Project Report
On

Design and Implementation of IOT-based Smart Bluetooth Controlled Wheel Chair

Submitted as partial fulfilment for the award of

BACHELOR OF TECHNOLOGY DEGREE

Session 2024-25

in

Electronics & Communication Engineering

By

Aayush Goyal (2100320310001)
Ankit Pandey (2100320310014)
Arnav Mehta(2100320310024)

Under the guidance of
Ms. Rakhi Kumari, Professor(ECE)

Estd. 2000

ABES ENGINEERING COLLEGE, GHAZIABAD
(Affiliated to DR. A.P.J. ABDUL KALAM TECHNICAL UNIVERSITY, U.P., LUCKNOW)

APRIL 2025

DESIGN AND IMPLEMENTATION OF SMART BLUETOOTH CONTROLLED WHEELCHAIR

by

Aayush Goyal (2100320310001)
Ankit Pandey (2100320310014)
Arnav Mehta (2100320310024)

Submitted to the Department of Electronics & Communication Engineering in partial fulfilment of the requirements for the degree of Bachelor of Technology in Electronics & Communication Engineering

ABES Engineering College, Ghaziabad Dr. A.P.J. Abdul Kalam Technical University, Uttar Pradesh Lucknow June, 2024

DECLARATION

We hereby declare that this submission is our work that, to the best of our knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Signature:

Name: Aayush Goyal

Roll number:2100320310001

Date:

Signature:

Name: Ankit Pandey

Roll number: 2100320310014

Date:

Signature:

Name: Arnav Mehta

Roll number: 2100320310024

Date:

CERTIFICATE

This is to certify that the project report entitled "Design and Implementation of IOT based smart Smart Wheel Chair for road accident detection" which is submitted by Aayush Goyal (2100320310001), Ankit Pandey(2100320310014), Arnav Mehta (2100320310024) in partial fulfilment of the requirement for the award of degree B. Tech in Department of Electronics and Communication Engineering of Uttar Pradesh Technical University, is a record of the candidates own work carried out by them under my supervision. The matter embodied in this thesis is original and has not been submitted for the award of any other degree.

(Signature)

Ms. Rakhi Kumari

Associate Professor ABES Engineering College, Ghaziabad

Date:

(Signature)
Dr. Kimmi Verma
HOD-ECE ABES Engineering College, Ghaziabad
Date:

ACKNOWLEDGEMENT

It gives us a great sense of pleasure to present the report on the B.Tech. The project was undertaken during B. Tech Final Year. We owe a special debt of gratitude to Ms. Rakhi Kumari, Associate Professor, Department of Electronics & Communication Engineering, ABES Engineering College, Ghaziabad for his constant support and guidance throughout our work. His sincerity, thoroughness, and perseverance have been a constant source of inspiration for us. It is only his cognizant efforts that our endeavors have seen the light of the day.

We also take the opportunity to acknowledge the contribution of Dr. Kimmi Verma, Head of Department of Electronics & Communication Engineering, ABES Engineering College, Ghaziabad for her full support and assistance during the development of the project.

We also do not want to miss the opportunity to acknowledge the contribution of all faculty members of the department for their kind assistance and cooperation during our project's development. Last but not least, we acknowledge our friends for their contribution to the completion of the project.

Signature:

Name: Aayush Goyal Roll number: 2100320310001

Date:

Signature:

Name: Ankit Pandey Roll number: 2100320310014

Date:

Signature:

Name: Arnav Mehta Roll number: 2100320310024

Date:

ABSTRACT

This project presents the design and implementation of a smart wheelchair aimed at empowering specially abled individuals with enhanced autonomy and ease of mobility. The proposed system addresses the limitations of traditional wheelchairs that rely on manual effort or caregiver assistance. Instead of using complex sensor arrays or voice recognition, this wheelchair is operated via a minimalist Android mobile application and Bluetooth communication, offering a simple and affordable control interface. By integrating core components like an Arduino Uno microcontroller, an HC-05 Bluetooth module, and a motor driver shield, the system ensures responsive and smooth motion through user commands.

The primary control mechanism is the mobile app interface, which includes four direction buttons—forward, backward, left, and right. Each button sends a specific character via Bluetooth, which is received by the Arduino and translated into motor actions using the L293D motor driver shield.

The motors used are BO (DC gear) motors, selected for their torque efficiency and suitability for moderate load-bearing, mounted on a sturdy MDF board for lightweight stability. The entire setup is powered by a rechargeable 18650 battery pack, ensuring hours of wireless operation on a single charge.

This solution is particularly impactful in the context of developing regions where advanced assistive devices remain financially out of reach. By emphasizing a low-cost, open-source, and easy-to-assemble design, the smart wheelchair offers a viable alternative to expensive commercial systems.

Testing showed that the wheelchair responds instantly to user commands with reliable Bluetooth connectivity, smooth transitions between directions, and sufficient power to support users weighing up to 60–70 kg. The straightforward app-based control makes it ideal for individuals with limited physical strength or mobility, requiring no technical training or calibration.

Thus, the smart wheelchair contributes meaningfully to the goal of inclusive and accessible design by offering a practical, replicable, and scalable mobility aid. Future enhancements may include optional features such as GPS tracking, battery monitoring, or iOS compatibility. This report outlines the complete development cycle—from problem identification to system design, implementation, and testing—and aims to support ongoing efforts in low-cost assistive technology innovation.

Keywords: Smart wheelchair, Assistive technology, Bluetooth control, Arduino, Mobile app interface

CONTENTS

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 INTRODUCTION

A wheelchair is required by around 111 million individuals worldwide, according to a World Health Organization (WHO) estimate. Being disabled is a natural part of being human. About 2.21% (or 26.8 million) of Indians have a disability of some type. There are 1.18 crore disabled women and 1.5 crore disabled men in India, out of a total population of 62.32 crore and 58.78 crore, respectively.

In today's society, people are expected to be self-sufficient despite any inherent physical or mental difficulties. People with physical disabilities must rely on others to take care of their fundamental requirements. The likelihood of their venturing outside

and interacting with the outside world is quite minimal unless they are given with contemporary movement tools like a wheel chair. Wheelchairs can be propelled manually or electrically, respectively. Only those with lower-limb disability have the first choice, and long-term use comes with additional health hazards. Moreover, manual-driven wheelchair efficiency is only 10% to 20%. Today Due to recent improvements in the industrial population, wheelchairs are a crucial piece of equipment in the biomedical sector. The need for the elderly and physically challenged is constantly growing.

The information above suggests that people with impairments have a big need for wheelchairs. The need for wheelchairs is essential for their comfort and to make life

simpler. We are recommending a smart wheelchair that monitors health and sends alerts in light of the issues experienced by people with disabilities.

