CoreID JOURNAL

Vol. 4 No. 1, March 2026 (38-45)
ISSN: 2987-6990
DOI: https://doi.org/10.60005/coreid.v4i1.160

Performance Analysis of MQTT and HTTP Protocols on Low-Power ESP32 Devices for IoT Applications

Syahrul Adriansyah[1], Sumarno[2], Ulya Mutiara Nurfuha[3], Abidzar Giffari[4], Tsulsi Khoirunisa[5], Winky Gunsan Subarkah[6]
[1,3,4,5,6]Informatics Department, UIN Sunan Gunung Djati Bandung, Indonesia
[2]UIN Sultan Aji Muhammad Idris Samarinda, Indonesia

Article Info

Article history:

Received April 27, 2026
Revised May 07, 2026
Accepted May 13, 2026

Keywords:

ESP32
HTTP
Internet of Things
MQTT
Low-Power Device

ABSTRACT

The choice of communication protocol plays a crucial role in determining the efficiency and operational lifetime of low-power Internet of Things (IoT) devices such as ESP32-based systems, particularly in Smart Home applications. This study conducts a Systematic Literature Review (SLR) following the PRISMA 2020 guidelines to analyze and compare the performance of MQTT and HTTP protocols on ESP32 microcontrollers, focusing on latency, power consumption, protocol overhead, network resilience, and security efficiency. From 166 initial records, 9 empirical studies met the inclusion criteria and were synthesized in the final review. The findings consistently show that MQTT outperforms HTTP across multiple performance dimensions. MQTT demonstrates significantly lower latency, ranging from 4.76× to 12.1× faster response times under high traffic and remote server conditions. In terms of energy efficiency, MQTT achieves approximately 6–8% lower power consumption than HTTP, contributing to longer battery life in IoT deployments. MQTT also imposes lower protocol overhead, especially for small-payload sensor communication and secure transmission scenarios using TLS. Furthermore, MQTT maintains more stable performance under constrained network conditions compared to HTTP. Although HTTP remains suitable for applications prioritizing compatibility and simplicity, the review concludes that MQTT is the more efficient and reliable communication protocol for resource-constrained ESP32-based IoT systems requiring low latency, energy efficiency, and robust network performance.

Corresponding Author:

Syahrul Adriansyah
Informatics Department, UIN Sunan Gunung Djati Bandung, Indonesia
Jl. A.H. Nasution No. 105 Cibiru Bandung, Indonesia, 40614
Email: 1247050038@student.uinsgd.ac.id

1. INTRODUCTION

The rapid development of the Internet of Things (IoT) has revolutionized home automation, with low-power microcontrollers such as the ESP32 serving as the backbone of smart home systems. While HTTP/REST-based communication models are widely used, their request–response architecture leads to increased energy consumption and latency due to continuous polling. To overcome these limitations, the Message Queuing Telemetry Transport (MQTT) protocol has emerged as a promising alternative.

Theoretically, MQTT provides higher efficiency, minimal data overhead, and improved responsiveness through its lightweight publish–subscribe architecture, making it an ideal candidate for power-sensitive IoT applications. Despite these theoretical advantages, a significant research gap remains in the current literature: while many studies have examined the implementation of either MQTT or HTTP independently, comprehensive empirical comparisons directly evaluating their practical performance on specific ESP32 hardware remain highly limited . There is a pressing need to synthesize existing data that explicitly compares critical metrics—particularly power consumption, latency, and throughput—between these two protocols in resource-constrained environments [1].

To address this gap, this study conducts a Systematic Literature Review (SLR) to synthesize existing evidence on the performance of both protocols in IoT applications using ESP32 devices. The findings are expected to provide evidence-based guidance for developers in selecting the most efficient communication protocol for smart home IoT development.and syntheses of the literature that directly compare the practical performance—particularly regarding power consumption, latency, and throughput—between MQTT and HTTP on specific ESP32 hardware.

