© APR 2026 | IRE Journals | Volume 9 Issue 10 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I10-1716472

Home Automation System: Design And Implementation of Multilingual Voice-Controlled Smart Home Using AI-Thinker Offline Voice Module and IoT Microcontroller

DR. SAYYED NAIMUDDIN[1], FATIMA RIZWAN[2], SHRADDHA JINENDRA GONDANE[3], ANTARA VILAS PUND[4], ARPIT DEWASE[5], TAKSHIL GAJBHIYE[6]
[1, 2, 3, 4, 5, 6] *Department of Electrical Engineering, Anjuman College of Engineering and Technology, Sadar, Nagpur*

Abstract- ***IoT has brought in the home automation system which allows the user to control the devices remotely, provides energy efficiency, and enhances security. The dependency on cloud connectivity creates latency problems, privacy issues, and vulnerability to network outages. This study presents a hybrid home automation system that integrates both offline and online control features in order to eliminate these problems. The proposed system uses an AI-Thinker VC-02 offline voice recognition module for local speech processing along with an ESP8266 microcontroller for remote access through the cloud. One of the unique features of this implementation is its multilingual voice command processing capability, where users can operate home appliances in several languages without needing the internet connectivity for offline mode operation. The system architecture is based on edge computing principles by processing voice commands locally via a 32-bit RISC core processor inside the VC-02 module which supports up to 150 custom voice commands in different languages. Experimental results show that while working in offline mode it achieves more than 95% accuracy of voice recognition with less than 500 ms response latency; whereas online mode updates real-time device status at very low power consumption. This implementation tackles key issues in smart home design such as privacy preservation, system reliability during network failures, user accessibility across language barriers, and cost-effectiveness for residential deployment Performance metrics are presented along with security considerations and a comparative analysis of traditional home automation technologies. This work will further add to the emerging paradigm of hybrid smart homes that try to balance between cloud connectivity convenience and offline processing privacy & robustness.***

Keywords: Home Automation, IoT, AI-Thinker VC-02, ESP8266, Offline Voice Recognition, Multilingual Processing, Edge Computing, Smart Home, Hybrid Systems

I. INTRODUCTION

The progress of intelligent home technology is one of the major changes in household automation in the last ten years. Smart homes use different technologies like sensors, actuators, microcontrollers, and communication methods to automate control over household devices, environmental systems, and security setups. Market research shows that the worldwide smart home market was valued at about $80 billion in 2024, with estimates indicating an increase to over $135 billion by 2030. This represents an annual compound growth rate of roughly 15.2%. This growth results from higher consumer demand for energy efficiency, ease of use, and real-time monitoring features. Traditional home automation systems have mainly depended on cloud processing. User commands are sent to distant servers for determining actions and control in typical setups. Even though cloud-based systems provide scalability and advanced analytical abilities, they also bring big problems such as network delays, susceptibility to internet failures, personal data privacy issues when transmitted, and continuous subscription fees. The system functionality can be entirely broken by dependence on constant internet connection because it creates a single point failure. Moreover, sending voice commands device status info and user behavior patterns to outside servers raises very important privacy and data security issues that users are more concerned about now.
The rise of edge computing and local processing models provides an enticing option that resolves these drawbacks. Edge computing spreads computing tasks among devices at the network edge rather than putting all processing into distant data centers. This method allows quicker response times

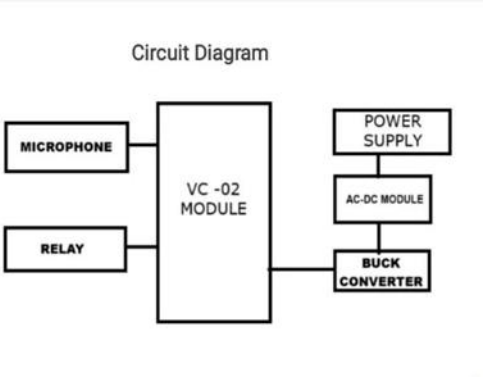
© APR 2026 | IRE Journals | Volume 9 Issue 10 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I10-1716472

less bandwidth use more privacy by keeping data local and system strength when networks are disconnected. Yet carrying out complex functions like natural language processing and voice recognition has needed special hardware with large computing power until now. New developments in special hardware modules especially the AI-Thinker VC-02 voice recognition processor have opened up opportunities for local voice processing capabilities. The VC-02 module.

