FEBRUARY 10, 2026

Intelligent Autonomous Wheelchair Robotics: Sensor-Based Navigation and Safety Control

School of Mathematics, Computer Science & Engineering

Liverpool Hope University

Supervisor: Emanuele Lindo Secco

SAFEER AHMED

MSC ROBOTICS ENGINEERING

Abstract

This dissertation presents the design, simulation, and experimental evaluation of a low-cost smart wheelchair system aimed at improving safety and usability in indoor environments. Instead of full autonomous navigation, the system adopts a shared autonomy approach, where the user remains in control while automatic safety intervention is applied only when unsafe conditions are detected. A simulation-first development methodology is used, with the system initially modelled and tested in the Webots robotics simulator using a corridor-based indoor environment. Ultrasonic sensors are employed for obstacle detection, while inertial measurement data is used for safety monitoring, including emergency braking and tilt detection. Both manual and autonomous operating modes are supported, with a safety layer that remains active at all times. Following simulation validation, a physical hardware prototype is developed using an ESP32 microcontroller, low-cost sensors, and a mobile application interface implemented through the Blynk platform. Hardware testing is conducted indoors without a human user to evaluate manual control, obstacle avoidance, emergency braking, and alert generation under real-world conditions. The results demonstrate that the system can reliably support basic navigation assistance, detect unsafe situations, and prioritise safety using affordable embedded components. Overall, the project shows that safety-focused shared autonomy can be achieved without complex autonomy or expensive hardware, providing a practical foundation for future low-cost assistive mobility systems.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

Abbreviation	**Full Form**
DC	Direct Current
ESP32	Espressif 32-bit Microcontroller
GPIO	General Purpose Input/Output
GPS	Global Positioning System
HC-SR04	Ultrasonic Distance Sensor Module
H-Bridge	Motor Driver Circuit for Direction Control
I^2C	Inter-Integrated Circuit
IMU	Inertial Measurement Unit
IoT	Internet of Things
LiDAR	Light Detection and Ranging
MPU6050	Motion Processing Unit (Accelerometer + Gyroscope)
PWM	Pulse Width Modulation
Wi-Fi	Wireless Fidelity

Chapter 1: INTRODUCTION

1.1 Background and Context

Mobility plays an important role in maintaining independence and quality of life for people with physical disabilities, long-term health conditions, or age-related impairments. Powered wheelchairs are commonly used to support mobility in these cases. However, most traditional powered wheelchairs depend fully on manual control, usually through a joystick or switch-based interface. This type of control assumes that the user can maintain steady motor skills, fast reaction time, and constant attention, which may not always be possible for users experiencing fatigue, muscle weakness, or neurological conditions.

Indoor environments such as homes, hospitals, and care facilities present additional challenges for wheelchair users. These spaces often include narrow corridors, doorways, and obstacles such as furniture or medical equipment. In such environments, even small control errors or delayed reactions can result in collisions or unsafe movements. Since powered wheelchairs operate very close to the user, these incidents can increase the risk of injury or loss of confidence. As a result, there has been growing interest in smart wheelchair systems that assist the user instead of fully replacing manual control.

Smart wheelchairs typically combine sensors, embedded controllers, and control algorithms to improve safety and ease of use. Common features include obstacle detection, assisted steering, and shared autonomy, where the system intervenes only when unsafe conditions are detected. Recent improvements in low-cost sensors and microcontrollers have made it possible to develop such systems at a lower cost, making them more accessible for research and practical use.

Despite these advances, many existing smart wheelchair solutions are still limited. Some systems rely on complex sensors or high-performance computing platforms, which increases cost and power consumption. Other systems focus mainly on navigation accuracy while giving less attention to safety monitoring, such as detecting tilt or sudden instability. In addition, while simulation tools are widely used during development, simulation results do not always match real-world hardware behaviour due to sensor noise and physical constraints.

In this context, this project focuses on the design and evaluation of a low-cost smart wheelchair system that prioritises safety, practicality, and accessibility. By combining simulation-based development with embedded hardware concepts, the project explores how navigation assistance, safety monitoring, and assisted interaction can be integrated into a realistic and responsible assistive mobility system.

1.2 Problem Statement

Powered wheelchairs are widely used to support mobility, but their safe operation often depends on continuous and accurate manual control from the user. For users with reduced motor skills, limited attention, or physical fatigue, maintaining precise control can be difficult, especially in indoor environments such as narrow corridors, doorways, or crowded rooms. While research on smart wheelchairs has introduced autonomous and semi-autonomous solutions to reduce these risks, many existing systems rely on expensive hardware, high processing power, or complex sensor configurations. This makes them less suitable for low-cost and practical deployment in real-world settings. In addition, safety features such as obstacle avoidance, tilt detection, and instability monitoring are often developed as separate functions rather than as part of a single integrated safety framework, which can reduce system reliability and consistency. Another important challenge is the gap between simulation-based testing and real-world behaviour, as simulation environments do not fully represent hardware limitations, sensor noise, surface irregularities, or power constraints. These challenges highlight the need for a low-cost smart wheelchair system that integrates assisted navigation and safety monitoring into a unified design, while being evaluated using both simulation-based development and real hardware considerations.

1.3 Motivation

The motivation for this project arises from the increasing need for assistive mobility systems that are both practical and affordable. During initial background research, it was observed that many smart wheelchair solutions focus on advanced autonomous navigation and high-performance hardware. While these systems demonstrate technical capability, they often increase cost, power consumption, and system complexity without significantly improving usability in everyday indoor environments. For many wheelchair users, reliable safety support and assisted control are more important than full autonomy. From an engineering perspective, user safety is a critical concern. Small control errors or delayed reactions can result in collisions, instability, or unsafe situations, particularly in narrow corridors and cluttered indoor spaces. However, many existing systems treat safety features such as obstacle avoidance, tilt detection, and instability monitoring as separate functions rather than as part of a unified safety framework. This separation can reduce overall system reliability and make real-world deployment more difficult. Another key challenge is the difference between simulation-based testing and real-world behaviour. Simulation environments often do not fully represent hardware limitations, sensor noise, surface irregularities, or power constraints. As a result, systems that perform well in simulation may not behave the same way when implemented on physical hardware. These limitations highlight the need for a low-cost smart wheelchair system that integrates assisted navigation and safety monitoring into a single, coherent design. Such a system should be evaluated using both simulation-based development and embedded hardware implementation to ensure practical and reliable operation in real environments.

1.4 Research Aim and Objectives

The main aim of this dissertation is to design, simulate, and evaluate a low-cost smart wheelchair system that improves user safety and ease of use through assisted autonomy. The system is intended to support basic indoor navigation, reduce the risk of collisions and instability, and provide additional safety support for the user. The design focuses on practical implementation using affordable embedded hardware, rather than complex or expensive autonomous systems. Instead of full autonomy, the wheelchair is designed to assist the user only when unsafe conditions are detected. This shared autonomy approach is chosen because it offers predictable behaviour, lower computational requirements, and greater user trust, which are important factors in assistive mobility applications.

To achieve this aim, several objectives are defined. First, a smart wheelchair model is developed in a robotics simulation environment to study navigation behaviour and safety responses under controlled and repeatable conditions. Second, obstacle detection and avoidance techniques using low-cost ultrasonic sensors are designed and tested for typical indoor environments. Third, a tilt and crash detection mechanism is implemented using inertial measurement data to identify unsafe operating situations. In addition, simple user interaction methods are explored to reduce reliance on continuous manual control. Finally, the proposed system is implemented on an embedded microcontroller platform, and its performance is evaluated through both simulation-based testing and real-world hardware experiments to assess practicality and safety.

1.5 Contributions of This Project

This project contributes to the field of assistive robotics by presenting the design and evaluation of a low-cost smart wheelchair system that places strong emphasis on safety and practical usability. Unlike many existing studies that focus on high levels of autonomy or expensive sensing and computing hardware, this work demonstrates how essential functions such as navigation assistance, obstacle avoidance, and safety monitoring can be integrated using affordable embedded components. The system follows a safety-first shared control approach, where intelligent assistance supports the user only when unsafe conditions are detected. This contribution highlights a realistic balance between autonomy, cost, and user trust, which is important for assistive mobility applications.

A further contribution of this work is the combined use of simulation-based development and physical hardware implementation. The project demonstrates how a robotics simulation environment can be used to model wheelchair movement, test safety behaviour, and refine control strategies before deployment on real embedded hardware. In addition, the inclusion of simple hands-free interaction through external voice-based commands provides a practical method for improving accessibility without increasing onboard computational requirements. Together, these contributions form a structured case study that can support future research, prototyping, and educational work focused on low-cost smart wheelchairs and safety-oriented assistive technologies.

Chapter 2: Literature Review

2.1 Scope, approach and paper selection

This literature review focuses on research that supports the development of a smart wheelchair system with three core functions: safe navigation assistance, detection of unsafe movement such as tilt or fall risk, and hands-free user interaction. The scope is deliberately limited to methods that have been evaluated through simulation, implemented on real wheelchair platforms, or can be realistically adapted to a low-cost embedded prototype. This limitation ensures that the reviewed studies remain relevant to practical engineering design rather than purely theoretical solutions.

The literature search was conducted primarily using Google Scholar due to its wide coverage of robotics, assistive technology, and healthcare engineering research. Highly cited and well-established papers were first reviewed to build a strong theoretical foundation. The search was then extended using reference lists and forward citation tracking to identify more recent and relevant studies. Final versions of selected papers were accessed and verified through recognized academic publishers, including IEEE Xplore, Elsevier (ScienceDirect), Springer, and MDPI Sensors, to ensure reliability and academic quality.

Search keywords were selected to reflect both the project objectives and commonly used research terminology. These included combinations such as “smart wheelchair” with “obstacle avoidance” or “shared control”, “powered wheelchair” with “IMU” or “LiDAR”, and “voice-controlled wheelchair”. Additional searches related to system validation and simulation were also included, such as “robot simulation validation” and “Webots mobile robot simulation”. This broader approach was necessary because smart wheelchair research spans multiple research areas rather than a single discipline.

Paper selection prioritised peer-reviewed journal articles and recognised conference publications that included evaluation through simulation, experimental testing, or user-based studies. Recent work was preferred where sensing and control technologies are rapidly evolving. However, several influential earlier studies were retained, as they provide important foundations in shared control and smart wheelchair development (Bourhis & Agostini, 1998); (Wang & Ishimatsu, 2004); (Simpson, et al., 2005); (Carlson & Demiris, 2012). Based on this selection process, the reviewed literature is organised into thematic sections to support critical comparison and identification of research gaps addressed in this dissertation.

2.2 Intelligent wheelchairs and autonomy levels

A smart wheelchair is commonly defined as a powered wheelchair that uses sensors and control algorithms to support safer movement and assist the user during navigation. Research shows that intelligence in wheelchair systems is usually based on a combination of environmental sensing, decision-making support, and user interaction, rather than full autonomous control (Gupta, et al., 2025); (Sahoo & Choudhury, 2023). Unlike many mobile robots used in laboratories, wheelchairs operate very close to the user and within narrow indoor spaces. For this reason, predictable, smooth, and comfortable behaviour is essential. As a result, autonomy in wheelchair systems is often described as a range, starting from fully manual control and extending to fully autonomous navigation, with shared or assisted control positioned between these two extremes.

Many studies identify shared control as the most suitable level of autonomy for assistive mobility systems. Early work, such as the VAHM wheelchair, showed that fully autonomous behaviour can reduce user confidence if system actions are not clearly understood (Bourhis & Agostini, 1998). Later research demonstrated that higher levels of autonomy can be achieved using vision-based or predictive control methods; however, these systems often depend on high computational power or structured environments (Wang & Ishimatsu, 2004); (Kawaguchi, et al., 2023). This can limit their practicality for low-cost or embedded wheelchair platforms.

More recent studies therefore describe smart wheelchairs as assistive systems that intervene only when safety is at risk, rather than replacing the user's control entirely (Carlson & Demiris, 2012); (Erturk, et al., 2024). This assisted autonomy approach allows the user to remain in control while the system provides protection against collisions or unsafe movement. Such a strategy improves safety and user trust while remaining suitable for low-cost embedded hardware. These findings directly support the design approach adopted in this dissertation, where safety and predictable behaviour are prioritised over full environmental autonomy.

2.3 Obstacle detection and collision avoidance

Obstacle detection and collision avoidance are essential safety functions in smart wheelchair systems, as collisions can occur during forward motion, turning, reversing, or when moving through narrow indoor spaces such as corridors and doorways. Unlike mobile robots operating in controlled environments, wheelchairs move very close to users, furniture, and other people. This increases the risk and impact of sensing or control errors. As a result, reliable obstacle detection is a critical requirement for assistive mobility systems. The literature reports a wide range of sensing technologies for collision avoidance, including ultrasonic sensors, infrared sensors, cameras, depth sensors, and LiDAR. Each sensing method presents trade-offs between cost, detection accuracy, system complexity, and computational requirements. For this reason, many studies focus on achieving sufficient safety performance rather than complete environmental perception.

LiDAR-based systems are frequently discussed in recent research because they provide accurate and detailed distance measurements in indoor environments. For example, two-dimensional LiDAR scanning has been used to detect nearby obstacles and reduce collision risk for wheelchair users and surrounding pedestrians (Szaj, et al., 2021). Other studies combine LiDAR sensing with real-time navigation methods, such as the Dynamic Window Approach, to improve maneuverability in cluttered spaces (Erturk, et al., 2024). While these systems demonstrate strong performance, they typically require higher system cost, greater power consumption, and increased onboard processing capability. These requirements can limit their suitability for low-cost embedded wheelchair platforms intended for everyday use. As a result, LiDAR-based solutions may be impractical for affordable assistive devices where simplicity and energy efficiency are key design constraints.

To address these limitations, several studies explore the use of low-cost sensors or simplified sensing strategies. Research on multi-sensor approaches shows that combining inexpensive sensors can improve reliability by compensating for individual weaknesses such as noise, limited range, or blind spots (Pu, et al., 2018). Other work focuses on assisting the user by restricting unsafe commands rather than performing full path planning, allowing user intent to be preserved while improving safety (Pieniazek & Szaj, 2023). Earlier studies also demonstrate that even simple obstacle detection systems can provide meaningful safety benefits in rehabilitation and assistive settings (Simpson, et al., 2005). Overall, the literature suggests that effective collision avoidance in smart wheelchairs depends not only on sensor selection, but also on designing safety interventions that are predictable, affordable, and compatible with shared control. This supports the use of low-cost sensing combined with reactive safety strategies, as adopted in this dissertation.

