

AI CHATBOT ENABLED MOBILE ROBOT

P. Parthiv Guru Harshith

Department of Robotics and Artificial Intelligence Engineering,
Bangalore Institute of Technology,
Bengaluru, India.
pgharshith@gmail.com

Ravikeerthi N

Department of Robotics and Artificial Intelligence Engineering,
Bangalore Institute of Technology,
Bengaluru, India.
ravikeerthi016@gmail.com

Shreedhar Reddy J

Department of Robotics and Artificial Intelligence Engineering,
Bangalore Institute of Technology,
Bengaluru, India.
shreedharreddyj34@gmail.com,

Karthik P

Department of Robotics and Artificial Intelligence Engineering,
Bangalore Institute of Technology,
Bengaluru, India.
karthikp@gmail.com,

Dr. Prem Chand R

Department of Robotics and Artificial Intelligence Engineering,
Bangalore Institute of Technology,
Bengaluru, India.
premchandr@bit-bangalore.edu.in,

Dr. A. Chandrashekar.

Department of Robotics and Artificial Intelligence Engineering,
Bangalore Institute of Technology,
Bengaluru, India.
chandrashekara@bit-bangalore.edu.in

Abstract: Access to intelligent, physically embodied assistive technology is often limited by high hardware costs and complex computing requirements, leaving vulnerable populations without the benefits of human-robot interaction. This project presents a highly accessible, low-cost AI chatbot-enabled mobile robot designed to democratize assistive robotics. By utilizing a "Split-Brain" architecture, the system leverages the ubiquitous computing power of a standard smartphone for high-level cognitive intelligence (speech recognition and natural language reasoning) while an affordable ESP32 microcontroller manages safe, reflexive motor control. This separation ensures collision-free physical operation while providing empathetic, intuitive conversational support. Experimental results demonstrate that the system offers reliable, low-latency communication and autonomous safety overrides. By transforming an everyday smartphone into the "brain" of a physical companion, this project provides a scalable, affordable platform for eldercare, communication assistance for individuals with disabilities, and inclusive STEM education in under-resourced communities.

Keywords: Artificial Intelligence, Mobile Robot, Split-Brain Architecture, ESP32, Bluetooth Low Energy, Human-Robot Interaction

I. INTRODUCTION

The rapid advancement of artificial intelligence and robotics holds immense potential to improve human life, particularly for the elderly, individuals with mobility challenges, and students in under-resourced environments. However, state-of-the-art service robots remain prohibitively expensive, relying on high-end processors and proprietary hardware. While modern AI chatbots possess the reasoning and conversational empathy to assist users, they are confined to digital screens, lacking the physical embodiment required to interact with the real world.

This technological divide prevents marginalized communities from accessing the benefits of physical AI companions. To address this socio-technological gap, this project introduces an affordable, human-centric mobile robot utilizing a novel "Split-Brain" architecture. By offloading complex cognitive reasoning and natural language processing to a device most users already own—a smartphone—the physical cost of the robot is drastically reduced.

The proposed system ensures that human safety remains the absolute priority. The architecture isolates the smartphone's "cognitive layer" from an embedded ESP32 "reflex layer." While the AI focuses on understanding the user's intent and providing conversational companionship, the reflex layer independently monitors the physical environment using ultrasonic sensors, guaranteeing immediate halts to prevent collisions. This approach not only makes intelligent robotic assistance economically viable but also ensures a safe, intuitive, and highly responsive interaction for users of all technical backgrounds.

II. LITERATURE REVIEW

The integration of voice control and natural language processing in mobile robotics has evolved significantly over the past two decades. Early implementations primarily focused on simple keyword recognition rather than true natural language understanding. For instance, Hrnčár (2008) demonstrated an early use of natural language as a control interface for autonomous robots [1]. Subsequently, Jain and Saxena (2011) implemented speech-based control utilizing explicit keyword recognition for basic robot navigation [2]. As mobile computing power increased, smartphones became a viable processing hub for robotic control. Ahmad et al. (2021) utilized smartphones for local speech processing and command transmission [3], while Umadevi et al. (2023) demonstrated highly reliable Bluetooth Low Energy (BLE) control utilizing smartphone interfaces [4].