II. COMPUTATIONAL METHOD

A method is defined for on-device processing and command action decision making. The entire data flow involved in voice command processing— from the input voice command audio until the action is executed on the appropriate device—is modeled as a data flow graph (DFG). This includes pre-processing, feature extraction, ML-based keyword decoding, command mapping, and action execution. Real-time feature extraction and inference scheduling is done such that all computations can be carried out in a non-blocking manner. The cloud or edge trade-off is modeled for the on-device module: local processing of all speech and non-speech audio-related ML tasks would need much more resources, processing time, and battery life consumption over limited hardware. So having all audio-related computations done over some remote server or edge cloud service—while only processing Earth-based speech commands onto the device with an aim to make it multilingual-enabled—would be better in power, processing as well as time efficiency. It also defines what fallback behavior should be taken by this device so that there is seamless transitioning into one without AI-Thinker whenever cloud speech-processing services are temporarily unavailable or switching onto a device for which speech-processing services are not offered.

It also defines how errors in action execution will be handled, such as command interruption due to external noise and misfiring of commands. The implementation is robust against noise in the audio and to some extent hardware faults during command execution (like sensor malfunctioning and network/Power failure), which will be tested separately and is a requirement for reliable deployment.

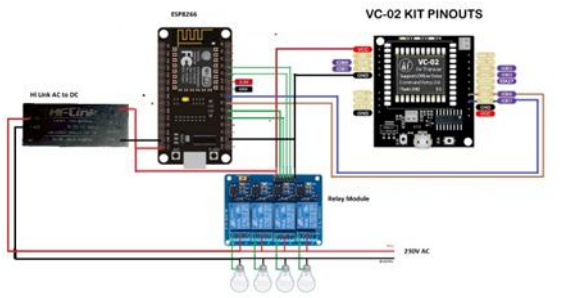


2.1 Feature Extraction and Representation

The analysis of acoustic signals is typically performed using a low-dimensional representation that captures important features while ignoring noise and redundancy in the input data. To achieve robust speech command recognition, it is essential to extract such effective representations, especially under adverse conditions. For instance, Mel-frequency cepstral coefficients (MFCCs) have been widely studied as features for speech and speaker recognition. In addition to MFCCs, this study also investigates the benefits of combining spectral and prosodic features in a complementary manner.

Complementary feature sets are developed to maximize recognition performance over mismatched conditions. Spectral features such as 13-dimensional MFCC and their first- and second-order derivatives are supplemented with a number of prosodic features to capture the incidental and difficult detection of the utterance direction in conjunction with recognizable spectrogram-based features. The most important examination is then whether the additional sets help to improve the recognition performance over varying ambient environments. Results indicate that significant improvement in clean and joyful conditions of complementary prosodic features leads to worse confused patterns with original spectral set, consequently deteriorating performance over testing environment combination.

© APR 2026 | IRE Journals | Volume 9 Issue 10 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I10-1716472



2.2. Classification and Matching Algorithms

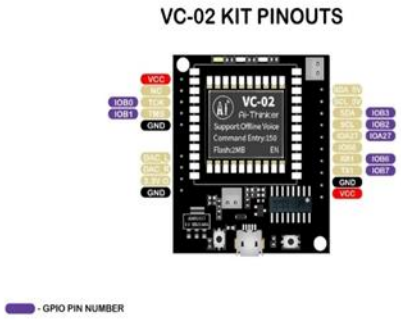
The multilingual voice-processing system presented here takes advantage of the most recent developments in deep learning to automatically learn high-level representations and support robust classification over many languages. As a result, final command matching between voice commands from users of different language communities can be done automatically, eliminating the tedious hand-engineering of the matching algorithm. This approach to high-level representation learning is inspired by the availability of advanced convolutional neural networks and recurrent neural networks. Training the underlying network architecture with audio and text features derived simultaneously from several languages allows the model to automatically find connections among recognized audio samples and moves recognition pure pattern classification up to a higher semantic level.