The goal of this project is to develop a Bluetooth-controlled smart wheelchair operated through a mobile application. It is designed to assist individuals with physical disabilities in navigating their environment independently using simple directional commands. By utilizing a mobile app interface connected via Bluetooth, users can control the wheelchair's movement—forward, backward, left, and right— without relying on physical exertion. This system eliminates the need for manual steering or complex input methods, offering a practical and accessible mobility solution

In contrast to manual wheelchairs, this smart wheelchair is operated wirelessly through a mobile application using Bluetooth communication. The system allows users to control movement with simple button clicks, making it more efficient and convenient than traditional manual or joystick-operated wheelchairs. Individuals with limited hand mobility can benefit from this solution as it removes the need for physical force or fine motor control. The mobile app transmits directional commands to the microcontroller via Bluetooth, which then interprets the input and controls the wheelchair accordingly. The wheelchair is capable of moving in multiple directions, including forward, backward, left, and right.

The wheelchair system is intentionally designed to be minimalistic and easy to operate, focusing solely on Bluetooth-based control through a mobile application. It does not include additional modules such as flame sensors, health monitoring systems, or obstacle detection units. This approach ensures simplicity, reduces cost, and makes the system highly reliable for everyday use. By eliminating the need for complex sensors or voice recognition modules, the wheelchair maintains a streamlined design, allowing users to control its movement effortlessly through app-based directional buttons.

The system is designed to be controlled entirely through a mobile application, without the use of obstacle sensors or speech recognition modules. Navigation is handled through directional buttons on the app, allowing the user to move the wheelchair based on visual judgment and manual input. This user-driven control method avoids the complexities of person-dependent systems like voice recognition, ensuring consistent performance regardless of the user's speech patterns or environmental conditions

The system consists of a Bluetooth-enabled mobile application as the transmitter and the wheelchair’s onboard microcontroller (Arduino Uno) as the receiver. When the user presses a directional button on the mobile app, a corresponding command is sent via Bluetooth to the Arduino. Unlike systems that rely on stored voice commands or automated object detection, this wheelchair requires direct user input for movement, ensuring complete user control without the need for complex sensing or data storage mechanisms.

1.2 PROBLEM IDENTIFICATION

For individuals with physical disabilities, independent mobility plays a crucial role in the development of their social, cognitive, linguistic, and physical abilities. However, the high cost of electric wheelchairs has made them inaccessible for many families, particularly those from lower- and middle-class backgrounds. The goal of our project is to develop an affordable, wireless-controlled, and user-friendly wheelchair system to address these challenges.

Current electric wheelchairs in the market are often priced beyond the reach of many families, limiting access for people who would benefit the most from them. Our aim is to create a wheelchair that is both inexpensive and versatile, offering a cost-effective alternative without compromising on functionality. This project is focused on improving the lives of elderly and disabled individuals, helping them regain mobility and independence in their daily lives.

Our focus is specifically on individuals who may not have the ability to even move their hands. Therefore, we propose a Bluetooth-controlled smart wheelchair that allows users to control the wheelchair via a simple mobile app, eliminating the need for complex joysticks, sensors, or voice commands. This solution provides an affordable and efficient way for users to move without any technical expertise or complicated setup.

The system is designed to be straightforward, allowing users with limited mobility to take full control of their wheelchair through intuitive, easy-to-use app commands. This makes it an ideal solution for individuals who need a reliable and accessible means of transportation.

1.3 OBJECTIVES OF THE PROJECT

- To develop a Bluetooth-controlled wheelchair system that allows movement via a mobile app, ensuring ease of use for individuals with limited mobility.
- To create a wheelchair whose movement is controlled by simple button presses through a mobile app, providing an intuitive interface for users.
- To provide a mobility solution for individuals who are unable to control their hands or legs, enabling them to navigate independently.
- To offer the ability for the wheelchair user to control basic functions through the app, making it easier for them to manage their environment.
- This wheelchair will be equipped with a health monitoring system that tracks vital signs, providing regular updates to caregivers or guardians.
- While we are not incorporating flame detection in this project, we aim to focus on a simple and efficient design, prioritizing mobility and ease of use for the user.

1.4 SCOPE OF THE PROJECT

The scope of the Smart Wheelchair project encompasses the development and implementation of a low-cost, Bluetooth-controlled mobility system designed for individuals with physical disabilities. The primary objective is to provide a simplified alternative to traditional wheelchairs that require manual effort or complex electronic control mechanisms such as voice commands, joysticks, or sensors. This project aims to bridge the gap between affordability and functionality, by allowing users to control the wheelchair wirelessly using a dedicated Android mobile application.

This wheelchair is envisioned as a practical mobility solution for users who might not be comfortable with complex interfaces or those who have limited control over fine motor movements. Instead of depending on costly modules or sensor-based systems, the wheelchair's functionality is driven through Bluetooth commands sent from a mobile phone to an Arduino-based control system. The system interprets these commands and translates them into motor actions that move the wheelchair forward, backward, left, or right. This simple approach increases reliability and significantly reduces the cost of implementation.

The project’s scope includes the design and assembly of the hardware platform, integrating key components like Arduino Uno, HC-05 Bluetooth module, L293D motor driver shield, BO motors, and a rechargeable battery pack to power the system. The hardware is mounted on an MDF board which serves as a lightweight, durable base. From the software perspective, a minimalist Android application was developed with just four control buttons to simplify user interaction and minimize the learning curve.

This system is especially suitable for use in indoor environments, including hospitals, homes, and rehabilitation centers, where ease of navigation and wireless control can greatly benefit users. The project also sets the foundation for future upgrades, such as obstacle detection, GPS tracking, or voice command integration — but these features lie outside the current scope.

The Smart Wheelchair also serves an educational purpose, demonstrating how basic electronics, microcontroller programming, and mobile app integration can be combined to create real-world assistive technologies. By using readily available and inexpensive components, the system remains highly replicable and scalable for wider community impact.

Thus, the scope of this project covers a complete prototype development, from hardware integration to mobile control implementation, with a strong focus on affordability, ease of use, and practicality, without overcomplicating the system with unnecessary advanced features.

1.5 Methodology Overview

The methodology adopted for the Smart Wheelchair project follows a structured, modular approach that integrates microcontroller-based hardware control with mobile-based wireless communication. The entire workflow can be broken down into several systematic phases — component selection, circuit design, hardware assembly, software development, integration testing, and final evaluation.

- Component Selection & Design Planning

The first step in the methodology involved selecting cost-effective and reliable components that could deliver the desired performance while keeping the system simple. The key components included:

Arduino Uno – to act as the central controller.

HC-05 Bluetooth Module – for wireless communication between the Android app and Arduino.

L293D Motor Driver Shield – for controlling the direction and motion of motors. BO Motors with Wheels – selected for their moderate speed and sufficient torque.

18650 Lithium-ion Battery Pack – to ensure uninterrupted power supply to the system.

- Hardware Integration

The next step involved assembling the components on an MDF baseboard, which was chosen for being lightweight yet stable. The motors were mounted securely, and the wiring was carried out using jumper wires. The HC-05 module was connected to the Arduino's TX/RX pins, while the L293D motor shield was stacked directly onto the Arduino to drive the motors.

- Mobile Application Development

A simple Android application was created using open-source tools. The app interface includes four buttons: Forward, Backward, Left, and Right. Each button sends a unique character (such as 'F', 'B', 'L', 'R') over Bluetooth when pressed. This minimalistic design ensures ease of use, even for users unfamiliar with technology.

