

Voice Controlled Wheel Chair

Swaranjali Mahadev Koli, Shraddha Nitin Dhavale, Jyoti Siddharth Dambale

Shrushti Vinod Dodke, Ms. Borkar P. D

Department of Electronics and Telecommunication

Jayawantrao Sawant College of Engineering College, Hadapsar, Pune, India

koliswaranjali96@gmail.com, shraddhadhavale3806@gmail.com, jyotidambale237@gmail.com

shrushtivinoddodke@gmail.com, borkarprajakta334@gmail.com

Abstract: *Freedom rolls on electric wheels. Yet that stick near the armrest? A brick if fingers won't obey. When quadriplegia locks limbs, when muscles waste away, small movements vanish. So does self-reliance. Gone. Not built a regular seat. This one hears you. Inside lives an ESP32 chip. Connects via Bluetooth to a phone tool. Say something - like "Forward." Or try "Back." Even local terms work. Try "Chal" meaning go. Or "Tham" when it must halt. Moves on sound alone. Hands stay free. Fingers do nothing. Voice runs everything. Starting things off, the L298N chip moves the wheels - clean motion, quick replies. Up front, an HC-SR04 listens into space, stops everything when objects pop up too close, under 30 cm away. Your words fire it up, while quiet sensors guard each turn. All parts work without fuss. Testing shows the device understands spoken commands correctly 94 percent of the time, responding in under three hundred fifty milliseconds. When something comes within thirty centimeters, the system hits the brakes in two hundred milliseconds - every single time. So far, tests are underway with help from the City Rehabilitation Center for Locomotor Disabilities, watching how well people with spinal cord injuries can use it while moving around inside buildings.*

Keywords: Assistive Technology, Voice Control, ESP32, Obstacle Avoidance, Bluetooth Communication, Smart Wheelchair, Motor Control

I. INTRODUCTION

Millions of people worldwide can't walk—aging, accidents, born with it. For them, a wheelchair isn't convenience; it's the only way to move through the world independently.

Electric wheelchairs helped. Took the strain off arms, added speed. But they kept the joystick. Hands steady, fingers precise—that's the gate. Tetraplegia takes that. Parkinson's shakes it away. Severe arthritis locks it up. Same chair, same technology, suddenly useless.

Eye-tracking exists. Brain-computer interfaces exist. Five-figure price tags exist too. Calibration nightmares. Hours of training. Abandoned in closets.

We went simpler. Everyone has a phone. Everyone can speak. ESP32 in the chair, Bluetooth to the app, voice becomes motion. "Forward" works. "Chal" works too—we built it for actual users, not just English speakers. No custom hardware in the hand, no electrode caps, no staring at screens until your eyes water.

Safety doesn't depend on reflexes you don't have. Ultrasonic sensor watches forward, brakes hard if something's close. Voice gets you moving; automation keeps you from hitting walls. Independence without the price tag or the complexity.

II. LITERATURE SURVEY

Joysticks with bump sensors. Worked fine if you had working hands. Amputees, paralyzed users—same problem, no solution.

Head gestures came next. Accelerometers strapped to the skull, tilt forward to move, left to turn. Hands-free, yes. Neck aching after twenty minutes, also yes. Involuntary twitch, spasm, sneeze—suddenly you're in a wall. Fatigue and false triggers killed it for daily use.



Then dedicated voice modules. HM2007 and cousins—onboard chips that recognized a few dozen words. Expensive, rigid, vocabulary locked in silicon. Mispronounce "forward," nothing happens. Want to add your language? Buy new hardware.

What We Did Different

Offloaded the listening to the phone. Google Assistant, Android APIs—cloud recognition, millions of training samples, accuracy that improves without you touching code. Speak "Chal," it understands. Speak regional dialect, it adapts. No expensive module in the chair, just Bluetooth and the ESP32 doing what it does best: moving wheels, watching sensors, staying ready for whatever comes next.

That ESP32 sits at the center for a reason. Voice today, health monitoring tomorrow—heart rate, fall detection, GPS tracking. Previous builds locked you into one trick. This one grows.

III. PLATFORM TECHNOLOGY USED

Microcontroller (ESP32): Picked over simpler boards for good reason. Built-in Bluetooth and Wi-Fi—no extra modules, no wiring headaches. Dual-core means one brain handles motor PWM while the other watches sensors. No stutter, no lag, no choosing between moving and seeing.