2.4 Shared control and personalized assistance

Shared control is a key concept in smart wheelchair research because it addresses the balance between user independence and system safety. Instead of fully replacing user input, shared control systems are designed to support the user by intervening only when unsafe situations occur. Research consistently shows that this approach can reduce collisions and navigation errors while allowing the user to remain actively involved in control. This is important for user trust and long-term acceptance of assistive technology (Carlson & Demiris, 2012). In indoor environments where layouts may change and sensing is not always reliable, shared control is often considered more practical than full autonomy, as it does not depend on complete and accurate knowledge of the environment at all times.

Personalised assistance is closely related to shared control because wheelchair users vary widely in physical ability, driving confidence, and preferred control style. Studies on adaptive assistance indicate that fixed safety rules can feel too restrictive for some users, while insufficient assistance may fail to prevent unsafe behaviour (Vanhooydonck, et al., 2010). To address this challenge, several researchers propose systems that adjust the level of assistance based on driving behaviour or environmental context. Decision-theoretic approaches further suggest that user intent is often uncertain and that safer outcomes can be achieved by selecting conservative actions when sensor data or user input is unreliable (Ghorbel, 2018). These methods prioritise predictability and safety rather than aggressive optimisation of navigation performance.

From a system design perspective, many studies separate low-level motion control from higher-level reasoning and user interaction. This architectural separation allows rapid safety responses, such as collision prevention, to be handled locally, while more complex processing, such as user modelling or interaction logic, is managed at a higher level. Such designs are especially suitable for embedded wheelchair systems, where computational resources are limited and reliability is critical. Overall, the literature supports shared control as a practical and user-centred approach that balances safety, usability, and technical feasibility. These principles directly inform the control strategy adopted in this dissertation, where conservative assistance and clearly defined intervention boundaries are prioritised over complex adaptive behaviour.

2.5 Interaction methods, including voice and hands-free control

User interaction is a key element of smart wheelchair systems because it directly influences usability, comfort, and safety. Although joystick control remains the most common interface, many users experience difficulty using it due to limited hand movement, muscle weakness, or fatigue. For this reason, research has explored alternative interaction methods such as head movement, gestures, bio-signals, mobile applications, and voice commands. While these methods aim to improve accessibility and independence, the literature highlights that interaction errors can lead to unsafe behaviour. As a result, interaction design is considered a safety-critical aspect of smart wheelchair systems.

Several studies investigate hands-free control based on body movement. Head-controlled wheelchairs using inertial sensors show that basic directional commands can be achieved using relatively simple and low-cost hardware (Haddouna, et al., 2025). Other research combines IMU and EMG signals with machine learning techniques to recognise user intent from body motion, allowing more flexible control for users with severe physical limitations (Kundu, et al., 2018). Although these systems demonstrate promising performance, they often require careful calibration and can be affected by sensor noise, changes in posture, or user fatigue. These factors reduce robustness in everyday use and highlight the importance of additional safety mechanisms.

Voice-based interaction is attractive because it requires minimal physical effort and can feel natural for many users. Studies on low-cost voice-controlled wheelchairs show that basic movement commands can be generated using speech recognition (Sahoo & Choudhury, 2023). However, challenges such as misrecognition, delayed responses, and reduced reliability in noisy environments are frequently reported. More advanced systems reduce these risks by using voice input for high-level commands, such as destination selection, while lower-level control is handled autonomously (Benayed & Masmoudi, 2025). An alternative approach uses mobile applications as an interaction layer, where speech processing and user interfaces are handled externally, allowing the wheelchair controller to focus on motion control and safety (Jayasekera, et al., 2024). Across the literature, there is strong agreement that hands-free interaction must be supported by clear emergency stop functions, predictable system behaviour, and conservative safety strategies. These principles directly guide the interaction design adopted in this dissertation.

2.6 Safety monitoring beyond collisions: fall and posture detection

While obstacle avoidance reduces the risk of collisions, it does not address other important safety risks such as tipping, falling, or unsafe posture. For wheelchair users, these situations can lead to serious injury, especially when moving over uneven surfaces, slopes, or during transfers. For this reason, recent research has extended safety monitoring beyond environmental perception to include the stability and posture of the wheelchair and user. These studies recognise that wheelchair safety depends not only on the surroundings, but also on the physical state and balance of the system during motion.

Many fall detection methods use data from inertial measurement units to identify sudden changes in acceleration, orientation, or angular velocity. Reviews of existing approaches show that machine learning techniques can achieve high detection accuracy; however, they often require large training datasets and careful model tuning (Kaur, et al., 2025). This makes such methods difficult to implement on low-cost embedded platforms with limited processing power and memory. More advanced systems that combine multiple sensors and learning-based models can further improve detection performance, but they also increase hardware cost, energy consumption, and overall system complexity (Yan, et al., 2023). These limitations reduce their suitability for simple and affordable wheelchair prototypes.

Several wheelchair-focused studies demonstrate that effective safety monitoring can still be achieved using low-cost embedded hardware by adopting simpler detection strategies. For example, threshold-based and anomaly detection methods using inertial sensor data aim to identify unsafe conditions such as excessive tilt or sudden instability, rather than recognising detailed user activities (Yousuf & Kadri, 2023). Other research highlights posture monitoring as an important long-term factor affecting both safety and comfort for wheelchair users (Vermander, et al., 2025).

Overall, the literature suggests that safety monitoring systems should be conservative, easy to interpret, and straightforward to validate. This supports the use of lightweight, threshold-based detection methods in practical smart wheelchair systems, as adopted in this dissertation.

2.7 Simulation and validation for robotics prototypes

Simulation is widely used in robotics to support system design, development, and testing before physical implementation. In the context of smart wheelchairs, simulation is particularly valuable because early testing in real environments can be unsafe, time-consuming, and difficult to control. Simulation allows experiments to be performed in a safe and repeatable manner, making it easier to evaluate navigation behaviour, safety responses, and sensor interaction. It also enables rapid modification of system parameters without the risk of damaging hardware or exposing users to unsafe situations.

Webots is a well-established simulation platform for mobile robotics and provides realistic modelling of sensors, robot kinematics, and control behaviour (Michel, 2004). Reviews of robotic simulation tools indicate that modern simulators can offer meaningful insight into system performance during early development stages (Camargo, et al., 2021); (Kargar, et al., 2024). However, the literature also highlights clear limitations of simulation-based evaluation. Real-world factors such as sensor noise, mounting inaccuracies, surface irregularities, communication delays, and power constraints are often simplified or not fully represented in simulated environments. As a result, simulated performance may differ from behaviour observed on physical hardware.

For this reason, many researchers emphasise that simulation should be used alongside real-world testing rather than as a complete replacement. Simulation is most effective for developing and refining control logic, while physical experiments are required to validate reliability, safety, and practical feasibility. This combined approach helps identify gaps between idealised simulation results and actual hardware performance. In line with this guidance, simulation in this dissertation is used as a development and validation tool to support system design, while recognising the importance of embedded hardware testing to ensure realistic and reliable evaluation.

2.8 Summary and identified gap

The reviewed literature provides strong evidence that smart wheelchairs can improve user safety and independence when autonomy is designed as assistance rather than a full replacement of user control. Research on intelligent wheelchairs consistently shows that shared or assisted autonomy is more suitable for indoor assistive mobility than full autonomous navigation. Indoor environments are often unpredictable, cluttered, and socially dynamic, which makes reliable perception and localisation difficult to achieve at all times. As a result, many studies support control strategies that intervene selectively to prevent unsafe actions while preserving user intent and manual control.

Obstacle detection and collision avoidance are widely studied topics, with many systems using LiDAR sensors and real-time planning methods such as the Dynamic Window Approach. These systems demonstrate strong performance in controlled and cluttered environments, but they often rely on expensive sensors and high computational power. In contrast, research on low-cost and multi-sensor approaches shows that acceptable safety performance can be achieved using simpler sensing and conservative control strategies. However, these systems often focus mainly on collision avoidance and provide limited integration with wider safety monitoring or user interaction functions.

Studies on shared control and personalised assistance highlight that safety is not only a technical challenge but also a user-centred one. Systems that reduce collisions while maintaining predictable behaviour and low cognitive workload are more likely to be accepted by users. Although adaptive and decision-theoretic approaches show promising results, many require complex modelling or extensive data collection, which may not be practical for low-cost embedded platforms. This reinforces the need for designs that prioritise transparency, reliability, and simple safety rules over highly complex autonomy.

The literature also shows that wheelchair safety extends beyond collision avoidance. Fall detection and posture monitoring are increasingly recognised as important safety features, yet many existing solutions rely on machine learning models, large datasets, or multiple sensors. These requirements can limit deployment on affordable embedded systems. Similarly, hands-free interaction methods such as voice or gesture control improve accessibility but introduce risks related to misinterpretation and system reliability, highlighting the need for strong safety layers and clear override mechanisms.

Simulation is widely accepted as a valuable development tool for early-stage testing of navigation and sensing behaviour. However, existing studies consistently emphasise that simulation cannot fully represent real-world conditions. This highlights the importance of combining simulation-based development with physical testing to validate feasibility and safety. Based on these findings, a clear research gap exists for a smart wheelchair system that integrates navigation assistance, safety monitoring, and hands-free interaction within a low-cost embedded architecture. There is a need for a practical prototype that prioritises safety-first shared autonomy, uses affordable sensors, and is evaluated through a combined simulation and hardware-based approach. Addressing this gap defines the focus and design choices of this dissertation.

Chapter 3: System Design and Methodology

3.1 Overall System Architecture

The smart wheelchair system developed in this project is designed with a clear focus on safety, reliability, and user support. The system is not intended to fully replace the user’s control. Instead, it follows a shared autonomy concept, where the wheelchair assists the user only when unsafe conditions are detected. This design is well suited for assistive mobility applications, where predictable behaviour, user confidence, and trust are more important than full autonomous operation. By limiting autonomy to safety-related assistance, the system remains simple, transparent, and easier for users to understand and accept.

The overall system architecture is organised into three main parts: sensing, control, and actuation. The sensing layer collects information about both the environment and the wheelchair’s motion. Ultrasonic sensors are used to detect nearby obstacles at short range, which is suitable for indoor environments such as corridors, rooms, and narrow passages. An inertial measurement unit is used to monitor the orientation and movement of the wheelchair, allowing detection of unsafe conditions such as tilt or sudden instability. These sensors provide real-time data that supports fast safety decisions without requiring high computational power.

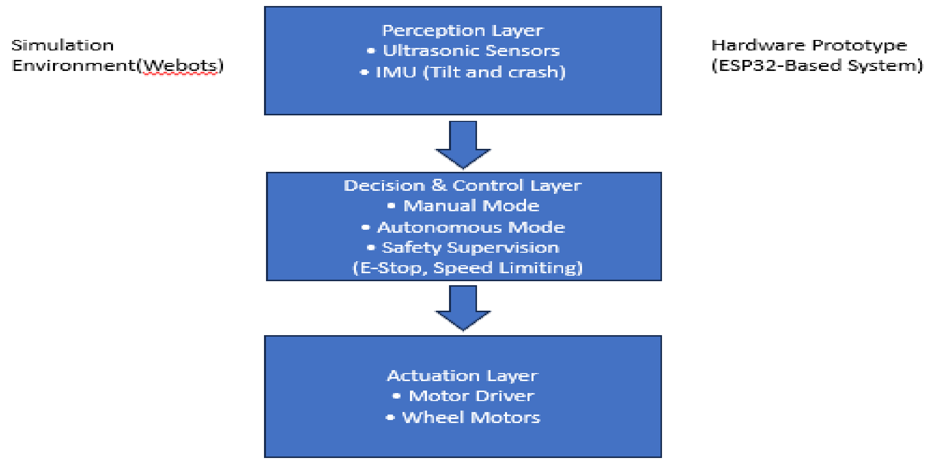

Figure 1: High-level system architecture of the intelligent smart wheelchair, illustrating the perception, decision and control, and actuation layers with continuous safety supervision across both simulation and hardware implementations.

The control layer processes sensor data and determines how the wheelchair should respond. Two operating modes are supported: manual mode and autonomous mode. In manual mode, the user directly controls the wheelchair, while in autonomous mode, the system assists with basic movement and obstacle avoidance. A key feature of the architecture is that a safety layer remains active in both modes. This layer can slow down or stop the wheelchair if a hazard is detected, regardless of user input. A simulation-first development approach is used to test and refine this architecture before hardware implementation. The physical prototype follows the same structure but with reduced complexity, ensuring that simulated behaviour can be realistically demonstrated on embedded hardware while maintaining a strong focus on safety and feasibility.

3.2 Control Modes and Safety Concept

The smart wheelchair system is designed to operate in two control modes: manual mode and autonomous mode. In manual mode, the user has direct control over the wheelchair movement, which provides a familiar and easy-to-understand driving experience. This mode is important for users who prefer full control or are operating in simple environments. In autonomous mode, the system provides assistance by handling basic navigation tasks, such as moving forward and avoiding nearby obstacles. These two modes are designed to address different user needs while keeping the system behaviour simple, predictable, and reliable.

A key design decision in this project is that the safety layer remains active at all times, independent of the selected control mode. The safety system continuously monitors data from the onboard sensors and checks for unsafe conditions. If a potential collision or dangerous situation is detected, the system can reduce the wheelchair speed or stop it completely. The emergency braking function can be activated either manually by the user or automatically by the system. Once engaged, the emergency brake immediately stops all motor movement and remains active until it is manually reset. This ensures that safety does not depend on fast user reactions or on the current operating mode.

Rather than aiming for full autonomous control, this project adopts a shared autonomy approach. Full autonomy in indoor environments can be unreliable due to changing layouts and sensor limitations, which may reduce user trust in assistive devices. Shared autonomy allows the user to stay in control while the system intervenes only, when necessary, mainly to prevent unsafe actions or collisions. This approach improves user confidence, increases system reliability, and aligns with established practices in assistive robotics research, as discussed in the literature review in Chapter 2.