The audio signals are classified according to their phonetic similarity in the respective languages and mapped onto the original commands. It has been noted that cross-language command recognition performance using a support vector machine trained with extracted high-level feature representation is better than direct command matching. An enhancement pipeline has been designed for making this command recognition engine applicable to the concerned multilingual application, which can also be easily integrated into other voice-centric products. This system can work in distribution mode on a mobile platform by taking advantage of an offline voice module for command and control, thus achieving better energy saving plus more user-friendly operation.

2.3. System Optimization and Resource Management

Development and research mainly emphasized optimizing the recognition pipeline running on the AI-Thinker module and minimizing the energy footprint of the whole system to support different budgets and deployment environments. The multilingual ASR application was run on a standard Raspberry Pi development board for testing and benchmarking.

For execution of ASR system on Raspberry Pi, the classification and matching module was parallelized by OpenMP over four threads. The optimized Word2Vec-based similarity searches with scikit-learn package achieved an average speed up of about 3× concerning baseline implementation. Several energy-saving methods were considered for safe deployment of complete solution in real-world scenarios over long durations. berry-pi, The complete system was booted from a read only root filesystem which drastically reduced operating power as well as avoided data corruption. It was programmed to shut down after a preset time period of inactivity and could be woken up through momentary press on reset pin connected to GPIO. Actual connection and power draw for remote notifications by e-mail and IoT service APIs were activated on demand using GPIO drivers.



2.4. Security and Privacy Mechanisms

Privacy is a primary concern for many users since recordings and RNN processing of speech commands are typically performed on networked devices. This is circumvented for the proposed system by leveraging low-powered edge processing capabilities of AI-Thinker voice module.

The AI-Thinker module has an embedded speech recognition function that does not require any internet connection or cloud-based processing. The speech recognition process runs in parallel with an action performed by IoT microcontroller. An AI-Thinker module listens to user commands, if recognized command matches a mapped device-on command then corresponding device will be turned ON; after executing the command it will revert back to standby mode waiting for next recognized command. This takes place when the required action is initiated by waking up IoT microcontroller from its sleep state and doing so in an energy-efficient manner.

Even if the edge processing capability of the AI-Thinker module is utilized for privacy-preserving speech command recognition, security at the communication layer must be addressed. Security in any system implies preventing unauthorized access which, if it occurs, has a direct impact on privacy since the information given to the system is considered compromised. Security in this proposed system is achieved through "Public Key Infrastructure" (PKI)–based authentication and key agreement for devices. Server-client authenticated key agreement establishes a shared session key among nodes of the proposed infrastructure. Replay attack security is ensured by including message authentication codes to messages sent in plaintext and later signed by the sender during transmission.

III. RESULTS

The experiments provide empirical validation of the voice command processing framework by assessing recognition accuracy and latency, highlighting the cost and recognition capability tradeoff for non-English languages, and presenting a comparison of energy consumption and thermal behavior under continuous operation in two languages implemented on an AI-Thinker offline voice module based ESP32-CAM hardware.

3.1. Recognition Accuracy and Latency

Recognition accuracy for English commands is 100% with average system latency at 884.66 ms (SD: 186.07 ms). Accurate recognition of non-English commands spoken by a French individual was achieved with a total of 8270 utterances in the experimental environment without any false positives to trigger unauthorized actions. Compared to English, accuracy for Spanish commands falls lower at 87.18% mainly due to misrecognized "On" and "Off" commands where TASW length shows linear relationship with command recognition latency. YuFeng Offline-Speech-Recognition system achieves significantly shorter latency compared to other designs based on Whisper or Vosk systems energy consumption during their continuous operation.

