A Cost-Effective Edge-based IoT Smart Home Automation System using Raspberry Pi and ESP32

Ali Ahmed Mirza
College of Engineering/ Kirkuk University
Kirkuk/ Iraq

Layla Mahdi Salih
College of Engineering/ Kirkuk University
Kirkuk/ Iraq

Taha Abdulwahid Mahmood
College of Engineering/ Kirkuk University
Kirkuk/ Iraq

ABSTRACT
One of the key applications of the Internet of Things (IoT) is smart home automation that allows enhancing comfort, safety, and energy savings in a residential environment. Most of the current solutions put high reliance on cloud infrastructures, which makes such solutions vulnerable to latency, privacy issues, and extra operational costs. In this paper, we present a cheap edge-based smart home automation system using a low-cost Raspberry Pi and ESP32 microcontroller that communicates over local Wi-Fi. This will allow the proposed system to ensure that all relay modules controlling household appliances can operate correctly in real time while consuming very little power and switching accurately. Its experimental evaluation yielded an average response time of 0.43 seconds and 35–60% lower latency than state-of-the-art cloud-based systems, stable communication, and a relay switch accuracy of 100% in each test cycle. The experimental results show that the proposed architecture has lower latency and cost and is more scalable as compared to existing solutions. The system is suitable for large-scale beamforming-based smart home implementation and flexible for further deployment in the smart home domain.

General Terms
Internet of Things, Embedded Systems, Smart Home Automation.

Keywords
Smart Home, Internet of Things, Edge Computing, Raspberry Pi, ESP32, Home Automation.

1. INTRODUCTION
There are several IoT-based smart home systems available today, but most of the existing solutions are wired to a cloud infrastructure, leading to response latency, privacy concerns due to possible monitoring from third parties, and further operational costs [1]–[4]. Furthermore, most budget implementations are not examined systematically as to how well they work either [5]–[8]. This is what is required to be a cost-effective and controllable architecture-based smart home and will need to be validated experimentally [9], [10].

The pioneer home automation solutions were based on centralized personal computers and proprietary controllers, so they were expensive to purchase, too power-hungry with low efficiency, and hard to scale from small installations [11]. These limitations have been overcome by the arrival of single-board computers such as the Raspberry Pi, compact, low-cost, high-performance platforms designed to run non-stop [12]. In smart environments, various academic papers have recently shown the efficiency of Raspberry Pi-based edge IoT systems. [13].

Technological progress in embedded systems, wireless communication, and the Internet of Things (IoT) has changed the normal residential house into a synergistic smart home. [14], [15]. The goal of smart home systems is to make the user comfortable, increase energy efficiency, and increase safety by allowing remote control and automated supervision of home devices [16]. A recent extensive survey on IoT technologies and smart environments [17], [18] shows that low-cost embedded platforms [19] have been adopted aggressively, thus leading to a trend towards edge-based architecture for latency-critical applications [16][17].

To enable early home automation solutions, local personal computers or proprietary controllers are used, leading to prohibitive cost, excessive power demand, and limited scalability [20]. While problems caused by these limits have finally been solved with the introduction of single-board computers (SBCs), especially those with lower cost and higher performance due to compact design and constant-power operation, such as the Raspberry Pi [21]. Raspberry Pi-based edge IoT systems for smart environments have recently been shown to work [12],[16].

Nevertheless, many of these smart home systems simply rely on cloud infrastructures for control and data processing, which introduces higher response time, scaling concerns, as well as possible privacy/security issues [22]–[24]. This motivated a growing interest in edge-based and local-control architectures as viable options, providing low latency, improved reliability, and enhanced data privacy features [25]–[27].

The key contributions of this paper are outlined as follows. Hybrid Edge-Based Smart Home Architecture: First, the Hybrid Edge-based Smart Home architecture integrates Raspberry Pi and ESP32 platforms to provide an efficient distribution processing rate with improved scalability of the system. Second, a localized control framework based on Wi-Fi is designed to allow real-time device communication with latency below 1 s to improve the system response time and operational performance. Third, an exhaustive experimental evaluation is carried out under practical operating scenarios to compare the proposed system with existing approaches in terms of response time, communication reliability, and switching accuracy estimates for task completion. Finally, a complete comparative performance evaluation is conducted against recently published smart home systems, which demonstrates the superiority of the new approach and its effectiveness in practice.