3.3 Simulation Environment Design

The simulation environment for this project was developed using Webots, which is a well-established robotics simulation platform. Webots was selected because it supports realistic robot motion, physics-based collision handling, and common sensors such as ultrasonic distance sensors. These features allow safe testing of navigation and safety behaviour before moving to real hardware. Using simulation also reduces risk and helps refine control logic in a controlled and repeatable way, which is important for assistive mobility systems.

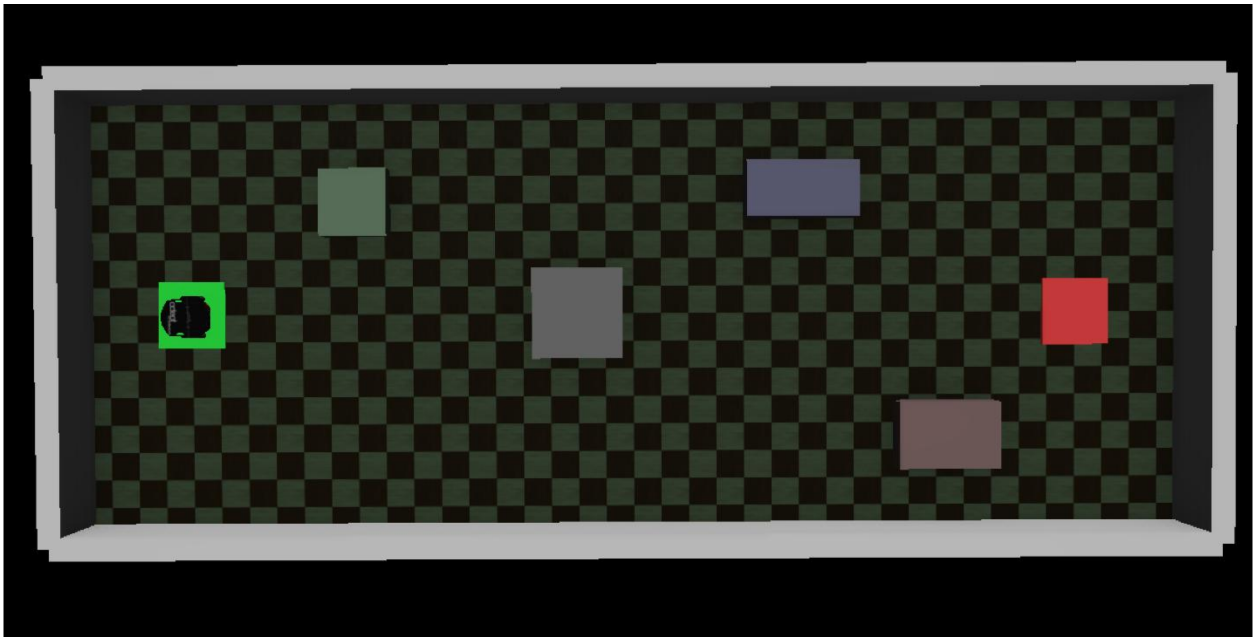

Figure 2:Figure 2: Corridor-based simulation environment created in Webots, showing static obstacles, start position, and goal location used for navigation testing.

A corridor-based indoor environment was created to represent typical spaces where powered wheelchairs are commonly used, such as hospitals, care facilities, and academic buildings. The corridor has fixed walls that restrict movement and increase the risk of collisions if control is inaccurate. This makes it suitable for evaluating obstacle detection and safety behaviour. As shown in Figure 2, several static obstacles are placed inside the corridor to create realistic navigation challenges. The figure also shows the wheelchair’s initial position and the goal location used during autonomous navigation tests. This layout allows clear observation of how the wheelchair reacts when moving close to walls and obstacles.

Only static obstacles were used in the simulation environment. This design choice ensures that each test is performed under the same conditions, which improves repeatability and allows fair comparison between different test runs. Dynamic obstacles were not included because the focus of this project is on basic navigation assistance and safety control rather than complex prediction or tracking. Start and goal markers were added to clearly define the navigation task and to check whether the wheelchair successfully reaches the target position. Overall, the environment is designed for controlled and repeatable evaluation, allowing system performance and safety behaviour to be analysed in a clear and reliable manner.

3.4 Simulated Wheelchair Model

In this project, the wheelchair is modelled in simulation using the Pioneer3-DX robot available in Webots. This robot is used as a proxy for a powered wheelchair because it has similar motion characteristics and is widely used in academic robotics research. Using an existing and well-tested model allows the focus of the work to remain on navigation, sensing, and safety behaviour rather than on detailed mechanical design. Figure 3 shows the simulated Pioneer3-DX model used in this project, including the wheel configuration and sensor placement around the body.

Figure 3: Pioneer3-DX robot model used as a proxy for a powered wheelchair in the simulation.

The Pioneer3-DX uses a differential drive system with two independently controlled main wheels. By adjusting the speed of the left and right wheels, the robot can move forward, reverse, and turn smoothly. This type of movement is commonly used in powered wheelchairs designed for indoor environments, where precise control and small turning radius are important. The front caster wheel, visible in the figure, supports balance while allowing flexible motion, which further matches the behaviour of real wheelchair platforms. Using a proxy robot model is an accepted practice in robotics research when the main objective is to study control strategies and system behaviour. In this project, the goal is to evaluate sensor-based navigation and safety control rather than mechanical performance. The Pioneer3-DX provides a stable and reliable platform for testing these functions in simulation. This approach keeps the simulation simple, repeatable, and realistic, and it supports the later transition from simulation results to a low-cost physical hardware prototype.

3.5 Sensor Configuration in Simulation

In the simulation, the wheelchair is equipped with multiple ultrasonic distance sensors to support safe movement in indoor environments. These sensors are selected because they are low cost, simple to use, and effective for short-range obstacle detection. Sensors are placed at the front, sides, and rear of the wheelchair to provide wide coverage and reduce blind spots during forward motion, turning, and reversing. Front sensors detect obstacles directly ahead and slightly to each side, while side sensors monitor the distance to corridor walls and nearby objects during steering. Rear sensors are used to detect obstacles when moving backward. The sensors are positioned to detect obstacles early, allowing the system to reduce speed or stop before a collision occurs. Distance thresholds are used to trigger different safety actions, with larger distances causing speed reduction and smaller distances activating emergency stopping. Ultrasonic sensors have known limitations, such as sensitivity to surface shape, angle, and noise, and they provide only distance information. Because of this, ultrasonic sensing alone is not sufficient for full autonomous navigation. To strengthen safety monitoring, an inertial measurement unit (IMU) is included conceptually in the simulation to represent tilt and abnormal motion detection. The IMU supports safety monitoring rather than navigation, reinforcing the system’s safety-first design.

3.6 Simulation Control Implementation

The simulation control system is implemented using a single Python controller within the Webots environment. This controller manages all wheelchair behaviors, including manual driving, autonomous navigation, and safety supervision. The control logic is designed in a modular way so that movement decisions, sensor processing, and safety checks are clearly separated. This structure improves readability and ensures that safety rules can always override motion commands when required. Figure 4 presents the high-level control architecture of the simulation controller, showing how sensor inputs, mode selection, safety supervision, and motion control are connected in a

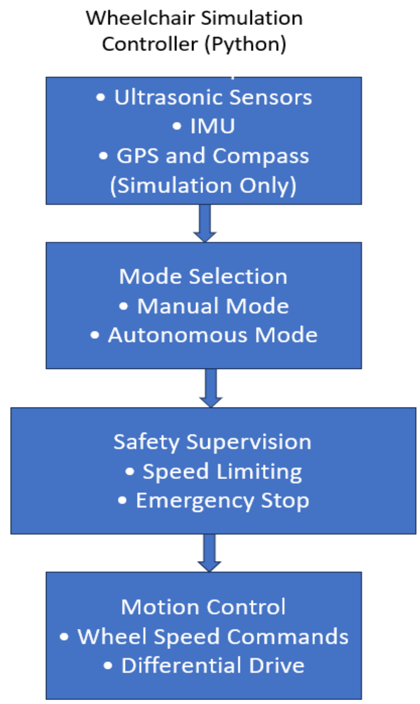

Figure 4: High-level simulation control architecture showing sensor inputs, mode selection, safety supervision, and motion control flow.

structured flow. At each simulation time step, the controller follows a fixed execution order. First, sensor data is read from the ultrasonic sensors and internal state variables are updated. Next, the active control mode is checked based on user input. Before any wheel commands are sent, safety conditions are evaluated. This sequence ensures that unsafe actions are blocked even if they are requested by the user or generated by autonomous logic. This design reflects a safety-first philosophy suitable for assistive mobility systems.

Mode Switching and Control Flow

The system supports two operating modes: manual mode and autonomous mode. Mode switching is handled explicitly in the controller, and internal state variables are reset when a mode change occurs to prevent unexpected behavior. Safety supervision remains active in both modes and does not depend on the selected mode. This ensures consistent behavior and predictable system response.

Manual Control Logic and Safety Scaling

In manual mode, keyboard inputs are mapped to forward motion and turning commands. To improve driving smoothness, short-term input memory is used so that combined forward and turning motions are possible. Speed scaling is applied based on obstacle distance. When objects are detected nearby, the wheelchair slows down automatically, and if an obstacle is detected very close, motion is stopped completely. This allows the user to remain in control while reducing collision risk.

Autonomous Navigation and Goal Seeking

In autonomous mode, the wheelchair moves toward a predefined goal position in the corridor. The controller calculates the heading direction and adjusts wheel speeds to maintain alignment with the goal. When no obstacles are present, the wheelchair moves forward with small steering corrections. Obstacle avoidance is handled using a reactive state-based approach, as illustrated in Figure 5. When an obstacle is detected, the system switches from goal-seeking to avoidance behavior, reduces speed, and turns toward the side with more free space. Separate distance thresholds are used for obstacle detection and clearance to avoid unstable switching between states.

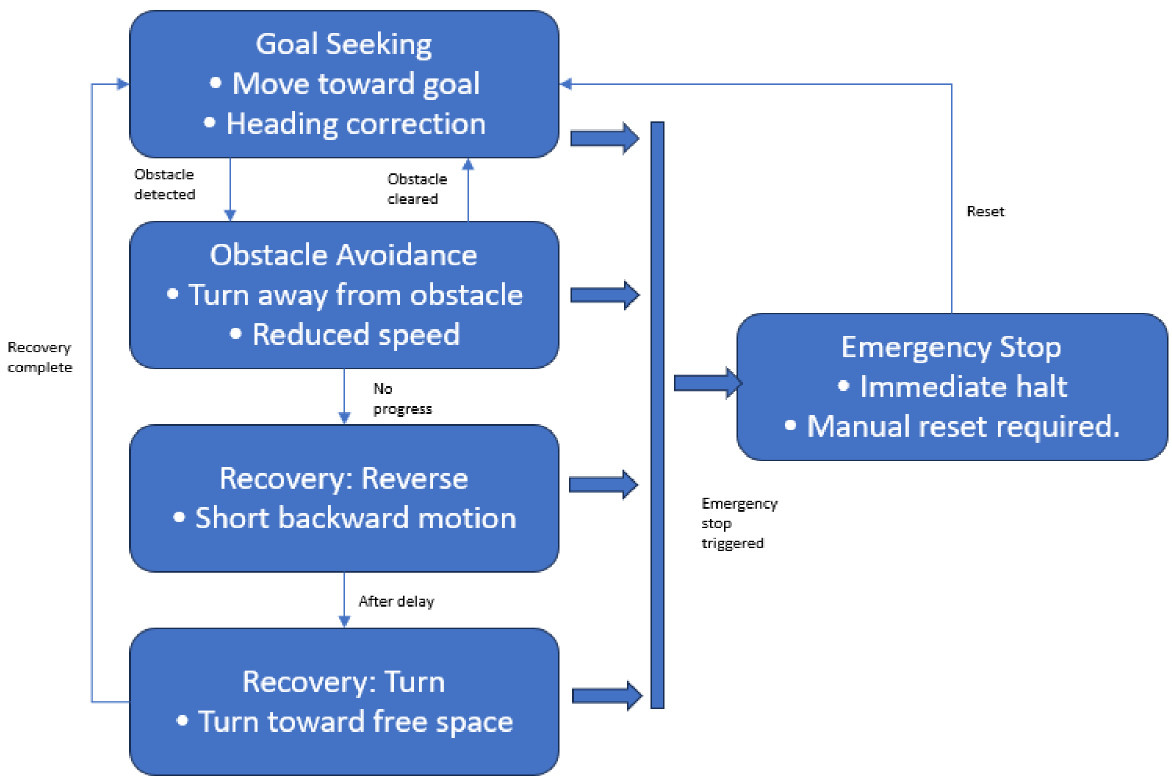

Figure 5:State-based autonomous navigation and safety behavior, illustrating goal seeking, obstacle avoidance, recovery actions, and emergency stop handling.

Speed Regulation and Emergency Stop Handling

Speed regulation is applied throughout both modes. The wheelchair slows down gradually as it approaches obstacles, improving stability and predictability. A hard emergency stop is triggered only when an immediate collision risk is detected or when the user activates the emergency brake. Once activated, all motion commands are cancelled, and the wheelchair remains stopped until the emergency state is manually cleared.

Stuck Detection and Recovery Behavior

In narrow corridor environments, the wheelchair may become stuck due to repeated avoidance actions. To address this, a stuck detection mechanism monitors whether progress toward the goal is being made. If no progress is detected for a fixed time, a recovery sequence is triggered. The wheelchair reverses slightly and then turns toward open space before resuming normal operation. This recovery logic improves robustness without requiring complex mapping or path planning.

Logging for Evaluation

To support evaluation, the controller records key data during simulation, including time, position, sensor readings, operating mode, navigation state, and motor commands. These logs are later used in Chapter 4 to analyze system behavior and safety performance.