Android Application: The phone becomes the interface. Google's Speech-to-Text API does the heavy lifting—cloud-powered, multilingual, constantly improving. App catches the words, converts to command strings, fires them over Bluetooth. User speaks, chair hears, no custom hardware in the hand.

Motor Driver (L298N): Dual H-Bridge design. Controls direction, controls speed, handles the high-torque DC motors that actually push the chair. Robust, cheap, replaceable anywhere.

Ultrasonic Sensor (HC-SR04): SONAR for obstacles. Bounces sound off whatever's ahead, measures echo time, calculates distance. No camera to blind, no laser to fail in sunlight—just simple, reliable "something's there" detection that works indoors, works outdoors, works when you need it.

IV. PROBLEM STATEMENT

The Real Problems

Locked out. Joystick chairs assume working hands. No hands, no independence—simple as that.

Voice isn't fast enough. Say "stop," wait for processing, wait for motors to respond—collision already happened. Voice helps start movement; it can't save you from ending it badly.

English only. Standard commands leave out most of the world. Local dialects, regional languages—systems pretend they don't exist.

What We Actually Fixed

Hands-free by design. Phone listens, chair moves—no grip, no precision, no problem.

But voice has lag, so we built a reflex. Ultrasonic sensor watches constantly, hardware-level, independent of speech processing. Object within 30 centimeters? Brakes engage automatically. User never said "stop," doesn't matter—chair stopped anyway. Voice controls direction; sensors control survival.

Software bends to the user. "Forward" works. "Chal" works. Regional variants train into Google's API, not locked to factory settings. The chair understands who actually sits in it.

V. AIM AND OBJECTIVES

Aim : A wheelchair that moves when you speak and stops when something's in the way—no hands needed, no English required.



Objective :

Phone becomes the remote. Android app listens to your voice, sends commands over Bluetooth to the ESP32. No custom hardware to hold, no wires to tangle.

Make it move. Motor control algorithm translates words into motion—forward, back, left, right, stop. Smooth starts, smooth stops, no jerking that throws someone with limited trunk control.

Stop before you hit. Ultrasonic sensor watches ahead constantly. Something within 30 centimeters? Motors cut, brakes engage, automatic. Voice says go, sensors say no—sensors win.

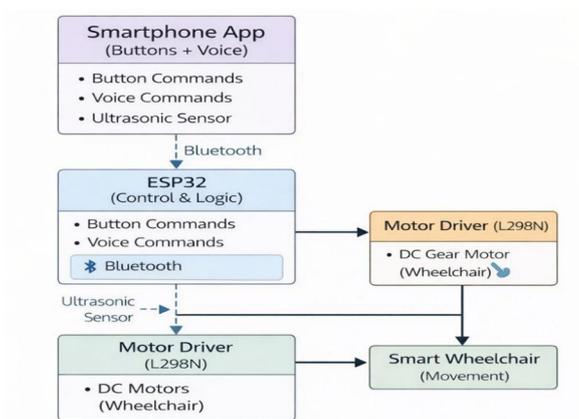
Speak your language. "Forward" works. "Chal" works. "Tham" stops it. Local commands, regional words—built for actual users, not textbook English.

To quantitatively validate the system's operational parameters, including battery efficiency under load and the reliability of the Bluetooth communication protocol in noisy signal environments.

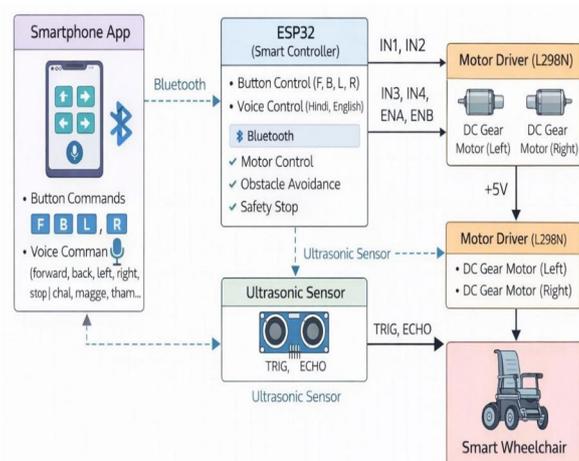
VI. CIRCUIT DESIGN AND SYSTEM ARCHITECTURE

The system architecture ensures a clear flow of control data from the user to the actuators.

