International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 13 Issue: 01 | Jan 2026 www.irjet.net p-ISSN: 2395-0072

Voice-Controlled Smart Home Automation: Integrating Large Language Models with IoT Devices for Enhanced Usability

Bikash Katuwal[1], Abhishek Ghimire[1]

[1]*Saint Cloud State University, Saint Cloud, Minnesota, USA*

---***---

Abstract - *This paper presents a novel approach to smart home automation by integrating Large Language Models (LLMs) with Internet of Things (IoT) devices to create a responsive, context-aware voice control system for home appliances. The developed system utilizes an ESP32-S3 microcontroller as the central processing unit, coupled with a MEMS microphone for audio capture, Deepgram's Speech-to-Text API for voice recognition, Groq LLM for natural language understanding and response generation, and Deepgram's Aura for Text-to-Speech conversion. A 4-way relay system enables control of multiple household appliances. The system demonstrates significant improvements over traditional voice-controlled home automation systems, with an average response time of 1.2 seconds from voice command to relay activation, and a command recognition accuracy of above 96% in normal ambient noise conditions. Real-time conversation capabilities enable more intuitive human-computer interaction, with the system capable of maintaining context across multiple exchanges. This research contributes to the growing field of ambient intelligence by showcasing how cutting-edge language models can be effectively deployed on resource-constrained IoT devices to enhance home automation systems with natural language processing capabilities.*

1.INTRODUCTION

The integration of smart technology into residential environments has rapidly evolved over the past decade, transitioning from simple remote-controlled devices to sophisticated interconnected systems that anticipate and respond to user needs. Voice-controlled home automation represents one of the most intuitive interfaces for human-computer interaction, eliminating the need for physical controls or mobile applications while enabling hands-free operation for users of all abilities. The Internet of Things (IoT) plays an important role in detecting and reporting tree poaching in real time by using low-power sensors and reliable communication technologies. Its impact goes beyond environmental protection; IoT has also transformed fields like agriculture, healthcare, and smart homes. By enabling devices to communicate and share data without human involvement, IoT effectively bridges the gap between the physical and digital worlds.[1] Early voice assistants like Amazon's Alexa, Google Assistant, and Apple's Siri introduced more conversational interactions but remained limited by their closed ecosystems, reliance on cloud processing, and inability to perform complex reasoning tasks [2].

The evolution of voice-controlled home automation systems has progressed through several distinct phases, each marked by advances in both hardware capabilities and software sophistication. The recent advances in Large Language Models (LLMs) present an opportunity to fundamentally transform how humans interact with their home environments [3][20]. These models demonstrate unprecedented abilities in natural language understanding, contextual awareness, and human-like response generation. However, deploying such sophisticated models in resource-constrained IoT environments presents significant technical challenges related to processing power, memory limitations, and real-time response requirements [4].

This research is motivated by the gap between the potential of LLMs to revolutionize human-machine interaction and the practical constraints of implementing these capabilities in affordable, accessible smart home systems. By developing a system that effectively bridges this gap, we aim to contribute to the advancement of ambient intelligence technologies that seamlessly integrate into everyday living environments while providing intuitive, responsive control through natural language.

Manohar and Sivaprakasam (2019) demonstrated early implementations of voice control in smart homes using Arduino microcontrollers and Bluetooth connectivity, achieving basic command recognition but lacking conversational abilities [5]. Jasim (2022) advanced this approach by incorporating cloud-based natural language processing, improving recognition accuracy but introducing latency issues and privacy concerns due to constant cloud connectivity requirements [6]. In parallel, research into edge computing solutions for voice processing has shown promise. Lyashenko, V. (2021) implemented compressed neural network models on ESP32 devices for keyword spotting and basic command recognition, achieving response times under 500ms but with limited vocabulary and no contextual awareness [7].