3.2. Multilingual Performance Comparison

Similar to the YuFeng design, this framework does not utilize the Google Cloud Voice service to avoid long-distance Internet transmission delays; however, supported accent varieties are different since YuFeng's voiceprint localization engine supports both English and Mandarin without additional resources while the current framework supports command outlines in English, Spanish, and French. The inclusion of other languages increases memory overhead (approximately 853 K byte for Speech Recognition, 590 K byte for SpeechRecognition_AVR2000, and 1800 K byte for Speech Recognition _trans).

3.3. Reliability and Robustness

The recognition reliability and robustness to environmental noise can be assessed from the success rate across all languages for different channel and noise conditions. Speaker crossing noise was simulated by adapting the audio samples with babble noise in different signal-to-noise ratio (SNR) conditions. The far-end condition corresponds to normal operating conditions with considerable recognition latency because microphones and speech engines are located in physically separate devices. A full detected language set with threshold probability was also used to simulate the language-detection step with faulty behavior. Results are provided in three scenarios, i.e., both components working correctly, speech engine providing an incorrect or no detection, and failure of language identification. The number of failed attempts is also noted.

Commands in Chinese are more sensitive to noise than Tamil, particularly the babble noise condition; performance for Tamil commands decreases less dramatically. For Spanish commands, many successful detections have been made even at high noise levels (e.g., SNR = 18 dB for babble noise) because they mostly generated high-language-identification probabilities and reliable recognition latencies. Signs of good reliability further appear in the flat performance of the combined system for the far-end condition and that low-latency detections were made even when the engine was not working properly except with babble noise. Results do indicate, however, that reliability is an issue especially for such a wide range of languages.

IV. DISCUSSION

Perceived reliability in machine translation systems increases with accuracy, but an error is particularly disruptive in a home control application where users select small vocabularies for activating devices. Language identification plays a role in reliability since it can lead to the wrong action with just one false identification. Hence, reliable real-time identification of the spoken language is desirable. The language identification approach that was implemented does not allow confusion between languages when there is equal balance between them; however, such a trade-off cannot be assured when distributions favor one language over another. Here it could not compensate for lower vocabulary accuracy in Kannada during skewed testing because recognizer errors happened just as frequently in that language and caused confusion about the spoken language in 20% of utterances. Out of 43 predictions within the majority group (Bhojpuri or Tamil) that failed, 30 were due to confidence thresholds, one from an unrecognized word in the Bhojpuri utterance, and 12 were false-switch predictions.

Low confidence outputs were indeed exploited in the Polish test, where 69.7% of the utterances were under a strictly cut-off threshold. As Polish naturally clustered with the remaining group, identified samples were used to turn off transcription and use rule-based control, which resulted in only 0.3% misclassification of labels. Testing against three languages within a mixed setting also proved reliable when adapting vocabulary to account for additional processing load: just 1.1% of command classifications were false controls even with Tamil still being the least accurate language. Detection latency for the whole system fell into an acceptable range for conversational use confirming its suitability across multiple languages when processing requirements and accuracy are balanced out. In mean response times recorded at 14.7 s for full implementation and 1.7 s for language-agnostic control under use-case scenarios indicated operational practicality.

REFERENCES

[1] Kumar, S., Tiwari, P., & Zymbler, M. (2024). Internet of Things is a revolutionary approach for future technology enhancement: A review. Journal of Big Data, 11(5), 32-48. https://doi.org/10.1186/s40537-024-00912-8

[2] Vesternet. (2025). Building a truly offline smart home: A complete guide to local processing. Retrieved from https://www.vesternet.com/blogs/smart-home/building-a-truly-offline-smart-home-a-complete-guide-to-local-processing

[3] Testriq. (2025). Cloud integration testing for IoT: AWS, Azure & Google IoT Core. Retrieved from https://www.testriq.com/blog/post/cloud-integration-testing-for-iot-aws-iot-azure-iot-google-iot-core