Therefore, this study aims to present a Systematic Literature Review (SLR) that analyzes existing empirical data to definitively compare and quantify the performance advantages of MQTT over HTTP in the context of low-power device usage for IoT applications. The results are expected to provide evidence-based guidance for developers in selecting the most efficient communication protocol for future ESP32-based smart home IoT systems [2].

In recent years, the ESP32 microcontroller has become one of the most widely utilized platforms in the development of Internet of Things (IoT) systems, particularly in the domain of smart homes. The popularity of the ESP32 stems from its integration of Wi-Fi and Bluetooth modules and its relatively efficient power consumption. Several studies in Indonesia indicate that ESP32 is frequently used in home automation systems, such as lighting control and environmental monitoring, due to its ease of implementation and compatibility with various network-based IoT services [1].

Nevertheless, the use of conventional communication protocols such as HTTP on low-power IoT devices still presents several technical challenges. The request–response communication pattern of HTTP requires devices to actively send periodic data requests, leading to increased processor workload and higher energy consumption [3][4]. Such conditions make HTTP less suitable for smart home systems that demand fast response times and long-term power efficiency. Therefore, a lighter and more adaptive communication approach is needed to optimize the capabilities of ESP32-based devices.

Therefore, a lighter and more adaptive communication approach is needed to optimize the capabilities of ESP32-based devices.

To address these challenges and the identified research gap, this study makes the following key contributions:

1. It provides a comprehensive Systematic Literature Review (SLR) that consolidates empirical data comparing MQTT and HTTP protocols specifically on ESP32 microcontrollers.
2. It systematically analyzes critical performance metrics, including energy consumption, latency, throughput, and protocol overhead in IoT application contexts.
3. It delivers evidence-based guidelines and practical recommendations for IoT developers to select the most efficient communication protocol for resource-constrained smart home systems.

2. METHOD

2.1. Search Strategy

This systematic review followed PRISMA 2020 (Preferred Reporting Items for Systematic Reviews and Meta-Analyses) framework to ensure a transparent, rigorous, and resproducible methodology. The PRISMA framework implemented in this study consists of four main phases, yaitu Identifying potential records through a comprehensive search across selected academic databases, Reviewing the titles and abstracts of the identified records to remove irrelevant studies and duplicates, Assessing the full-text articles of the screened records against strict, predefined inclusion and exclusion criteria, and Finalizing the subset of studies that meet all criteria for qualitative data extraction and synthesis.

For the initial identification phase, a comprehensive literature search was conducted on January 27, 2026, across three major academic databases and repositories:

- **SciSpace**: A multidisciplinary academic search engine covering peer-reviewed journals, conference proceedings, and preprints
- **Google Scholar**: A broad academic search engine indexing scholarly literature across disciplines and formats
- **ArXiv**: A preprint repository for physics, computer science, mathematics, and related fields

The search query was designed to capture studies comparing MQTT and HTTP protocols on ESP32 devices for IoT applications: "Performance Analysis of MQTT and HTTP Protocols on Low-Power ESP32 Devices for IoT Applications". This query was applied consistently across all three databases to identify relevant empirical studies, comparative analyses, and performance evaluations. No date restrictions were applied to maximize coverage of the available literature. The search was limited to English-language publications due to resource constraints for translation and validation.

2.2. Inclusion and Exclusion Criteria

Articles included in this review were selected based on predetermined inclusion and exclusion criteria to align with the research objectives. Inclusion criteria were used to ensure that each analyzed study was relevant to the performance comparison of the MQTT and HTTP protocols on ESP32 devices in an Internet of Things (IoT) environment. Selected studies had to meet several requirements: (1) discuss the performance comparison of the MQTT and HTTP protocols or their variants, such as MQTT-TLS and HTTPS; (2) use an ESP32 microcontroller or its derivatives as a testing platform; (3) be applied to IoT application contexts, such as smart homes, environmental monitoring, or sensor systems; and (4) present quantitative performance measurement results, such as latency, power consumption, throughput, packet loss, or protocol overhead.

