SMATIKA : STIKI Informatika Jurnal
Accredited Sinta 4 Ministry of Higher Education, Science and Technology Republic of Indonesia SK No. 79/E/KPT/2023
E-ISSN: 2580-6939 || P-ISSN: 2087-0256

Design and Implementation of an ESP32-Based Smart Home Electricity Monitoring and Control Module Using MQTT Dash and Telegram Integration

Muh. Asnoer Laagu[1*], Bayu Retno Joyo Nendriyo[2], Iftiqori Fiqi Anshori[3], Immawan Wicaksono[4], Ali Rizal Chaidir[5]

[1,2,3,4,5] *University of Jember, Faculty of Engineering, Department Electrical Engineering, Kalimantan Street No. 37, Krajan Timur, Sumbersari, Sumbersari District, Jember Regency, East Java 68121, Indonesia*

Keyword

ESP32; Energy Monitoring; Iot; MQTT; Smarthome

**Corresponding Author:*

asnoer@unej.ac.id

Abstract

This study presents the design and implementation of an ESP32-based smart home box module that integrates multi-channel load control, real-time electrical energy monitoring, and automated user reporting within a single compact and modular system. The proposed module employs an eight-channel relay for household appliance control using the MQTT communication protocol, with MQTT Dash providing real-time visualization and manual control. Electrical parameters, including voltage, current, active power, and cumulative energy consumption (kWh), are measured using a PZEM-004T sensor and transmitted continuously via MQTT. The main contribution of this work lies in the unified system architecture that consolidates device control, energy measurement, and user interaction without reliance on proprietary cloud platforms. In addition to dashboard-based monitoring, a Telegram bot is integrated to deliver real-time system status notifications, on-demand energy reports summarizing average power and total consumption, and fully automated 24-hour periodic reports. This dual interaction mechanism enables both active real-time supervision and passive long-term monitoring, addressing limitations commonly found in prior IoT smart home implementations. Experimental evaluations confirm that the proposed system operates reliably, responsively, and with acceptable measurement accuracy for residential applications. All relays on the eight-channel module respond stably to ON/OFF commands from the MQTT Dash interface with delays of less than one second, ensuring dependable real-time control. The PZEM-004T-based monitoring subsystem provides voltage, current, power, and energy data with power measurement errors ranging from 0.83% to 2.66%, which is suitable for daily household energy monitoring. Tests under various single and combined load conditions demonstrate consistent detection of power variations. Overall, the results indicate that the proposed smart home box module provides a practical, low-cost, and scalable solution for integrated IoT-based residential energy control and monitoring.

Cite: Laagu, et al., (2026). Design and Implementation of an ESP32-Based Smart Home Electricity Monitoring and Control Module Using MQTT Dash and Telegram Integration. SMATIKA : STIKI Informatika Jurnal,16(2). doi: https://doi.org/10.32664/smatika.v16i02.2321

1. Introduction

The development of Internet of Things (IoT) technology has driven innovations in home automation (smarthome). Through IoT, electrical devices can be controlled, monitored, and integrated in real time via the internet. This system offers increased convenience, energy efficiency, and more adaptive electrical device management capabilities than conventional systems. Several studies have shown that implementing IoT in smart homes can facilitate centralized device control and energy monitoring at a relatively low cost compared to commercial devices [1]. The application of Internet of Things (IoT) technology in smart home systems has been widely studied due to its potential to improve convenience, energy efficiency, and centralized device management. Previous research demonstrates that microcontroller-based solutions, particularly those using ESP32, can provide cost-effective and flexible alternatives to commercial smart home systems by enabling real-time control and monitoring over the internet [2]. Several studies have focused specifically on energy monitoring, employing sensors such as the PZEM-004T to measure electrical parameters including voltage, current, power, and energy consumption with acceptable accuracy for residential applications [3] .Other works emphasize reliable and low-latency device control through lightweight communication protocols such as MQTT, which is widely adopted due to its efficiency in constrained IoT environments [4]. In addition, messaging platforms such as Telegram have been explored as user interfaces for remote interaction, notifications, and simple control mechanisms due to their accessibility and cross-platform support [13][5].,

However, despite these advancements, most existing approaches are implemented as function-specific solutions, where control, energy monitoring, and user interaction are treated as separate or loosely integrated subsystems rather than as a cohesive platform [6]. Telegram-based implementations are typically limited to event notifications or basic commands and rarely support structured or automated energy reporting, while energy monitoring systems often lack direct actuator control capabilities [7],[8]. On the other hand, commercial smart home products attempt to integrate these functionalities but are generally expensive, proprietary, and dependent on closed ecosystems or cloud services, reducing flexibility, customization, and scalability for users and developers [9].