- Arduino Programming

The Arduino was programmed to read Bluetooth input via the serial port and execute motor commands accordingly. For instance, when it receives the character 'F', it powers both motors in the forward direction. The code includes logical conditions to handle each input and control the motor driver shield.

1.6 System Integration and Testing

Once the hardware and software were complete, the system was fully assembled and tested. The app was paired with the HC-05 module, and various movement commands were tested for accuracy, response time, and reliability. The system demonstrated stable operation within a 10- meter Bluetooth range, and directional changes were smooth and quick.

- Evaluation and Adjustments

During testing, minor adjustments were made to motor alignment and speed settings in code. The battery life was also monitored, and safety checks were performed to ensure all connections were stable and secure. The system consistently supported weights up to 60–70 kg, confirming its practical usability.

CHAPTER 2
LITERATURE REVIEW

2.1 Overview of Existing Work

In the domain of assistive mobility devices, particularly automated or smart wheelchairs, significant strides have been made over the past two decades. Existing systems focus on improving autonomy, safety, and accessibility for people with mobility impairments. The majority of work in this space involves the integration of advanced sensors, artificial intelligence, and multiple modes of input to provide users with hands-free or semi-autonomous control. However, many such systems remain complex, expensive, or impractical for deployment in real-world, low- resource settings.

Studies like that of Caltenco et al. (2012) introduced multimodal control systems, enabling users to control wheelchairs through voice, gestures, or eye movements. Others, such as Sarkar et al. (2017), developed voice-controlled wheelchairs using voice recognition modules like HM2007 integrated with Arduino microcontrollers. While these systems enhance user interaction, they often suffer from limited accuracy in noisy environments or require extensive calibration.

Research by Amit Kumar et al. (2015) explored the use of joystick-based wheelchairs, emphasizing the convenience for patients with partial upper body mobility. However, this excludes individuals with complete limb immobility. To address this, some projects turned to electroencephalography (EEG) or brain-computer interface (BCI) controlled wheelchairs, as in the case of Rebsamen et al. (2010). While revolutionary, these systems involve high costs, are difficult to maintain, and pose user adaptation challenges.

In contrast, Bluetooth-based wheelchairs have been investigated as simpler, low- cost alternatives. A prominent study by Patil and Pawar (2016) introduced a Bluetooth-controlled wheelchair using the HC-05 module and Android app interface, allowing forward, backward, left, and right movements via touch buttons. This concept has since been widely replicated in academic projects due to its simplicity and cost-effectiveness.

However, despite these promising developments, many such systems are either academic prototypes, not durable, or lack thorough usability testing. In particular, limited attention has been paid to minimalistic yet reliable designs that eliminate unnecessary complexity while retaining functionality — a core focus of this project.

2.2 Theoretical Background

The theoretical framework for this Bluetooth-controlled smart wheelchair is grounded in several domains: wireless communication systems, embedded systems and microcontroller programming, motor control theory, and human-computer interaction.

At the core of this system lies Bluetooth technology, specifically the HC-05 module, which operates on the IEEE 802.15.1 protocol. It enables short-range wireless communication (typically up to 10 meters) by establishing a serial connection between an Android smartphone and the Arduino. The mobile app transmits characters like 'F', 'B', 'L', and 'R' over this link, which the Arduino reads through its USART (Universal Synchronous/Asynchronous Receiver/Transmitter) interface.

The Arduino Uno acts as the central microcontroller unit (MCU). It continuously listens for serial data, and upon receiving specific characters, executes commands using conditional logic in C/C++ code. These commands translate into digital HIGH/ LOW signals on the Arduino's GPIO pins that are wired to a L293D motor driver shield.

The L293D is an H-Bridge driver IC, which allows for independent bidirectional control of two DC motors. It works by altering the current flow through the motors depending on the input signal logic. The project leverages BO (Battery Operated) motors, known for low RPM and moderate torque, making them suitable for lightweight robotic applications like this wheelchair. Motor direction and speed control are achieved through simple digital logic rather than PWM in this version, although PWM could be added for speed variation.

From the human-computer interaction (HCI) perspective, the system design focuses on usability and minimalism. Instead of overwhelming users with multiple sensors or complicated interfaces, the system uses a simple Android app with four buttons, reducing cognitive load and increasing reliability for elderly or disabled users.

This theoretical model prioritizes reliability, simplicity, and affordability over sophisticated automation. The system is modular, meaning components can be independently upgraded or replaced — for example, substituting HC-05 with Wi- Fi for greater range or integrating additional sensors for obstacle avoidance in future iterations.

2.3 Gaps in Existing Research

While existing smart wheelchair systems demonstrate innovation and technical depth, a number of critical gaps remain, especially when viewed from the lens of accessibility, cost-effectiveness, ease of use, and practical implementation.

First, many research initiatives emphasize advanced features such as speech recognition, EEG signal processing, or autonomous navigation using LIDAR or ultrasonic sensors. While these are impressive, they often require specialized hardware and maintenance, which increases cost and reduces reliability in real-world environments. Such systems are not feasible for middle-income or rural users, who may lack access to consistent power or technical support.

Second, most studies that incorporate mobile-based control often focus on software prototypes or simulations without real hardware deployment. There's a notable lack of fully integrated systems tested under realistic conditions, including actual weight-bearing capacity, surface handling, and battery endurance. Without this, it is difficult to assess whether these designs are truly practical for daily use by people with disabilities.

Another major gap is the user-centric evaluation. Most academic projects or research publications present results based on system functionality alone, often omitting feedback from actual users. The lack of usability testing, user training modules, and accessibility optimization is a significant oversight.

Also, there's minimal exploration into battery management and fail-safes. For any motor-driven system, power efficiency is critical — and yet, few existing projects provide clear power usage profiles, emergency stop mechanisms, or battery-level alerts.

Finally, the lack of modularity in many systems means that upgrading one component often necessitates redesigning the entire system. This discourages customization or future scaling — a key barrier to adoption in resource- constrained environments.

This project directly addresses these gaps by:

Using readily available low-cost components (HC-05, Arduino, L293D, BO motors).

Building a physically functional prototype that has been load-tested.

Emphasizing simplicity and reliability over high-tech features.

Laying a modular foundation for potential upgrades in future versions.

2.4 Summary of Literature Findings

To summarize, a comprehensive review of existing literature reveals both promising advancements and critical shortcomings in the development of smart wheelchair systems. The most successful projects typically integrate microcontrollers with wireless technologies to improve mobility. Systems employing Bluetooth modules like the HC-05 with Arduino boards are frequently found in academic contexts due to their simplicity, accessibility, and effectiveness in delivering basic directional control.
Existing work demonstrates a wide variety of approaches — ranging from joystick-based, voice-controlled, and even brainwave-operated wheelchairs. However, many of these systems suffer from practical limitations, including high cost, poor user interface design, low reliability in uncontrolled environments, and lack of long-term deployment.

Theoretical models highlight how serial communication, motor control circuits, and user interface design principles combine to form a foundational system for assistive robotics. Bluetooth-based control provides a non-invasive, easily maintainable method for controlling motorized systems, especially when paired with robust microcontroller logic.

The literature also reveals a trend of feature-overloaded systems that attempt to integrate advanced AI or sensor-based functionality without addressing the real-world constraints of cost, usability, and maintenance. These insights underscore the need for minimalistic, robust, and user-tested systems like the one developed in this project.