Meanwhile, several types of studies were excluded from the review process. Studies that only used platforms other than the ESP32, such as the ESP8266, Arduino, Raspberry Pi, or STM32, were not included in the analysis. Studies that only discussed a single protocol without comparing MQTT and HTTP were also not considered. Additionally, articles that focused solely on system implementation, architectural design, or conceptual discussion without including empirical performance testing were excluded. Studies outside the IoT context, literature review articles, and simulation-based research without real-world hardware validation were also excluded.

2.3. Study Selection Process

The article selection process was conducted in stages, following a systematic literature review approach. The initial stage began with article identification through searches of the SciSpace, Google Scholar, and ArXiv databases using predetermined keywords. This search yielded 166 articles. Duplicate articles were then removed before proceeding to the screening stage.

During the screening stage, the titles and abstracts of all articles were reviewed to assess their compliance with the inclusion and exclusion criteria. Articles deemed relevant then proceeded to the eligibility stage, a comprehensive assessment of the article's content (full-text review). This stage evaluated the research methods, hardware platforms used, the types of protocols compared, and the availability of performance data for analysis.

The final selection process yielded nine articles that met all criteria and were used in the data synthesis process. To maintain consistency and reduce potential bias, the selection process was conducted using predetermined assessment guidelines based on the research objectives.

2.4. Data Extraction and Synthesis

Data from each selected article was systematically extracted to obtain information relevant to the research objectives. The collected information included study characteristics, the type and variant of ESP32 used, communication protocol configurations, network test scenarios, and performance measurement results such as latency, power consumption, throughput, and protocol overhead. Furthermore, the limitations of the studies presented by each author were also recorded for evaluation.

The collected data were then analyzed using narrative synthesis. This approach was chosen because variations in experimental design, measurement methods, and test scenarios between studies precluded direct quantitative meta-analysis. The results from each study were compared based on the same performance category to identify patterns, differences in results, and factors influencing MQTT and HTTP performance on ESP32 devices. This approach provided a comprehensive overview of the advantages and limitations of each protocol in ESP32-based IoT implementations.

3. RESULTS AND DISCUSSION

3.1. Study Selection

The article selection process was conducted in stages according to the Systematic Literature Review (SLR) method using the PRISMA flow. Search results in the SciSpace, Google Scholar, and ArXiv databases yielded 166 articles relevant to the research keywords. A filtering process was then conducted to eliminate articles that did not align with the research objectives.

Most of the eliminated articles were studies that did not use the ESP32 platform as the primary testing device. A further 140 articles used other platforms such as the ESP8266, Arduino, Raspberry Pi, NodeMCU, or other common IoT devices, thus failing to meet the study's inclusion criteria. Furthermore, several articles discussed only one protocol, either MQTT or HTTP, without conducting a direct performance comparison, making them unusable in the analysis.

Several other articles were also excluded because they did not present empirical performance test results. These articles generally only discussed system implementation, architectural design, or IoT communication concepts without comparable measurement data. Additionally, a small number of articles were eliminated due to the use of simulations without real-world hardware validation, incomplete methodological details, or inaccessibility of the full text.

After completing all selection steps, nine articles met all inclusion criteria and were used in the analysis. These articles provide empirical data comparing the performance of the MQTT and HTTP protocols on ESP32 devices in various Internet of Things (IoT) application scenarios.