3.7 Hardware System Architecture

The hardware prototype developed in this project is designed to validate the feasibility of the proposed smart wheelchair system using real, low-cost components. The goal of the hardware implementation is not to reproduce the full complexity of the simulation, but to demonstrate that the core concepts especially safety control, obstacle detection, and emergency braking can be achieved on an embedded platform. This practical approach allows the system design to be tested under real electrical, sensing, and timing constraints.

The overall hardware architecture is centered around an ESP32 microcontroller, which acts as the main control unit for the system. As shown in Figure 6, the ESP32 receives sensor data from ultrasonic distance sensors and an inertial measurement unit (MPU6050) and generates control signals for the motor driver. The ultrasonic sensors provide short-range obstacle information from the front and sides of the wheelchair, while the IMU is used to detect tilt, sudden motion, or crash conditions. These sensor inputs are processed continuously to support safety monitoring in both manual and autonomous modes.

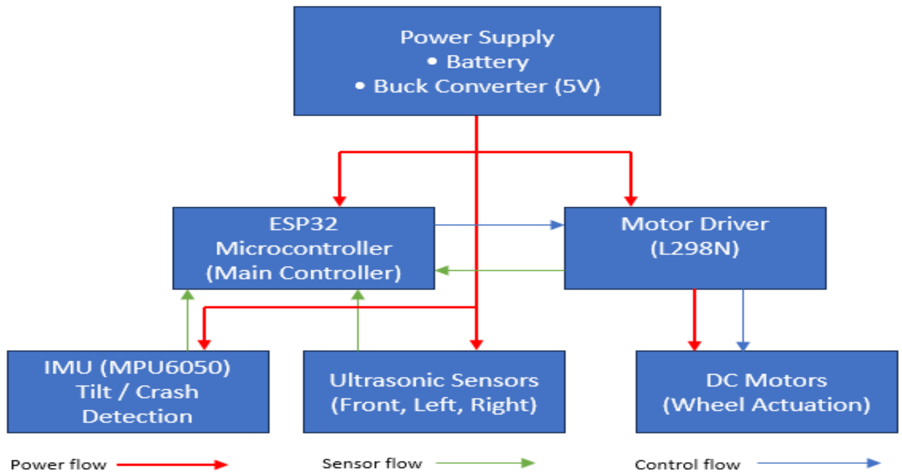

Figure 6:Hardware system architecture of the smart wheelchair prototype, showing power distribution, sensor inputs, control flow, and motor actuation centered around the ESP32 microcontroller.

Power for the system is supplied by a battery, with a buck converter used to regulate the voltage for the ESP32 and sensors. The motor driver (L298N) receives control signals from the ESP32 and provides sufficient current to drive the DC motors. Control signals, sensor data flow, and power distribution are clearly separated in the architecture to improve system stability and reduce interference. Compared to the simulation model, the hardware system focuses on reactive safety behavior and short-range navigation rather than full goal-based planning. This design reflects practical limitations such as cost, power consumption, and hardware simplicity, while still proving that a safety-first smart wheelchair concept can be implemented using affordable embedded components.

3.8 Hardware Components and Design Rationale

The hardware components used in this project are selected to validate the proposed smart wheelchair system using real, low-cost embedded hardware. The purpose of the hardware prototype is not to build a full commercial wheelchair, but to demonstrate that the safety and control concepts developed in simulation can be implemented reliably on physical hardware. For this reason, component selection is guided by cost, availability, simplicity, and suitability for assistive mobility applications.

Complete Components List

Component	Quantity	Purpose
ESP32 DevKit V1	1	Main microcontroller (WiFi + Blynk)
L298N Motor Driver	1	Controls 2 DC motors
DC Motors	2	Left and right wheelchair wheels
HC-SR04 Ultrasonic	3	Front, Left, Right obstacle detection
MPU6050 Sensor	1	Crash and tilt detection
Battery (7-12V)	1	Powers motors through L298N
5V Power Supply	1	Powers ESP32 (via USB or regulator)
1kΩ Resistors	6	Voltage dividers for HC-SR04 ECHO pins (2 per sensor)
Jumper Wires	~30	All connections

Table 1:Complete hardware component list used in the smart wheelchair prototype.

ESP32 DevKit V1.

The ESP32 DevKit V1 is used as the main microcontroller of the system. It acts as the central controller, reading sensor data, executing control and safety logic, and sending commands to the motor driver. The ESP32 is selected because it provides sufficient processing power for real-time control while remaining affordable and energy efficient. Its built-in Wi-Fi capability allows direct communication with the mobile application through the Blynk platform, which supports remote monitoring and emergency control. Figure 7 shows the ESP32 board and its pin layout, while Table 2 summarizes the key pins used for sensor input, motor control, and communication in this project.

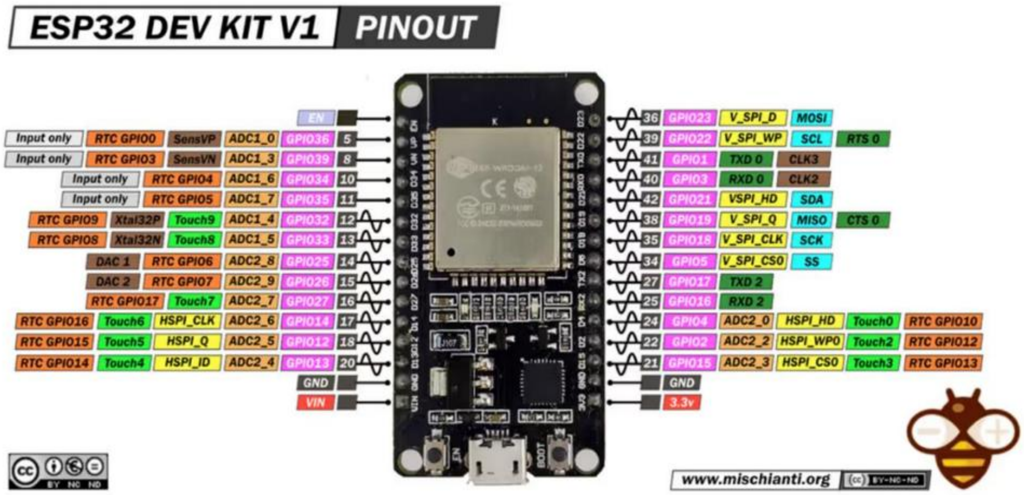

Figure 7:ESP32 DevKit V1 microcontroller and pin configuration used for sensor and motor interfacing (source: components101.com).

PIN	DESCRIPTION
Power Supply	
3.3V	3.3V power output from regulator.
5V	5V power input for the board.
GND	Ground connection.
EN	Reset button. Pull LOW to restart ESP32.
Digital I/O Pins	
GPIO 0-5	General input/output pins. Can read sensors or control devices.
GPIO 12-19	General input/output pins. Can read sensors or control devices.
GPIO 21-27	General input/output pins. Can read sensors or control devices.
GPIO 32-33	General input/output pins. Can read sensors or control devices.
Input Only Pins	
GPIO 34-39	Input only. Used to read sensors. Cannot control output devices.
Communication Pins	
GPIO 21	I2C data line (SDA). Used for sensor communication.
GPIO 22	I2C clock line (SCL). Used for sensor communication.
GPIO 1 (TX)	Serial transmit pin for data transfer.
GPIO 3 (RX)	Serial receive pin for data transfer.
Analog Input	
ADC Pins	Read analog values from sensors (0-3.3V range).
PWM Output	
All GPIO	All pins support PWM for motor speed control or LED dimming.

Table 2:ESP32 pin usage summary for sensors, motor driver, and communication interfaces.

HC-SR04 ultrasonic distance sensor.

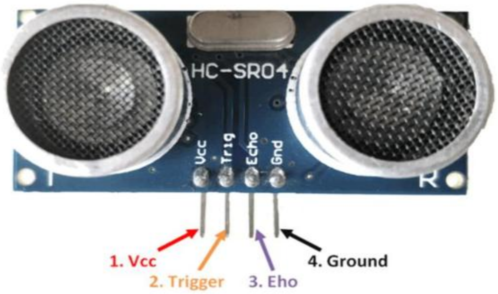

Figure 8:HC-SR04 ultrasonic sensor module showing pin configuration (source: components101.com).

PIN	DESCRIPTION
VCC	Power supply pin. Connect to 5V.
TRIG	Trigger input pin. Send 10µs HIGH pulse to start measurement.
ECHO	Echo output pin. Sends HIGH pulse with duration based on distance.
GND	Ground connection.

Table 4:Pin description of HC-SR04 ultrasonic distance sensor.

Technical Specifications

Operating Voltage:	5V DC
Operating Current:	15mA
Measuring Range:	2cm to 400cm
Accuracy:	±3mm
Measuring Angle:	15 degrees
Trigger Pulse:	10µs HIGH signal
Echo Pulse:	Proportional to distance

Table 3:Technical specifications of the HC-SR04 ultrasonic distance sensor.

Obstacle detection is implemented using three HC-SR04 ultrasonic sensors placed at the front, left, and right sides of the wheelchair prototype. These sensors provide short-range distance measurements that are suitable for indoor environments such as corridors and rooms. Ultrasonic sensors are chosen instead of cameras or LiDAR because they are low cost, simple to interface, and require minimal processing. Figure 8 illustrates the HC-SR04 sensor module and pin configuration, while the Table 3 and 4 describe its pins and lists down its main technical specifications. Using multiple sensors improves coverage and reduces blind spots, which supports safer movement during both manual and assisted operation.

MPU6050.

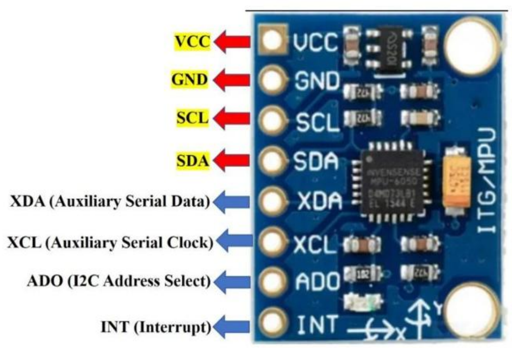

Figure 9:MPU6050 inertial measurement unit (IMU) with I²C communication pins (source: components101.com).

PIN	DESCRIPTION
VCC	Power supply pin. Connect to 3.3V or 5V.
GND	Ground connection.
SCL	I2C clock line. Connect to microcontroller SCL pin.
SDA	I2C data line. Connect to microcontroller SDA pin.
XDA	Auxiliary I2C data line. Used to connect external sensors.
XCL	Auxiliary I2C clock line. Used to connect external sensors.
AD0	I2C address select pin. LOW = 0x68, HIGH = 0x69.
INT	Interrupt output pin. Sends signal when data is ready.

Table 5:table showing pin description of MPU6050

Technical Specifications

Operating Voltage:	3V to 5V
Communication:	I2C protocol
I2C Address:	0x68 (default) or 0x69
Accelerometer Range:	±2g, ±4g, ±8g, ±16g
Gyroscope Range:	±250, ±500, ±1000, ±2000 °/s
Temperature Sensor:	Built-in (-40°C to +85°C)
Operating Current:	3.5mA
Data Update Rate:	Up to 8kHz

Table 6:Technical specifications of the MPU6050 sensor.

To monitor stability and detect unsafe motion, an MPU6050 inertial measurement unit (IMU) is included in the hardware design. The IMU provides acceleration and angular velocity data, which is used to detect excessive tilt or sudden impacts that may indicate a crash or tipping risk. In this project, the IMU is used only for safety monitoring and not for navigation or localization. This design choice reduces system complexity while still improving safety. Figure 9 shows the MPU6050 module and its I²C interface pins, table 5 describe its pin and Table 6 presents its operating ranges and key features.

L298N motor driver

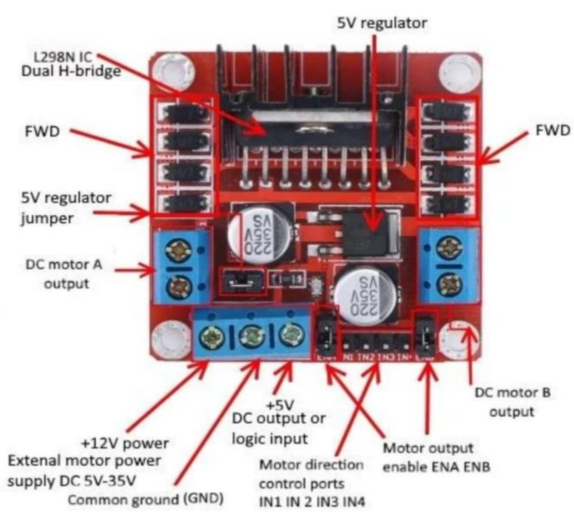

Figure 10:L298N dual H-bridge motor driver showing control and power connections (source: components101.com).

PIN	DESCRIPTION
VCC	Power supply for motors (5V to 35V). Add 2V extra if 5V-EN jumper is connected to run motors at full speed.
GND	Common ground pin for the circuit.
5V	Powers the internal logic circuit. Works as output if 5V-EN jumper is on. Connect to Arduino 5V if jumper is removed.
Motor A Control Pins	
ENA	Controls speed of Motor A. HIGH = motor runs, LOW = motor stops. Connect to PWM pin for speed control.
IN1	Direction control for Motor A. Use with IN2 to set rotation direction.
IN2	Direction control for Motor A. IN1 HIGH + IN2 LOW = forward. IN1 LOW + IN2 HIGH = reverse.
OUT1	Connect to Motor A (positive terminal).
OUT2	Connect to Motor A (negative terminal).
Motor B Control Pins	
ENB	Controls speed of Motor B. HIGH = motor runs, LOW = motor stops. Connect to PWM pin for speed control.
IN3	Direction control for Motor B. Use with IN4 to set rotation direction.
IN4	Direction control for Motor B. IN3 HIGH + IN4 LOW = forward. IN3 LOW + IN4 HIGH = reverse.
OUT3	Connect to Motor B (positive terminal).
OUT4	Connect to Motor B (negative terminal).

Table 7:L298N motor driver pin usage summary for motor control and communication interfaces.

Motor control is handled using an L298N dual H-bridge motor driver. This driver allows the ESP32 to control the speed and direction of two DC motors independently, enabling differential drive movement. The L298N is selected because it is widely used in educational and research prototypes, easy to connect, and capable of handling the current required by small DC motors. Although it is less efficient than modern drivers, its simplicity and reliability make it suitable for this prototype. Figure 10 shows the L298N module with its control and power connections.

