, D. Alghazzawi, K. Shaban, *et al.*, "Towards sustainable energy :a systematic review of renewable energy sources, technologies, and public opinions," *IEEE access,* Vol. 7, pp. 63837-63851,2019. ttps://doi.org/10.1109/ACCESS.2019.2906402

[6] Farosyid. (2018, December 15). *Energi Terbarukan vs Energi Non-Terbarukan*. Kata Pengetahuan. https://katapengetahuan.wordpress.com/2018/12/14/energi-terbarukan-vs-energi-non-terbarukan/

[7] Y. G. Rashid, F. M. Tuaimah, and M. Salem, "Contingency Assessment for Power System Integrated with Wind Power", *Diyala Journal of Engineering Sciences (DJES),* Vol. 16, Issue: 4, pp. 59–71, Dec. 2023. http://dx.doi.org/10.24237/djes.2023.160405

[8] A. Kekre and S. K. Gawre, "Solar photovoltaic remote monitoring system using IOT," *International conference on recent innovations in signal processing and embedded systems (RISE),* 2017, pp. 619-623. https://doi.org/10.1109/RISE.2017.8378227.

[9] R. M. Hagem, D. V. Thiel, S. G. O'Keefe, N. Dahm, A. Stamm, and T. Fickenscher, "Smart optical wireless sensor for real time swimmers feedback," in *SENSORS, 2012 IEEE*, 2012, pp. 1-4. https://doi.org/10.1109/ICSENS.2012.6411208.

[10] R. Hagem, D. Thiel, S. O'Keefe, and T. Fickenscher, "Real-time swimmers' feedback based on smart infrared (SSIR) optical wireless sensor," *Electronics Letters,* Vol. 49,pp.340-341,2013, https://doi.org/10.1049/el.2012.3222.

[11] B. M. H. Alhafidh, R. M. Hagem, and A. I. Daood, "Face detection and recognition techniques analysis," in *2022 International Conference on Computer Science and Software Engineering (CSASE)*, 2022, pp. 265-270, https://doi.org/10.1109/CSASE51777.2022.9759573.

[12] B. S. Abdullah and S .Y. Ameen, "Off-Grid Photovoltaic System for a Villa at AVRO City in Duhok," *AL-Rafidain Engineering Journal (AREJ),* Vol. 28, Issue: 1, March 2023 https://doi.org/10.33899/RENGJ.2022.134729.11

[13] A. Zipperer, P. A. Aloise-Young, S. Suryanarayanan, R. Roche, L. Earle, D. Christensen, *et al.*, "Electric energy management in the smart home: Perspectives on enabling technologies and consumer behavior," *Proceedings of the IEEE,* Vol. 101, pp. 2397-2408, 2013, https://doi.org/10.1109/JPROC.2013.2270172.

[14] Z. Abbas and W. Yoon, "A survey on energy conserving mechanisms for the internet of things: Wireless networking aspects," *Sensors,* Vol. 15, Issue 10, pp. 24818-24847, 2015, https://doi.org/10.3390/s151024818.

[15] J. Kim, "HEMS (home energy management system) base on the IoT smart home," *Contemporary Engineering Sciences,* Vol. 9, pp. 21-28, 2016, http://dx.doi.org/10.12988/ces.2016.512316.

[16] M. V. Moreno, B. Úbeda, A. F. Skarmeta, and M. A. Zamora, "How can we tackle energy efficiency in IoT based smart buildings?," *Sensors,* Vol. 14, Issue: 6, pp. 9582-9614, 2014, https://doi.org/10.3390/s140609582

[17] R. Arshad, S. Zahoor, M. A. Shah, A. Wahid, and H. Yu, "Green IoT: An investigation on energy saving practices for 2020 and beyond," *Ieee Access,* Vol. 5, pp. 15667-15681, 2017 DOI: 10.1109/ACCESS.2017.2686092.

[18] V. K. S. Princy S, Aiswarya C ,Sujith A H ,Sithara Krishnan, "IoT Based Smart Home Using Renewable Solar Energy," *International Research Journal of Engineering and Technology (IRJET)*, Vol. 06, Issue: 4, pp. 4333-4335, 2019. Retrieved from https://www.irjet.net/archives/V6/i4/IRJET-V6I41294.pdf

[19] B. Mataloto, J. C. Ferreira, and N. Cruz, "LoBEMS—IoT for building and energy management systems," *Electronics*, Vol. 8, Issue: 7, p. 763, 2019, https://doi.org/10.3390/electronics8070763.

[20] M. M. Ibrahim and A. M. Elwany, "Design and installation off-grid solar electrification system for residential school building in Egypt: future of green energy," *Energy Systems*, pp. 1-28, 2025. https://doi.org/10.1007/s12667-025-00740-9.