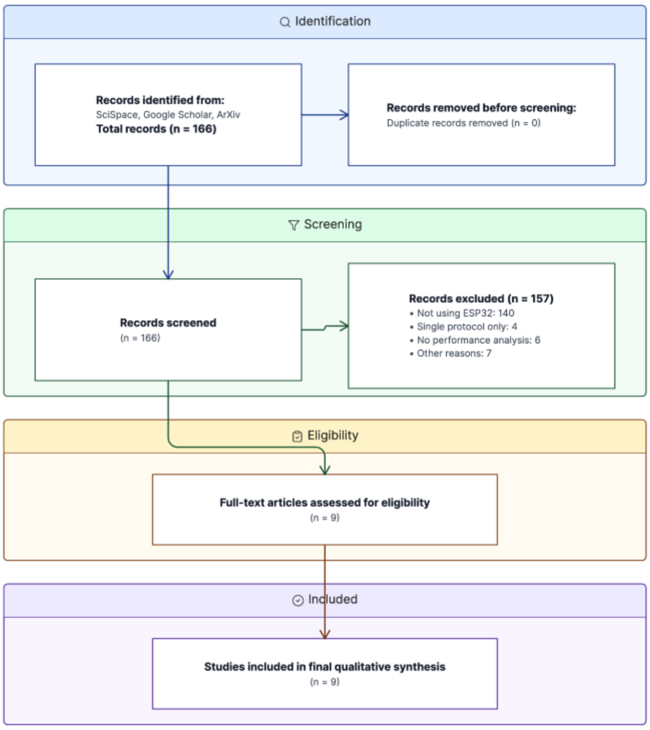

Figure 1. PRISMA 2020 flow diagram showing the study selection process.

3.2. Study Characteristics

Table 1 summarizes the characteristics of the 9 included studies. Publication years ranged from 2018 to 2026, with the majority (6 studies, 66.7%) published in 2023 or later, reflecting growing research interest in ESP32-based IoT protocol performance. Studies employed diverse experimental designs including controlled laboratory experiments, field deployments, and hybrid approaches combining real hardware with network emulation.

Table 1. Characteristics of included studies (n=9)

Study	Year	ESP32 Variant	Application Domain	Study Design	Key Metrics
Rassekhnia [1]	2026	ESP32-C6	IoT/IIoT with PQC	Lab experiment with Raspberry Pi server	Latency, CPU, energy, key size, TLS overhead
Kartha et al. [2]	—	ESP32	General IoT architectures	Hardware validation, low-power setup	Implementation, performance, difficulties

Study	Year	ESP32 Variant	Application Domain	Study Design	Key Metrics
Balamanikandan et al. [3]	—	ESP32	Sensor data transmission	Real hardware benchmarking	Latency, throughput
Study [4]	2023	ESP32-S3	AIoT/IIoT Industry 4.0	Real hardware, sensor scenario	Power consumption, data transfer
Cusin [6]	—	ESP32-C6	General IoT	Comparative protocol study	Latency, energy, network/application layers
Silva et al. [9]	2023	ESP32	Environmental sensing	Power meter measurements	Power, message delay, message length
Luthfi et al. [18]	2018	IoT device	Air quality control	Real device testing	Latency, current consumption
Hanif et al. [19]	2023	IoT device	Flood detection	Performance measurement	Throughput, delay, packet loss
Rönty [27]	—	ESP32	Crane data access	Embedded system, MQTT broker hosting	Low-power setup, IoT context

Note: Some studies did not explicitly report publication year or specific ESP32 variant in available metadata.

Hardware configurations varied across studies, with most employing standard ESP32 development boards, while more recent studies utilized newer variants (ESP32-C6 with RISC-V architecture and Wi-Fi 6 support, ESP32-S3 with enhanced AI capabilities). Application domains spanned environmental monitoring, industrial IoT (IIoT), smart home automation, flood detection systems, and general-purpose sensor networks, demonstrating the breadth of ESP32 deployment scenarios.

3.3. Performance Metrics Analysis

3.3.1. Latency and Response Time

Latency measurements consistently favored MQTT over HTTP across all included studies that reported this metric. The magnitude of the latency advantage varied substantially based on network topology, server placement, and traffic conditions.

The most dramatic latency differences were observed in fog/cloud deployment scenarios. One study reported HTTP response times 12.1 times higher than MQTT for fog server placement and 4.76 times higher for cloud server placement under high-traffic conditions [5]. These findings demonstrate that the latency gap between protocols widens significantly when servers are deployed remotely rather than on local networks, and when network load increases. The stateless nature of HTTP, requiring complete request-response cycles for each transaction, imposes greater overhead under these conditions compared to MQTT's persistent connection model [6].