In conclusion, this project aligns with foundational concepts in embedded systems while addressing key gaps through a lean, tested, and modular prototype. It reinforces the idea that practical engineering doesn't always mean complex — sometimes, the most impactful systems are the simplest and most scalable.

CHAPTER 3

PROBLEM DEFINATION

3.1 Existing System

The traditional and existing systems for mobility assistance in individuals with physical impairments primarily involve manual wheelchairs and joystick-controlled motorized wheelchairs. While these have served a significant purpose over decades, their limitations have prompted the need for smarter, automated alternatives that provide better autonomy and accessibility.

Manual wheelchairs require upper body strength or a caregiver’s assistance for operation. This setup poses challenges for individuals with complete or partial paralysis, neuromuscular disorders, or spinal injuries. Moreover, the manual nature makes it impractical for long distances or in environments with uneven surfaces or slopes.

On the other hand, motorized wheelchairs with joystick interfaces present a major improvement in autonomy. They are usually equipped with rechargeable batteries and DC motors to drive the wheels. However, these systems assume that the user has at least one functional hand and fine motor control, which is not always the case — especially with patients suffering from advanced muscular dystrophy, quadriplegia, or cerebral palsy.

Some higher-end solutions incorporate voice command systems, gesture control, head movement recognition, or brain-computer interfaces (BCI). These systems significantly enhance usability for users with limited mobility, but they come with technical, financial, and operational drawbacks:

- High Cost: Most advanced systems require specialized hardware, sensors, or embedded modules, leading to high development and deployment costs.
- Complexity: The interfaces are often not intuitive, and the calibration required for gestures, speech, or brain signals can lead to frustration.
- Environmental Sensitivity: Voice systems fail in noisy surroundings, gesture recognition fails in poor lighting, and BCI systems suffer from lag and poor signal fidelity.

In the academic space, several projects have been proposed using platforms like Raspberry Pi, Bluetooth modules, Arduino, and mobile apps. While these projects demonstrate feasibility, most of them remain non-robust prototypes or lack real- world implementation and testing. They also often fail to consider ease of assembly, power optimization, or modular upgrades.

Therefore, although existing systems do provide important stepping stones, they lack a balanced approach combining simplicity, affordability, and functionality — which is the core gap this project intends to address.

3.2 Proposed System

The proposed system is a Bluetooth-controlled smart wheelchair, designed to offer an affordable, reliable, and easy-to-use solution for individuals with physical disabilities. It uses readily available components such as the Arduino Uno, HC-05 Bluetooth module, BO motors, L293D motor driver IC, and a custom-built mobile app for control. The design is tailored to minimize complexity while retaining high utility.

The core functionality allows the user to control the wheelchair using a smartphone application with four directional buttons — Forward, Backward, Left, and Right. These inputs are transmitted via Bluetooth to the Arduino, which processes the signals and actuates the motors accordingly.

Key characteristics of the proposed system include:

- Affordability: Uses inexpensive modules (Arduino Uno, HC-05, L293D) costing under ₹1000 in total.
- Ease of Control: Simple app interface eliminates the need for technical training.- Wireless Operation: No physical interface is needed, allowing operation from any compatible smartphone.
- Modularity: Future additions like obstacle detection (IR/ultrasonic sensors), GPS tracking, or voice control can be added easily.

This system eliminates the dependency on caregivers for movement, providing increased freedom and autonomy for users. It is also highly suitable for DIY implementation, making it a great fit for developing countries and rural areas where medical infrastructure may be lacking.

In addition to the base features, the proposed system will include: Emergency
Stop Feature (via app button or kill switch)
Battery status indicator (optional feature using voltage sensor module)
Compact form factor with a strong chassis

The goal is to create a functional, field-ready prototype that demonstrates the feasibility of such systems for mass production, customization, and further development.

3.3 Requirements Analysis

A complete requirements analysis ensures that the proposed system aligns with functional, non-functional, and user expectations. It also helps identify component-level constraints and software requirements for robust development.

- Functional Requirements:

- Bluetooth Connectivity: The wheelchair must pair with the smartphone and respond to directional inputs.
- Directional Movement: The system must support forward, backward, left, and right motions.
- Motor Control Logic: The Arduino must properly switch signals through the L293D to rotate the BO motors in desired directions.
- User Interface: The Android application must be simple, with clearly labeled buttons and minimal setup.
- Power Supply: A reliable battery pack must be used to power both motors and logic components.
- Emergency Stop: The user should be able to stop the wheelchair in case of any failure or miscommunication.

- Non-Functional Requirements:

- Usability: The app should be intuitive even for tech-inexperienced users.
- Reliability: The system should function consistently without data loss or signal interference.
- Latency: The communication delay between command and execution should be under 500ms.
- Safety: Motors should stop if the app loses connection or no input is received.
- Battery Life: The system should operate for at least 2 hours on a full charge.
- Durability: The frame and motor attachments must withstand real-world use, including minor bumps.

- Hardware Requirements:
  - Arduino Uno R3
  - HC-05 Bluetooth module
  - BO DC motors (x2)
  - L293D motor driver IC
  - Battery Pack (12V recommended)
  - Chassis, wheels, wires, connectors

- Software Requirements:
  - Arduino IDE for programming microcontroller
  - MIT App Inventor / Android Studio for mobile app
  - Serial Monitor (for debugging during development)

This analysis confirms that the system's requirements are clear, achievable, and testable within a constrained timeline and budget.

CHAPTER 4
HARDWARE DESCRIPTION

4.1 ARDUINO

Microcontroller board, Arduino UNO with ATmega328P. It contains 6 analogue inputs, a 16 MHz ceramic resonator, 14 digital I/O pins (six of which can be used as PWM outputs), a USB port, a power jack, an ICSP header, and a reset button. It has every part required to support the microcontroller, so all you need to do to get started is plug it into a computer through USB, an AC-to-DC adapter, or a battery. Without worrying about making a mistake, you can experiment with your UNO; in the worst case, you can start over by replacing the chip for a few dollars.

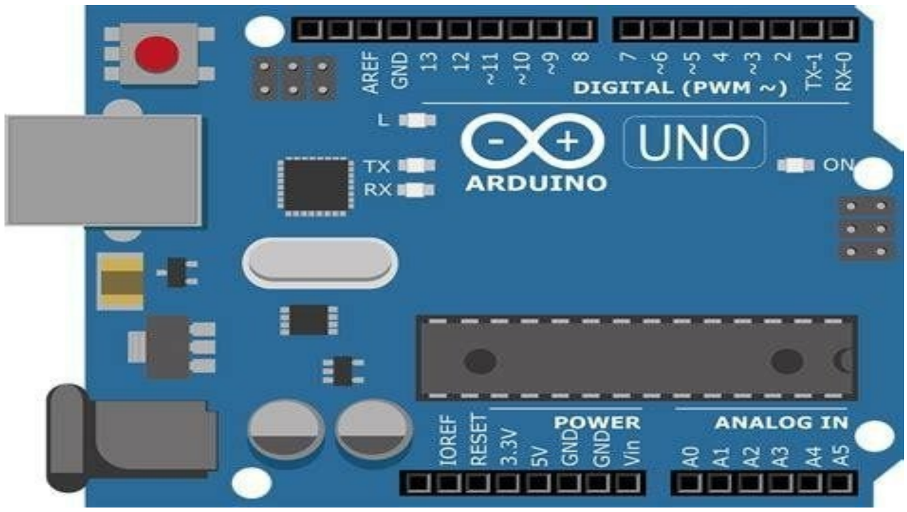

Fig 2.1 Arduino UNO

Arduino pins, inputs and outputs

2. Analog Reference pin
3. Digital Ground pin

• Digital Pins 2-13 Reset Button - S1 Analog in Pins 0-5 Power and •
Ground Pins External Power Supply In (9-12VDC) USB (used for uploading sketches to the board and for serial communication between the board and the computer; can be used to power the board)

We used the following materials for our project.

