Smart Voice Assistant Robot Using Arduino and AI

R.Nandhini[1], A.Vijayalakshmi[2]

R.Nandhini[1],BCA.,M.Sc.,M.Phil.,Assistant Professor,

Department of Computer Science and Applications

D.K.M College For Women (Autonomous)

A.Vijayalakshmi[2],PG Student,Department of Computer Science and Applications

D.K.M College For Women (Autonomous),Vellore,Tamil Nadu,India

Abstract — This paper presents the design and implementation of a Smart Voice Assistant Robot that integrates Arduino UNO-based hardware control with artificial intelligence (AI) driven voice processing. The proposed system enables hands-free, real-time control of a mobile robot through natural spoken commands processed by an AI voice assistant (Google Gemini) on a paired mobile device. Voice commands are transmitted wirelessly via an HC-05 Bluetooth module to the Arduino UNO, which controls four DC gear motors through an L293D dual H-bridge motor driver. The robot supports bidirectional movement, speed regulation, and audio feedback through an onboard speaker and audio player module. The entire system is powered by rechargeable lithium batteries, making it portable and self-contained. Experimental evaluation confirms reliable command recognition, low response latency, and robust motor control under varied operating conditions. The system achieves a voice recognition accuracy of approximately 93% in indoor environments and an average command-to-action response time of under 350 ms. The proposed platform is cost-effective, scalable, and suitable for applications in home automation, assistive technology, educational robotics, and human–robot interaction research.

Keywords — Arduino UNO, Voice Control, HC-05 Bluetooth, L293D Motor Driver, AI Voice Assistant, Google Gemini, Human–Robot Interaction, Embedded Systems.

I. INTRODUCTION

Robotics is a rapidly advancing interdisciplinary field that merges electronics, mechanical engineering, sensor technology, and software to create machines capable of autonomous or semi-autonomous operation. With the proliferation of embedded computing platforms such as Arduino and the democratization of artificial intelligence, robotic systems that were once confined to industrial settings are increasingly being deployed in homes, hospitals, educational institutions, and public spaces [1].

One of the most impactful developments in this landscape is voice-based human–robot interaction. Natural language communication removes the barrier of specialized controllers, enabling users with limited technical expertise—including the elderly and individuals with physical disabilities—to interact with robotic systems intuitively [2]. Voice assistants powered by deep learning models, such as Google Gemini, can parse complex spoken instructions with high accuracy and generalize across different accents, speaking rates, and ambient noise conditions.

Despite these advances, many low-cost robotic platforms still rely on button-based mobile applications or predefined command sets with rigid syntax. There is a clear need for systems that combine affordable hardware with intelligent, flexible voice control. This paper addresses that need by presenting the design, implementation, and evaluation of a Smart Voice Assistant Robot that couples an Arduino UNO microcontroller with AI-powered voice processing to achieve responsive, natural, and reliable voice-controlled mobility.

The main contributions of this work are: (i) a modular hardware architecture that integrates Arduino UNO, HC-05 Bluetooth, L293D motor driver, DC gear motors, and an audio feedback unit into a compact, battery-operated mobile platform; (ii) a software pipeline that offloads speech recognition and natural language understanding to a smartphone AI assistant, reducing onboard computational requirements; (iii) a quantitative evaluation of command recognition accuracy, response latency, and battery performance; and (iv) a discussion of real-world application scenarios and pathways for future enhancement.

II. LITERATURE SURVEY

A substantial body of prior research has explored voice-controlled and Bluetooth-enabled robotic systems. Kannan and Selvakumar [3] developed a walking robot governed by an EasyVR speech-recognition module communicating over a ZigBee link to an ATmega2560-based controller, demonstrating that off-board speech processing preserves main-CPU resources. Saravanan et al. [4] extended this concept to a wheeled vehicle in which a dedicated Android application handled voice recognition and relayed commands to the robot over Bluetooth, achieving usable control at ranges up to 100 m.

Bharani et al. [5] integrated voice-controlled locomotion with a robotic arm, illustrating that Arduino UNO is capable of coordinating multiple actuator groups in parallel. Complementary work on IoT-based monitoring by Sharma and Gupta [6] showed that integrating PIR sensors with an Arducam and Firebase API enables cost-effective motion-triggered surveillance—an architecture this paper extends by adding voice-command navigation. Manolescu et al. [7] demonstrated interactive conversational AI applied to IoT-connected devices, highlighting the feasibility of large language models as robot control interfaces.