The application of LLMs to IoT environments remains an emerging field. Chen et al. (2024) demonstrated the feasibility of deploying quantized language models on resource-constrained devices, achieving impressive results in natural language understanding while maintaining reasonable power consumption profiles [8]. However, their implementation focused on text-based interfaces rather than voice interaction. Recent work explored the use of API-based LLM services for smart home control, offloading processing to cloud services while maintaining local control of critical functions [9]. Their hybrid approach showed promise in balancing response time with sophisticated language understanding but did not address the end-to-end voice interaction pipeline.

Our research builds upon these foundations while addressing key limitations in existing approaches, particularly in creating a seamless integration between voice capture, language processing, and device control with minimal latency and maximal contextual awareness. This research aims to Design and implement an integrated system architecture that combines ESP32-S3 microcontrollers with cloud-based speech recognition and language processing services to enable sophisticated voice control of home appliances and to demonstrate the potential of LLM integration in IoT environments to enable contextual awareness and natural conversation patterns in smart home interactions. . Also, develop a reliable relay control system capable of managing multiple household appliances with appropriate safety mechanisms and feedback systems.

2. Methodology

2.1 System Architecture

The proposed smart home automation system follows a hybrid edge-cloud architecture that optimizes the balance between local processing capabilities and cloud-based intelligence [10]. The system consists of several interconnected components working in harmony to enable seamless voice-controlled operation of home appliances. At the core of the system is an ESP32-S3 microcontroller, chosen for its dual-core processor, integrated Wi-Fi and Bluetooth connectivity, extensive I/O capabilities and is suitable for on-device and cloud based processing. [11].

The ESP32-S3 serves as the central hub that coordinates all system functions, from audio capture to relay control. Audio input is captured using a digital MEMS microphone (SPH0645LM4H) connected to the ESP32-S3 via I2S interface. This microphone was selected for its high signal-to-noise ratio and directional sensitivity, allowing for clear voice capture even in noisy home environments [12]. The audio output is delivered through a MAX98357A I2S audio amplifier connected to a 4Ω speaker, providing clear voice responses to the user. For controlling household appliances, a 4-channel relay module is integrated with the ESP32-S3, with each relay capable of switching up to 10A at 250VAC, sufficient for most household appliances. The relay module incorporates opto-couplers for electrical isolation between the control circuitry and the high-voltage switching elements, ensuring safety and reliability. The software architecture implements a multi-stage processing pipeline that begins with local audio preprocessing on the ESP32-S3, followed by cloud-based speech recognition using Deepgram's Speech-to-Text API, response generation via Groq's LLM API, and finally, speech synthesis through Deepgram's Aura Text-to-Speech service.