Consequently, there remains a gap in the development of an open, low-cost, and fully integrated smart home solution that combines multi-channel device control, real-time energy monitoring, and flexible user interaction within a single deployable system. This study addresses these limitations by proposing an ESP32-based smart home box module that consolidates eight-channel relay control, PZEM-004T-based energy monitoring, MQTT-based real-time communication, dashboard-level control via MQTT Dash, and Telegram bot integration for both manual and automated 24-hour energy reporting. By unifying these functionalities, the proposed system bridges the gap between isolated IoT prototypes and inflexible commercial systems, offering a practical and scalable solution for residential smart home applications.

Based on these needs, this research developed an ESP32-based smart home box module that combines 8-channel relay control, energy monitoring using the PZEM-004T sensor, MQTT communication, a control interface via the MQTT Dash application, and Telegram bot integration for manual and automated reporting. With its flexible design and lower cost compared to commercial smart home devices, this module is expected to be an alternative solution for both home users and IoT system developers, consistent with prior findings that commercial smart home products remain comparatively expensive and inaccessible for many households, motivating low-cost microcontroller-based alternatives [19].

2. Research Method

2.1 Research Design

This study adopts an experimental research methodology to design, implement, and evaluate an IoT-based smart home electricity monitoring and control system. The system architecture is based on the four-layer IoT model, consisting of sensing, processing, communication, and application layers. The PZEM-004T sensor functions as the sensing unit to measure electrical parameters such as voltage, current, power, and energy consumption, while the ESP32 microcontroller serves as the central processing unit responsible for data acquisition, control logic, and communication. Data transmission between devices is achieved using the

MQTT protocol, which enables efficient and real-time communication through a publish–subscribe mechanism [11]. The system is integrated with two user interfaces: MQTT Dash for real-time monitoring and relay control, and a Telegram bot for notification, manual reporting, and automated 24-hour energy reporting. The experimental setup includes an ESP32, PZEM-004T sensor, an 8-channel relay module, and various household electrical loads tested under single and combined conditions. System performance is evaluated based on control responsiveness (relay delay), measurement accuracy (comparison of sensor readings with reference values), communication reliability (stability and latency of MQTT transmission), and overall system functionality (dashboard and Telegram features). Validation is carried out through functional and performance testing, including verification of relay operation, consistency of sensor measurements across load variations, reliability of data transmission, and correctness of user interaction via MQTT Dash and Telegram. The system is considered valid if it demonstrates stable operation, acceptable measurement accuracy for residential use, and reliable real-time control and reporting capabilities.

2.2 System Design

MQTT is widely used in IoT systems because of its lightweight architecture and low communication overhead, making it suitable for resource-constrained devices such as ESP32. However, the use of MQTT in smart home environments also introduces security challenges, particularly related to unauthorized access, data interception, and device manipulation over the network. Previous studies have emphasized that IoT-based smart home systems should incorporate secure communication protocols, authentication mechanisms, encryption methods, and access control policies to ensure system reliability and protect user privacy [5]. For instance, one IoT smart home network architecture addressed these requirements by deploying a dedicated authentication, authorization, and accounting (AAA) server alongside AES-128 encryption for data integrity protection, illustrating a more comprehensive security approach than the basic username/password broker authentication adopted in the present implementation [12]. These issues become increasingly important when IoT systems are capable of remotely controlling physical devices in residential environments. The system design employs an IoT-based Smart Home Box Module, which integrates components such as the relay module, sensors, and communication interfaces to enable control and monitoring. This architecture facilitates seamless data flow between hardware and software layers. The 8-channel relay module functions as an electronic switch, opening and closing circuits controlled via an ESP32 using the MQTT protocol and the MQTT Dash application. The PZEM-004T sensor is used to monitor electrical parameters connected to the relay module circuit. Parameters displayed include voltage, current, power, and energy, which are displayed in real time in the MQTT Dash application or can be requested via the Telegram Bot [11]. An automatic report feature is sent via the Telegram Bot every 24 hours, containing average power usage and total energy consumption over the past 24 hours. Although the proposed system demonstrated reliable functionality for smart home control and energy monitoring, several security-related limitations remain. The current implementation employs basic MQTT broker authentication using username and password but does not yet implement TLS/SSL encryption or topic-based Access Control Lists (ACLs). Consequently, the system is more suitable for deployment in private or controlled network environments. Future work will focus on enhancing communication security through encrypted MQTT transmission, stronger authentication mechanisms, and fine-grained access control to improve protection against unauthorized access and cyber threats in IoT-based smart home applications. The system flowchart is shown in Figure 1, and the pin mapping is shown in Figure 2 below.