Figure 11:Power supply setup including battery and DC–DC buck converter for regulated voltage delivery.

Power for the system is supplied using a 7.2 V battery. A DC–DC buck converter is used to step down the battery voltage to 5 V for the ESP32 and sensors. This ensures stable operation and protects sensitive electronic components from over-voltage. The motors are powered directly through the motor driver to provide sufficient current. Figure 11 illustrates the power supply arrangement, including the battery and voltage regulator. Overall, the selected hardware components provide a balanced solution that supports safety, reliability, and low-cost implementation, making them suitable for validating the proposed smart wheelchair system

3.9 Hardware Circuit Design and System Operation

The hardware circuit of the smart wheelchair prototype is designed to provide safe operation and stable control using low-cost embedded components. Figure 12 presents the complete circuit diagram of the system, where the ESP32 microcontroller acts as the central controller. All sensor inputs, safety checks, and motor control commands are handled by the ESP32. The circuit design separates power lines, sensor signals, and motor control connections, which helps reduce electrical noise and improves overall system reliability during operation.

Power is supplied using a 7.2 V battery, as shown in Figure 12. A DC-DC buck converter is used to step this voltage down to 5 V for the ESP32 and the sensors. This ensures a stable power supply and protects sensitive electronic components from over-voltage. The L298N motor driver is powered directly from the battery so that it can deliver sufficient current to the DC motors. A common ground connection is shared between the ESP32, sensors, and motor driver to ensure correct signal reference and reliable communication.

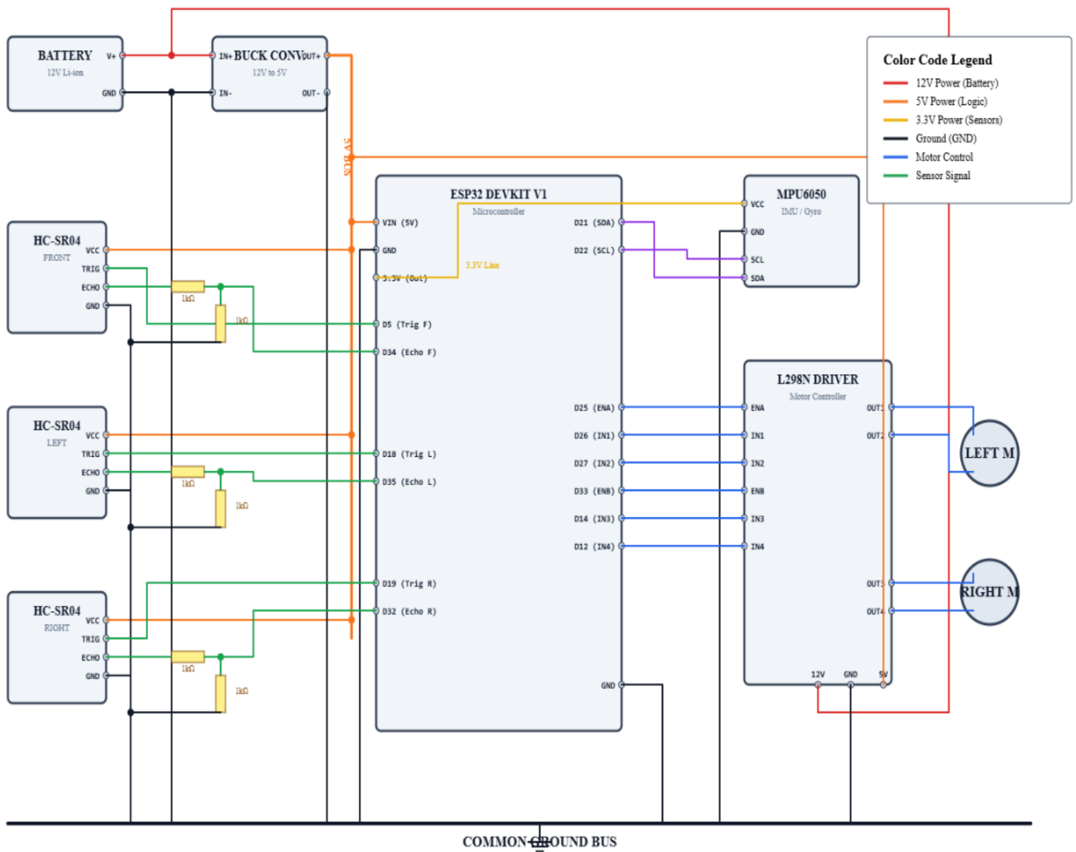

Figure 12:Complete hardware circuit diagram of the smart wheelchair prototype, showing power distribution, sensor connections, ESP32 control logic, and motor driver interfacing.

Three HC-SR04 ultrasonic sensors are connected to the ESP32 using separate trigger and echo pins, as illustrated in Figure 12. These sensors measure distance by sending ultrasonic pulses and receiving the reflected signals. The ESP32 calculates the distance based on the time delay and uses averaged readings to reduce noise. These distance measurements allow the system to detect obstacles in front of and on the sides of the wheelchair and to activate safety responses when required.

The MPU6050 inertial measurement unit is connected to the ESP32 through the I²C communication interface. This sensor provides acceleration and rotation data that is used to detect unsafe conditions such as tilt, sudden impact, or abnormal motion. The IMU is used only for safety monitoring and does not control movement directly. Motor control is handled by the L298N driver, where direction pins control wheel rotation and PWM signals control speed. When an emergency condition is detected, the ESP32 immediately disables motor outputs to stop the wheelchair safely.

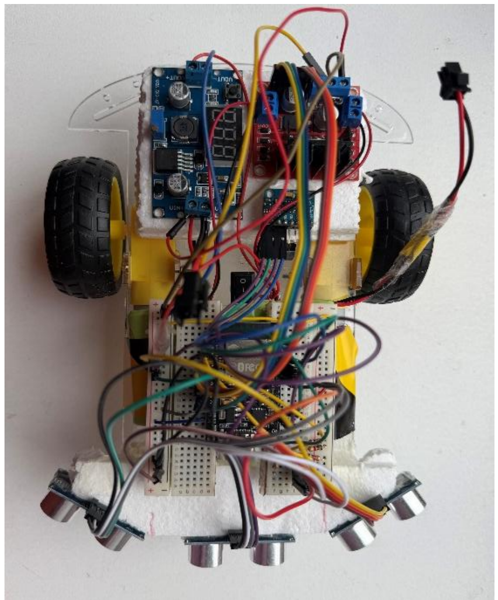

Figure 13: Assembled smart wheelchair hardware prototype showing real component placement, wiring, sensors, and drive system used for experimental testing.

Figure 13 shows the completed hardware prototype after assembly. This figure clearly shows the physical placement of the ESP32, motor driver, buck converter, breadboard, battery, and ultrasonic sensors on the mobile platform. Although the wiring is visible due to the prototype nature of the system, the hardware layout directly follows the circuit design shown in Figure 12. During operation, sensor data is processed first, safety conditions are checked next, and motor commands are applied only if the system is safe. This operating sequence ensures that safety always has priority over movement.

3.10 Embedded Software Implementation

The embedded software for the smart wheelchair is developed using an ESP32 microcontroller programmed through the Arduino platform. The main role of the software is to read sensor data, receive user commands from the mobile application, control the motors, and ensure that safety rules are always enforced. The overall program logic is illustrated in Figure 14, which presents the complete control flow of the system from initialization to continuous operation.

As shown at the top of the flowchart, the software begins by initializing serial communication, input and output pins, and PWM channels required for motor speed control. After this, the MPU6050 inertial sensor is initialized, followed by the Wi-Fi connection and Blynk application setup. This initialization stage ensures that all sensors, communication interfaces, and control outputs are ready before the wheelchair is allowed to move. Once setup is completed, the program enters the main execution loop, which runs continuously during system operation.

Inside the main loop, the software first updates all inputs received from the Blynk mobile application. These inputs include control mode selection, joystick values, and the emergency brake command. The software then checks whether the wheelchair is operating in manual mode or autonomous mode. Based on this selection, different safety checks are performed. In manual mode, the software monitors tilt conditions using the MPU6050 to detect possible tipping. In autonomous mode, additional checks are applied, including tilt detection, impact detection, and sudden motion changes. If any unsafe condition is detected, the emergency brake is activated, an alert message is sent to the mobile application, and the motors are stopped immediately.

When no emergency condition is present, the software proceeds with motion control. In manual mode, joystick X and Y inputs are converted into forward motion and turning commands. Dead-zone filtering and gradual speed changes are applied to avoid sudden or unstable movement. In autonomous mode, ultrasonic sensors placed at the front and sides are used to detect obstacles. If a front obstacle is detected at a close distance, the wheelchair performs a stop-reverse-turn sequence to avoid collision. If obstacles are detected on the sides, steering adjustments are applied while moving forward. After completing each control cycle, the loop repeats, ensuring continuous monitoring and safe operation.

Overall, the software structure shown in Figure 14 demonstrates a safety-first design where emergency braking and accident detection always override movement commands. This approach ensures reliable operation, predictable behavior, and strong user safety in both manual and autonomous modes.

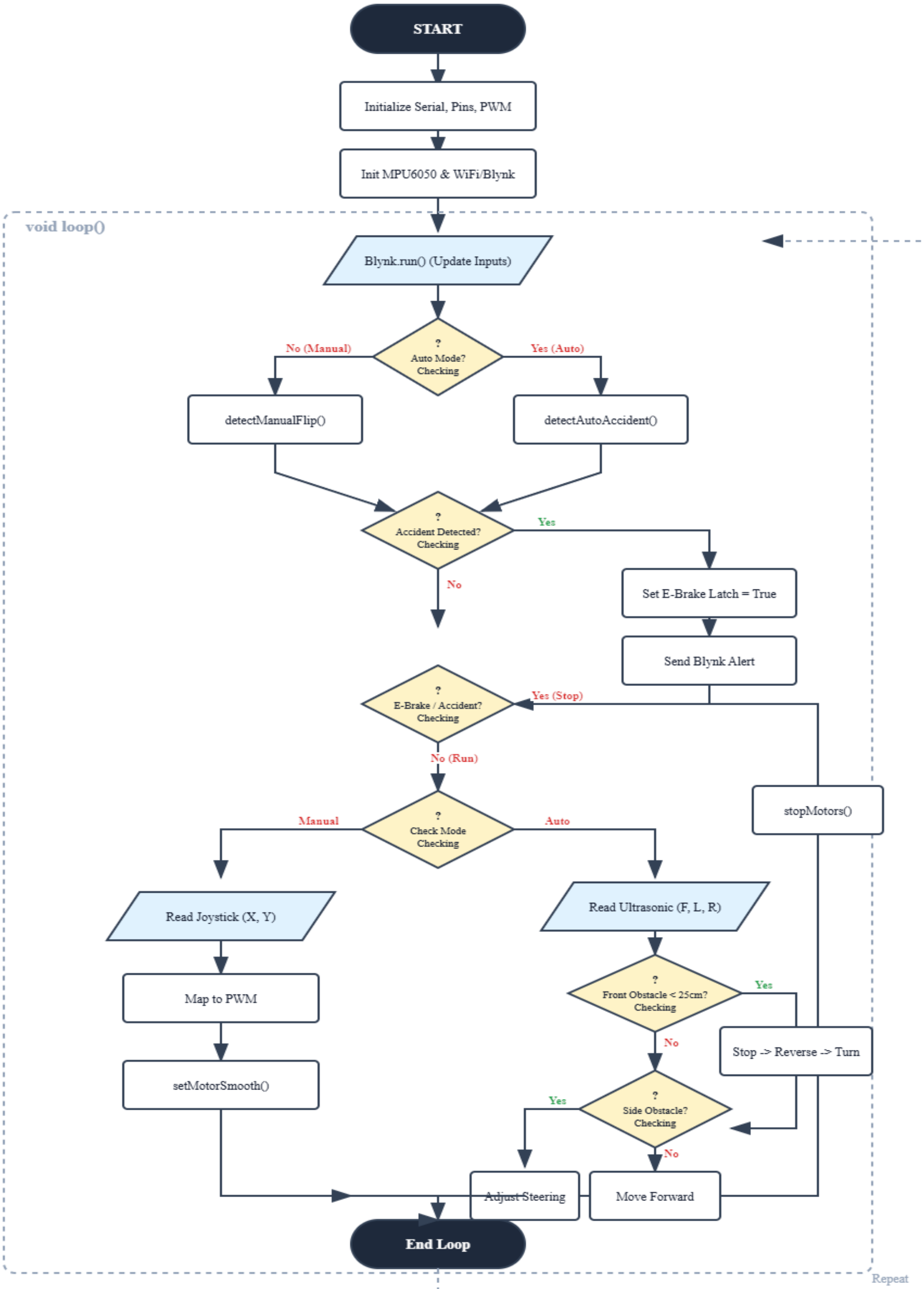

Figure 14:Embedded software flowchart showing system initialization, mode selection, accident detection, emergency brake logic, and manual and autonomous motor control.

3.11 Mobile Application Interface

In this project, a mobile application is developed using the Blynk platform to control and monitor the smart wheelchair prototype. Blynk is an Internet of Things (IoT) platform that allows communication between a smartphone and a microcontroller through a cloud-based system. It is selected because it provides a reliable, low-cost, and fast method to create a custom control interface without developing a mobile application from scratch. This makes it well suited for academic prototypes and embedded system projects where stability and ease of integration are important.

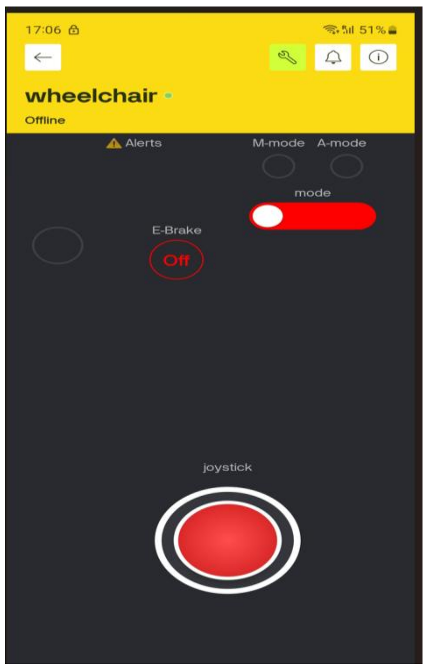

Figure 15:Custom Blynk mobile application interface developed for the smart wheelchair prototype, showing manual joystick control, mode selection (manual/autonomous), emergency brake (E-Brake) control.