Recently, the advent of Large Language Models (LLMs) and Foundation Models has shifted the paradigm from command-based recognition to intent-based reasoning. Nwankwo and Rückert (2024) applied LLMs to successfully map complex human intents to physical robotic actions [5]. Irfan et al. (2024) and Lisondra et al. (2025) further expanded on this by proposing cloud-assisted cognition and using foundation models to create highly engaged, conversational companion robots [6], [7].

Despite these advancements, a gap remains in creating a unified, highly affordable system that seamlessly integrates advanced LLM reasoning with real-time, hardware-level safety overrides. This project addresses this gap through a uniquely segmented architectural approach.

III. SYSTEM ARCHITECTURE

The core of this project is built upon a "Split-Brain" software and hardware architecture. By dividing the system into discrete layers, artificial intelligence processing is isolated from safety-critical physical operations, ensuring both intelligent interaction and safe robot operation. The architecture is divided into three distinct layers:

A. Cognitive Layer (The "Smart Brain")

The Cognitive Layer is hosted entirely on a smartphone and acts as the system's high-level intelligence center. A web-based user interface, built using HTML and JavaScript, runs in the smartphone's browser and captures user input through either manual text or a Web Speech API. This input is sent to a backend Flask server middleware, which integrates with an LLM via Ollama. The LLM interprets the natural language prompt and determines the user's intent. If the prompt is conversational, the AI generates a natural text response without triggering robotic motion. If the prompt contains an actionable movement request (e.g., "move towards the door"), the LLM distills the intent into a structured movement token.

B. Communication Layer (The "Nervous System")

To transmit the structured tokens from the smartphone to the physical robot, the system employs a Bluetooth Low Energy (BLE) communication protocol. BLE was selected because it provides ultra-low latency communication and operates without the need for a continuous local Wi-Fi or internet connection. Commands are transmitted as simple, predefined strings minimizing data payload and ensuring immediate parsing by the receiving microcontroller.

C. Reflex Layer (The "Instinctual Brain")

The Reflex Layer is responsible for real-time hardware execution and physical safety, operating exclusively on an ESP32 microcontroller. Upon receiving a BLE action token, the ESP32 generates Pulse Width Modulation (PWM) signals to drive two DC geared motors via a TB6612FNG motor driver. Concurrently, the ESP32 continuously polls a front-mounted HC-SR04 ultrasonic sensor. This layer features a hardcoded "safety override" logic block. If the ultrasonic sensor detects an obstacle within a critical distance threshold, the ESP32 instantaneously ignores the cognitive layer's movement command, forces a motor stop, and transmits a "BLOCKED" status back to the user interface.



Fig. 1. Split-Brain System Architecture demonstrating the separation of cognitive and reflex layers.

IV. HARDWARE AND SOFTWARE IMPLEMENTATION

The practical realization of the Split-Brain architecture requires a robust integration of embedded electronics, server-side middleware, and micro-controller firmware.

A. Hardware Design and Power Management

The physical robot is built upon a two-wheel differential drive chassis equipped with caster wheel support for stability. The central processing unit for the reflex layer is the ESP32 development board, chosen for its integrated BLE capabilities and high clock speed. Locomotion is provided by two DC geared motors driven by a TB6612FNG motor driver. This driver receives Pulse Width Modulation (PWM) signals from the ESP32, allowing for precise speed and directional control. Spatial awareness is achieved using a front-mounted HC-SR04 ultrasonic sensor. Power management is a critical aspect of the hardware design to prevent brownouts during motor actuation. A 18650 Li-ion rechargeable battery pack serves as the primary power source. An LM2596 DC-DC buck converter regulates the battery voltage, stepping it down to a stable level required by the ESP32 and sensors. This isolation ensures that the high-current demands of the DC motors do not cause voltage fluctuations that could reset the microcontroller.