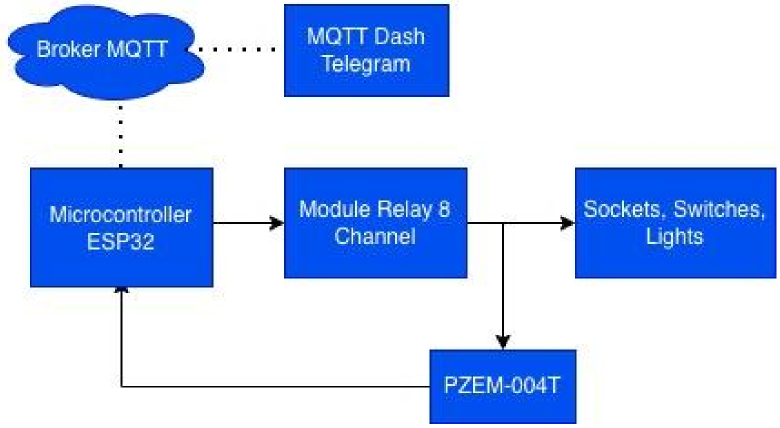

Figure 1. The Flowchart of System

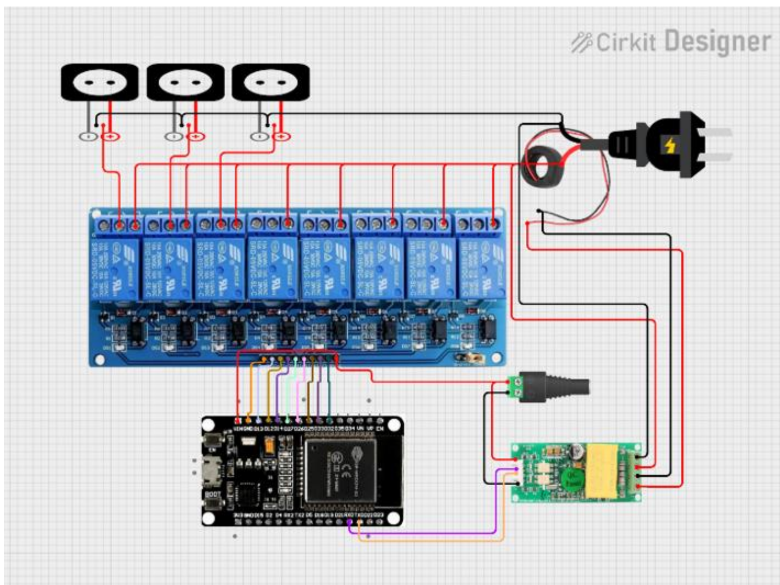

Figure 2. Pin Mapping of The System

After designing the main system diagram, the initial scenario is to initialize the entire system, starting with the serial port, relay pin, WiFi client, and MQTT client. Once all systems have been properly initialized, the next step is to connect to the WiFi set and MQTT server, along with callbacks. In this main system, MQTT will always reconnect, ensuring continuous operation. Once MQTT is connected, the relay can be controlled on the MQTT Dash. MQTT Dash readings are delayed by 2 seconds for PZEM sensor readings. The sensor readings are then sent to MQTT, which then reads the total power, total energy, and total sample statistics. The resulting daily average power and total energy reports are then sent to Telegram. Figure 3 below shows a flowchart of the main system, focusing on ESP32 core 1.