Cryptographic protocols substantially increased latency for both MQTT and HTTP, with HTTPS and TLS variants showing particularly pronounced overhead. Studies examining post-quantum cryptography (PQC) implementations found that HTTPS incurred higher latency increases compared to MQTT-TLS under equivalent security configurations . The choice of cryptographic algorithms also materially affected message delay on ESP32 devices, with stronger or more complex algorithms imposing greater latency penalties [3]. These findings highlight the trade-off between security and performance, with HTTP-based protocols bearing disproportionate overhead when cryptographic protection is required [7][8].

Several studies reported latency measurements for specific IoT application scenarios. In air quality control systems, MQTT demonstrated lower latency compared to HTTP for sensor data transmission [18]. Similarly, flood detection systems showed reduced delay with MQTT relative to HTTP [19]. Benchmarking studies on ESP32 microcontrollers consistently reported lower latency for MQTT across various sensor data transmission scenarios [9][10].

3.3.2. Power and Energy Consumption

Power consumption represents a critical performance dimension for battery-powered IoT devices, and the included studies provide consistent evidence of MQTT's energy efficiency advantages over HTTP.
A long-duration field test comparing MQTT and HTTP on a NodeMCU platform (ESP8266-based, closely related to ESP32) found that MQTT saved approximately 6.03% power at QoS level 0 and 8.33% power at QoS level 1 compared to HTTP [4]. While these percentage savings may appear modest, they translate to meaningful battery life extensions in long-term deployments. The study's field testing methodology and extended duration provide high ecological validity for these findings.

Experiments specifically measuring ESP32 devices demonstrated that cryptographic algorithm choices materially affect energy consumption for both protocols [3]. Stronger or more complex cryptographic algorithms increased power consumption, with the magnitude of increase varying by algorithm type and implementation. Under post-quantum cryptography and TLS conditions, MQTT was judged more energy-efficient while HTTPS provided stronger security at higher energy cost [2]. This finding reinforces the security-efficiency trade-off, with MQTT maintaining better energy performance even when cryptographic overhead is added[11].

The power consumption advantage of MQTT stems from several architectural factors: persistent TCP connections eliminate repeated handshake overhead, smaller message framing reduces transmission time and energy, and the publish-subscribe model allows devices to remain in low-power states between messages rather than maintaining active polling connections as often required with HTTP.

Studies examining power consumption in specific application contexts reported consistent findings. Environmental sensing applications using ESP32 with power meters showed MQTT consumed less power than HTTP [9]. Industrial IoT scenarios using ESP32-S3 demonstrated that communication protocol choice significantly impacted power consumption, with MQTT showing advantages [4][12][13].

3.3.3. Throughput and Bandwidth Efficiency

Throughput measurements, while reported less frequently than latency and power metrics, generally favored MQTT over HTTP. Benchmarking studies on ESP32 microcontrollers reported higher throughput for MQTT compared to HTTP in sensor data transmission scenarios [3]. Flood detection systems measured throughput differences between MQTT and HTTP, with MQTT demonstrating superior performance [19].
The throughput advantage of MQTT derives primarily from reduced protocol overhead. MQTT's binary encoding and minimal fixed headers (2 bytes) result in smaller message sizes compared to HTTP's text-based headers and verbose request-response structure. This efficiency becomes particularly pronounced for small payloads typical of IoT sensor data, where HTTP header overhead can exceed the actual data payload.[1][14]

3.3.4. Protocol Overhead and Message Size

Protocol overhead—the ratio of protocol control information to actual application data—represents a key efficiency metric for resource-constrained IoT devices. The included studies consistently demonstrated that MQTT imposes lower overhead than HTTP, with the gap widening when security protocols are added.