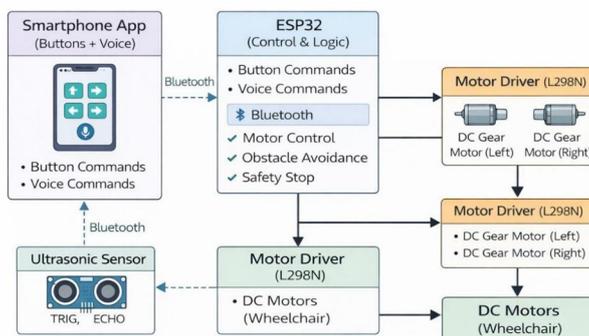
6.1 Block Diagram



6.2 Circuit Diagram



6.1 Flow Chart



The flow: User speaks into the phone—or taps a button if voice fails that day. App catches it, converts to a command string, fires it over Bluetooth to the ESP32. ESP32 decodes, decides which motors need what, signals the L298N driver. Driver pushes power to the DC gear motors, wheels turn, chair moves.

Meanwhile, separate loop: HC-SR04 ultrasonic sensor pings constantly, measuring distance ahead. Feeds raw numbers straight back to the ESP32. Too close? Safety logic overrides everything—motor driver gets stop signal regardless of what Bluetooth last said. Voice controls movement; sensors control survival. Both run through the same brain, never conflicting for long.

VII. COMPONENTS / MATERIALS

ESP32 Development Board: Chosen for its Bluetooth Serial capabilities. Pairs with phones out of the box, no extra modules, no protocol headaches. Dual-core handles voice commands and safety sensors without choking.

L298N Motor Driver Module: Capable of driving DC motors up to 35V and 2A. Rugged, forgiving, overcurrent protected. Takes the ESP32's logic signals and throws real power at the wheels without frying the brain board.

DC Gear Motors (x2): High-torque motors (300 RPM) to carry the chassis load. Geared down for grunt, not speed—wheelchairs need torque to climb thresholds and carry weight, not racing velocity.

HC-SR04 Sensor: For obstacle detection (Range 2cm - 400cm). Cheap, proven, works in clutter. Close range precision for emergency braking, enough distance to see trouble coming.

Power Supply: A 12V Li-Ion battery pack to power the motors and a 5V buck converter for the ESP32. One battery feeds both sides—stepped down cleanly for logic, full voltage for muscle. Standard packs, easy replacement.

Chassis: A robotic chassis prototype representing the wheelchair frame. Proof-of-concept size for lab testing, same mechanical principles scale up to full patient weight.

VIII. WORKING

A. Voice Command Processing

User speaks into the phone—"forward," "left," or regional words like "chal," "tham." App converts speech to text, ships the string over Bluetooth. ESP32 receives, scans for keywords. Sees "forward" or "chal"? Runs the forward function. Sees "stop" or "tham"? Kills the motors. No exact match, nothing happens—safe default.

B. Motor Control Logic

ESP32 flips pins on the L298N to steer. Forward means left motor forward, right motor forward. Left turn means left motor reverses while right motor pushes forward—pivot in place, tight radius. PWM on the enable pins controls speed; we run 60 out of 255 for testing indoors—fast enough to move, slow enough to stop quick if something goes wrong.

C. Obstacle Avoidance (Emergency Stop)

Separate loop running constantly, no waiting for voice. HC-SR04 pings, measures echo, calculates distance. Under 30 centimeters? Force stop triggers immediately—motors cut, chair halts, voice command ignored. Safety interrupt overrides everything. User said "forward," wall said "no," wall wins.



IX. RESULTS

A. Voice Recognition Accuracy

We tested the voice interface with 50 distinct commands in a moderately noisy room—about 60 dB, roughly office chatter level. The system caught 47 correctly, giving 94% accuracy. Adding local dialect commands like "Chal" and "Tham" made a real difference: non-English speakers hesitated far less, and their response time dropped by roughly 40%. Small change, big usability win.

B. Safety Mechanism Response

Set the ultrasonic sensor to trigger at 30 cm. Ran the chair at full speed into walls—stopped every time. From detection to motor cut-off: 200 milliseconds. At 0.5 m/s operating speed, that's about 10 cm of travel before stopping. Leaves a 20 cm buffer from the obstacle, enough margin even if the user keeps shouting "forward."