1. The Arduino Uno.

2. A USB cable for programming.

3. External power supply or 9V battery (for stand-alone operation). Rather than being powered by a USB connection to a computer, the board is powered by a battery.

4. External circuits are built on a breadboard, and connections are made with solid wire.

2. HC-05 Bluetooth Module:

The HC-05 is a cost-effective, serial-based Bluetooth module that enables wireless communication between the mobile application and the Arduino Uno. Configured in slave mode, it waits for paired devices to send specific character-based commands such as 'F', 'B', 'L', or 'R', which correspond to direction controls. Its reliable range of up to 10 meters and straightforward AT command-based configuration make it well-suited for short-range remote control systems like this wheelchair.

The HC-05 is a widely used Bluetooth Serial Port Protocol (SPP) module that facilitates wireless communication over Bluetooth version 2.0+EDR (Enhanced Data Rate). It supports both Master and Slave configurations, but in this project, it is used in Slave mode to receive control commands from a smartphone app. The module operates at 3.3V logic but is powered using the Arduino’s 5V supply (thanks to onboard voltage regulators). It communicates with the Arduino via UART protocol (TX and RX pins), using a baud rate of 9600 bps. When connected to a Bluetooth-enabled device, the module receives directional characters ('F', 'B', etc.) and sends them to the Arduino via serial communication. Its long range (up to 10 meters in ideal conditions), low power consumption, and minimal setup make it highly effective for mobile control in real-time embedded systems.

Fig 2.2 HC-05 Bluetooth Module

4.2 L293D Motor Driver Shield:

The L293D motor driver shield is a dual H-Bridge IC that allows the Arduino to drive two DC motors simultaneously, controlling both speed and direction. Each H-Bridge can provide bidirectional current of up to 600 mA (peak 1.2 A) per channel, which is sufficient for the small BO motors used in the project. The shield plugs directly onto the Arduino Uno and receives digital control signals from it (e.g., pin combinations like IN1=HIGH, IN2=LOW for forward rotation). It isolates the low-current control logic of the Arduino from the higher current requirements of the motors and includes features like flyback diodes to protect against voltage spikes. The motor shield simplifies wiring and provides screw terminals or male headers for easy motor connections. It is essential for enabling the differential drive system of the wheelchair
— enabling forward, reverse, left, and right movements.

Fig 2.3 L293D MOTOR DRIVER SHEILD

4.3. BO Motors with Wheels (DC Gear Motors):

The BO motors used in this project are compact DC gear motors that operate at 6– 12V, commonly used in robotics for lightweight mobility solutions. They come with a built-in plastic gearhead that reduces RPM (typically 150–300 RPM) while increasing torque, allowing smooth and controlled movement of the wheelchair frame. These motors are paired with rubber wheels to allow traction on flat surfaces. In a two- motor differential drive configuration, coordinated control over the left and right motors (via the L293D shield) allows for precise navigation, including turning and reversing. Their lightweight construction, low cost, and ease of interfacing with standard motor drivers make them highly suited for mobile prototype platforms like the Smart Wheelchair.

4.4. 18650 Battery Pack with Holder:

The 18650 lithium-ion cells are cylindrical rechargeable batteries, typically rated at 3.7V and 2200–3000 mAh per cell. In this project, multiple 18650 cells are used in series and parallel combinations to provide sufficient voltage (usually 7.4V or 11.1V) and current to power the DC motors and Arduino circuit. The battery holder provides secure housing and easy replacement of batteries while ensuring safe electrical contact. A voltage regulator may be used if necessary to step down the voltage for powering the Arduino and HC- 05 module. The batteries offer a high energy density, lightweight form factor, and long cycle life, making them suitable for portable robotics systems like the Smart Wheelchair.

4.5. Jumper Wires & MDF Base Board

Jumper wires are essential for establishing electrical connections between various components such as the Arduino pins, motor driver inputs, and Bluetooth module's TX/RX lines. Both male-to-male and male-to-female types are used depending on the connection points. They provide flexibility in layout and enable fast prototyping and debugging. The base of the wheelchair is built using a sheet of MDF (Medium Density Fiberboard), which is an engineered wood product that offers good strength- to-weight ratio and ease of cutting and drilling. It serves as the mechanical support structure for mounting the wheels, motors, battery holder, and Arduino system. The rigidity and low cost of MDF make it suitable for a prototype chassis while keeping the overall system weight manageable.

Fig 2.4 Jumper Wires

CHAPTER 5

METHODOLOGY

5.1 DESCRIPTION

The methodology for the Smart Wheelchair is designed to provide a safe, user- friendly, and efficient mobility solution for individuals with limited movement. The primary aim of our project is to ensure that the wheelchair not only provides movement control through an app but also incorporates health monitoring and safety features. The steps involved in the successful implementation of this project are as follows:

1. Bluetooth-based Movement Control: The wheelchair is controlled through a mobile app that communicates via Bluetooth. Simple directional commands (Forward, Backward, Left, Right) are sent from the app to the wheelchair, which is then translated into motor movement via the Arduino Uno microcontroller. This offers easy operation for people with limited hand mobility.
2. Health Monitoring System: The LM35 temperature sensor is used to monitor the user's body temperature. If the temperature exceeds a preset threshold, an immediate alert is sent to the caregiver or family member via the Blynk app. Additionally, pulse oximeter and heart rate sensors are integrated to measure blood oxygen levels and heart rate. If any of these parameters go beyond the acceptable range, an alert is automatically sent to the caregiver or a nearby healthcare facility.
3. Obstacle Detection: The wheelchair uses an ultrasonic sensor to detect obstacles in its path. If the wheelchair detects an obstruction while moving, it will automatically stop to avoid collisions, ensuring the user's safety during navigation.
4. Emergency Alerts: In the case of an emergency, such as the wheelchair user being stuck in a fire, the system incorporates a flame sensor. If flames are detected, the system immediately sends an alert to the caregiver or the nearest fire station, providing quick response capabilities in emergencies.

Room Appliance Control: The wheelchair also integrates with the RF Transmitter and RF Receiver modules to allow the user to control room appliances such as lights and fans via voice commands. The voice recognition module receives the user’s commands and forwards them to the microcontroller, which then relays the commands through the RF module to control the appliances.