HTTPS and TLS handshakes add substantial message length and computational overhead compared to MQTT-TLS [2], [3]. Post-quantum cryptography implementations further increased overhead, with HTTPS showing greater increases in key size and TLS handshake overhead compared to MQTT [1]. The study by Rassekhnia measured specific overhead metrics including key size and TLS handshake overhead for MQTT, HTTP, and HTTPS under post-quantum cryptography conditions, providing quantitative evidence of HTTPS's disproportionate overhead burden [1][15].

Environmental sensing applications using ESP32 devices measured additional message length imposed by cryptographic algorithms, finding that HTTP incurred greater overhead than MQTT [9]. The overhead differences become particularly significant in bandwidth-constrained networks or high-frequency data transmission scenarios where cumulative overhead substantially impacts network utilization and energy consumption [13][6].

3.3.5. Reliability and Packet Loss

Reliability metrics, including packet loss rates and successful delivery percentages, were reported in a subset of included studies. MQTT demonstrated advantages in reliability, particularly under challenging network conditions. Studies comparing protocols in fog/cloud scenarios found that MQTT carried data with lower packet loss compared to HTTP [1]. Flood detection systems measured packet loss for both protocols, with MQTT showing superior reliability [16][1].

MQTT's reliability advantages stem from its Quality of Service (QoS) mechanisms, which provide explicit delivery guarantees (at most once, at least once, exactly once) and automatic retransmission at the application layer. HTTP, lacking built-in QoS mechanisms, relies on TCP's transport-layer reliability, which may be insufficient for applications requiring guaranteed message delivery or ordered processing[10][17][18].

4. CONCLUSION

This study conducted a Systematic Literature Review (SLR) to analyze and compare the performance of the MQTT and HTTP protocols on ESP32 devices in the context of Internet of Things (IoT) applications. Based on a literature selection process using the PRISMA 2020 guidelines, nine articles met the inclusion criteria and were used in the analysis. The study results indicate that MQTT generally performs better than HTTP in various testing areas.

In terms of latency, MQTT provides lower response times than HTTP, especially in high-traffic network conditions or when using remote servers. Regarding power consumption, several studies have shown that MQTT is more energy efficient, making it more suitable for use on battery-powered IoT devices. Furthermore, MQTT has lower protocol overhead due to its lightweight publish-subscribe communication mechanism, making it more efficient for sending small, periodic sensor data.

In limited network conditions, such as low bandwidth or unstable connections, MQTT also demonstrates better reliability than HTTP. The use of Quality of Service (QoS) mechanisms in MQTT helps maintain stable data communications without significantly increasing resource consumption. From a security perspective, implementing MQTT-TLS is considered more efficient than HTTPS because it results in lower additional overhead while maintaining a high level of data communication security. Nevertheless, HTTP can still be used in IoT applications that do not have specific requirements for energy efficiency, low latency, or bandwidth optimization. HTTP also offers advantages in terms of compatibility and ease of integration with conventional web services.

Based on the results of this study, it can be concluded that MQTT is a more suitable communication protocol for ESP32-based IoT implementations, particularly in systems that require power efficiency, real-time communication, and optimal network utilization. However, the number of studies meeting these criteria is still limited, so further research with more diverse test scenarios is needed, particularly regarding protocol evaluation in large-scale IoT implementations and the use of the latest security technologies..

REFERENCES

[1] C. Austin, M. Mulyadi, and S. Octaviani, "Implementasi IoT dengan ESP 32 Untuk Pemantauan Kondisi Suhu Secara Jarak Jauh Menggunakan MQTT Pada AWS," *J. Elektro*, vol. 15, no. 2, pp. 46–55, Jan. 2024, doi: 10.25170/jurnalelektro.v15i2.5141.

[2] B. Khoeriah Utami, M. Rizqi Fajri, and A. Dwi Chaerani, "Jurnal Sains dan Teknologi Sistem Internet Of Things (IoT) Untuk Pemantauan Kualitas Udara Dalam Ruangan," *J. Sains Dan Teknol.*, vol. 02, no. 02, pp. 25–29, 2025.