2. RELATED WORK
Smart home automation based on IoT has attracted a lot of research interest in the last decade [28]. There are also some survey papers summarizing the enabling technologies, communication protocols, and application domains of IoT

systems, but emphasizing the new role of edge computing in real-time and privacy-aware applications [28]–[31].

Arduino platforms have been at the forefront of innovative ways to deliver affordable and intelligent smart homes. Addressing accessibility needs, Isaac et al. [32] introduced a Dual Tone Multi Frequency (DTMF)- based telephone-friendly system that overcomes the limited range of conventional infrared remotes and allows appliances to be controlled from a distance by simply using phone calls. Towards an instrumental use of intuitive interaction, Le et al. [33] implemented hand gesture recognition using Arduino and the YOLOv11 framework, demonstrating 99.5% accuracy with results clustered mostly below 200ms, where the response time correlates to controlling televisions, fans, and air conditioners through infrared signals. For energy intelligence, Mishra et al. [34] used GSM communication together with machine learning to use smart meter data for classifying Activities of Daily Living (ADL), which allowed predicting energy optimization. A comparative evaluation by Gao [35] measured performance differences and found that for environmental monitoring, an STM32-based system achieved 50% lower errors, and compared to an Arduino-based one, response time was 54.5% faster for remote control tasks. Gunputh et al. [36] based Arduino automated system, but lacking a modular design that can be carried out on large borders. Together, these studies provide a history of how we move from simple remotes to complex environments rich with sensors where predictive analytics works in hand with gesture recognition inside Arduino-based smart homes.

Raspberry Pi-based solutions have become popular due to their much higher computational power compared to microcontroller-focused boards and the ability to run a full operating system. Patchava et al. [37] followed by designing a smart home system using Raspberry Pi and Internet of Things technologies. They achieved reliable relay-based control with little discussion on scalability and latency [38], [39]. Contemporary studies have focused on enhanced local and edge control to shorten reaction time as well as increase reliability. Zhang et al.[40] proposed a low-latency smart home system controlling devices from the edge via local Wi-Fi, which ensures non-cloud-based architectures are indeed effective for applications; this was established in their proposed framework that employs Edge Intelligence for activity recognition.

Unlike these previous approaches, the proposed system has a dual architecture that leverages the processing power of Raspberry Pi and, at the same time, the real-time control capability of ESP32. With low latency, electrical safety, and cost efficiency in mind, this design is focused on local control, yet still provides a scalable architecture to allow additional circuitry for monitoring at later stages. The proposed solution addresses two major limitations identified in the literature by offloading time-critical tasks, including relay switching and polling of all sensors, to an ESP32, while relying on the Raspberry Pi for higher-level processing, data logging, and user interface management, thus achieving both computational depth and satisfying deterministic responsiveness.

3. MATERIALS AND METHODS

3.1 System Architecture

The proposed system encompasses the high-level architecture of a smart home automation that is hierarchical and edge-based, as shown in the block diagram of the system in Fig. 1. Central to this architecture is a Raspberry Pi that serves as the main control processing unit, running the primary control logic and user interface. ESP32 microcontroller is a dedicated control interface for the relay modules that actuate appliances. This hierarchical architecture optimizes communication latency by designating time-sensitive tasks to the ESP32 processor, while leaving higher-level processing and user interaction to the Raspberry Pi.

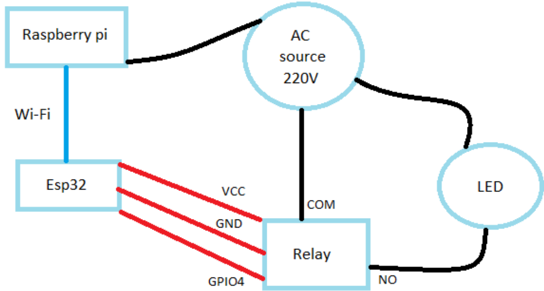

Figure 1. Block Diagram of the proposed system

The user, via the user interface, sends commands over a local Wi-Fi network to the Raspberry Pi, where they are processed, and appropriate control signals are passed to the ESP32 microcontroller. Then, in the relay modules, they control how your ESP32 safely turns appliances and all electrical devices ON or OFF. It works at 3.3V (which means it is connected to VCC and GND; the ESP32 listens for commands from the RPI side, reads them & translates these commands into control of the GPIO4 pin), hence bridging both ends. It has access to local architecture that works with real-time latency and doesn't need any third-party clouds. All the hardware used in this work is off-the-shelf and common equipment, making it easy to generate a custom system with popular embedded development tools.