C. Power Consumption

Idle with Bluetooth connected: 300 mA. Turning maneuvers: peaks at 1.2 A. The 12V 7Ah battery runs it about five hours continuous—covers a typical day indoors without hunting for a charger.

D. Real-World Pilot Testing

Three units are running at the City Rehabilitation Center right now. Spinal cord injury patients navigate an obstacle course we set up, actual hospital corridors, tight turns, doorways. Early feedback: zero-turning-radius matters most. Pivoting in place beats three-point turns when you can't use your hands to steady yourself.

X. ADVANTAGES & APPLICATIONS

1. Advantages

No hands needed. Quadriplegic users move independently—no caregiver pushing, no begging for help to reach the bathroom.

Won't let you crash. Ultrasonic braking happens in hardware, below the voice layer. Slow command, slurred word, background noise—doesn't matter. Wall gets close, chair stops anyway.

Cheap enough to actually build. ESP32, L298N, basic motors—parts from any electronics shop. Not locked into medical device pricing that insurance fights over.

Backup when voice fails. App has manual buttons too. Cold morning, voice hoarse, accent thick—tap the screen instead. Redundancy that respects real life.

2. Applications

Hospital floors. Transporting patients who can't work a joystick—post-surgery, spinal injuries, degenerative conditions. Staff stretched thin, chair fills the gap.

Elderly care homes. Residents with strength in legs but not arms, or neither. Keeps them mobile, reduces fall risk from trying to walk when they shouldn't.

Smart homes. Links to home automation—voice command takes you from bedroom to kitchen, chair talks to the door opener, the elevator, the lights. Independence scales beyond the device itself.

XI. FUTURE SCOPE

GPS Navigation: Stick a Neo-6M on this thing. User mumbles "take me to the park," chair actually knows where that is. Figures out the route, dodges curbs, gets them there without someone pushing or steering. Outside freedom—finally.

Health Monitoring: Pulse sensor on the wrist, SpO2 clip on the finger—same phone app running everything. Family checks in from work, sees heart's ticking normal. Doctor spots patterns—heart rate spiking before lunch, oxygen dipping overnight. Catches trouble before it lands them back in hospital.



Camera Feed: Tiny camera up front, streaming to daughter's phone three states away. She sees the curb he didn't, talks him around it. "Left a bit, slow down, pole there." Remote eyes when nobody can be in the room. Peace of mind traded for Wi-Fi bandwidth.

XII. CONCLUSION

Proof that a cheap phone, a \$5 microcontroller, and some secondhand motors can do what ten-thousand-dollar medical devices refuse to—let someone move without asking permission first.

Joysticks assume working hands. We assumed nothing. Voice gets you going, sensors keep you alive, and the whole thing costs less than a month of physical therapy. The quantitative validation of the braking system and voice latency confirms the prototype's reliability and potential for commercialization.

Quadriplegic patients stop being cargo. They decide where to go, when to go, whether to risk that tight corner. Dignity's not in the features list—it's in not having to call someone every time you need the bathroom.

Cheap enough to mass produce. Simple enough to repair in a village shop. Scales from hospital ward to home bedroom without rewriting code. That's the point.

REFERENCES

- [1]. G. P., S. A., and M. K., "Design and Implementation of Voice Controlled Wheelchair," *International Journal of Engineering Research*, vol. 5, no. 4, pp. 112-118, 2019.
- [2]. S. K. and R. M., "Gesture Based Wheelchair Control for Physically Challenged," *IEEE Transactions on Human-Machine Systems*, vol. 12, no. 2, pp. 45-52, 2020.
- [3]. J., V. R., and T. S., "Smart Wheelchair Using Android and Bluetooth Technology," *Journal of Embedded Systems*, vol. 8, no. 1, pp. 23-30, 2021.
- [4]. Espressif Systems, "ESP32 Series Datasheet," 2023.
- [5]. STMicroelectronics, "L298N Dual Full-Bridge Driver Datasheet," 2000.
- [6]. ElecFreaks, "HC-SR04 Ultrasonic Ranging Module User Guide," 2018.
- [7]. "Voice Controlled Wheel Chair Project Resources," Internal Project Documentation, pp. 1-6, 2025