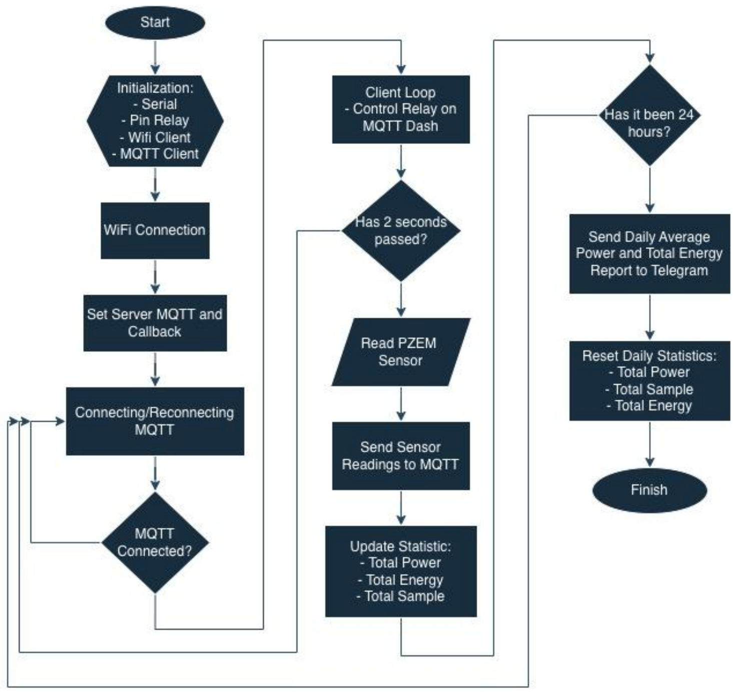

Figure 3. The Flowchart of Main System (Core 1 ESP32)

The main system will send the daily average total power and energy reports to the Telegram notification system. The system begins by initializing the bot token and chatID. Once the WiFi connection is established, any incoming messages with the chatID will be verified for validity. If the chatID has been verified as valid, the Telegram bot will send 3 types of messages, namely /start, /status, and /report. If the user sends a Telegram bot message with the /start command, Telegram will provide 2 options, namely /status and /report. The /status message will send data to the user regarding voltage, current, power, and energy data. The /report message will send data regarding the average daily power and total energy in the system. The Telegram notification system diagram on the system (core 0 ESP32) can be shown in Figure 4.

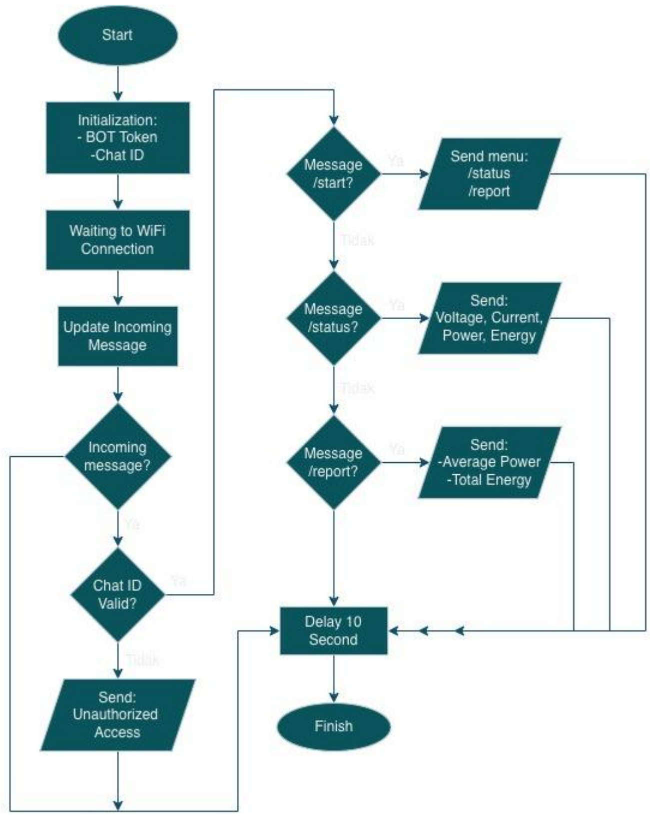

Figure 4. The Flowchart of Telegram Notification System (Core 0 ESP32)

3. Result and Discussions

The tests were conducted with several different load variables to verify the MQTT connection and the integration between the control system and electrical parameter monitoring, ensuring system reliability [2]. The test configuration is shown in Table 1 below.