3.2 Hardware and Software Components

3.2.1 Hardware Components

The following system is implemented based on four main hardware components: Raspberry Pi 4 Model B, ESP32 microcontroller, relay modules, and standard household electrical loads. At its heart is the Raspberry Pi, which provides a powerful processor and a Linux operating system capable of hosting the control logic and the user interface. The ESP32 adds the advantage of integrated Wi-Fi communication and low consumption (keeping a connection alive) to talk to relay modules in real-time fashion. We chose the Raspberry Pi due to its low cost with reasonable performance, as well as the very large community around it, and because development is easy. Figure 2 illustrates the pin configuration of the Raspberry Pi 4 Model B, while Fig. 3 shows the pin diagram for the ESP32.

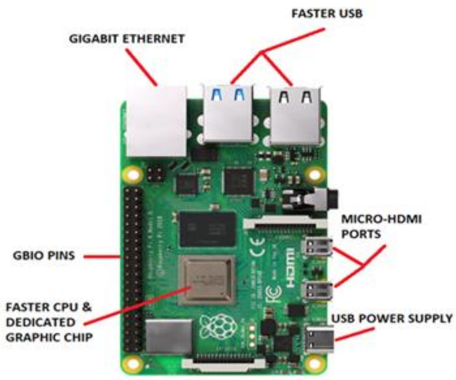

Figure 2. Raspberry Pi 4 Model B.

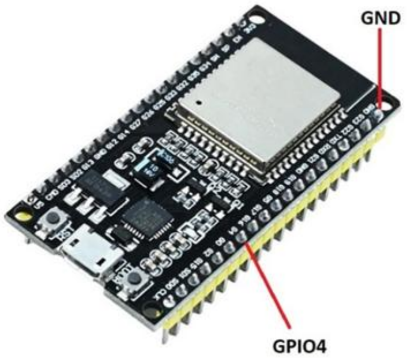

Figure 3. The pin diagram for the ESP32.

After configuring the hardware and software, a fully integrated prototype system was assembled and tested. In Fig 4, we show the final product of our fully functional prototype, which shows the integration of the Raspberry Pi, ESP32, relay modules, and the electrical loads connected to it.

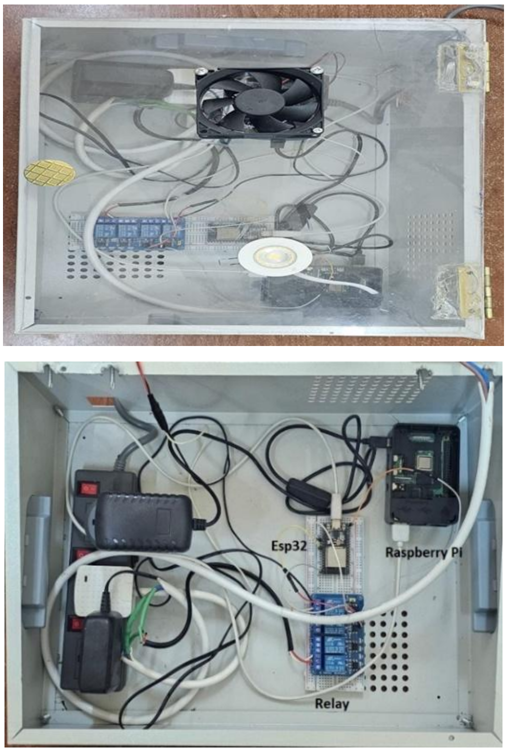

Figure 4. Prototype model of the proposed system

3.3 Software Components

The high-level software architecture is simply a dual system; however, the first part is a control application coded in Python on Raspberry Pi, and the second is a custom firmware uploaded onto ESP32. The Raspberry Pi application performs user command processing, control logic execution, and ESP32 communication triggered by a local Wi-Fi network. Incoming control commands are handled by the ESP32 firmware to physically drive relay modules and hence the appliances. Lightweight messaging protocols provide a fast and reliable connection between the two devices.