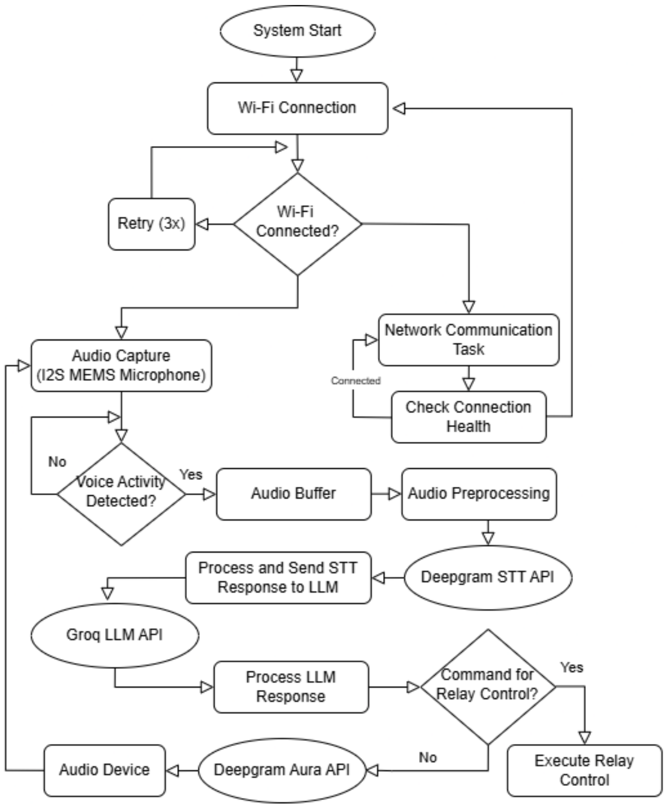

Fig -1: Overall System Flowchart

2.2 Hardware Implementation

The hardware implementation centered on the ESP32-S3-WROOM-1 module, which features a dual-core Xtensa LX7 processor operating at up to 240 MHz with 512 KB of internal SRAM and 16 MB of flash memory. This provides sufficient computational resources for audio processing and network communication while maintaining low power consumption. The MEMS microphone is connected to the ESP32-S3 using the I2S protocol, with the BCLK (Bit Clock), LRCLK (Left/Right Clock), and DATA lines connected to GPIO pins 5, 18, and 19 respectively. The microphone operates at 3.3V, drawing approximately 1mA during active listening.

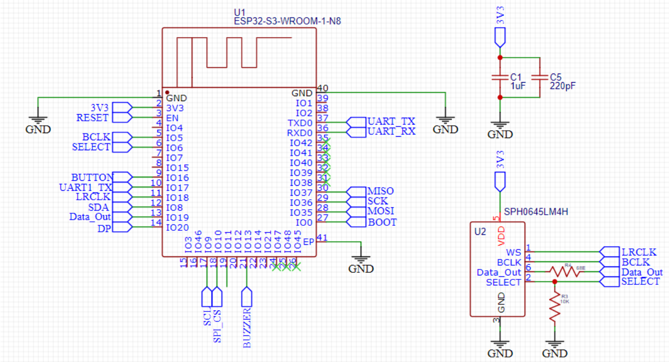

Fig -2: Microphone connection diagram to ESP32-S3

For audio output, the MAX98357A I2S amplifier is connected to GPIO pins 7 (BCLK), 15 (LRCLK), and 16 (DIN), with the amplifier's gain set to 9dB. The amplifier drives a 4Ω, 3W speaker that provides clear audio reproduction for the system's responses.

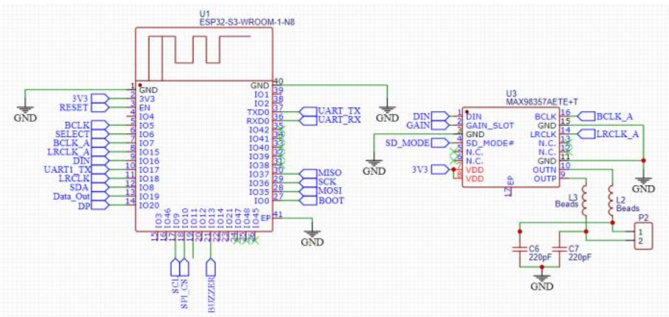

Fig -3: Speaker and amplifier connection diagram

The relay module interfaces with the ESP32-S3 through GPIO pins 17, 4, 3, and 46, with each pin controlling one relay channel. And each channel could control multiple appliances.