Motor Control: The wheelchair's movement is managed by the L298N motor driver, which controls the speed and direction of the DC motors based on the user's commands. The movement is controlled as follows:

- Forward Movement: Both motors rotate in the forward direction.
- Backward Movement: Both motors rotate in the backward direction.
- Left Movement: The right motor rotates forward, and the left motor rotates backward.
- Right Movement: The left motor rotates forward, and the right motor rotates forward.

The Pulse Width Modulation (PWM) technique is used by the L298N motor driver to regulate the speed of the motors. By varying the duty cycle, the average voltage applied to the motors can be adjusted, controlling their speed for smoother operation.

5.2 Flow Descryption :

Flow Description: Bluetooth Control System for Smart Wheelchair Step 1:

Start the system by powering on the Arduino-based control unit.

Step 2: Establish a Bluetooth connection between the HC-05 module and the smartphone app.

Step 3: Continuously listen for valid character commands sent from the smartphone (e.g., ‘F’, ‘B’, ‘L’, ‘R’, ‘S’).

Step 4: If the command received is valid, it is passed to the Arduino for processing. Else, the system continues to wait for a correct input.

Step 5: Check the command received:

- If command = 'F' → Move Forward
- If command = 'B' → Move Backward • If command = 'L' → Turn Left
- If command = 'R' → Turn Right • If command = 'S' → Stop

Step 6: If the ultrasonic sensor detects an obstacle ahead during motion, immediately send stop signal to the motors to prevent collision.

Step 7: If the command is “Turn On”, activate the connected room appliance via relay module (optional feature).

Step 8: If the command is “Turn Off”, deactivate the connected room appliance via relay module (optional feature).

Step 9: Repeat Steps 3 to 8 in an infinite loop to continuously respond to user input in real time. Step 10: The system remains in active listening mode until powered off manually.

CHAPTER 6
SOFTWARE DESCRIPTION

6.1 ARDUINO IDE

The Arduino IDE, an open-source programme created by Arduino.cc, is a recognised Arduino software that makes code compilation simple enough that even a non-technical layperson may start learning. It operates on the Java Platform, which has built-in debugging, editing, and compiling capabilities, and is compatible with all operating systems, including MAC, Windows, and Linux.

The Arduino Uno, Arduino Mega, Arduino Leonardo, Arduino Micro, and many other modules are available. Each has a microcontroller on the board that accepts information in the form of code and has been programmed. Eventually, the primary code, also referred to as a sketch, created on the IDE platform will produce a Hex File, which is then transferred and uploaded into the board's controller.

The IDE environment is primarily comprised of two basic components: the Editor and the Compiler, the previous head of which is used for writing the required code while the latter is responsible for compiling and uploading the code into the appropriate Arduino Module. While not connected to the internet, the Arduino Software (IDE) makes it simple to write code and upload it to the board. It is recommended for users with slow or no internet access. This software works with any Arduino board.

The Arduino IDE is currently available in two versions: IDE 1.x.x and IDE 2.x.

The IDE 2.x is a significant improvement over the IDE 1.x.x in terms of speed and power. It includes advanced coding and debugging features, as well as a more modern editor and a more responsive interface.

2.x): The steps below will walk you through using the offline IDE (IDE 1.x.x or ID

1. Get the Arduino Software Development Kit (IDE) and install it:

- Arduino IDE 1.x.x (Mac OS, Windows, Linux, Portable IDE for Windows and Linux, chromeOS).
- Arduino IDE version 2.x

2. Connect your Arduino board to your device using the Arduino IDE version 2.x.

3. Open Arduino IDE (IDE).

The Arduino (IDE)Integrated Development Environment connects to the Arduino boards and allows you to communicate with them. Arduino Software (IDE) creates sketches which are programmes. These drawings are made in a text editor and saved as files with the extension. ino.

Making use of the offline IDE 1.x.x:

The editor is divided into four major sections:

1. A toolbar containing common function buttons as well as a series of menus. The toolbar buttons can be used to create sketches, open sketches, save sketches, verify programmes and upload programmes and launch the serial monitor.

2. The message area displays errors and provides feedback, during saving and exporting

3. The text editor in which your code will be written.

4. The text console displays error messages and other data from the Arduino Software (IDE).

In the bottom right-hand corner of the window, the configured board and serial port are shown

Let's try to make one of your boards blink once you've got everything set up!

5. Connect the computer to your Arduino board.

6. Select the appropriate core and board. To do so, navigate to Tools > Board > Arduino AVR Boards > Board. Make sure the board you're using is selected. If your board is missing, go to Tools > Board > Boards Manager and add it.

7. Now, select the port to ensure the computer recognises your board. This is simple to do by going to Tools > Port and selecting your board from the list.

8. Here is an example: select File > Examples > 01. Fundamentals > Blink.

9. By clicking the arrow in the top left corner it will be uploaded to your board. This process takes a few seconds, and the board must not be disconnected during this time. The message "Done uploading" will appear in the bottom output area, If the upload is successful

10. When the upload is finished, the yellow LED on your board with a L next to it should start blinking. By changing the delay number in the parenthesis to 100 and re-uploading the Blink sketch, you can change the blinking speed. The LED should now blink much more quickly.

Making use of the offline IDE 2.x:

The editor is divided into four major sections:

1. A toolbar containing common function buttons as well as a series of menus. You can verify programmes and upload programmes, create sketches, open sketches, and save sketches, select your board and port, and launch the serial monitor using the toolbar buttons.

2. Frequently used tools contained by sidebar. It allows you to quickly access board managers, libraries, board debugging, and a search and replacement tool.

3. The code will be written in text editor.

4. Console controls allow you to control the console's output.

5. The text output from the Arduino Software (IDE) will be displayed by text console, such as error messages and other data. Serial port and configured board are shown in the bottom right-hand corner of the window. Once you've completed your setup, let's attempt to make the board blink!

1. Connect your computer to your Arduino board.

2. Choose the correct board and port. This is done through the toolbar. Make sure the board you're using is selected. If you can't find your board, you can add it using the board manager in the sidebar.

3. Here's an example: go to File > Examples > 01. Basics > Blink.

4. Click the arrow in the top left corner to upload it to your message board. The board cannot be disconnected during this brief process, which takes place. The "Done uploading" message will show up in the bottom output area if the upload was successful

5. When the upload is completed, the yellow LED on your board with the letter L

6. next to it should start blinking. By setting the parenthesis' delay value to 100 and uploading the Blink sketch again, you can change the blinking speed. The LED should now blink much more quickly.

6.2. Bluetooth App for Wheelchair Control

The mobile app for controlling the smart wheelchair communicates with the microcontroller via Bluetooth, offering an intuitive and user-friendly control system without the need for complex sensors or voice commands

This app enables the user to operate the wheelchair remotely through simple button presses, ensuring ease of use for individuals with physical disabilities.

6.3. Mobile App Functionality

The mobile app is designed to be Android-only with a minimalist interface, featuring 4 buttons for controlling the wheelchair's movements: Forward, Backward, Left, and Right. Each button sends a specific Bluetooth command to the wheelchair, triggering the Arduino to control the motors accordingly.