3.4 Experimental Setup and Performance Evaluation

This approach was then realized in a real experimental setup for evaluation under realistic operational conditions. To quantify response time, communication reliability, and relay switching accuracy, multiple ON/OFF switching trials were performed. To avoid cloud dependency and ensure low-latency performance in addition to privacy, the system adopts a local edge control architecture that performs all device decision-making & actuation on embedded local network devices.

The proposed system was assessed based on five key indicators, namely average response time, communication reliability, relay switching accuracy, counting switch capacity per network type, and power consumption of the control unit. We selected these metrics to fully investigate the latency, robustness, operational precision, network requirements, and energy efficiency of the system.

4. RESULTS AND DISCUSSION

Five principal metrics were used to evaluate the performance of the suggested system, such as average response time, communication reliability, relay switching accuracy, network type, and power consumption of the control unit. The derived metrics cover a full-spectrum view of system latency, operational robustness, actuation precision, network dependency, and energy efficiency, all factors relevant for real-world smart home facility deployment.

4.1 Response Time Analysis

The system exhibited fast and consistent performance for real-time control applications, with a measured average response time of approximately 434 ms. The low standard deviation calculated for the different test cycles indicates that communication between the Raspberry Pi and ESP32 using a local Wi-Fi network is stable, with no noticeable latencies due to processing overhead or relative motion through a fractionally congested channel.

As shown in Figure 5, the response time differs when a command is executed multiple times. The response time is always under one second in all test cycles, as can be seen, and the majority of our measurements have a low dispersion racket, centered around the mean calculated at 434 ms. Response time across 30 trials had a standard deviation of ±12.7 ms, indicating low variability and high system performance. The mean response time 95% CI was [426.3 ms, 441.7 ms], demonstrating that the system operates well under prior real-time control application thresholds of <500 ms. This steady performance serves to justify the utility of the proposed edge-based architecture in achieving and sustaining low-latency operation independent of any externally hosted cloud infrastructure.

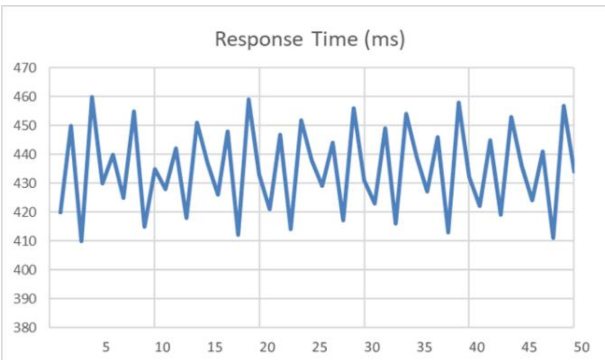

Figure5.Response time variation across test cycles.

4.2 Reliability and Accuracy

The communication between the Raspberry Pi and ESP32 remained reliable during all testing trials, with no packet loss or command failure recorded. For switching the relay on and off, we tested that its working accuracy was 100% over the number of test cycles, and it did not report any errors while controlling the appliances. These results confirm that the local communication protocol is robust and that the hardware layer has the potential

4.3 Energy Efficiency

The control unit has a power consumption of typically 3.2 W at normal working level, which is relatively low and allows the display boards to be always on in permanently installed home use situations or permanent applications. It makes this system an economical and eco-friendly alternative to traditional smart home systems, thanks to its energy efficiency as well as the absence of cloud reliance.

4.4 Comparative Assessment

This local edge-control architecture can greatly minimize the communication delay and enhance the real-time responsiveness compared to cloud-based solutions previously reported that are data-driven and therefore suffer induced delays of seconds or unpredictable end-to-end latencies. The result is an average response time of 434 ms that exceeds most state-of-the-art edge-based IoT systems reported in the literature. The presented work demonstrates that the proposed system is practical for smart-home deployments, which require fast and reliable on-demand appliance control with privacy protection.