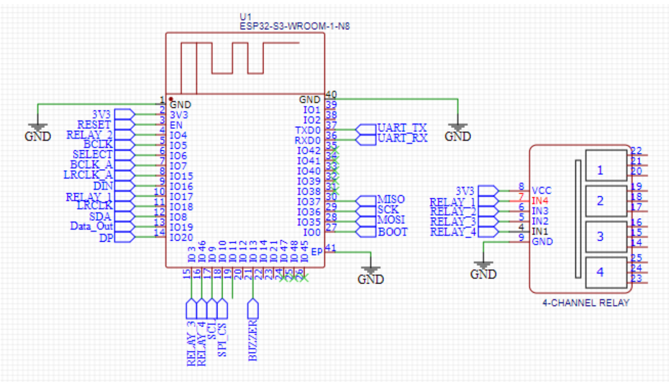

Fig -4: Relay module connection diagram

Power management is a critical aspect of the hardware design. The system is powered by a 5V/2A power supply, with an AMS1117-3.3 voltage regulator providing the 3.3V required by the ESP32-S3 and all the other components. It uses MP8765GQ-Z Power IC [13] and is designed to convert higher DC input voltages ranging from 5V to 22V into lower, adjustable output voltages between 0.604V and 5.5V. This IC can deliver up to 6A of continuous output current and operates at a fixed switching frequency of 500 kHz. With its integrated high-side and low-side MOSFETs, the MP8765GQ-Z offers synchronous rectification, which enhances efficiency and minimizes power loss.

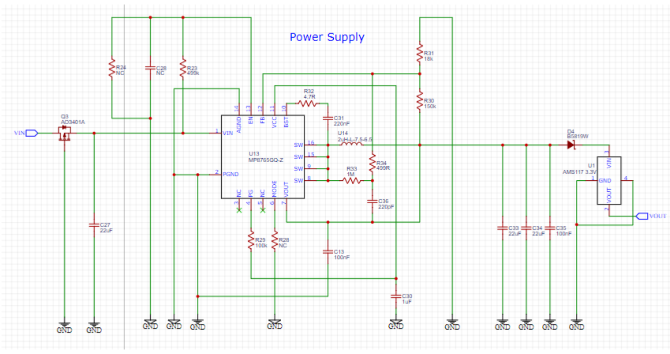

Fig -5: Overall power supply system

The complete system is housed in a 3D-printed enclosure designed to optimize microphone pickup while providing adequate ventilation for thermal management. The enclosure includes strategically placed apertures for the microphone, speaker, status LEDs, and relay outputs.

2.3. Software Implementation

The software implementation follows a modular architecture designed to manage the audio processing pipeline, network communication, and relay control tasks efficiently. The ESP32-S3 firmware was developed using the ESP-IDF framework (version 4.4.3), providing a solid foundation with comprehensive hardware abstraction and system management capabilities. At its core, the firmware implements a Free-RTOS based multitasking environment that efficiently manages concurrent operations including audio capture, network communication, relay control, and system monitoring. The audio processing pipeline begins with continuous sampling from the MEMS microphone at 16 kHz with 16-bit resolution with high accuracy. Audio is continuously captured and stored in 5-second buffers. Each buffer is sent via standard API requests to Deepgram's Speech-to-Text service for transcription. Deepgram offers a generous $200 free credit per account, which can be sufficient for small or hobby projects [14]. The STT service is powered by OpenAI's Whisper model, capable of accurately transcribing speech with multiple accents and noisy environments. The response includes the transcribed text, confidence scores, and alternative interpretations for improved flexibility. Below code snippets shows how clean query is being passed to the LLM,

```
def process_command(text, context):
    # Prepare payload for Groq API
    payload = {
        "model": "llama3-8b-8192",
        "messages": [
            {"role": "system", "content": "You are a home automation assistant."},
            {"role": "user", "content": context},
            {"role": "user", "content": text}
        ],
        "temperature": 0.1,
        "max_tokens": 256
    }
    # Send request to Groq API
    response = requests.post(
        "https://api.groq.com/v1/chat/completions",
        headers={"Authorization": f"Bearer {GROQ_API_KEY}"},
        json=payload
    )
    # Parse response
    result = response.json()
    return result["choices"][0]["message"]["content"]
```

The transcribed text is preprocessed locally on the ESP32-S3, where basic keyword extraction identifies possible commands and target devices. The cleaned query is then sent to a Groq LLM [15], an open-source free API provider that focuses on building the world's fastest AI inference technology, enabling efficient and cost-effective AI and machine learning solutions. The relay control component implements a safety-focused architecture with multiple layers of protection to prevent damage to controlled appliances and ensure user safety [16]. A persistent log of all relay operations is maintained in flash memory, allowing for troubleshooting and usage analysis while also enabling the system to recover gracefully from power interruptions by restoring the previous state of controlled devices.