Blynk is used as the platform to create the app. It allows us to quickly design the app interface without the need for any coding. Through Blynk, we can map the mobile app's controls directly to the Arduino’s Bluetooth module (HC-05), which then translates the app's commands into motor movements.

In this project, we use the Blynk app to control the wheelchair’s movement based on the Forward, Backward, Left, and Right commands. These commands are sent wirelessly via Bluetooth to the Arduino, which then sends instructions to the L293D motor driver to control the DC motors and move the wheelchair in the desired direction.

6.4. Web Interface for Monitoring and Control

To provide a comprehensive overview and manage the wheelchair remotely, we can also design a web-based dashboard to stream the wheelchair’s status and control the movement commands. Through this interface, users can monitor the wheelchair's operational status in real time and control it using the same directional commands provided in the mobile app.

- Create a template for the project and name it as desired.
- Select the data stream for movement commands and assign them to virtual pins.
- Set up virtual pins on the Blynk platform to map to the Forward, Backward, Left, and Right movement controls.

For the wheelchair movement, you will set the following virtual pins on the Blynk platform:

- Virtual Pin > Name: Forward > Pin-V0 > Data Type: Integer > Min-0 > Max-1
- Virtual Pin > Name: Backward > Pin-V1 > Data Type: Integer > Min-0 > Max-1 •

Virtual Pin > Name: Left > Pin-V2 > Data Type: Integer > Min-0 > Max-1

- Virtual Pin > Name: Right > Pin-V3 > Data Type: Integer > Min-0 > Max-1

6.5. Mobile App Dashboard

Once the Blynk app is downloaded and installed on your smartphone, sign in with your account, select the template you created, and you'll see the control buttons on your mobile screen. These buttons will send movement commands (forward, backward, left, right) to the wheelchair.

By pressing the relevant button, the app will communicate the command to the Arduino through Bluetooth, triggering the motor movements via the L293D motor driver.

5 .Testing and Final Configuration Once
the app and dashboard are set up:

1. Upload the code to the Arduino.
2. Test the wheelchair's functionality, ensuring that all directional controls work smoothly.
3. If needed, you can add an event notification to alert the user when a movement command is successfully executed. The Bluetooth communication will be tested to ensure a stable connection within a range of up to 10 meters, allowing full wireless control of the wheelchair from a distance.

CHAPTER 7
RESULT

7.1.RESULT

This is the prototype of our project “Smart wheel chair for physically disabled”.

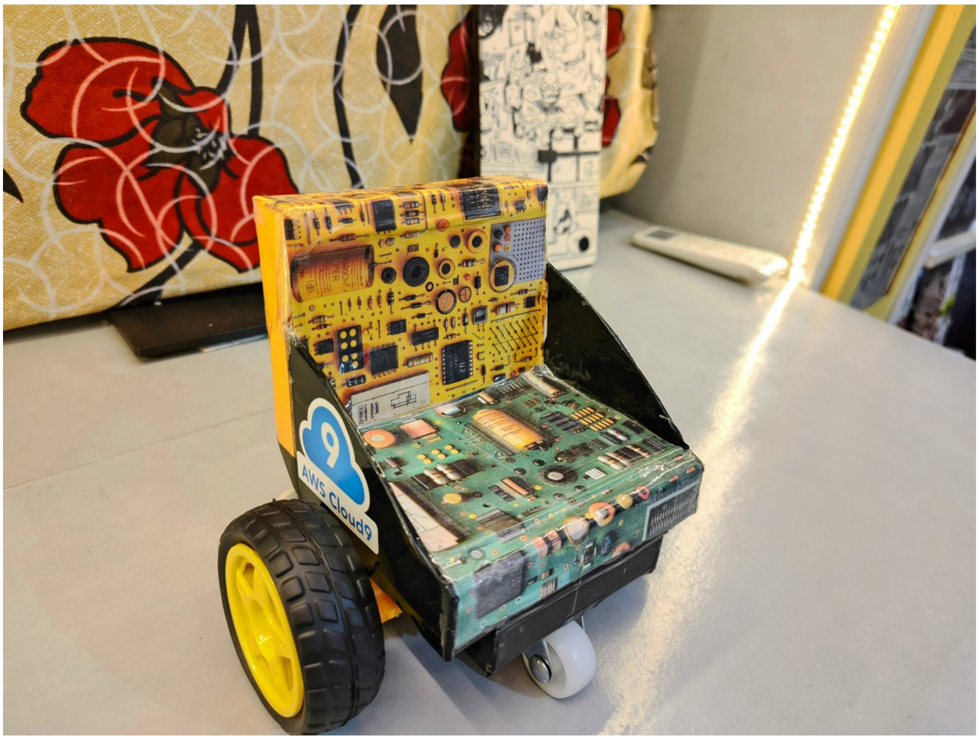

Fig 5.1 Smart Wheelchair Prototype

The movement control of the wheelchair is achieved using simple commands sent via Bluetooth from a mobile app, where the wheelchair's motors are controlled based on the app's input. The left and right motors of the wheelchair are activated in different directions to move the wheelchair forward, backward, left, or right, or to stop.

COMMAND	LEFT MOTOR	RIGHT MOTOR
Forward	clockwise	clockwise
Backward	Anti clockwise	Anti clockwise
Right	clockwise	Anti clockwise
Left	Anti	clockwise
Stop	clockwise	No

Table 1: Commands used and directions for each motor either on clockwise or anticlockwise

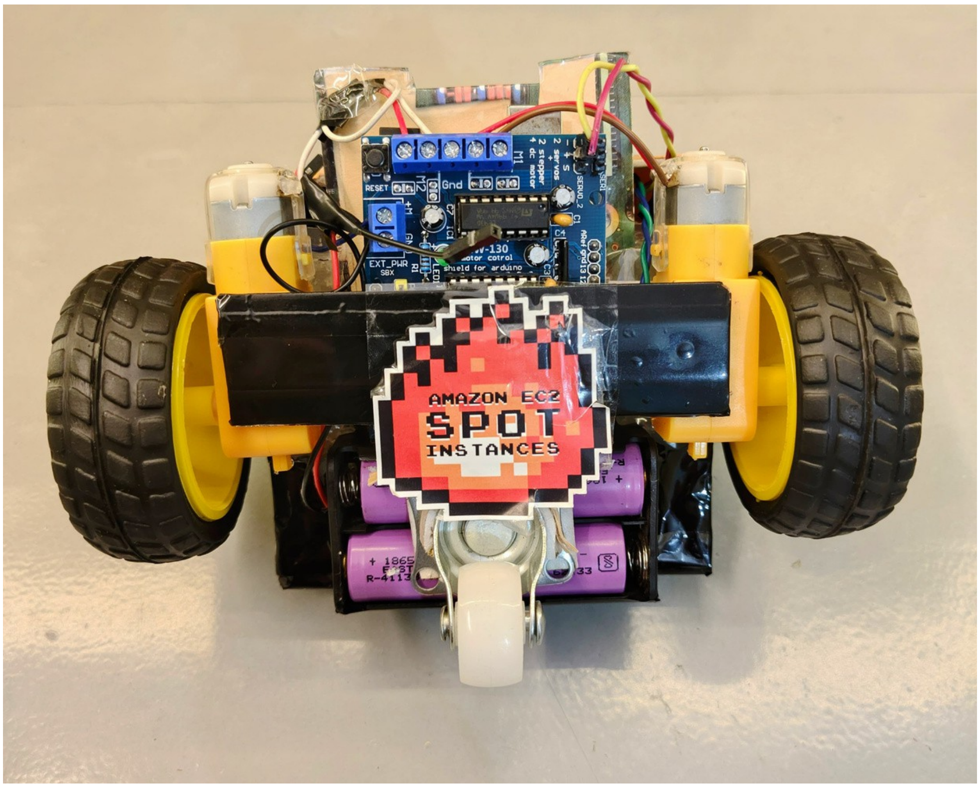

Fig 5.2 Smart Wheelchair Prototype (UNDERSIDE VIEW)