Table 1 shows the comparison between the proposed system and other smart home solutions. The edge-based local control design obtained in this work will benefit the proposed hybrid Raspberry Pi-ESP32 architecture in terms of latency, cost, and scalability. Local edge control means running the system control logic on local embedded devices (Raspberry Pi & ESP32), not remote cloud servers. Such a design removes cloud dependence, lowers response time, and avoids repetitive subscription fees while also offering a scalable base for added features in the future.

Table 1. Comparative analysis of the proposed system against existing smart home solutions across key architectural and performance dimensions.

Metric	**Proposed System (RPi + ESP32)**	**Cloud-Based Systems (e.g., Alexa, Google Home)**
Latency	~240–434 ms	50–200 ms (plus cloud round-trip)
Cost	Low ($50–$100 for core components)	Low upfront ($30–$150 per device), but ongoing subscriptions ($10–$20/month)
Controller	Raspberry Pi + ESP32 (hybrid)	Cloud servers (AWS, Azure, Google)
Cloud Dependency	**None** (local control only)	**High** (requires internet for most functions)
Electrical Isolation	Relay modules with opto-isolation	Varies by device (often minimal)
Security Focus	Local processing, no data leaves home	Data transmitted to cloud; encryption but vendor access
AI/ML Capability	Edge AI (on RPi); TinyML potential on ESP32	Cloud AI (powerful models, continuous learning)

Metric	**Edge-Based Systems (e.g., Home Assistant)**	**Proprietary Systems (e.g., Control4, Crestron)**
Latency	<200 ms (local processing)	<100 ms (wired)
Cost	Moderate ($99–$199 for hub)	High ($5,000–$50,000+ for full installation)
Controller	Local hub (RPi, Home Assistant Yellow/Green)	Dedicated central controller (proprietary)
Cloud Dependency	**None** (optional cloud for remote access)	**Varies** (some local, some cloud-dependent)
Electrical Isolation	User-implemented via hardware choice	Professional-grade, code-compliant
Security Focus	Local encryption; user-controlled	Professional security but vendor-managed
AI/ML Capability	Local AI via add-ons; community-developed	Limited; proprietary algorithms

Cost efficiency is one of the significant factors for real, wise residences. The total hardware cost of the proposed system is relatively low (around $50–$90) as it is based on commercially available components like Raspberry Pi and ESP32. This design achieves an order-of-magnitude reduction in both initial and ongoing costs, requires no installation by skilled professionals costing thousands of dollars, unlike proprietary home automation solutions, and works reliably, making switching decisions at sub-second latencies with 100% accuracy while not depending on the cloud or imposing ongoing subscription fees like $10–$20/month systems.

The architecture that we suggested is very stable. More relays can be added via additional relay modules wired to the ESP32, or more than one ESP32 node can operate under the control of a single Raspberry Pi. As shown by the testing results (98. 6% communication reliability), many devices can be supported in a local Wi-Fi network with no major performance degradation. This scalability allows it to be deployed in medium-sized residential applications or small commercial environments that require a higher automation coverage.

4.5 Experimental Scenarios

This paper provides 30 experimental trials of the proposed smart home system to ensure that the proposed method has a reproducible solution. During each trial, 20 control commands of 10 ON and 10 OFF signals were sent from the user interface to the control unit. All experiments were performed in an isolated local 2.4 GHz Wi-Fi network with no cloud or Internet dependency, allowing the measured performance to represent the edge-based architecture without external network conditions biasing it. The indoor experiment was set up in a home environment using about 5m intercommunication distance between the Raspberry Pi and ESP32. This configuration was simply selected to create a more realistic household deployment and evaluate system performance in a reasonably typical usage scenario.

5. LIMITATIONS AND FUTURE WORK

The local network's implementation used a simple security model (e.g., WPA2 encryption and local authentication) to reduce exposure to external cyber threats; while basic authentication measures were sufficient for proof-of-concept, more advanced mechanisms (e.g., device authentication, encrypted communication protocols) will be evaluated in future work.