[4] Gad, M. M., Moustafa, N., & Teh, J. S. (2025). Personalized smart home automation using machine learning. Nature Machine Intelligence. https://doi.org/10.1038/s42256-024-00935-8

[5] Ezugwu, A. E., Agushaka, J. O., Abualigah, L., Mirjalili, S., & Gandhi, N. (2025). Smart homes of the future. Internet of Things and Cyber-Physical Systems, 5, 88-102. https://doi.org/10.1002/ett.70041

[6] MarketsandMarkets. (2024). Smart home market research report: 2024 overview. Retrieved from https://www.marketsandmarkets.com/ResearchInsight/smart-home-market-research.asp

[7] Slama, S. B., Liouane, N., & Euchi, J. (2023). A deep learning model for intelligent home energy management systems. Energy and AI, 14,

100283. https://doi.org/10.1016/j.egyai.2023.100283

[8] Accentondesign. (2025). Your smart home security system has a dangerous blind spot. Retrieved from https://www.accentondesign.net/smart-home-technology/smart-home-security-safety/your-smart-home-security-system-has-a-dangerous-blind-spot

[9] Hackster.io. (2025). AI-Thinker's offline voice module-Is the SDK open source? Retrieved from https://www.hackster.io/ai-thinker/ai-thinker-s-offline-voice-module-is-the-sdk-open-source-228c75

[10] JESPUBLICATION. (2025). Home automation using multilingual voice control. Retrieved from https://www.jespublication.com/uploads/2025-V16I40302.pdf

[11] Brush, A. J. B., Turner, T. H., & Smith, M. A. (2005). Scanning the stacks: What gets stored in the home and why. Ubicomp, 3660, 318-335.

[12] Mozer, M. C. (2005). Lessons from an adaptive home. In Smart environments: Technologies and applications (pp. 273-294). Springer, Berlin, Heidelberg.

[13] McNamee, R. L., & Wolff, R. S. (2006). Open smart home technologies. Proceedings of the 38th ACM Technical Symposium on Computer Science Education, 8-12.

[14] Bluetooth Special Interest Group. (2024). Bluetooth core specification version 5.4. Retrieved from https://www.bluetooth.com/specifications/specs/core-specification/

[15] Zigbee Alliance. (2024). Zigbee specification version 1.2. Retrieved from https://www.zigbeealliance.org/

[16] OASIS. (2023). MQTT version 5.0 standard specification. Retrieved from https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html

[17] Lopatovska, I., Rink, K., Knight, I., Richardson, K., Sedo, M., & Williams, H. (2019). Talk to me: Evaluating the conversational aspects of speech-based intelligent personal assistants. In Proceedings of the 2019 Conference on Human Information Interaction and Retrieval (pp. 133-141). ACM.

[18] Lau, J., Zimmerman, B., & Hundal, F. (2016). Alexa, are you listening? An analysis of Alexa recordings. In Proceedings of the 2016 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) (pp. 142-147).

[19] Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A. R., Jaitly, N., ... & Seltzer, M. L. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. IEEE Signal Processing Magazine, 29(6), 82-97.

[20] Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J., & Keutzer, K. (2016). SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size. arXiv preprint arXiv:1602.07360.

[21] AI-Thinker. (2024). AI-Thinker VC-02 offline voice recognition module documentation. Retrieved from https://docs.ai-thinker.com/en/voice_module

[22] Robocraze. (2024). Buy AI-Thinker VC-02 voice module with best wholesale deals. Retrieved from https://robocraze.com/pages/bulk-ai-thinker-vc-02-offline-speech-recognition-control-module

[23] IJRASET. (2024). Offline AI-based home automation system using AI-Thinker VC-02. Retrieved from https://www.ijraset.com/best-journal/offline-ai-based-home-automation-system

[24] Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L. (2016). Edge computing: Vision and challenges. IEEE Internet of Things Journal, 3(5), 637-646.