More recently, Arntz [8] studied task-driven voice interaction using Amazon Alexa within virtual reality environments, underscoring the importance of latency and recognition accuracy for user experience. Gupta et al. [9] provided a rigorous analysis of speech-recognition wireless control for mobile robots, reporting that hybrid cloud-edge processing yields the best accuracy-latency trade-off. Patil et al. [10] presented a voice assistant–based food-serving robot, confirming the practical applicability of AI voice control in service robotics. Collectively, these works validate the architectural choices made in the present system while revealing gaps—particularly the lack of integrated audio feedback and quantitative latency benchmarking—that this paper addresses.

A. Comparison of Related Systems

Author / Year	Controller	Comm. Link	Voice Processing	Limitation
Kannan et al. [3], 2015	ATmega2560	ZigBee	EasyVR module	Fixed vocabulary, no audio feedback
Saravanan et al. [4], 2020	Arduino UNO	Bluetooth	Android App	No AI; button-based fallback only
Bharani et al. [5], 2021	Arduino UNO	Bluetooth	Android App	Limited to arm + wheels; no feedback
Gupta et al. [9], 2025	Raspberry Pi	Wi-Fi	Cloud ASR	High cost; requires internet
Proposed System	Arduino UNO	Bluetooth HC-05	AI Gemini (on-device)	Short-range; no obstacle detection

III. PROPOSED SYSTEM

The proposed Smart Voice Assistant Robot adopts a three-layer architecture—voice processing, wireless communication, and hardware control—as illustrated in Fig. 1. This layered design separates concerns, simplifies debugging, and allows each layer to be upgraded independently.

A. System Architecture

Layer 1 (Voice Processing) resides entirely on the user's smartphone. The AI voice assistant (Google Gemini) continuously listens for wake-word activation, captures spoken utterances, and maps them to discrete command tokens (FORWARD, BACKWARD, LEFT, RIGHT, STOP) or speed-adjustment values. This offloading strategy eliminates the need for a power-hungry processor or cloud connectivity on the robot itself, keeping hardware cost and weight low.

Layer 2 (Wireless Communication) is implemented by the HC-05 Bluetooth module operating in master/slave mode. Encoded command bytes are transmitted over the Bluetooth Serial Port Profile (SPP) at 9600 baud. The HC-05 was selected for its proven reliability in indoor short-range (up to ~10 m) applications, low power draw (~40 mA during transmission), and straightforward AT-command configuration.

Layer 3 (Hardware Control) is centred on the Arduino UNO (ATmega328P, 16 MHz). The firmware continuously polls the hardware serial buffer; upon receipt of a valid command byte, it drives the appropriate output pins connected to the L293D motor driver to achieve the desired motion. An audio response is simultaneously triggered via the DFPlayer Mini audio module, providing verbal confirmation of the executed command.

B. Hardware Architecture

Table II summarises the key hardware components and their specifications. The Arduino UNO operates at 5 V and provides 14 digital I/O pins, of which six support PWM output—sufficient to control motor speed and direction. The L293D H-bridge can supply up to 600 mA per channel (1.2 A peak), adequate for the N20 DC gear motors (operating at ~200 mA stall). Two separate battery rails are used: a 7.4 V Li-ion pack for motors, and a 5 V USB power bank for the Arduino and logic circuitry, preventing motor-induced voltage noise from corrupting the microcontroller.

Component	Specification	Function
Arduino UNO	ATmega328P, 16 MHz, 5 V, 14 digital I/O	Central controller
HC-05 Bluetooth	SPP, 9600 baud, up to 10 m range	Wireless communication
L293D Motor Driver	Dual H-bridge, 600 mA/ch, 4.5–36 V	Motor speed & direction
N20 DC Gear Motors (×4)	6 V, 200 RPM, ~200 mA stall current	Wheel drive
DFPlayer Mini + Speaker	SD card playback, 3 W, 8 Ω speaker	Audio feedback
Li-ion Battery	7.4 V, 2200 mAh (motors); 5 V USB bank (logic)	Power supply

Table II: Hardware Components and Specifications

IV. SOFTWARE IMPLEMENTATION

A. Arduino Firmware

The Arduino firmware is developed in the Arduino IDE using embedded C/C++. At startup, the setup() function initialises the hardware serial port at 9600 baud for Bluetooth communication and configures the PWM outputs that drive the L293D enable pins for speed control. A global speed variable (default = 130 out of 255) is adjustable at runtime by transmitting byte values greater than 10 from the mobile application, allowing the user to tune robot velocity without reflashing the firmware.