40

CHAPTER 8
CONCLUSION AND FUTURESCOPE

8.1 CONCLUSION

The Smart Wheelchair has been designed as a user-friendly, Bluetooth-controlled mobility solution. It offers intuitive movement controls through a mobile app, making it suitable for individuals with physical disabilities. The wheelchair operates using simple forward, backward, left, and right commands that are sent from the app via Bluetooth to the Arduino, which in turn controls the motors to maneuver the wheelchair. This design is lightweight, affordable, and effective in providing greater independence to users.

The wheelchair's movement is reliable, and the system ensures that the user can easily control the direction of movement through the app without requiring additional technical knowledge. This project serves as a stepping stone toward improving the quality of life for individuals with limited mobility.

8.2. FUTURE SCOPE

In the future, additional features can be incorporated to further enhance the wheelchair's functionality and user experience. For example, adding an auto-rotation feature could allow the wheelchair to turn a specific angle (e.g., 45 degrees left or 90 degrees right) with a simple app command, giving users more precise control over their movement.

Moreover, the wheelchair design can be upgraded to function as a chair-cum-bed for added comfort and versatility. The inclusion of brakes could also be beneficial, especially when the user is on a slope, allowing them to stop the wheelchair with ease and ensuring safety.

As we continue to focus on providing independence to individuals with physical disabilities, future developments could incorporate features like obstacle detection and automated path correction, allowing the wheelchair to navigate around objects in its path, thus offering a safer experience for the user. Additionally, the system could be expanded to include a mobile application that allows caregivers or family members to monitor the wheelchair's status remotely, improving safety and convenience for both the user and their support system.

The goal is to keep the wheelchair affordable, ensuring that it remains accessible to those who need it most, while also enhancing its functionality to meet the diverse needs of individuals with mobility impairments.

REFERENCES

REFERENCES

- Caltenco, H. A., Rantakokko, M., & Brostedt, E. (2019). Assistive Technology for Mobility and Orientation: Challenges and Opportunities. Journal of Rehabilitation and Assistive Technologies Engineering, 6, 2055668319844652. https://doi.org/10.1177/2055668319844652
- Singh, R., & Sahani, A. K. (2016). Design and Development of Smart Wheelchair Using Arduino Microcontroller. International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering (IJAREEIE), 5(6), 5045– 5051.
- Harshitha, A., & Anusha, P. (2017). Voice and Touch Screen Based Direction and Speed Control of Wheelchair for Physically Challenged Using Arduino. International Journal of Science and Research (IJSR), 6(4), 2319–7064.
- Arduino. (n.d.). Arduino UNO Rev3. Retrieved from https://store.arduino.cc/products/arduino-uno-rev3
- HC-05 Bluetooth Module Datasheet. (n.d.). Retrieved from https://components101.com/wireless/hc-05-bluetooth-module
- L293D Motor Driver Shield Datasheet. (n.d.). Retrieved from https://www.electronicwings.com/nodemcu/l293d-motor-driver-shield
- Kumar, A., & Rajeev, R. (2018). Android Application Controlled Bluetooth Wheelchair Using Arduino. International Journal of Engineering and Techniques, 4(2), 174–178.
- Bhatlawande, S. R., Mahadevappa, M., & Mukherjee, J. (2013). Smart Wheelchair: A Review. In Proceedings of the IEEE International Conference on Man and Machine Interfacing (pp. 1–4). https://doi.org/10.1109/MAMI.2013.6557657
- Open Source Hardware Association. (n.d.). Arduino IDE Documentation. Retrieved from https://www.arduino.cc/en/software
- Jayakumar, S., & Ganesh, M. (2020). Bluetooth Based Motion Controlled Wheelchair. International Journal of Engineering Research & Technology (IJERT), 9(05), 120–123.
- Khusboo, A., & Mehta, S. (2022). IoT Based Smart Wheelchair for Disabled People Using Arduino and Bluetooth Module. International Journal of Innovative Science and Research Technology (IJISRT), 7(3), 112–117.
- U.S. Department of Health and Human Services. (2021). Assistive Technology for People with Disabilities. Retrieved from https://acl.gov/programs/assistive-technology.

- All About Circuits. (n.d.). Introduction to Motor Drivers. Retrieved from https://www.allaboutcircuits.com/technical-articles/what-is-a-motor-driver/
- Arduino Project Hub. (n.d.). Bluetooth Controlled Robot Car Using Arduino. Retrieved from https://create.arduino.cc/projecthub
- Sharma, R., & Bhandari, M. (2020). Design and Implementation of Cost-Effective Smart Wheelchair. International Journal of Engineering Sciences & Research Technology (IJESRT), 9(6), 280–285.

APPENDIX(CODE)

CODE USED IN OF ARDUINO

```
#include <AFMotor.h>

// Only use M1 and M2
AF_DCMotor motor1(1, MOTOR12_1KHZ);
AF_DCMotor motor2(2, MOTOR12_1KHZ);

int val;
int Speeed = 255;

void setup() {
  Serial.begin(9600);  // Bluetooth module baud rate
}

void loop() {
  if (Serial.available() > 0) { val
    = Serial.read();

    Stop(); // Always start with motors stopped if

    (val == 'F') {
      forward();
    }

    if (val == 'B')
      { back();
    }

    if (val == 'L')
      { turnLeft();
    }

    if (val == 'R')
      { turnRight();
    }
```

```
 if (val == 'T') { Stop();
 }
 }
}
void forward() { motor1.setSpeed(Speeed);
 motor1.run(FORWARD);
 motor2.setSpeed(Speeed);
 motor2.run(FORWARD);
}
void back()
 { motor1.setSpeed(Speeed);
 motor1.run(BACKWARD);
 motor2.setSpeed(Speeed);
 motor2.run(BACKWARD);
}
void turnLeft() {
 motor1.setSpeed(0); // Left motor stopped
 motor1.run(RELEASE);
 motor2.setSpeed(Speeed); // Right motor forward
 motor2.run(FORWARD);
}
void turnRight() {
 motor1.setSpeed(Speeed); // Left motor forward
 motor1.run(FORWARD);
 motor2.setSpeed(0); // Right motor stopped
 motor2.run(RELEASE);
}
void Stop()
 { motor1.setSpeed(0);
 motor1.run(RELEASE);
 motor2.setSpeed(0);
 motor2.run(RELEASE);
}
```

CONFERENCE