[21] P. Sankhwar, "Energy reduction in residential housing units" *International Journal of Advanced Research*, vol. 12, Issue: 8, A, pp. 1-15. 2024, https://doi.org/10.21474/ijar01/19298.

[22] C. Krishna 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, p. 20230015, 2024, https://doi.org/10.1515/ehs-2023-0015.

[23] A. Wren, "Scheduling, timetabling and rostering—a special relationship?," in *International conference on the practice and theory of automated timetabling*, pp. 46-75, 1995, https://doi.org/10.1007/3-540-61794-9_51.

[24] A. Shewale, A. Mokhade, N. Funde, and N. D. Bokde, "A survey of efficient demand-side management techniques for the residential appliance scheduling problem in smart homes," *Energies*, Vol. 15, p. 2863, 2022, https://doi.org/10.3390/en15082863.

[25] S. N .Makhadmeh, A. T. Khader, M. A. Al-Betar, S. Naim, A. K. Abasi, and Z. A. A. Alyasseri, "Optimization methods for power scheduling problems in smart home: Survey," *Renewable and Sustainable Energy Reviews*, Vol. 115, p. 109362, 201, https://doi.org/10.1016/j.rser.2019.109362.

عمليات إعطاء الأفضلية والأولوية لموارد الطاقة المستدامة للأحمال السكنية في المدن الخضراء

بسمان محمود حسن الحافظ *
bm.alhafidh@uomosul.edu.iq

ربيع موفق حاجم *
rabeehagem@uomosul.edu.iq

مصطفى سهام قصاب*
mustafa.qassab@uomosul.edu.iq

محمد باسل شكر**
mohammed.basil@ntu.edu.iq

* قسم هندسة الحاسوب، كلية الهندسة، جامعة الموصل، الموصل، العراق

** قسم هندسة الأمن السيبراني وتقنيات الحوسبة السحابية، الكلية التقنية الهندسية للحاسوب والذكاء الاصطناعي، الجامعة التقنية الشمالية، الموصل، العراق

تاريخ الاستلام : 9 مارس 2025 استلم بصيغته المنقحة : 28 يونيو 2025 تاريخ القبول: 27 يوليو 2025

الملخص

إن حماية وإدارة أنظمة طاقة الألواح الشمسية السكنية في المدن الذكية التي تعتمد على إنترنت الأشياء (IoT) أمر حيوي. يعد التحكم في استهلاك الطاقة السكنية يدويًا أمرًا صعبًا. بالإضافة إلى ذلك، يجب معالجة التحديات الكبيرة، بما في ذلك تسهيل الاتصال الفعال بين الأجهزة وإجراء تحليل البيانات في الوقت الفعلي وضمان الأمان. بمعنى آخر، يجب التغلب على القيود المرتبطة بالتشغيل البيني وتحليل البيانات في الوقت الفعلي وسهولة الاستخدام وقابلية التوسع والفعالية من حيث التكلفة في أي تصميم لنظام إدارة موثوق. في هذه الورقة، يتم تقديم نظام إدارة طاقة ذكي يعتمد على إنترنت الأشياء لتحسين كفاءة وسلامة حالة الألواح الشمسية السكنية. يركز النهج المقترح على إعطاء الأولوية لفصل الأحمال، والذي يلعب دورًا حاسمًا في تعزيز ممارسات الطاقة المستدامة في المدن الذكية. تم تصميم النظام على أساس Particle Photon والذي يتضمن متحكمًا دقيقًا مع Wi-Fi ومحولات التيار (CTs) لإدارة استخدام الطاقة ديناميكيًا، مما يساعد على تقليل الهدر وإطالة عمر مصادر الطاقة المحدودة مثل الألواح الشمسية أو مولدات الطاقة المحلية. يُحسب استهلاك الطاقة لنظام الإدارة المقترح بافتراض سعة بطارية 3576 مللي أمبير/ساعة، وعمر بطارية يصل إلى 1000 ساعة. يُستخدم مُحاكي الشبكة OPNET 14.5 لمحاكاة النظام المقترح. ويُقترح استخدام منصة Blynk لنشر البيانات على السحابة، والتي يمكن حفظها لمزيد من التحليل. كما يُقترح تقديم توصيات بشأن خصوصية البيانات وأمنها. ونتيجةً لذلك، يُمكن لأصحاب المصلحة مراقبة استهلاكهم للطاقة والتحكم فيه بسلاسة من خلال واجهة جوال سهلة الاستخدام. ويُظهر البحث أن دمج تقنية إنترنت الأشياء بهذه الطريقة يُعزز إدارة الطاقة ويدعم تطوير بيئات حضرية أكثر خضرةً واستدامة.

الكلمات الداله:

الاتصالات اللاسلكية المدمجة، نظام إدارة الطاقة الذكية، المدن الذكية، الألواح الشمسية، فصل الاحمال.