Table 1. The Configuration of MQTT Dash, Module Relay and Load

MQTT Dash	Module Relay	Load Item	Load Capaity (W)
Button 1	Channel 1	Phone Charger	12
Button 2	Channel 2	Blue Light	20
Button 3	Channel 3	Red Light	15
Button 4	Channel 4	-	-
Button 5	Channel 5	-	-
Button 6	Channel 6	-	-
Button 7	Channel 7	-	-
Button 8	Channel 8	-	-

Table 2 compares the proposed system with several related IoT-based smart home and energy monitoring studies. Compared with previous Telegram-based smart home systems, the proposed system provides additional capabilities through integrated electrical parameter monitoring and automated energy reporting [8]. Likewise, previous ESP32- and PZEM-004T-based monitoring systems primarily focused on energy monitoring, whereas the proposed system integrates real-time energy monitoring with MQTT-based relay control within a unified platform [2]. Unlike existing systems that mainly emphasize either monitoring or control, the proposed module combines MQTT-based real-time control, multi-channel relay operation, energy monitoring, and Telegram-based notification and reporting into a single integrated solution. Experimental results demonstrate stable relay response times of less than 1 s under various load configurations, including a 12 W phone charger, a 20 W blue light, and a 15 W red light, while maintaining acceptable power measurement accuracy for household applications.

Table 2. The Comparation of the proposed system with several related IoT-based smart home and energy monitoring studies

Study	Relay Control Delay	Mesurement Error	Energy Monitoring	MQTT-Based Control	Telegram Integration	Automatic Reporting	Number of Channel
Hutajulu et al. [8]	< 2 s	Not evaluated	No	No	Yes	No	4
Yasa et al. [2]	Not discussed	0.35%–1.45%	Yes	No	No	No	1
Esposito et al. [17]	~1 s	Not discussed	Partial	Yes	No	No	Multi-device
Proposed System	< 1 s	0.83%–2.66%	Yes	Yes	Yes	24-hours automatic report	8

Table 2 highlights that the proposed system achieves a relay response time of less than 1 s, which is comparable to previously reported MQTT-based smart home systems and faster than Telegram-based implementations that reported response times below 2 s [8], [17]. This performance is likely attributable to the lightweight MQTT publish–subscribe architecture and the use of a local Wi-Fi network, which minimizes communication latency. In terms of energy monitoring, the proposed system achieved power measurement errors ranging from 0.83% to 2.66%, which are acceptable for residential energy monitoring applications. Although lower measurement errors of 0.35%–1.45% have been reported in previous ESP32- and PZEM-004T-based monitoring systems [2], the slightly higher error observed in the present study is likely influenced by differences in load characteristics, supply voltage fluctuations, sensor calibration, wiring conditions, and measurement timing. This pattern is broadly consistent with an independent validation study that benchmarked the PZEM-004T against a reference smart meter (EVM430-F6736) across multiple load types, reporting power measurement errors between 0.37% and 2.21% for purely resistive loads [20], a range that overlaps closely with the lower end of the error observed in the present study, reinforcing that the PZEM-004T provides accuracy suitable for residential energy monitoring despite the absence of a reference-grade calibration setup in the present experimental design. By comparison, a separate evaluation of a PZEM-004T-based kWh monitoring prototype tested against nameplate specifications across ten household appliances reported a substantially higher average deviation of 15.8%, with only 45% of tests achieving deviations of 5% or less [16], indicating that the 0.83%–2.66% error range achieved in the present study compares favorably against PZEM-004T deployments evaluated under different testing protocols and reference standards. In terms of communication reliability, the present system maintained stable MQTT transmission throughout testing, which contrasts with findings from a separate ESP32-MQTT-AWS monitoring implementation that reported a 16.67% data transmission failure rate under stress-test network conditions despite achieving 100% reliability under normal conditions [10], suggesting that transmission reliability under load remains a common challenge across MQTT-based IoT monitoring systems regardless of the cloud or local broker architecture employed. Overall, the results demonstrate that the proposed system provides a balanced trade-off between communication responsiveness, monitoring accuracy, and integrated smart home functionality.