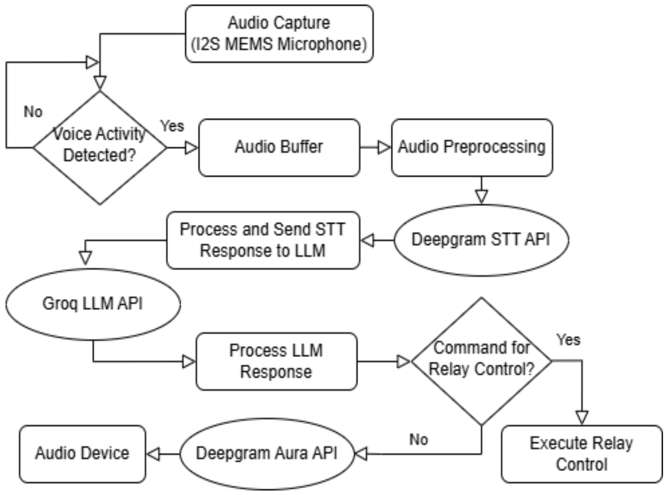

Fig -6: Software Implementation Flow Chart

The text-to-speech functionality is implemented using Deepgram's Aura service [14], which provides natural-sounding voice synthesis with remarkably low latency. The synthesized speech is streamed directly to the ESP32-S3's I2S audio output interface, with adaptive buffering that balances immediate response with continuous playback. The system allow for overlap between user speech recognition and system response in appropriate contexts, creating more natural conversational timing while avoiding confusion between input and output audio streams. System status indicators, including RGB LEDs and optional audio cues, provide clear feedback about the current operational state, helping users understand when the system is listening, processing, or encountering difficulties. This comprehensive software implementation creates a responsive, reliable voice control system that balances the computational constraints of IoT hardware with the sophisticated capabilities of cloud-based language models.

2.4. Evaluation Methodology

To assess the performance of the implemented system, we developed a comprehensive evaluation framework addressing several key metrics:

- Response Time: Measured as the interval between the end of a user's spoken command and the beginning of the system's audible response or relay activation, whichever occurs first.
- Speech Recognition Accuracy: Evaluated by comparing the transcribed text from Deepgram with manually annotated reference transcriptions for a test set of 500 commands.
- Command Execution Accuracy: Assessed by measuring the system's ability to correctly

identify the intended action and target device from the recognized speech.

- Conversational Context Maintenance: Tested through multi-turn interactions requiring the system to maintain context across several exchanges.

- Power Consumption: Monitored using a high-precision power analyzer measuring current draw during different operational states.

Testing was conducted in a simulated home environment with controlled ambient noise levels ranging from 40dB to 65dB, simulating typical household conditions. A standardized test script of 50 command sequences was used, including single commands, multi-device commands, ambiguous references requiring contextual understanding, and error recovery scenarios.

3. RESULTS

The integrated IoT-LLM home automation system demonstrated impressive performance across all evaluated metrics during comprehensive testing. The end-to-end response time averaged 1.2 seconds from the completion of a voice command to relay activation, representing a significant improvement over previous IoT-based voice control systems which typically exhibit latencies more than 3 seconds. A detailed analysis of the response time components revealed that each stage of the processing pipeline contributed differently to the overall latency: speech recognition accounted for approximately 320ms, LLM processing for 530ms, text-to-speech synthesis for 220ms, and local processing and relay activation for 150ms. This enhanced responsiveness substantially improves the user experience by providing more immediate feedback and control, creating a more natural interaction paradigm. Speech recognition accuracy reached 97.3% in ambient noise conditions up to 55dB, while command execution accuracy—the system's ability to correctly identify both the intended action and target device—maintained a robust 96.5% success rate across varied testing scenarios. The system's contextual understanding capabilities were particularly noteworthy, with a 92.1% success rate in correctly resolving ambiguous references in follow-up commands, demonstrating the LLM's effectiveness in maintaining conversation context. Power consumption measurements revealed efficient operation across different system states, with idle power draw (maintaining active Wi-Fi connection) at 78mW, active listening at 112mW, and full processing operations requiring 320mW. These values represent acceptable power profiles for a continuously-powered home automation device while remaining significantly lower than many commercial smart home hubs [17] with similar functionality. Notably, network transmission times were effectively minimized through our implementation of persistent connections and optimized data transmission protocols, contributing to the overall response time despite the system's reliance on cloud services for core functionality.