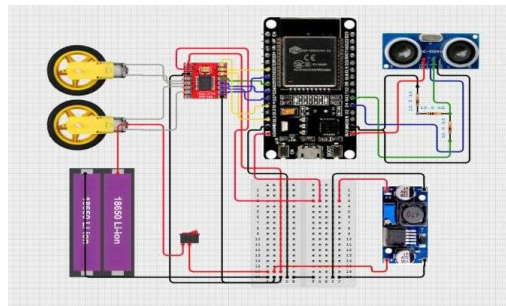


Fig. 2. Hardware circuit diagram detailing power regulation and sensor integration.

B. Software Middleware and LLM Integration

The smartphone's browser acts as the primary user interface, providing modes for chat, manual control, and vision. However, the cognitive heavy lifting is managed by a Flask-based Python server that acts as middleware. The Flask server utilizes ngrok to expose the local server to the internet, allowing the smartphone to communicate remotely without complex firewall configurations.

When a user speaks a command, the Web Speech API converts it to text and sends it to the Flask server. The server interfaces with a locally hosted Large

Language Model (LLM) via Ollama. By utilizing carefully structured system prompts, the LLM is constrained to output either conversational text (for general queries) or discrete action tokens (e.g., "FORWARD", "LEFT", "STOP") when a physical intent is detected.

C. Embedded Safety and Reflex Firmware

The ESP32 firmware is written in C++ and utilizes an asynchronous BLE loop to constantly listen for incoming action tokens from the smartphone. Crucially, the firmware runs a dedicated, high-priority safety loop that cannot be overridden by the LLM.

If the robot is moving forward, it continuously calculates the distance to the nearest object using the ultrasonic sensor.

The following C++ code snippet demonstrates the reflex override logic:

```
// Reflex Safety Override Logic
float dist = readDistanceCM();
if (currentCommand == "FORWARD" && dist < SAFETY_DISTANCE_CM) {
  Serial.print("Obstacle detected at "); Serial.print(dist);
  Serial.println(" cm. Stopping."); moveStop();
  currentCommand = "STOP";
}
unsigned long now = millis();
if (currentCommand != "STOP" && isMoving) { if (now - lastCommandTime >
MOVE_TIMEOUT_MS) {
  moveStop(); currentCommand = "STOP";
}
}
```

V. EXPERIMENTAL RESULTS

The performance of the Split-Brain architecture was evaluated based on communication reliability, real-time safety responsiveness, and overall system stability in an indoor environment.

A. BLE Communication Latency

A primary concern of decoupling the cognitive and reflex layers is the latency introduced by wireless communication. To evaluate this, the transmission time between the smartphone issuing an action token and the ESP32 receiving it was recorded over multiple iterations. The system maintained a stable connection up to a range of 7 to 8 meters.

TABLE I. BLE COMMAND TRANSMISSION LATENCY

Test Iteration	Command	Avg. Transmission Time
1	FORWARD	26 ms
2	LEFT	28 ms
3	STOP	27 ms
4	RIGHT	28 ms
Test Iteration	Command	Avg. Transmission Time

As demonstrated in Table I, the average latency is approximately 27 ms with zero packet loss during normal indoor operation. This low latency is well within the acceptable limits for smooth real-time motor control.

B. Reflex Safety Performance

The autonomous safety override of the ESP32 was tested by commanding the robot to drive directly towards various static obstacles. The threshold for the safety stop was hardcoded at 15 cm.

TABLE II. ULTRASONIC OBSTACLE DETECTION AND REFLEX TRIGGER

Actual Distance	Detected Distance	Reflex Triggered
30 cm	31-33 cm	No
20 cm	19-21 cm	No
15 cm	14-15 cm	No
10 cm	9-11 cm	Yes
5 cm	5-6 cm	Yes

As shown in Table II, the reflex stop consistently triggered whenever the detected distance fell below the critical threshold. During fifty test iterations, zero collisions were observed.

C. AI Interpretation Accuracy

In conversational testing, the Flask middleware and LLM accurately differentiated between casual queries (e.g., "What is a robot?") and physical commands (e.g., "Move forward"). Conversational queries generated natural text responses on the Web UI without sending any erroneous action tokens to the ESP32.