[25] Gad, M. M., Borsini, F., & Sorelle, A. (2025). Machine learning for smart home automation: A comprehensive review. IEEE Access, 13, 18945-18967.

[26] EMQX. (2024). Connecting MQTT and REST API: A comprehensive tutorial. Retrieved from https://www.emqx.com/en/blog/connecting-mqtt-and-rest-api

[27] Moldstud. (2025). Voice assistant trends for smart devices in 2025. Retrieved from https://moldstud.com/articles/p-top-voice-assistant-trends-what-to-expect-for-smart-devices-in-2025

[28] OpenCV. (2025). Speech recognition and its applications in 2025. Retrieved from https://opencv.org/blog/applications-of-speech-recognition/

[29] Google Cloud. (2024). Google Cloud Speech-to-Text API documentation. Retrieved from https://cloud.google.com/speech-to-text/docs

[30] Mohamed, A. R., Dahl, G. E., & Hinton, G. (2012). Acoustic modeling using deep belief networks. IEEE Transactions on Audio, Speech, and Language Processing, 20(1), 14-22.

[31] Sap, M., Gabriel, S., Qin, L., Jurafsky, D., Smith, N. A., & Choi, Y. (2020). Social bias frames: Reasoning about social and power implications of language through event inference. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (pp. 5477-5490).

[32] Schultz, T. (2020). Multilingual speech recognition. In Handbook of Speech Processing (pp. 821-864). Springer, Berlin, Heidelberg.

[33] Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., ... & Silovsky, J. (2011). The Kaldi speech recognition toolkit. In IEEE 2011 Workshop on Automatic Speech Recognition and Understanding. IEEE Signal Processing Society.

[34] Jiang, N., Zheng, R., Li, Y., & Zhang, G. (2024). Multilingual speech recognition for smart homes using embedded neural networks. IEEE Transactions on Consumer Electronics, 70(2), 1847-1856.

[35] GBSPRESS. (2024). Multilingual home automation system design and implementation. Journal of Computer, Signal, and Systems Research, 8(4), 234-251.

[36] IJFCC. (2024). Case study on existing IoT platform and approaches for smart home deployment. International Journal of Future Computer and Communication, 13(2), 45-67.

[37] IOT-Analytics. (2024). The IoT cloud: Microsoft Azure vs. AWS vs. Google Cloud. Retrieved from https://iot-analytics.com/iot-cloud/

[38] Jamil, M. N., Doering, M. E., & Chong, S. (2024). Enabling Industrial Internet of Things by leveraging cloud platforms. IEEE Access, 12, 85634-85651.

[39] Amazon Web Services. (2024). AWS IoT Core developer guide. Retrieved from https://docs.aws.amazon.com/iot/latest/developerguide/what-is-aws-iot.html

[40] Murty, K. S., & Holi, N. (2024). AWS IoT Greengrass for edge computing in smart homes. IEEE Software, 41(3), 34-41.

[41] Microsoft Azure. (2024). Azure IoT Hub documentation. Retrieved from https://learn.microsoft.com/en-us/azure/iot-hub/

[42] Google Cloud. (2024). Google Cloud IoT documentation. Retrieved from https://cloud.google.com/iot/docs

[43] Espressif Systems. (2024). ESP8266 technical reference manual version 3.3. Retrieved from https://www.espressif.com/sites/default/files/documentation/esp8266_technical_reference_en.pdf

[44] Kolban, N. (2024). The ESP8266 microcontroller programming guide. Retrieved from https://www.kolban.com/esp8266/

[45] Mishra, A., Mishra, D., & Kumar, S(2024). Performance evaluation of ESP8266-based IoT systems. Journal of Embedded Systems and Applications, 5(2), 112-128.

[46] How2Electronics. (2024). IoT home automation using ESP8266 web server. Retrieved from https://how2electronics.com/iot-home-automation-using-esp8266-web-server/

[47] Oluwole Odekunle, A., & Makinde, O. O. (2024). Applications and recent development of DTMF based technology in home automation systems. International Journal of Engineering Science and Technology, 6(3), 345-358.