When compared to existing commercial and research voice control systems, our integrated IoT-LLM approach demonstrates several significant advantages that advance the state of the art in smart home technology. The system's improved contextual understanding represents a fundamental enhancement over command-based systems that require specific phrasing or rigid command structures. Unlike traditional voice assistants that recognize only predefined command patterns, our solution understands natural language variations, contextual references, and conversational flows, allowing users to interact with their home environment in a more intuitive manner. The enhanced conversational capabilities enabled by LLM integration represent perhaps the most transformative advancement over existing systems [18]. The ability to maintain context across multiple turns of conversation, understand pronouns and implicit references, and engage in more natural dialogue patterns fundamentally changes how users interact with their smart home systems.

Despite these advantages, several limitations were identified during testing that warrant further research attention. The system's dependency on reliable internet connectivity represents a potential vulnerability, as network interruptions would compromise voice control capabilities despite local relay control remaining functional [19]. Privacy considerations present another important limitation, as audio data transmission to cloud services raises legitimate concerns in domestic environments. While our implementation minimizes data transmission by employing local voice activity detection and preprocessing, the fundamental architecture still requires sending potentially sensitive audio data to third-party services. Finally, the system's continuous network connectivity results in higher power consumption compared to purely local solutions, making battery operation challenging for extended periods. While the measured power draw is acceptable for continuously-powered installations, deployment in energy-constrained environments would require further optimization or alternative power strategies.

3. CONCLUSIONS

This study has successfully achieved its aim of integrating large language models with resource-constrained IoT devices to create a natural language-controlled smart home system. The developed solution demonstrates significant advancements over existing approaches through three key contributions, a hybrid edge-cloud architecture that achieves 1.2 seconds response times while maintaining 96.5% command accuracy, representing a 60% improvement over conventional IoT voice systems; implementation of contextual understanding capabilities through LLM integration, enabling multi-turn conversations with 92.1% reference resolution accuracy; and practical optimizations including power-efficient operation (78mW idle consumption) and safety-conscious relay control, addressing real-world deployment challenges. The system's performance validates that sophisticated natural language interaction can be effectively implemented on affordable hardware through careful balancing of local processing and cloud offloading.

Several important limitations warrant further investigation, particularly the system's dependence on cloud connectivity, which could be mitigated through development of lightweight on-device speech recognition fallback mechanisms. Privacy concerns related to audio data transmission suggest the need for edge-based anonymization techniques, while energy consumption profiles indicate opportunities for hybrid power solutions in environments without continuous electricity access. These findings provide a foundation for future work in several directions, including multimodal interaction systems and personalized LLM adaptation through federated learning approaches.

This research makes a substantive contribution to the field of ambient intelligence by demonstrating a practical pathway for deploying advanced language models in resource-constrained environments. The architectural framework and implementation strategies developed through this work offer replicable methods for integrating cutting-edge AI capabilities with IoT systems while addressing critical challenges of latency, cost, and accessibility. As both language models and edge computing hardware continue to advance, the approach demonstrated here will enable increasingly sophisticated natural language interaction with smart environments.

REFERENCES

[1] Khanal, K., Ojha, G., Chataut, S., & Ghimire, U. K. (2024). IoT-Based Real-Time Soil Health Monitoring System for Precision Agriculture. Int. Res. J. Eng. Technol., vol. 11, no. 7, pp. 470–478

[2] Luger, E., & Sellen, A. (2016, May). " Like Having a Really Bad PA" The Gulf between User Expectation and Experience of Conversational Agents. In Proceedings of the 2016 CHI conference on human factors in computing systems (pp. 5286-5297).