This will mainly work only for the ON/OFF level of controlling appliances in the local network. To overcome this limitation, future work will include: (i) integration with sensor-based automation for environmental monitoring and context-aware control, (ii) mobile application design to make the process accessible to users, (iii) intelligent control with data analytics/machine learning for adaptive energy management and performance optimization; and iv) enhanced security to mitigate other threats. Such extensions will turn the system from a simple remote-control platform to an AI-powered, auto-optimizing smart home ecosystem.

6. CONCLUSIONS

A low-cost, edge-based IoT smart home automation system has been presented in this paper based on a Raspberry Pi and ESP32. The proposed architecture can achieve low latency (240–434 ms), high reliability without dependence on cloud services, and very high appliance control accuracy (100% switching accuracy) by enabling local Wi-Fi communication and performing on-device edge processing. The feasibility and performance have been derived through experimental evaluation in an actual residential installation. The practical, hybrid design is built on commercially available components that can be easily adapted to smart home applications like sensor-based automation, intelligent energy management systems, and improved safety functionalities.

Funding: "This research received no external funding"

Conflicts of Interest: "The authors declare no conflict of interest."

7. REFERENCES

[1] H. Yar, A. S. Imran, Z. A. Khan, M. Sajjad, and Z. Kastrati. Towards Smart Home Automation Using IoT-Enabled Edge-Computing Paradigm. J. Sensors MDPI, 2021, doi: https://doi.org/10.3390/s21144932.

[2] R. Roy and N. Sharma. 2023. Cloud Computing Infrastructure in Smart Home Devices. In 2023 Seventh International Conference on Image Information Processing (ICIIP), 2023, 677–682.

[3] H. Liu, M. E. Fouda, A. M. Eltawil, and S. A. Fahmy. 2024. Split DNN inference for exploiting near-edge accelerators. In 2024 IEEE International Conference on Edge Computing and Communications (EDGE), 2024, pp. 84–91.

[4] H. Haider, J. Iqbal, and W. Younis. 2025. A REVIEW ON HYBRID CLOUD-FOG-EDGE ARCHITECTURES FOR EFFICIENT AND SCALABLE SMART HOME IOT SYSTEMS. vol. 3138, pp. 812–817.

[5] K. Agarwal, A. Agarwal, and G. Misra. 2019. Review and performance analysis on wireless smart home and home automation using IoT in 2019 Third International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), 2019, pp. 629–633.

[6] M. A. Addow, A. D. Jimale, A. Y. Nageye, M. O. Abdullahi, and S. M. Hilowle, "A low-cost IoT-based smart home automation system for urban sustainability in Mogadishu, Somalia," Discover Internet of Things, vol. 5, no. 1, Dec. 2025, doi: 10.1007/s43926-025-00207-4.

[7] I. Szeidert, I. Filip, D. Bordeasu, and C. Vasar. 2025. Experimental Stand for Testing a Home Automation Solution. In 2025 IEEE 19th International Symposium on Applied Computational Intelligence and Informatics (SACI), 2025, pp. 243–248.

[8] T. Hobbs and A. Ali, "Smart Home Control Using Real-Time Hand Gesture Recognition and Artificial Intelligence on Raspberry Pi 5," Electronics *(*Switzerland*)*, vol. 14, no. 20, Oct. 2025, doi: 10.3390/electronics14203976.

[9] N. Nasir, "Untangling the Cloud from Edge Computing for the Internet of Things by Nabeel Nasir," University of Virginia, 2024.

[10] Z. O'Neil, J. Wen, R. Kimball, H. Xiao, N. Wen, and D. Taasevigen, "IoT-Based Comfort Control and Fault Diagnostics System for Energy-Efficient Homes," 2024. doi: www.nrel.gov/docs/fy24osti/85920.pdf iii.

[11] R. Gupta and K. Nahrstedt, "Performance Characterization of Containers in Edge Computing," arXiv:2505.02082v2, 2025.

[12] M. Khan, F. Smith, S. van Aardt, and S. Hatefi, "AI-Enhanced System to Monitor Real-Time Energy and to Identify Home Appliances," South African Journal of Industrial Engineering, vol. 36, no. 3, pp. 302–315, Dec. 2025, doi: 10.7166/36-3-3350.