[3] A. A. Fikhri, M. Ula, M. Sayuti, Taufik, and Nurdin, "Perbandingan Kinerja Protokol MQTT dan HTTP Dalam Komunikasi Data Internet of Things Artikel Penelitian," *J. Infomedia Tek. Inform. Multimedia, dan Jar.*, vol. 10, no. 1, pp. 1–10, 2025.

[4] H. J. Jara Ochoa, R. Peña, Y. Ledo Mezquita, E. Gonzalez, and S. Camacho-Leon, "Comparative Analysis of Power Consumption between MQTT and HTTP Protocols in an IoT Platform Designed and Implemented for Remote Real-Time Monitoring of Long-Term Cold Chain Transport Operations," *Sensors*, vol. 23, no. 10, p. 4896, May 2023, doi: 10.3390/s23104896.

[5] B. Dwivedy, M. Moharana, and M. Nayak, "Internet of Things over the last decade: a review of frameworks, architecture, services, and platforms," *Smart Sci.*, pp. 1–36, Mar. 2026, doi: 10.1080/23080477.2026.2640359.

[6] C. D'Ortona, D. Tarchi, and C. Raffaelli, "Open-Source MQTT-Based End-to-End IoT System for Smart City Scenarios," *Futur. Internet*, vol. 14, no. 2, p. 57, Feb. 2022, doi: 10.3390/fi14020057.

[7] S. S. Fithria, M. A. P. Tresna, K. Saddami, and N. Nasaruddin, "Performance Analysis of IoT Message Queuing Telemetry Transport Implementation in Smart Home Systems," in *2024 4th International Conference of Science and Information Technology in Smart Administration (ICSINTESA)*, IEEE, Jul. 2024, pp. 329–334. doi: 10.1109/ICSINTESA62455.2024.10748081.

[8] B. A, T. Natuva, M. V. Shaik, M. Karthik Thippirisetti, V. K. Atte, and S. M, "Latency and Throughput Benchmarking of IoT Protocols on ESP32 Microcontrollers," in *2025 IEEE 6th Global Conference for Advancement in Technology (GCAT)*, IEEE, Oct. 2025, pp. 1–6. doi: 10.1109/GCAT66372.2025.11368454.

[9] F. Abou-Mehdi-Hassani, A. Zaguia, H. Ait Bouh, and A. Mkhida, "Systematic literature review of smart greenhouse monitoring," *SN Comput. Sci.*, vol. 6, no. 2, p. 95, Jan. 2025, doi: 10.1007/s42979-024-03640-4.

[10] D. Bramasta, I. Ramadhan, and S. Ageng Prasetyo, "Perancangan dan Implementasi Smart Home untuk Pengendalian Lampu Otomatis Menggunakan ESP32 dan Aplikasi Blynk," *Pros. Semin. Nas. Teknol. Inf. dan Bisnis*, pp. 653–658, Jul. 2025, doi: 10.47701/64s9j942.

[11] M. F. Tsaqief and J. Sutopo, "Comparative Performance Analysis Between the MQTT and WebSocket Protocols," *bit-Tech*, vol. 8, no. 2, pp. 2227–2237, Dec. 2025, doi: 10.32877/bt.v8i2.3223.

[12] R. B. Ginting, M. A. Sulaiman, and A. Kristanto, "RANCANGAN CHANGEOVER OTOMATIS BERBASIS WAKTU PADA TRANSMITTER VHF AIR TO GROUND R&S SU4200," *J. Rev. Pendidik. dan Pengajaran*, vol. 8, no. 2, pp. 5563–5577, Jun. 2025, doi: 10.31004/jrpp.v8i2.46797.