Nevertheless, the PZEM-004T sensor consistently detected power variations under both single and combined load conditions, demonstrating its suitability for practical smart home applications. Compared with previous studies, the main added value of the proposed system lies in its integration of real-time device control, energy monitoring, and Telegram-based communication within a single platform. While prior Telegram implementations were generally limited to remote control or event notifications — for instance, one home security system used Telegram Bot solely to deliver intrusion alerts and to issue basic start/stop commands, without any structured or periodic energy- or status-reporting functionality [14], the proposed system extends this functionality by providing real-time status updates, on-demand energy reports, and automated 24-hour energy consumption summaries. These features improve user accessibility and energy-awareness without requiring continuous access to a dashboard application. From both practical and technical perspectives, the results demonstrate that integrating eight-channel relay control, MQTT communication, energy monitoring, and automated reporting into a single ESP32-based module provides a cost-effective and flexible alternative to commercial smart home systems while maintaining reliable control performance and acceptable measurement accuracy for residential deployment.

The relay control experiment using MQTT Dash to receive ON/OFF commands aimed to measure the delay in command execution, highlighting system responsiveness. The results of the experiment are shown in Table 3 below.

Table 3. The Measurement of Delay Relay Control

Button	Module Relay	Status	Delay
1	Channel 1	ON	0.63 s
2	Channel 2	ON	0.56 s
3	Channel 3	ON	0.49 s
1	Channel 4	OFF	0.43 s
2	Channel 5	OFF	0.89 s
3	Channel 6	OFF	0.58 s

Data collection was conducted through five experiments, including testing with three different loads and recording voltage, current, and power values displayed on MQTT Dash to evaluate system performance. The results of the experiment can be seen in Table 4.

Table 4. The Monitoring of measurement on MQTT Dash

No	Load Item	Volt (V)	Ampere (A)	Power (W)	Error
1.	No Load	0	0	0	0%
2.	Phone Charger	220.30	0.090	12.10	0.83%
3.	Blue Light	221.10	0.081	18.20	2,41%
4.	Red Light	221.40	0.068	15.40	2.66%
5.	Phone Charger, Blue Light, Red Light	221.20	0.219	45.90	2.34%

Furthermore, we document the fifth experiment, where the cell phone charger, blue light, and red-light loads were controlled by buttons and PZEM-004T sensor readings displayed on MQTT Dash. The results of the experiment can be seen in Figure 5 below.

Figure 5. Documentation of Experimental Results

The developed smart home system uses an 8-channel relay to control various electrical devices such as lights and cell phone chargers. Control is performed through the MQTT Dash application, where each relay channel has its own thread for receiving ON/OFF commands. Table 2 shows that test data demonstrates that all relays can respond to commands in real time without significant delay, typically less than 1 second. This consistent performance reassures the audience about system stability and reliability, essential for trust in practical applications.

The PZEM-004T sensor reads electrical parameters including voltage (V), current (A), power (W), and energy (Wh). The read data is automatically sent to MQTT Dash and recorded by the Telegram bot system to display real-time status and automatic daily reports. Measurement results under various load conditions are presented in Table 3. For example, in the second data point, the charger power is 12.10 W, consistent with the load specifications in Table 1. The low error value indicates that the sensor readings are highly accurate under low loads. In general, the PZEM is capable of producing stable and consistent data, with power errors ranging from 0.83% to 2.66%. This error value is still reasonable for an IoT device not intended for commercial metering purposes.

The developed system not only provides control and monitoring via MQTT Dash but also features automatic and manual reporting via Telegram, offering flexible remote management [18]. The average power report feature accumulates data during device operation, giving users a comprehensive view of daily power consumption. This versatility enables users to feel more in control and confident in managing their energy use effectively.

Figure 6. Average power and total energy consumption report

Furthermore, the daily total energy feature, calculated based on kWh readings from the sensor, provides direct information on the actual amount of energy consumed in a 24-hour period. A manual report option is available to determine how much energy has been used without waiting for the 24-hour period. The system also performs an automatic energy reset after the daily report is sent, providing users with separate, clean data for each day. This way, users can monitor the efficiency of electrical devices and changes in electricity consumption from day to day.