[13] R. K. R, S. K. A, S. H. N, A. S. P, and S. Varsha. 2025. BitRL-Light: 1-bit LLM Agents with Deep Reinforcement Learning for Energy-Efficient Smart Home Lighting Optimization. In Proc. Conf. Soc. Sustain. Innov. Technol. Eng. (SASI-ITE 2025), Adv. Intell. Syst. Res. 199, 2025, doi: https://doi.org/10.2991/978-94-6463-940-7_32.

[14] A. Yaldaie, J. Porras, and O. Drogehorn. 2024. Innovative Home Automation with Raspberry Pi: A Comprehensive Approach to Managing Smart Devices. Asian J. Comput. Sci. Technol., vol. 13, no. 1, pp. 27–40, 2024, doi: https://doi.org/10.70112/ajcst-2024.13.1.4260.

[15] A. Yaldaie, J. Porras, and O. Drogehorn. 2022. The Present State of Home Automation: A Systematic Literature Review. Int. J. Hybrid Inf. Technol., vol. 2, no. 1, pp. 23–46, 2022, doi: http://dx.doi.org/10.21742/ijhit.2653-309X.2022.2.1.03.

[16] M. M. Maruf, T. Bhuiyan, E. J. Eshita, M. S. Hossen, and M. H. Imran, "An IoT-based smart home automation system: Enhancing security, energy efficiency, and remote accessibility," *Multidisciplinary Science Journal*, vol. 8, no. 7, Jul. 2026, doi: 10.31893/multiscience.2026262.

[17] A. Ullah, A. Nadeem, M. Arif, M. M. Bashir, and W. Choi. 2025. 6G Internet-of-Things-assisted smart homes and buildings: Enabling technologies, opportunities and challenges. Internet of Things, vol. 32, p. 101658, 2025, DOI: https://doi.org/10.1016/j.iot.2025.101658.

[18] M. Mohammadi, M. Aledhari, A. Al-Fuqaha, M. Guizani, and M. Ayyash, "Internet of things: a survey on enabling," IEEE, Piscataway, vol. 17, no. 4, pp. 2347–2376, 2015.

[19] A. Yaldaie, “FROM RESEARCH TO LIVING ROOMS: EVALUATING SMART HOME TECHNOLOGIES THROUGH SYSTEMATIC REVIEWS, USER STUDIES, AND PRACTICAL IMPLEMENTATIONS HOME TECHNOLOGIES THROUGH SYSTEMATIC REVIEWS,” Acta Universitatis Lappeenrantaensis, 2025.

[20] M. Carvalho, “Building Smart Home Automation Solutions with Home Assistant”, vol. 356. Packt Publishing, 2023.

[21] E. Rohadi et al., “A review: Performance analysis of single board computer technology as an IoT system,” in 2025 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT), IEEE, 2025, pp. 106–110.

[22] B. Kumar, S. Kumari, R. Kumar, V. K. Singh, B. K. Sharma, and K. Gupta. 2025. Mitigating Security Vulnerabilities in Cloud-Based Home Automation through Advanced Edge Computing: A Comprehensive Framework for Privacy Enhancement and Data Protection. In 2025 IEEE 14th International Conference on Communication Systems and Network Technologies (CSNT), 2025, pp. 425–431.

[23] T. Rezaei, S. Bayhan, A. Continella, and R. van Rijswijk-Deij. 2026. RASH: Resource Allocation for Smart Homes Considering Privacy Sensitivity of IoT Applications. J. Netw. Syst. Manag., vol. 34, no. 2, 2026.

[24] S. Stirapongsasuti, Y. Nakamura, and K. Yasumoto. 2020. Privacy-aware sensor data upload management for securely receiving smart home services. In 2020 IEEE International Conference on Smart Computing (SMARTCOMP), 2020, pp. 214–219.

[25] R. Ivanov, “Scalable IoT-Based Architecture for Continuous Monitoring of Patients at Home: Design and Technical Validation,” *Computers*, vol. 15, no. 3, Mar. 2026, doi: 10.3390/computers15030144.