The mobile control interface is created using widgets provided by the Blynk mobile application. Figure 15 shows the custom wheelchair control interface designed for this project. The interface includes a virtual joystick for manual driving, buttons for switching between manual mode and autonomous mode, and a dedicated emergency brake (E-Brake) control. A text display is also used to show real-time safety alerts. These elements allow the user to control movement, change operating modes, and respond quickly to emergency situations. The layout is designed to be simple and clear, so that important safety controls are always easy to access.

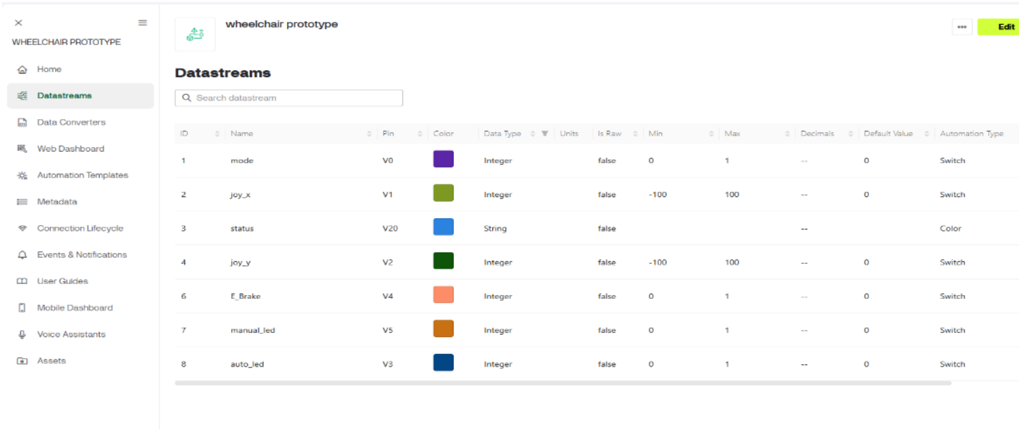

Figure 16:Blynk cloud datastream configuration for the smart wheelchair, showing virtual pin assignments linked to ESP32 control and monitoring.

To support this interface, virtual data channels are configured on the Blynk cloud platform**.** Figure 16 shows the datastream configuration created for the project, where each control and indicator are assigned to a specific virtual pin. These datastreams include mode selection, joystick X and Y values, emergency brake input, status messages, and visual indicators for the active mode. On the hardware side, the ESP32 reads these virtual pin values using the Blynk library. When the user interacts with the mobile app, the data is sent to the Blynk cloud and then forwarded to the ESP32 over Wi-Fi, enabling real-time control of the wheelchair.

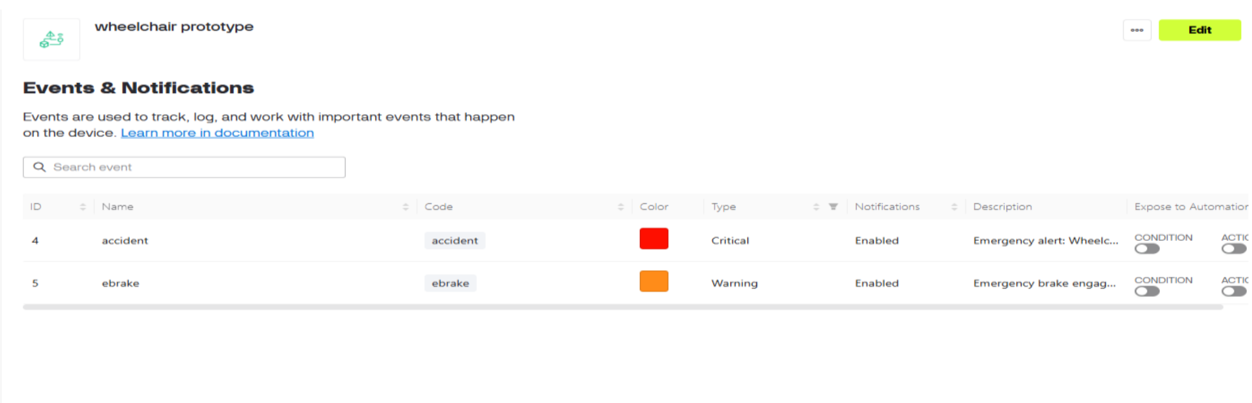

Figure 17:Blynk cloud events and notification setup for accident detection and emergency brake alerts in the smart wheelchair system.

In addition to control, the mobile application plays a key role in system safety. Figure 17 shows the events and notification setup configured on the Blynk cloud. Two main safety events are defined: accident detection and emergency brake activation. When the embedded software detects unsafe conditions such as tilt, crash, or sudden impact, the ESP32 automatically engages the emergency brake and sends an alert through the Blynk server. These notifications are immediately displayed on the mobile application, informing the user that the wheelchair has stopped due to a safety event. This design ensures that safety feedback is clear and immediate, supporting the safety-first shared control approach used throughout the system.

Communication security is handled using a unique authentication token assigned by the Blynk platform, which ensures that only the authorised application can access and control the wheelchair. The ESP32 connects to the Blynk cloud using this token and a secured Wi-Fi connection. In addition to control commands, the system also sends feedback and safety alerts to the user. The third screenshot shows the events and notifications configured in the Blynk console, including crash and emergency brake alerts. When the embedded software detects unsafe conditions such as tilt or impact, the E-Brake is activated automatically and an alert message is sent to the mobile application. These figures together demonstrate how the Blynk platform supports safety-first shared control by combining user input, cloud communication, and real-time safety notifications in a single interface.

Chapter 4: Simulation and Experimental Results

4.1 Evaluation Methodology

The evaluation of this project is conducted to assess the behaviour of the smart wheelchair system under different operating conditions, with particular emphasis on safety performance, assisted navigation, and control reliability rather than navigation efficiency. A simulation-first evaluation approach is adopted, as it enables controlled and repeatable testing without physical risk, which is appropriate for early-stage assistive robotics development. The Webots simulation environment is used to evaluate manual control, autonomous movement, obstacle avoidance, emergency braking, and recovery behaviour under consistent indoor conditions. Hardware testing is then carried out as a secondary stage to verify the practical feasibility of the proposed design using a real embedded prototype. Due to limitations related to available space, power constraints, and prototype-level construction, hardware experiments focus on short-range motion, safety response, and alert generation instead of full autonomous navigation. Ethical considerations are fully addressed, as no human participants are involved, all tests are performed without a user seated on the wheelchair, and no personal or sensitive data is collected, ensuring that the evaluation process remains low-risk and compliant with university ethical guidelines.

4.2 Simulation Test Scenarios

Several test scenarios are defined within the Webots simulation environment to evaluate the behaviour of the smart wheelchair system under different operating conditions. All scenarios are executed in the same corridor-based indoor environment to ensure consistency and repeatability across tests. The simulation layout, obstacle placement, and wheelchair position are shown in Figures 18 and 19. These scenarios are designed to evaluate manual control, autonomous movement, obstacle avoidance, emergency braking, and recovery behaviour, in line with the control and safety logic described in Chapter 3.

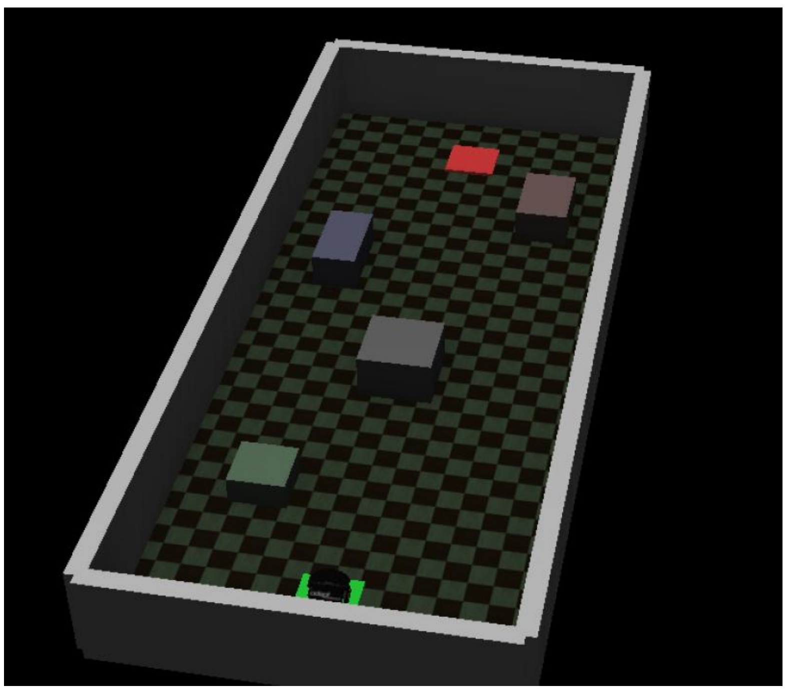

Figure 18:Top-down view of the corridor-based Webots simulation environment showing static obstacle placement, wheelchair position, and navigation layout used for evaluation of safety and obstacle avoidance behaviour.

The first scenario evaluates manual driving with the safety layer active. The wheelchair is controlled using keyboard input representing user commands and is intentionally driven close to corridor walls and static obstacles. The expected behaviour is that the system automatically reduces speed or stops when unsafe distances are detected, while still allowing normal user control when conditions are safe. This scenario verifies that the shared-control safety layer operates correctly during manual operation.

The second scenario evaluates autonomous navigation. In this mode, the wheelchair moves from a defined start position toward a goal location without user input. The purpose of this scenario is to observe whether the wheelchair can move smoothly toward the goal while avoiding collisions. Direction and speed are adjusted based on ultrasonic sensor readings, demonstrating basic navigation assistance in an indoor environment.

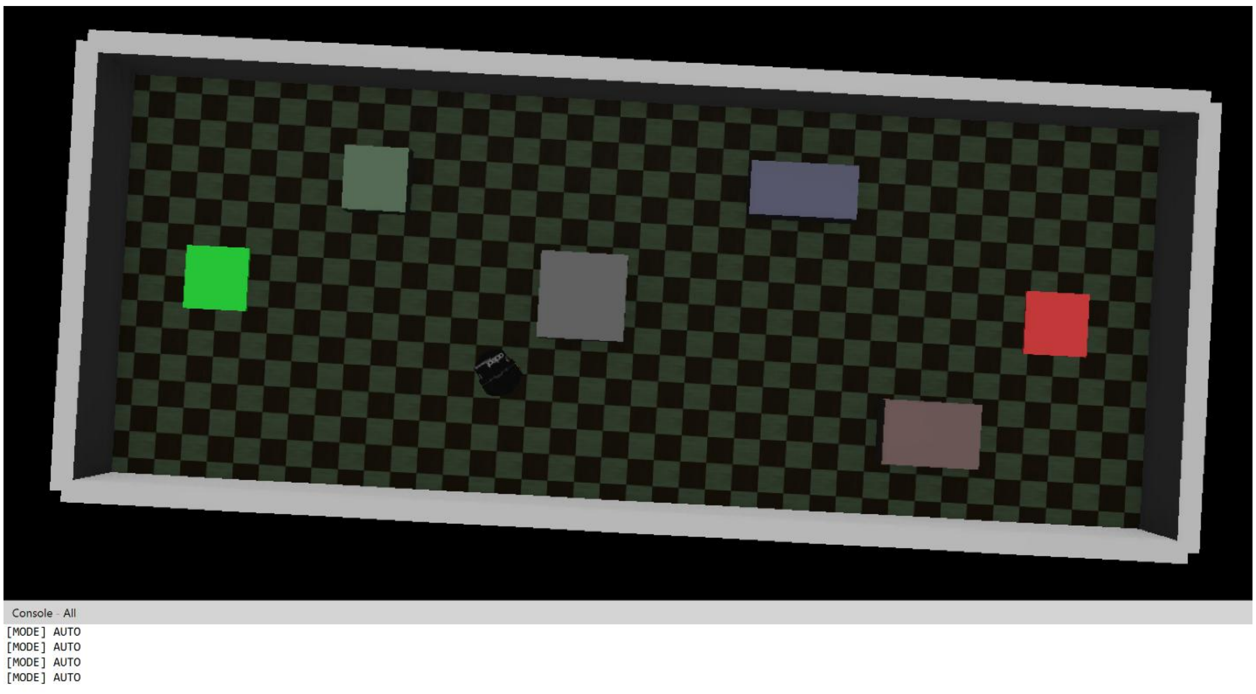

Figure 19:Perspective view of the same Webots simulation environment illustrating corridor boundaries, obstacle distribution, and the initial wheelchair start position during autonomous navigation testing.

Another scenario focuses specifically on obstacle avoidance using static obstacles placed within the corridor. When an obstacle is detected in front of the wheelchair, the system is expected to slow down, turn toward the side with more free space, and continue moving once the path is clear. This scenario checks whether the avoidance logic functions correctly without causing oscillation or the system becoming stuck.

An emergency stopping scenario is also tested. In this case, the wheelchair approaches an obstacle at a very close distance to deliberately trigger the emergency stop condition. The expected result is an immediate stop, confirming that emergency braking has the highest priority over all other control actions.

The final scenario evaluates recovery behaviour. When the wheelchair is unable to make progress due to its position relative to obstacles, the system is expected to reverse slightly and then turn before continuing forward. This scenario verifies the controller's ability to recover safely from difficult situations and maintain stable navigation.

4.3 Simulation Results

The simulation results demonstrate that the smart wheelchair system behaves safely and predictably across all defined test scenarios. During manual driving tests, the wheelchair responded correctly to user commands while the safety layer remained continuously active. When the wheelchair approached walls or obstacles, its speed was automatically reduced, and when the distance became unsafe, the system stopped completely. This behaviour confirms that the safety layer can successfully override user input when required, without restricting control when the environment is safe. The emergency stop behaviour observed in manual mode is illustrated in Figure 20.