Overall, these additional features through Telegram expand the capabilities of the smart home system to be more interactive, informative, and accessible, while making the energy monitoring system much more comprehensive than conventional methods — for example, a conventional ESP32-based energy monitoring system relying on the Blynk platform with WhatsApp notifications provides real-time monitoring but lacks structured on-demand reporting and has acknowledged scalability limitations for larger deployments [15], in contrast to the structured /status and /report commands, automatic 24-hour summaries, and energy-reset functionality offered by the proposed system.

4. Conclusions and Future Works

Based on the design, implementation, and testing results, the Smarthome Box Module system and IoT-based electricity monitoring system using ESP32 and MQTT have proven to be effective, responsive, and accurate in controlling and monitoring energy. All relays on the 8-channel module operate stably, receiving ON/OFF commands from the MQTT Dash application with a delay of less than 1 second, which should reassure users about system reliability. The PZEM-004T-based monitoring system successfully displays voltage, current, power, and energy parameters with a fairly good level of accuracy, as evidenced by power errors ranging from 0.83% to 2.66%, which is still suitable for daily monitoring needs. The obstacle encountered during system testing was the instability of the internet connection, resulting in an increase in the error percentage of up to 2.66%. we assumed three relays were sufficient to obtain preliminary data. Test results on various load variations indicate that the sensor is able to consistently detect power changes, both for single and combined loads. Integration with a Telegram bot further enhances the system's functionality, allowing users to obtain real-time status, manual reports, and automatic reports every 24 hours.

Overall, the designed system demonstrates reliable, economical, and flexible performance, inspiring confidence in its potential as a practical smarthome solution that is easy to implement in household environments to save energy and improve user comfort. A limitation of this research is that the number of relays is limited to eight channel modules and the number of load testing. The three tested channels were considered representative because all relay channels in the module share the same hardware architecture, control logic, power configuration, and MQTT communication mechanism. Each channel is driven by the same ESP32 control scheme and relay driver circuit, meaning that the operational behavior of one channel is theoretically identical to the others under similar load conditions. Therefore, the observed relay response time, MQTT communication performance, and integration behavior can reasonably represent the functionality of the remaining channels. We acknowledge that simultaneous testing of all eight channels under varying load combinations would provide a more comprehensive evaluation of system scalability, stability, thermal performance, and potential communication bottlenecks. Consequently, this limitation has now been explicitly stated in the manuscript. The current results mainly validate the functional feasibility and reliability of the proposed architecture rather than full-scale stress performance of all channels operating concurrently. Future work will therefore include comprehensive multi-channel loading scenarios and long-duration testing to further evaluate system robustness under maximum operational conditions. System integration with artificial intelligence (AI) could also support daily load predictions, given the large amount of data generated and implemented in the smarthome system.

5. References

[1] M. Asvial, A. Cracias, M. Laagu, and A. S. Arifin, "Design and Analysis of Low Power and Lossy Network Routing System for Internet of Things Network," *Int. J. Intell. Eng. Syst.*, vol. 14, no. 4, 2021, doi: 10.22266/IJIES2021.0831.48.

[2] K. A. Yasa, I. M. Purbhawa, I. M. S. Yasa, I. W. Teresna, A. Nugroho, and S. Winardi, "IoT-based Electrical Power Recording using ESP32 and PZEM-004T Microcontrollers," *J. Comput. Sci. Technol. Stud.*, vol. 5, no. 4, pp. 62–68, 2023, doi: 10.32996/jcsts.2023.5.4.7.

[3] S. Khan, H. Ali, and Z. Shah, "Systematic Analysis of Smart Homes: Current Trends and Future Recommendations," *Cogent Eng.*, 2024, doi: 10.1080/23311916.2024.2344452.

[4] S. Watmah, "Pemantauan Catu Daya untuk Proteksi Listrik Menggunakan ESP32 Berbasis Internet of Things (IoT)," *IMTechno J. Ind. Manag. Technol.*, vol. 6, no. 2, pp. 94–99, 2025, doi: 10.31294/imtechno.v6i2.9518.

[5] M. F. Zia and others, "Securing the Future: A Survey on Smart Home Security in IoT," *Adv. Networks*,

2025, doi: 10.11648/j.net.20251201.11.