The main loop() function polls Serial.available() on every iteration. When a byte is present, it is read and compared against a lookup table of command codes. Command codes 1–5 correspond to FORWARD, BACKWARD, LEFT, RIGHT, and STOP respectively, while codes 6 and 7 implement timed quarter-turns (400 ms pulse) for voice-issued relative rotation commands. This architecture ensures an end-to-end software latency of less than 5 ms from byte receipt to pin assertion, which is negligible compared to Bluetooth transmission delay.

B. Motor Control Logic

Each movement function (forword, backword, turnLeft, turnRight, Stop) directly calls the AF_DCMotor API on all four motor objects. For turning manoeuvres, opposing motor pairs are driven in reverse while the other pair moves forward, producing tank-style differential steering. The RELEASE command is issued on Stop(), actively de-energising the motor windings rather than applying dynamic braking, which reduces current draw and heat generation during prolonged stationary periods.

C. Mobile Application and AI Voice Pipeline

The companion Android application is built on a Bluetooth SPP stack. On launch, the user selects the HC-05 device from a paired-device list, and the app establishes a socket connection. A dedicated voice-recognition button activates the Google Gemini voice assistant, which processes the utterance, maps it to one of the supported command intents (directional movement, stop, speed adjustment), and encodes the result as a single byte transmitted over the Bluetooth socket. The app also exposes manual D-pad buttons for redundant control, useful in noisy acoustic environments where voice recognition confidence falls below threshold.

V. RESULTS AND DISCUSSION

The system was evaluated in a controlled indoor environment (15 m × 10 m laboratory) over 150 test trials covering five command classes. Performance was assessed across three metrics: (i) voice recognition accuracy, (ii) command-to-action response time, and (iii) operating range. Table III summarises the results.

Command	Trials	Correct (%)	Avg. Latency (ms)
FORWARD	30	96.7	318
BACKWARD	30	93.3	325
LEFT	30	90.0	341
RIGHT	30	90.0	339
STOP	30	96.7	298
Overall	**150**	**93.3**	**324**

Table III: Experimental Results — Command Recognition and Latency

The overall voice recognition accuracy of 93.3% is consistent with prior smartphone-based voice-control systems (Gupta et al. [9] reported 91–95% for similar indoor conditions) and significantly exceeds hardware EasyVR-based systems (~85%) reported by Kannan et al. [3]. The slight reduction in accuracy for LEFT and RIGHT commands (90%) is attributed to phonetic similarity in Tamil-accented English, where the two words share stressed vowel sounds; a post-processing disambiguation step will be explored in future work.

The mean command-to-action latency of 324 ms encompasses Gemini processing time (~180 ms), Bluetooth transmission (~80 ms), and Arduino serial parsing and pin assertion (~5 ms). The remaining ~59 ms is attributed to Android socket buffering. This latency is imperceptible for navigation tasks where commands are issued at a natural conversational cadence, and falls well below the 500 ms threshold identified in human-factors literature as the point at which users perceive unacceptable lag in teleoperation contexts.

Effective Bluetooth communication was maintained across the full test area up to a range of 9.8 m, confirming the HC-05's rated 10 m specification. No packet loss was observed within 8 m; beyond 8.5 m, occasional single-byte drops were detected but did not cause unsafe robot behaviour due to the STOP-on-no-signal logic implemented in the firmware. Battery endurance testing showed that the 2200 mAh Li-ion pack sustained continuous operation for approximately 2.4 hours at default speed, sufficient for typical demonstration and assistive-use scenarios.

VI. APPLICATIONS

The Smart Voice Assistant Robot is suitable for a range of real-world domains. In assistive technology, it enables individuals with reduced motor function to command a mobile platform for object retrieval or environmental interaction without physical effort. In healthcare settings, the robot can deliver lightweight items—medication, documents, or water—in ward corridors, reducing nurse workload. In educational robotics, the transparent hardware-software stack provides students with a concrete, hands-on platform for learning embedded programming, wireless communication, and AI integration. In smart home automation, the robot can serve as a mobile IoT hub, responding to voice commands to navigate to specific rooms and interact with home devices via IR or RF relay modules. Finally, as a research prototype the system provides a reproducible baseline for exploring multi-modal control, obstacle avoidance, and large-language-model–based task planning in mobile robotics.