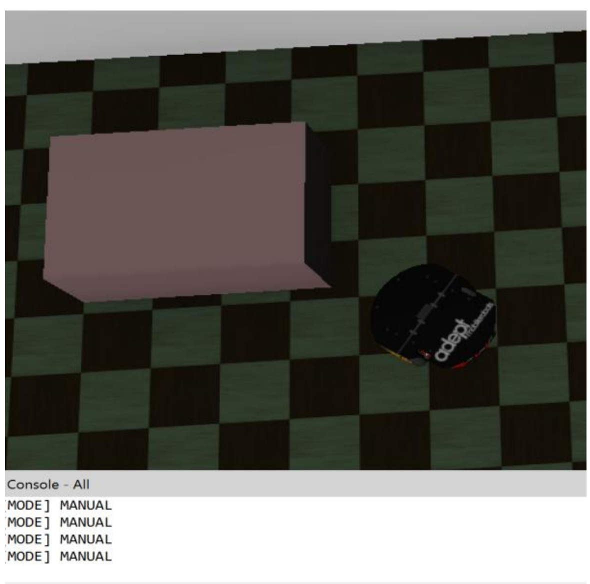

Figure 20:Simulation screenshot showing the wheelchair stopped close to an obstacle during manual control, illustrating emergency braking and safety override behaviour when a collision risk is detected.

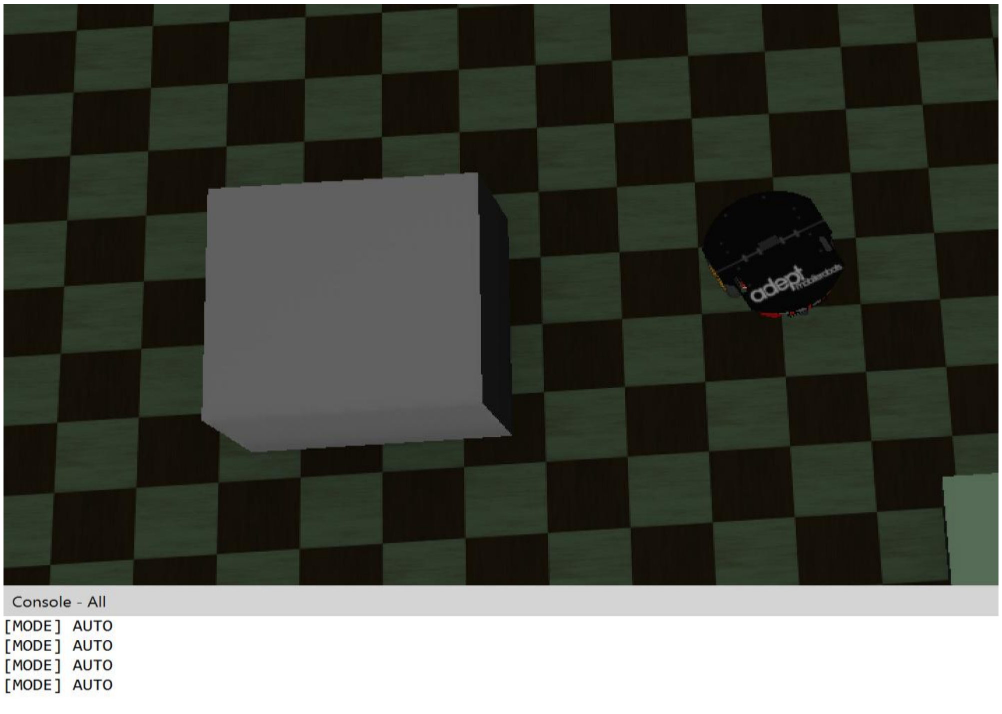

Figure 21:Simulation screenshot showing the wheelchair actively avoiding a static obstacle by changing direction, demonstrating the obstacle avoidance behaviour during autonomous navigation.

During autonomous navigation tests, the wheelchair was able to move from the defined start position toward the goal location without colliding with obstacles. The system adjusted its direction smoothly based on sensor readings, demonstrating stable goal-seeking behaviour. When obstacles were detected, the wheelchair slowed down and turned away before continuing forward. This confirms that the goal-seeking and obstacle avoidance states described in Chapter 3 operate together as intended in a corridor environment.

Obstacle avoidance behaviour remained stable across most simulation runs. When a static obstacle appeared in front of the wheelchair, the controller selected the direction with more available space and adjusted the wheel speeds accordingly. After passing the obstacle, the wheelchair returned to its normal navigation behaviour. This avoidance process is shown in Figure 21, where the wheelchair changes direction to maintain a safe distance from the obstacle. In narrower areas, the wheelchair reduced its speed, which improved safety and prevented collision.

Emergency stopping was consistently reliable during simulation testing. When an obstacle was detected at a very short distance, the wheelchair stopped immediately, regardless of whether it was operating in manual or autonomous mode. The warning indicator was also activated during these events, confirming that emergency conditions were correctly detected and prioritised. This behaviour demonstrates that emergency braking has the highest control priority in the system.

Recovery behaviour was observed in situations where the wheelchair could not make progress toward the goal. In these cases, the system reversed slightly and then turned before moving forward again. This allowed the wheelchair to escape difficult positions and continue navigation safely. Overall, the simulation results confirm that the system can safely manage navigation, obstacle avoidance, emergency stopping, and recovery behaviour within a controlled indoor environment.

4.4 Hardware Testing Scenarios

Hardware testing was conducted to evaluate the behaviour of the smart wheelchair prototype in real indoor conditions. All tests were performed in a controlled environment on a flat, carpeted surface, and no person was seated on the wheelchair during testing. The primary aim of these tests was to verify basic movement, obstacle detection, autonomous response, emergency braking, and alert generation rather than full autonomous navigation. The physical hardware setup used during testing, including the motors, sensors, and electronic components mounted on the prototype, is shown in Figure 22.

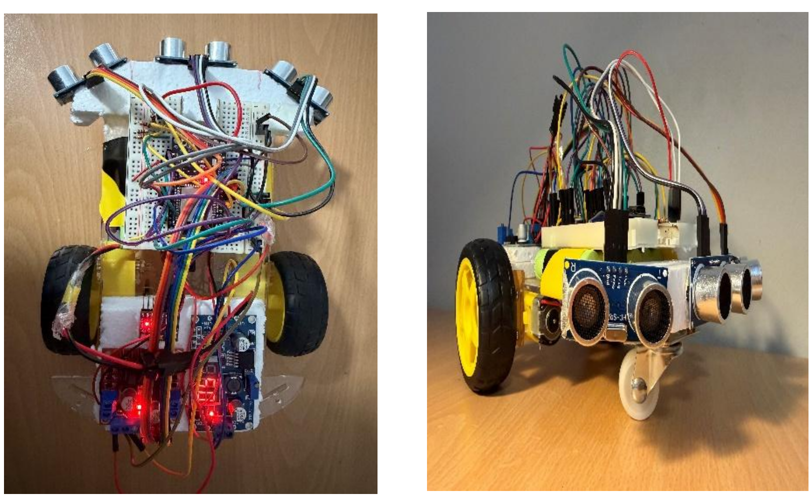

Figure 22:Hardware prototype of the smart wheelchair showing the mounted ESP32 controller, ultrasonic sensors, motor driver, and power components used during indoor testing.

The first test focused on manual control using the Blynk mobile application. The wheelchair was commanded to move forward, backward, and turn using the virtual joystick interface. This test verified that control commands sent from the mobile application were correctly received by the ESP32 and translated into motor actions. The wheelchair's response to joystick input during manual driving is illustrated in Figure 23. During manual operation, the emergency brake (E-Brake) was also tested. While the wheelchair was moving, the E-Brake button on the mobile application was activated. The wheelchair stopped immediately and remained stationary until the brake was manually reset. This confirms that the emergency brake has the highest priority over all motion commands. The E-Brake activation and reset state displayed on the mobile application are shown in Figure 24.

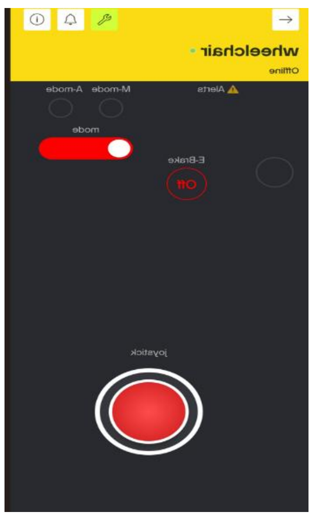

Figure 24:Manual wheelchair control using the Blynk mobile application joystick during hardware testing.

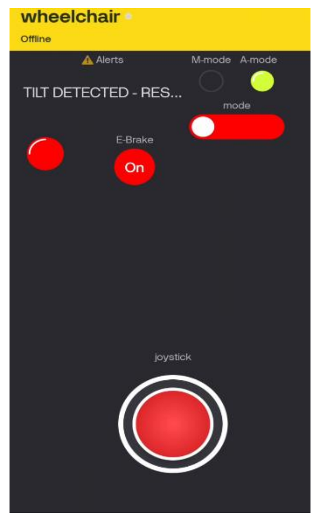

Figure 23:Emergency brake activation and lock state displayed on the Blynk mobile application.

Obstacle detection and avoidance were then evaluated using a solid object placed in front of the wheelchair. As the wheelchair approached the obstacle, the ultrasonic sensors detected it and the system responded by slowing down and turning to avoid a collision. This sequence of behaviour is shown in Figures 25, 26, and 27, where the wheelchair approaches the obstacle, changes direction, and then moves away after successful avoidance.

Figure 26:Wheelchair approaching a static obstacle during hardware obstacle detection testing.

Figure 27: Wheelchair turning away from the obstacle after detection, demonstrating avoidance behaviour.

Figure 25:Wheelchair moving away safely after completing the obstacle avoidance manoeuvre.

Autonomous movement was also tested over a short distance due to space limitations. In this mode, the wheelchair moved forward without user input and adjusted its direction automatically when obstacles were detected. The purpose of this test was to confirm that the autonomous control logic operates safely in real conditions. The avoidance manoeuvre during autonomous mode is again visible in Figure 26.

The final test focused on safety detection and automatic emergency braking. The wheelchair was slightly tilted or disturbed to simulate unsafe conditions. When this occurred, the IMU detected the event, the emergency brake was engaged automatically, and the motors were disabled. At the same time, an alert notification was sent to the mobile application. Examples of these safety alerts, including accident and tilt detection messages intended to notify caregivers or family members, are shown in Figures 28 and 29.

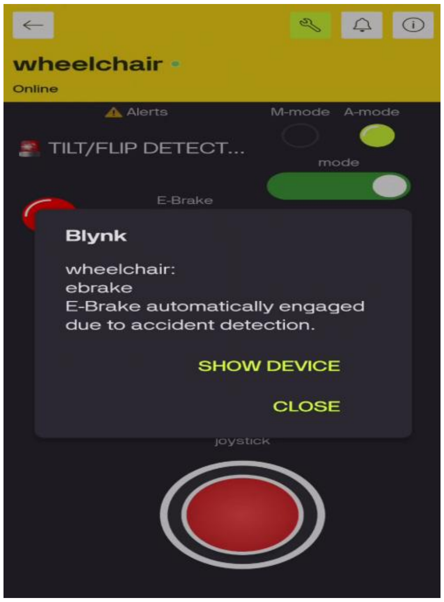

Figure 28:Blynk alert notification showing automatic emergency brake engagement due to detected accident or unsafe condition

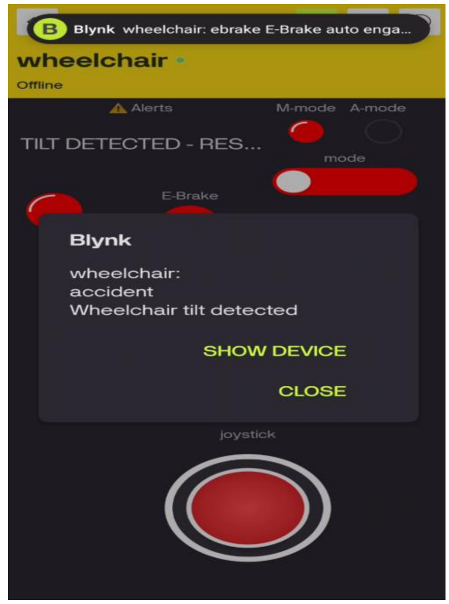

Figure 29:Blynk alert notification showing wheelchair tilt detection.

4.5 Hardware Results

The hardware test results indicate that the smart wheelchair prototype operated safely and reliably across all test scenarios. Manual control using the mobile application was responsive and stable, with the wheelchair moving forward, backward, and turning correctly in response to joystick commands. This confirms that communication between the mobile application, the ESP32 controller, and the motor driver functioned correctly. The physical movement observed in Figure 24 clearly demonstrates this behaviour.

Obstacle detection and avoidance were also effective in the real indoor environment. When an object was placed in front of the wheelchair, the ultrasonic sensors detected it at short range. Instead of continuing forward, the wheelchair reduced its speed and turned away from the obstacle. This behaviour is shown in Figures 25 and 26, where the wheelchair changes direction to avoid a collision. These results confirm that the basic autonomous avoidance logic validated in simulation can be successfully implemented on low-cost embedded hardware.

The emergency braking system proved to be one of the most reliable safety features of the prototype. The emergency brake could be activated manually at any time using the mobile application, resulting in an immediate stop. In addition, the emergency brake was triggered automatically when unsafe conditions, such as tilt, were detected. An example of this automatic response and alert notification is shown in Figure 29. This demonstrates that the system can detect hazardous situations and stop the wheelchair without requiring user input.

Some limitations were observed during hardware testing. Minor surface irregularities, such as carpet bumps, occasionally caused brief changes in sensor readings. During early tests, this resulted in false emergency brake activation. These effects were reduced by adjusting safety thresholds during testing. Despite these limitations, the system remained conservative in its responses, prioritising safety, which is appropriate for assistive wheelchair applications.