[6] M. L. U. A. Usi'il Arzaq, A. Budi Laksono, and U. Ilmi, "Perancangan Sistem Monitoring Peralatan Listrik Pada Peternakan Ayam Berbasis Internet of Things (IoT)," *J. FORTECH*, vol. 6, no. 2, pp. 93–98, 2025.

[7] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions," *Futur. Gener. Comput. Syst.*, vol. 29, no. 7, pp. 1645–1660, 2013, doi: 10.1016/j.future.2013.01.010.

[8] O. Y. Hutajulu and others, "Feasibility Study of Smart House Design with ESP32 and Telegram," in *EAI Conference Proceedings*, 2022. doi: 10.4108/eai.11-10-2022.2325332.

[9] M. T. Muslihi, "Pengembangan dan Evaluasi Sistem Monitoring Konsumsi Daya Listrik Berbasis IoT dengan Sensor PZEM-004T dan ESP8266," *J. FASILKOM*, vol. 15, no. 1, pp. 77–83, 2025, doi: 10.37859/jf.v15i1.8508.

[10] C. Austin, M. Mulyadi, and S. Octaviani, "Implementasi IoT dengan ESP32 Untuk Pemantauan Kondisi Suhu Secara Jarak Jauh Menggunakan MQTT Pada AWS," *J. Elektro*, vol. 15, no. 2, pp. 46–55, 2022, doi: 10.25170/jurnalelektro.v15i2.5141.

[11] A. Banks and R. Gupta, "MQTT Version 3.1.1," 2014.

[12] A. S. Ibrahim, A. M. Abbas, A. M. A. Hassan, W. M. Abdel-Rehim, A. Emam, and S. Mohsen, "Design and Implementation of a Pilot Model for IoT Smart Home Networks," *IEEE Access*, vol. 11, pp. 59701–59728, 2023, doi: 10.1109/ACCESS.2023.3282095.

[13] F. P. Putra, S. Sabirin, and H. Soetanto, "Prototype of Internet of Things-Based Control System Using Telegram with Bot API Method," *J. Syntax Transform.*, vol. 6, no. 2, pp. 87–109, 2025, doi: 10.46799/jst.v6i2.1055.

[14] M. N. Osman, M. H. F. Ismail, K. A. Sedek, N. A. Othman, and M. Maghribi, "A Low-Cost Home Security Notification System Using IoT and Telegram Bot: A Design and Implementation," *J. Comput. Res. Innov.*, vol. 7, no. 2, pp. 327–337, 2022, doi: 10.24191/jcrinn.v7i2.325.

[15] H. J. El-Khozondar, S. Y. Mtair, K. O. Qoffa, O. I. Qasem, and A. H. Munyarawi, "A Smart Energy Monitoring System Using ESP32 Microcontroller," *e-Prime - Adv. Electr. Eng. Electron. Energy*, vol. 9, p. 100666, 2024, doi: 10.1016/j.prime.2024.100666.

[16] N. P. Putra, "IoT-based kWh Monitoring Prototype using PZEM-004T Sensor," *J. Emerg. Innov. Eng.*, 2025, doi: 10.65664/jeie.v1i02.11.

[17] M. Esposito, A. Belli, L. Palma, and P. Pierleoni, "Design and Implementation of a Framework for Smart Home Automation Based on Cellular IoT, MQTT, and Serverless Functions," *Sensors*, vol. 23, no. 9, 2023, doi: 10.3390/s23094459.

[18] D. N. Bestari and A. Wibowo, "IoT-Based Real-Time Monitoring System Using Telegram Bot," *Int. J. Interact. Mob. Technol.*, vol. 17, no. 6, 2023, doi: 10.3991/ijim.v17i06.34129.

[19] B. Mustafa, M. W. Iqbal, M. Saeed, A. R. Shafqat, H. Sajjad, and M. R. Naqvi, "IOT Based Low-Cost Smart Home Automation System," in *2021 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, 2021, pp. 1–6, doi: 10.1109/HORA52670.2021.9461276.

[20] J. A. Medina Molina, B. Arzate-Rueda, E. Reyes-Archundia, and J. C. Olivares-Rojas, "Comparison of Electrical Measurements between Different Devices for Smart Meter Applications," *Glob. J. Eng. Technol. Adv.*, vol. 19, no. 3, pp. 71–78, 2024, doi: 10.30574/gjeta.2024.19.3.0089.