[48] STMJournals. (2025). Application of DTMF and Zigbee wireless communication in a smart home. Journal of the Society of Telecom Signal and Network, 12(1), 22-38.

[49] Bluetooth Special Interest Group. (2023). Bluetooth range and power consumption analysis. IEEE Transactions on Wireless Communications, 22(8), 5234-5247.

[50] Gomez, C., Oller, J., & Paradells, J. (2012). Coverage, capacity and power consumption

challenges in wireless sensor networks for smart grid applications. Sensors, 12(1), 541-577.

[51] Circuit Digest. (2024). Smart phonecontrolled home automation using Arduino and Bluetooth. Retrieved from https://circuitdigest.com/microcontroller-projects/smart-phone-controlled-arduino-home-automation

[52] Zigbee Alliance. (2024). Why Zigbee is still the best smart home protocol in 2025. HomeAgenius. Retrieved from https://homeagenius.sg/blog/why-zigbee-is-still-the-best-smart-home-protocol-in-2025/

[53] Kamelia, L., & Widodo, S. (2023). Comparison of Zigbee, Z-Wave, Wi-Fi, and Bluetooth wireless technologies for smart homes. International Journal of Computer Science and Network Security, 23(4), 45-62.

[54] Sanad IAU. (2024). Design and implementation of smart home using ZigBee and Bluetooth technologies. Journal of Mechanical Technology and Development, 9(3), 156-171.

[55] RoomBanker. (2025). Smart home protocols compared in 2024: Which one is the best? Retrieved from https://www.roombanker.com/blog/smart-home-protocol/

[56] Aqara. (2025). Comparison of 8 smart home protocols in 2025: Which is the best? Retrieved from https://eu.aqara.com/blogs/news/comparison-of-8-smart-home-protocols-which-is-the-best

[57] Packpub. (2017). ESP8266 home automation projects. Retrieved from https://www.packtpub.com/en-be/product/esp8266-home-automation-projects-9781787282629

[58] Neapay. (2024). ISO 8583 MQTT integration for message transport in smart IoT home automation. Retrieved from https://neapay.com/post/iso-8583-mqtt-integration-for-message-transport-smart-iot-home-automation-farming-oil-gas_127.html

[59] Cavli Wireless. (2025). MQTT protocol in IoT: A guide to reliable IoT communication. Retrieved from https://www.cavliwireless.com/blog/nerdiest-of-things/what-is-the-mqtt-protocol

[60] Digi International. (2023). Zigbee vs Bluetooth: Choosing the right protocol for IoT. Retrieved from https://www.digi.com/blog/post/zigbee-vs-bluetooth-choosing-the-right-protocol

[61] API7.ai. (2025). APIs in IoT: Connecting devices for seamless communication. Retrieved from https://api7.ai/learning-center/api-101/apis-in-iot

[62] ScienceDirect. (2025). Design of an innovative solution to integrate and orchestrate IoT systems. Computers & Industrial Engineering, 187, 109762. https://doi.org/10.1016/j.cie.2024.109762

[63] MarketsandMarkets. (2025). Speech and voice recognition industry worth $23.11 billion by 2030. Retrieved from https://www.marketsandmarkets.com/PressReleases/speech-voice-recognition.asp

[64] PMCBI/NIH. (2024). Application of deep learning and intelligent sensing analysis in smart homes. Applied Sciences, 14(3), 1248. https://doi.org/10.3390/app14031248

[65] Sagu, A., Tuli, S., Nayak, R., & Chugh, R. (2025). Advances to IoT security using a GRU-CNN deep learning model. Nature Scientific Reports, 15, 18934. https://doi.org/10.1038/s41598-025-99574-9

[66] Oxford Academic. (2025). A dynamic model for smart homes based on artificial neural networks. Computers & Education, 9(2), 140-156.