Overall, the hardware results confirm that safe manual control, basic autonomous obstacle avoidance, and reliable emergency braking can be achieved using low-cost components. Although the hardware system is less precise than the simulation environment, it successfully demonstrates the key safety and control principles of the proposed design under real operating conditions.

4.6 Simulation vs Hardware Comparison

In simulation testing, the wheelchair was provided with a defined start position and a fixed goal location, allowing consistent evaluation of autonomous navigation and safety behaviour. The simulated environment was flat and fully controlled, which resulted in smooth motion, stable obstacle avoidance, and reliable emergency stopping. These conditions made it possible to repeat the same tests multiple times with similar outcomes, supporting effective validation of the control logic and safety mechanisms. During hardware testing, the same control logic was applied; however, differences were observed due to real-world conditions. The floor surface was uneven and included small bumps, which caused vibration and occasional fluctuations in sensor readings. In some cases, this led to emergency brake activation even when no immediate danger was present. To address this, safety thresholds were adjusted during testing to reduce false emergency stops while maintaining a conservative safety response. This comparison highlights that simulation is valuable for early development and controlled validation, but hardware testing is essential to account for physical effects such as surface irregularities, vibration, and sensor noise.

Chapter 5: Discussion

5.1 Navigation Performance

The navigation performance of the smart wheelchair system indicates that the proposed control design is suitable for safe indoor movement. In manual mode, the wheelchair responded smoothly and consistently to user inputs, demonstrating reliable communication between the mobile application, the ESP32 controller, and the motor system. This confirms that the system is capable of real-time control without noticeable delay. In autonomous mode, the wheelchair was able to move forward and adjust its direction when obstacles were detected, showing that simple reactive navigation can be effective for short-range indoor assistance. When comparing simulation and hardware results, the simulation environment produced more stable and predictable behaviour due to the absence of sensor noise and surface irregularities. In contrast, real hardware testing introduced small variations caused by uneven ground and sensor uncertainty. However, these effects did not prevent successful movement or safe operation. Overall, the results suggest that a safety-focused, reactive navigation approach is appropriate for assistive wheelchair applications, where reliable and controlled motion is more important than speed or full autonomy.

5.2 Safety System Effectiveness

The safety system was a central component of the smart wheelchair design, and the results demonstrate that it functioned reliably in both simulation and hardware testing. The emergency brake consistently had the highest priority, ensuring that the wheelchair stopped immediately whenever a potentially dangerous situation was detected. This confirms that the safety logic was correctly implemented and that unsafe movement was prevented. The system responded effectively to both manual emergency brake commands from the mobile application and automatic triggers such as tilt or sudden motion detected by the sensors. From an engineering perspective, this shows that combining sensor-based safety monitoring with an emergency latch is an effective and practical solution. During early hardware tests, some false emergency stops occurred due to uneven surfaces and sensor noise. These events were reduced by adjusting safety thresholds while maintaining conservative behaviour. Overall, the safety system proved to be fast, dependable, and protective, supporting the design decision to prioritise safety over continuous motion, which is essential for assistive wheelchair systems.

5.3 Shared Autonomy Evaluation

The shared autonomy approach adopted in this project successfully balanced user control with automatic safety support. The wheelchair allowed the user to control movement manually while continuously monitoring for unsafe conditions. System intervention occurred only when necessary, such as slowing down or stopping to avoid a hazard. This behaviour reduced the risk of accidents while maintaining user involvement in control, which is important for trust and predictability in assistive systems. The results from both simulation and hardware testing show that the system was able to intervene effectively without fully taking control from the user. Full autonomy was deliberately avoided, as it would require more advanced sensors, higher computational resources, and more complex decision-making, increasing both cost and risk. From an engineering point of view, the results confirm that shared autonomy is a more suitable approach for a low-cost smart wheelchair system. Overall, the findings demonstrate that combining human input with automatic safety assistance provides a practical and reliable solution for indoor assistive mobility.

5.4 Limitations

Although the smart wheelchair system performed well during testing, several limitations were identified. These limitations are expected for a prototype system and highlight areas for future improvement. One limitation relates to obstacle detection using ultrasonic sensors. While these sensors are low-cost and simple to implement, they have a limited sensing range and are affected by object shape and surface angle. As a result, thin or angled objects were not always detected accurately, reducing detection reliability in some situations.

Another limitation concerns the inertial measurement unit used for safety monitoring. During hardware testing, small floor bumps and vibrations caused sudden changes in acceleration readings, which initially led to false emergency brake activation. Although this issue was reduced through threshold tuning, it shows that IMU-based safety detection can be sensitive to environmental conditions. More advanced filtering or sensor fusion could improve robustness in future designs.

The hardware design itself also introduced limitations. The use of low-cost motors and the absence of a suspension system meant that uneven surfaces affected system stability and sensor readings. In addition, sensor placement was constrained by the physical size of the prototype, limiting full coverage around the wheelchair. Testing conditions further limited evaluation, as all experiments were performed indoors and without a human user. This ensured safety and ethical compliance but prevented assessment of user comfort, weight effects, and long-term operation. Despite these limitations, the system achieved its intended goals and provides a realistic foundation for further development.

5.5 Ethical & Practical Considerations

Ethical and practical considerations were carefully addressed throughout this project. All testing was conducted without a human user seated on the wheelchair, ensuring that no risk was posed to individuals. This approach aligns with ethical requirements for early-stage assistive robotics research. The system was designed with safety as the highest priority, supported by both manual and automatic emergency braking. From a practical perspective, the prototype is intended solely for research and demonstration purposes and is not suitable for medical or real-world deployment. Any future testing involving human users would require formal ethical approval, additional safety mechanisms, and more extensive validation to ensure user protection.

Chapter 6: Conclusion & Future Work

6.1 Conclusion

This dissertation presented the design, simulation, and evaluation of a low-cost smart wheelchair system that prioritises safety and practical usability through a shared autonomy approach. Rather than aiming for full autonomous navigation, the system was designed to support the user while automatically intervening only when unsafe conditions are detected. This design choice reflects the requirements of assistive mobility systems, where reliability, predictability, and user trust are more important than complex autonomous behaviour.

A simulation-first development methodology was successfully applied using the Webots robotics simulator. The simulation environment enabled controlled and repeatable testing of key system functions, including manual control, autonomous navigation, obstacle avoidance, emergency braking, and recovery behaviour in an indoor corridor setting. The simulation results showed smooth navigation behaviour, stable obstacle avoidance, and correct prioritisation of safety actions. These findings confirmed that the proposed control logic and safety mechanisms functioned as intended under idealised conditions.

To evaluate real-world feasibility, a physical hardware prototype was developed using low-cost embedded components, including an ESP32 microcontroller, ultrasonic sensors, and an inertial measurement unit. Hardware testing demonstrated that the system could safely support manual control, basic autonomous obstacle avoidance, and reliable emergency braking. The integration of a mobile application interface allowed real-time control, emergency stop activation, and safety alert notifications, further strengthening the safety-focused design. Differences observed between simulation and hardware behaviour highlighted the effects of sensor noise, surface irregularities, and vibration, reinforcing the importance of physical testing alongside simulation.

Overall, the results demonstrate that a safety-first smart wheelchair system can be implemented using affordable hardware without relying on complex autonomy or expensive sensing technologies. The project confirms that shared autonomy is a practical and effective strategy for assistive wheelchair applications, particularly in indoor environments. By combining simulation-based development with embedded hardware validation, this work provides a realistic foundation for future research and development of accessible, low-cost assistive mobility systems.

6.2 future work

The current system successfully demonstrates important features such as emergency braking, alert notifications to family members, and low-cost autonomous navigation. In future work, the system can be extended to support outdoor navigation using GPS and digital maps. By integrating services such as Google Maps, the wheelchair could help users travel from one location to another, for example from home to a nearby grocery store or clinic. The user could select a destination using a mobile application, and the wheelchair would assist with navigation while still keeping safety as the main priority.

Another possible improvement is the use of artificial intelligence for better interaction and control. Voice commands could allow users to control the wheelchair without using a joystick or mobile app. Simple commands such as “go,” “stop,” or “turn left” would make the system easier to use. An AI assistant could also provide useful information such as news updates, weather reports, or live scores, and offer basic interaction to improve the user experience. These features could be especially helpful for elderly users who may find traditional controls difficult.

Future versions of the system could also use more advanced sensors to improve navigation safety. Sensors such as LiDAR or depth cameras would provide more accurate information about the environment and help detect obstacles more reliably. AI-based navigation methods could help the wheelchair plan safer paths and adapt to different environments. When combined with existing safety features like emergency braking and alert systems, these improvements could make the wheelchair more secure, intelligent, and suitable for real-world use.

References

Benayed, W. & Masmoudi, M. S., 2025. Voice-controlled autonomous navigation for smart wheelchairs using ROS-based SLAM. *Scientific Reports.*

Bourhis, G. & Agostini, Y., 1998. The VAHM robotized wheelchair: system architecture and human–machine interaction. *Journal of Intelligent and Robotic Systems,* Volume 22, p. 39–50.

Camargo, C. et al., 2021. A systematic literature review of realistic simulators applied in educational robotics context. *Sensors,* 21(12), p. 4031.

Carlson, T. & Demiris, Y., 2012. Collaborative control for a robotic wheelchair: Evaluation of performance, attention and workload. *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics,* 42(3), p. 876–888.

Erturk, E., Kim, S. & Lee, D., 2024. Driving assistance system with obstacle avoidance for electric wheelchairs. *Sensors,* 24(14), p. 4644.

Ghorbel, M., 2018. A decision-theoretic approach for shared control of a robotic wheelchair. *International Journal of Social Robotics,* 10(1), p. 131–145.

Gupta, A., Chowdhury, K., Farheen, N. & Schoen, M., 2025. Evolution and emerging trends in intelligent wheelchair control: A comprehensive review. *Machines,* 14(1), p. 33.

Haddouna, A., Djabri, D., Saidani, M. & Benbouzid, M., 2025. Development and evaluation of a head-controlled wheelchair system for users with severe motor impairments. *MethodsX,* Volume 15, p. 103485.

Jayasekera, N. et al., 2024. Revolutionizing accessibility: Smart wheelchair robot and mobile application for mobility, assistance, and home management. *Journal of Robotics and Control,* 5(1), p. 1–12.

Ji, Y., Hwang, J. & Kim, E., 2013. An intelligent wheelchair using situation awareness and obstacle detection. *Procedia - Social and Behavioral Sciences,* Volume 97, p. 620–628.

Kargar, S., Yordanov, B., Harvey, C. & Asadipour, A., 2024. Emerging Trends in Realistic Robotic Simulations: A Comprehensive Systematic Literature Review. *IEEE Access.*

Kaur, A. P. et al., 2025. State-of-the-art fall detection techniques with emphasis on floor-based systems — A review. *Biomedical Engineering Advances,* Volume 9, p. 100179.

Kawaguchi, E., Sekiguchi, K. & Nonaka, K., 2023. Self-driving electric wheelchair in crowded environments using a fuzzy potential model predictive. *IFAC-PapersOnLine,* 56(2), p. 11827–11833.

Kundu, A., Mazumder, O., Lenka, P. & Bhaumik, S., 2018. Hand gesture recognition based omnidirectional wheelchair control using IMU and EMG sensors. *Journal of Intelligent and Robotic Systems,* Volume 91, p. 529–541.

Li, R., Oskoei, M., McDonald-Maier, K. & Hu, H., 2013. *ROS based multi-sensor navigation of intelligent wheelchair.* s.l., IEEE, p. 1–6.

Michel, O., 2004. Webots™: Professional Mobile Robot Simulation. *International Journal of Advanced Robotic Systems,* 1(1), p. 39–42.

Ntakolia, C. & Iakovidis, D. K., 2021. A route planning framework for smart wearable assistive navigation systems. *SN Applied Sciences,* Volume 3, p. 104.

Pieniazek, J. & Szaj, W., 2023. Augmented wheelchair control for collision avoidance. *Mechatronics,* Volume 96, p. 103082.

Pu, J. et al., 2018. Low cost sensor network for obstacle avoidance in share-controlled smart wheelchairs under daily scenarios. *Microelectronics Reliability,* p. 180–186.

Sahoo, S. & Choudhury, B., 2023. Voice-activated wheelchair: An affordable solution for individuals with physical disabilities. *Management Science Letters,* Volume 13, p. 175–192.

Sahoo, S. & Choudhury, B., 2024. Autonomous navigation and obstacle avoidance in smart robotic wheelchairs. *Journal of Decision Analytics and Intelligent Computing,* Issue 1, pp. 47-66.

Simpson, R. et al., 2005. A prototype power assist wheelchair that provides for obstacle detection and avoidance for those with visual impairments. *Journal of NeuroEngineering and Rehabilitation,* Volume 2, p. 30.

Szaj, W., Fudali, P., Wojnarowska, W. & Miechowicz, S., 2021. Mechatronic anti-collision system for electric wheelchairs based on 2D LiDAR laser scan. *Sensors,* 21(24), p. 8461.

Vanhooydonck, D. et al., 2010. Adaptable navigational assistance for intelligent wheelchairs by means of an implicit personalized user model. *Robotics and Autonomous Systems,* p. 963–977.

Vermander, P. et al., 2025. Intelligent sitting postural anomaly detection system for wheelchair users with unsupervised techniques. *Digital Communications and Networks,* Volume 11, p. 622–633.

Wang, H. & Ishimatsu, T., 2004. Vision-based navigation for an electric wheelchair using ceiling light landmark. *Journal of Intelligent and Robotic Systems.*

Yan, J., Wang, X., Shi, J. & Hu, S., 2023. Skeleton-Based Fall Detection with Multiple Inertial Sensors Using Spatial-Temporal Graph Convolutional Networks. *Sensors,* 23(4), p. 2153.

Yousuf, S. & Kadri, M. B., 2023. A ubiquitous architecture for wheelchair fall anomaly detection using low-cost embedded sensors and isolation forest algorithm. *Computers and Electrical Engineering,* Volume 105, p. 108518.

Zhang, Y. et al., 2026. Multi-sensor fusion-based intelligent auxiliary system of power wheelchairs for individuals with limbs disabilities: design and implementation. *Measurement,* Volume 257, p. 118573.