VII. CONCLUSION

This paper presented a Smart Voice Assistant Robot that successfully integrates Arduino UNO hardware control with AI-powered voice processing via the Google Gemini assistant and Bluetooth wireless communication. The system achieves a voice command recognition accuracy of 93.3% and an average command-to-action latency of 324 ms under indoor test conditions, demonstrating performance competitive with more expensive platforms. The modular, layered architecture keeps total component cost below INR 2,500 (~USD 30), making the system accessible for educational and assistive applications. Future work will focus on integrating ultrasonic obstacle avoidance, extending communication range using Wi-Fi (ESP8266/ESP32), adding natural language task chaining via LLM-based planning, and conducting user studies with individuals with motor disabilities to evaluate real-world assistive impact.

REFERENCES

[1] V. D. Manolescu, H. AlZu'bi, and E. L. Secco, "Interactive Conversational AI with IoT Devices for Enhanced Human-Robot Interaction," J. Intell. Commun., 2025.

[2] A. Arntz, "Enhancing Human-Robot Collaboration in Virtual Reality: A Task-Driven Communication System Using Alexa Voice Service," in Proc. 2025 IEEE Int. Conf. AIxVR, pp. 277–280, Jan. 2025.

[3] K. Kannan and J. Selvakumar, "Arduino Based Voice Controlled Robot," Int. J. Eng. Res., 2015.

[4] M. Saravanan, B. Selvababu, A. Jayan, A. Anand, and A. Raj, "Arduino Based Voice Controlled Robot Vehicle," Int. J. Eng. Res. Technol., 2020.

[5] S. Bharani, K. V. Sujith, S. Tanushree, and Y. Tambe, "Arduino Based Voice Controlled Robotic Arm and Wheels," Int. J. Eng. Sci. Comput., 2021.

[6] M. Sharma and S. C. Gupta, "An IoT Based Smart Surveillance and Monitoring System Using Arduino," Int. J. Comput. Appl., 2018.

[7] V. D. Manolescu, H. AlZu'bi, and E. L. Secco, "Interactive Conversational AI with IoT Devices," J. Intell. Commun., 2025.

[8] A. Arntz, "Enhancing Human-Robot Collaboration in VR," IEEE AIxVR 2025, pp. 277–280.

[9] S. Gupta, U. Mamodiya, and A. J. Al-Gburi, "Speech Recognition-Based Wireless Control System for Mobile Robotics," Automation, vol. 6, no. 3, p. 25, 2025.

[10] L. Patil et al., "Voice Assistant Based Food Serving Robot," Int. J. Adv. Comput. Theory Eng., vol. 14, no. 1, pp. 195–200, 2025.

[11] B. Sarada et al., "Voice Controlled Robot using Bluetooth," IJERT, Vol. 12, May 2023.

[12] V. Teeda et al., "Robot Voice: A Voice Controlled Robot using Arduino," arXiv Preprint, 2024.

[13] L. Basyal, "Voice Recognition Robot with Real-Time Surveillance and Automation," arXiv Preprint, 2023.

[14] R. Zhang et al., "Walk along: Controlling mobile robot with voice and gestures," ACM Trans. Hum.-Robot Interact., 2025.

[15] G. Likitha et al., "Implementation of Humanoid Robot Using Arduino Mega," in Proc. STCR 2025, pp. 1–6, IEEE, May 2025.

[16] Y. Lai et al., "Natural multimodal fusion-based human–robot interaction via large language model," IEEE Robot. Autom. Mag., 2025.

[17] G. Pandy, V. J. Pugazhenthi, and J. A. Lourdusamy, "Human-Robot Interfaces: A Comprehensive Study," Eur. J. Comput. Sci. Inf. Technol., vol. 13, no. 2, pp. 51–63, 2025.

[18] H. T. Wu et al., "Developing a Consumer Electronics Robotics with a Large Language Model Based on a Trustworthy AI Framework," IEEE Tra
ns. Consum. Electron., 2025.