[13] P. Yan, "Architectural Evolution and Performance Optimization in Embedded Systems: A Comparative Analysis of ESP8266, ESP32-S3, and ESP32-C6 Platforms," *Int. J. Adv. Appl. Sci. Res.*, vol. 4, no. 9, pp. 51–56, 2025, [Online]. Available: www.h-tsp.com

[14] R. Nethanani, Y. Ndou, W. Nchabeleng, and S. Ndlovu, "IoT and Cloud Computing in Biological Water Monitoring: A Systematic Review of Challenges, Architectures, and Emerging Trends," Jun. 12, 2025. doi: 10.21203/rs.3.rs-6848919/v1.

[15] R. Mirza and K. Khairuni, "Analisis Komparatif Konsumsi Daya Baterai pada Perangkat IoT Menggunakan Protokol Komunikasi MQTT dan HTTP," *J. Sains dan Teknol. 4.0*, vol. 2, no. 2, pp. 1–4, 2025.

[16] P. G. Agbulu and G. Joselin Retna Kumar, "An Ultra-low Power IoT System for Indoor Air Quality Monitoring," *J. Phys. Conf. Ser.*, vol. 2007, no. 1, p. 012053, Aug. 2021, doi: 10.1088/1742-6596/2007/1/012053.

[17] A. Hanif, R. Amri, and R. Amri, "Implementasi Internet Of Things Pada Protokol MQTT Dan HTTP Dalam Sistem Pendeteksi Banjir," *INOVTEK Polbeng - Seri Inform.*, vol. 8, no. 2, p. 498, Nov. 2023, doi: 10.35314/isi.v8i2.3767.

[18] L. O. Aghenta and M. Tariq Iqbal, "Design and implementation of a low-cost, open source IoT-based SCADA system using ESP32 with OLED, ThingsBoard and MQTT protocol," *AIMS Electron. Electr. Eng.*, vol. 4, no. 1, pp. 57–86, 2020, doi: 10.3934/ElectrEng.2020.1.57.

[19] J. Toldinas, B. Lozinskis, E. Baranauskas, and A. Dobrovolskis, "MQTT Quality of Service versus Energy Consumption," in *2019 23rd International Conference Electronics*, IEEE, Jun. 2019, pp. 1–4. doi: 10.1109/ELECTRONICS.2019.8765692.

[20] C. Silva, V. A. Cunha, J. P. Barraca, and R. L. Aguiar, "Analysis of the Cryptographic Algorithms in IoT Communications," *Inf. Syst. Front.*, vol. 26, no. 4, pp. 1243–1260, Aug. 2024, doi: 10.1007/s10796-023-10383-9.

[21] F. Luthfi, E. A. Juanda, and I. Kustiawan, "Optimization of Data Communication on Air Control Device Based on Internet of Things with Application of HTTP and MQTT Protocols," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 384, p. 012009, Jul. 2018, doi: 10.1088/1757-899X/384/1/012009.

[22] L. Milić, L. Jelenković, and I. Magdalenić, "Minimizing IoT Energy Consumption Using BLE and Metaprotocol Architecture," *Int. J. Distrib. Sens. Networks*, vol. 2025, no. 1, Jan. 2025, doi: 10.1155/dsn/7859748.

[23] P. N. Bideh, J. Sönnerup, and M. Hell, "Energy consumption for securing lightweight IoT protocols," in *Proceedings of the 10th International Conference on the Internet of Things*, New York, NY, USA: ACM, Oct. 2020, pp. 1–8. doi: 10.1145/3410992.3411008.

[24] A. Ramschie, J. Makal, R. Katuuk, and ..., "Pemanfaatan ESP32 Pada Sistem Keamanan Rumah Tinggal Berbasis IoT," *... Work. Natl. ...*, pp. 4–5, 2021, [Online]. Available: https://jurnal.polban.ac.id/ojs-3.1.2/proceeding/article/view/2688/2076

[25] U. Krcadinac, P. Pasquier, J. Jovanovic, and V. Devedzic, "Synesketch: An Open Source Library for Sentence-Based Emotion Recognition," *IEEE Trans. Affect. Comput.*, vol. 4, no. 3, pp. 312–325, 2013, doi: 10.1109/T-AFFC.2013.18.