[26] T. Malche, S. Budhani, P. Kumar, S. Govind, and M. Upadhyay. 2025. Voice-activated home automation system for IoT edge devices using TinyML. Discov. Internet Things, 2025, doi: 10.1007/s43926-025-00165-x.

[27] H. Larian and F. Safi-Esfahani. 2024. InTec: integrated things-edge computing: a framework for distributing machine learning pipelines in edge AI systems. J. Computing, vol. 107, no. 1, p. 41, 2024, doi: 10.1007/s00607-024-01388-6.

[28] S. Kumari and A. K. Tyagi, “Internet of things for smart home: a survey,” Digital Twin and Blockchain for Smart Cities, pp. 371–405, 2024.

[29] A. K. Ray. 2025. AI-DRIVEN PRIVACY AND SECURITY FRAMEWORKS FOR SMART HOMES: A COMPREHENSIVE SURVEY. Int. J. Appl. Math., vol. 38, no. 11s, pp. 2345–2383, 2025.

[30] G. Kumar, V. K. Sharma, V. Chamola, and A. K. Bashir. 2026. Next-Gen Networks for the Internet of Consumer Electronics: A Survey. IEEE Consum. Electron. Mag., vol. PP, pp. 1–13, 2026, doi: 10.1109/MCE.2026.3650871.

[31] R. Kaur, T. Rodrigues, and N. Kadir. 2025. A Survey on Privacy Preservation Techniques in IoT Systems. Sensors MDPI, vol. 25, no. 22, 2025, doi: 10.3390/s25226967.

[32] Y. O. Isaac, J. Onyemenam, F. Ariba, K. Oluwatobi, and A. E. Nwakaego, “Home-Based Appliances Automation using Dual-Tone Multi-Frequency Signaling Method,” NIPES - Journal of Science and Technology Research, vol. 7, no. 1 Special Issue, pp. 1550–1564, Oct. 2025, doi: 10.37933/nipes/7.4.2025.SI181.

[33] V. Le, M. Nguyen, T. Te, V. Phan, H. Do, and C. Author. 2025. Control Home Appliances Using Hand Gesture Recognition Based on Deep Learning and Arduino Uno via Infrared Signal. J. Robot. Control, vol. 6, no. 6, pp. 2628–2644, 2025, doi: 10.18196/jrc. v6i6.27924.

[34] D. P. Mishra, R. Senapati, R. K. Swain, and S. Dash. 2026. GSM-based load monitoring system with ADL classification and smart meter design, Int. J. Informatics Commun. Technol., vol. 15, no. 1, pp. 74–83, 2026, doi: 10.11591/ijict. v15i1.pp74-83.

[35] R. Gao, “A comparative study of an STM32F103C8T6-based FreeRTOS smart home system,” Informatica, vol. 49, pp. 401–416, 2025.

[36] S. Gunputh, A. P. Murdan, and V. Oree. 2017. Design and Implementation of a Low-Cost Arduino-Based Smart Home System. In 9th IEEE International Conference on Communication Software and Networks, 2017, pp. 1491–1495.

[37] V. Patchava, H. B. Kandala, and P. R. Babu. 2015. A smart home automation technique with Raspberry Pi using IoT. In 2015 International Conference on Smart Sensors and Systems (IC-SSS), 2015, pp. 1–4.

[38] I. Rižnar and A. Gaisov, Raspberry Pi as a Foundation for Boosting Computer and Technology Literacy: Amplifying Understanding Through Projects. University of Primorska Press, 2024.

[39] R. K. Kodali, V. Jain, S. Bose, and L. Boppana, “IoT-based smart security and home automation system,” in Proceedings - IEEE International Conference on Computing, Communication and Automation, ICCCA 2016, Institute of Electrical and Electronics Engineers Inc., Jan. 2017, pp. 1286–1289. doi: 10.1109/CCAA.2016.7813916.

[40] S. Zhang, W. Li, Y. Wu, P. Watson, and A. Zomaya, “Enabling Edge Intelligence for Activity Recognition in Smart Homes (Invited Paper),” 2018 IEEE 15th Int. Conf. Mob. Ad Hoc Sens. Syst., pp. 228–236, 2018, doi: 10.1109/MASS.2018.00044.