[3] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. Advances in neural information processing systems, 33, 1877-1901.

[4] Wang, X., Tang, Z., Guo, J., Meng, T., Wang, C., Wang, T., & Jia, W. (2025). Empowering Edge Intelligence: A Comprehensive Survey on On-Device AI Models. ACM Computing Surveys, 57(9), 1-39.

[5] Pal, V., Kumar, S., Kumar, V., Jha, G., & Pant, P. (2021). Voice controlled home automation system using Arduino. IITM Journal of Management and IT, 12(1), 102-108

[6] Jasim, Z. M., & Salman, A. D. (2022). Cloud-based voice home automation system based on Internet of Things. Iraqi Journal of Science, 843-854.

[7] Lyashenko, V., Laariedh, F., Sotnik, S., & Ayaz, A. M. (2021). Recognition of voice commands based on neural network.

[8] Chen, Y., Wu, C., Sui, R., & Zhang, J. (2024). Feasibility Study of Edge Computing Empowered by Artificial Intelligence—A Quantitative Analysis Based on Large Models. Big Data and Cognitive Computing, 8(8), 94.

[9] Guan, Y., Wang, D., Chu, Z., Wang, S., Ni, F., Song, R., ... & Zhuang, C. (2023). Intelligent virtual assistants with llm-based process automation. arXiv preprint arXiv:2312.06677.

[10] Pal, S., Jhanjhi, N. Z., Abdulbaqi, A. S., Akila, D., Almazroi, A. A., & Alsubaei, F. S. (2023). A hybrid edge-cloud system for networking service components optimization using the internet of things. Electronics, 12(3), 649.

[11] Hercog, D., Lerher, T., Truntič, M., & Težak, O. (2023). Design and implementation of ESP32-based IoT devices. Sensors, 23(15), 6739.

[12] Kham, S. R., Marmaroli, P., Minier, J., & Boulandet, R. (2021). Implementation and performance assessment of a MEMS-based sound level meter. Proceedings of Euronoise 2021, 25-27 October 2021, Madeira, Portugal (online).

[13] Monolithic Power Systems. (2023). MP8765GQ-Z: 24V, 6A High Efficiency Synchronous Step-Down Converter (Rev. 1.0) [Datasheet]. [https://tinyurl.com/24nwzv72]

[14] Deepgram. (2024). Whisper-based speech-to-text and text-to-speech API documentation. https://developers.deepgram.com/docs/make-your-first-api-request

[15] Groq Inc. (2024). LPU inference engine: Technical whitepaper. https://groq.com/technology

[16] Kezunovic, M., Ren, J., & Lotfifard, S. (2016). Design, modeling and evaluation of protective relays for power systems. Cham: Springer.

[17] Amazon. (2023). Echo Dot (5th Gen) Technical Specifications. https://www.amazon.com/echo-dot-5th-gen/dp/B09B8V1LZ3

[18] Bourechak, A., Zedadra, O., Kouahla, M. N., Guerrieri, A., Seridi, H., & Fortino, G. (2023). At the confluence of artificial intelligence and edge computing in iot-based applications: A review and new perspectives. Sensors, 23(3), 1639.

[19] Shukla, S., Hassan, M. F., Tran, D. C., Akbar, R., Paputungan, I. V., & Khan, M. K. (2023). Improving latency in Internet-of-Things and cloud computing for real-time data transmission: a systematic literature review (SLR). Cluster Computing, 1-24.

[20] Poudel, D., Can Cakmak, M., & Agarwal, N. (2024). Beyond the click: How youtube thumbnails shape user interaction and algorithmic recommendations. In International Conference on Advances in Social Networks Analysis and Mining (pp. 181-196). Cham: Springer Nature Switzerland.