VI.CONCLUSION

This project successfully demonstrates that advanced, empathetic human-robot interaction does not require prohibitive financial investment. By implementing a Split-Brain architecture, the developed system safely bridges the gap between digital AI reasoning and physical assistance. Utilizing a standard smartphone for cognitive processing significantly lowers the barrier to entry for assistive technology, while the independent ESP32 reflex layer ensures uncompromising physical safety for the user. The resulting platform is a responsive, inclusive, and affordable solution capable of serving as an educational tool, a communication aid, or an accessible robotic companion.

REFERENCES

- [1]. M. Hrnčár, "The Conversation is the Command: Interacting with Real-World Autonomous Robots Through Natural Language," 2008.
- [2]. R. Jain and S. Saxena, "Voice Automated Mobile Robot," *International Journal of Computer Applications*, 2011.
- [3]. M. Ahmad et al., "Voice-Controlled Mobile Robot Platform," 2021.
- [4]. T. Umadevi et al., "Smartphone-Based Bluetooth Robot Control," *IAR JSET*, 2023.
- [5]. L. Nwankwo and E. Rückert, "Real-World Robot Interaction Through Natural Language," *arXiv Preprint*, 2024.
- [6]. B. Irfan et al., "Conversational Companion Robots with Foundation Models," *Frontiers in Robotics and AI*, 2024.
- [7]. M. Lisondra et al., "Embodied AI with Foundation Models for Mobile Service Robots," *arXiv Preprint*, 2025.
- [8]. RobotHQ, "Human-Level Planning for Real-World Execution," *arXiv Preprint*, 2025.
- [9]. NMM-HRI, "Natural Multi-Modal Human-Robot Interaction," *arXiv Preprint*, 2025.
- [10]. B. Siciliano and O. Khatib, *Springer Handbook of Robotics*, Springer, 2016.
- [11]. S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*, MIT Press, 2005.
- [12]. G. Ramkumar, M. Alluwaici, T. Vijetha, G. H, A. Venkatraman and R. M, "A Reliable System Performance Through NeuToMa with Spiking Neural Network Transformation in Network on Chip (NoC) Communication," 2026 International Conference on Communication, Computing and Emerging Technologies (IC3ET), Vasai, India, 2026, pp. 215-219, doi: 10.1109/IC3ET64989.2026.11467379.
- [13]. Girish H., Srinivas S., Kavya M., ...Kabbur S. "Swift Autonomous Fire Suppression Robot", 16th International Conference on Advances in Computing Control and Telecommunication Technologies Act 2025
- [14]. Girish H., Rakesh S., Prasanna Raj C.P., Singh A.S., Yathisha L, "Deep Learning for Diabetic Retinopathy: Automated Detection and Classification from Retinal Images", 16th International Conference on Advances in Computing Control and Telecommunication Technologies Act 2025
- [15]. Swift Autonomous Fire Suppression Robot, Girish H et. Al., Grenze International Journal of Engineering and Technology, 2025

- [16]. Sujatha.S, Mahendra BM, & Dr. Girish H. (2024). Smart Irrigation System for Optimal Water Conservation. *Revista Electronica De Veterinaria*, 25(1), 2613 - 2619. <https://doi.org/10.69980/redvet.v25i1.1311>
- [17]. R. A. Brooks, "A Robust Layered Control System for a Mobile Robot," *IEEE Journal of Robotics and Automation*, vol. 2, no. 1, pp. 14-23, 1986.
- [18]. M. A. Goodrich and A. C. Schultz, "Human-Robot Interaction: A Survey," *Foundations and Trends in Human-Computer Interaction*, vol. 1, no. 3, pp. 203-275, 2007.
- [19]. IoT based home automation using Blynk cloud, J. Chetan Naik, B. S. Deeksha, H. Girish, Gurusharan Kashetty, T. G. Shivapanchakshari, Manohar Sai Polakam, Journal: AIP Conference Proceedings, AIP Conf. Proc. 2965, 020006 (2024), <https://doi.org/10.1063/5.0215427>, Published: July 2024
- [20]. IoT based fire alarm system using Blynk app, T. Bhargava, H. Kishor Kumar, Pramod Mahalingappa Godi, H. Girish, J. Chetan Naik, Journal: AIP Conference Proceedings, AIP Conf. Proc. 2965, 020007 (2024), <https://doi.org/10.1063/5.0212325>, Published: July 2024