VIETNAM NATIONAL UNIVERSITY, HANOI

INTERNATIONAL SCHOOL

GRADUATION PROJECT

PROJECT NAME

Hybrid Navigation System for Mobile Robots: Manual Control and Autonomous Obstacle Detection

Student's name

Truong Gia Huy

Hanoi – Year 2025.

VIETNAM NATIONAL UNIVERSITY, HANOI

INTERNATIONAL SCHOOL

GRADUATION PROJECT

PROJECT NAME

Hybrid Navigation System for Mobile Robots: Manual Control and Autonomous Obstacle Detection

SUPERVISOR: Dr. Pham Ngoc Thanh

STUDENT: Truong Gia Huy

STUDENT ID: 21070439

COHORT: QHQ2021

SUBJECT CODE: INS401401

MAJOR: Information and Computer Engineering

Hanoi – Year 2025

Table of Contents

Abstract

In the modern day, with the rapid development of Internet of Things (IoT) has instant influences across industries, technology, and have variety of applications. A domain where this converge holds great potential is robotics and autonomous vehicles, especially the development of Automation vehicles. This project emphasizes the integration of IoT-based control mechanisms in small-scale autonomous vehicles, demonstrating potential applications in robotics, smart transportation, industrial automation, and education.

This project presents the development of a hybrid navigation system for mobile robot that integrates Bluetooth remote control and autonomous navigation mode using the HC-SR04 ultrasonic sensor. The system is built on the Arduino Uno microcontroller, utilizing the HC-05 Bluetooth module for remote communication and the HC-SR04 sensor for obstacle detection in autonomous mode.

In Bluetooth mode, users can control the vehicle manually using mobile applications, such as smartphones, providing flexibility and manual operation. In autonomous mode, the vehicle uses real-time distance measurements from the ultrasonic sensor to navigate safely and avoid obstacles without human intervention. The results show that combining Bluetooth control and sensor-based navigation enhances flexibility, usability, and efficiency in vehicle operations.

Acknowledgements

I want to express my sincere thanks to my thesis supervisor, Dr. Pham Ngoc Thanh, for his guidance, support and valuable advice throughout the research. He took the time to understand my goals and encouragement for the project, also providing clear directions and expectations.

I am grateful to the faculty and staff at VNU-IS for their support and for providing the essential resources that help me to complete this project.

Finally, I am profoundly thankful to my family and friends for their constant support, understanding, inspiration and encouragement. Their insights, constructive feedback and encouragement have been invaluable throughout this journey. I feel truly fortunate to have them by my side.

Declaration

I hereby declare that this graduation project, titled “Hybrid Navigation System for Mobile Robots: Manual Control and Autonomous Obstacle Detection” is my original work and has been conducted under the guidance of Dr. Pham Ngoc Thanh. To the best of my knowledge, all the information and data presented in this project is true and authentic.

I affirm that all sources of information have been properly acknowledged and cited to ensure academic integrity and to prevent any instances of plagiarism. Furthermore, this project has not been submitted, either in whole or in part, for any other degree or diploma at any university or institution.

I take full responsibility for the content of this project and the research conducted. Any errors or omissions are solely my own.

List of Figures

List of Tables

Chapter 1: Introduction and Overview

I. Research Rationale

The field of mobile robotics is experiencing rapid advancement, with a growing demand for automated systems across various sectors. The global autonomous mobile robot (AMR) market, for instance, is projected to reach USD 3.49 billion by 2023, with a robust compound annual growth rate (CAGR) of 15.3% forecasted from 2024 to 2030 [1]. In Vietnam, this trend is also clearly observable, driven by the nation's rapid industrialization, growing investment in technology, and ambitious digital transformation initiatives that underscore the importance of automation and smart solutions in diverse sectors [2]. This significant growth is primarily driven by the increasing need for automation in manufacturing, logistics, and other industrial environments, where AMRs are instrumental in enhancing operational efficiency, reducing reliance on manual labor, and optimizing complex workflows. The ability of these robots to perform tasks such as picking, transporting, and sorting goods autonomously transforms intralogistics and contributes to intelligent manufacturing facilities [1].

Despite the strong market demand and the crucial role autonomous mobile robotics technology plays in enhancing operational safety and efficiency [3], the development of fully autonomous mobile robots remains a formidable challenge, particularly in unstructured and dynamic environments. Academic research consistently highlights the complexity of achieving sophisticated scene understanding in such settings [4][5]. Robots operating in "open-world scenarios", which is uncertain, dynamic, and complex environments, could face significant difficulties, including adapting to dynamic changes, managing real-time issues caused by massive data collection, and overcoming limited computing resources [4]. Even advanced control strategies like Model Predictive Control (MPC), while powerful for high-precision control, are limited by high computational complexity, model dependency, and sensitivity to noise [3]. In complex scenarios, the performance of purely autonomous systems can remain below that of an average human operator, especially in off-road or highly unpredictable areas [5]. With highly needs in several fields, especially in logistics and transportations, a prototype for hybrid robot development is highlighted.

Conversely, purely remote-controlled systems, while offering direct human control and adaptability for complex tasks, introduce their own set of limitations. These include human factors such as fatigue, the potential for operational errors over extended periods, and inherent inefficiencies when performing repetitive or long-duration tasks. Such systems also lack the scalability and inherent efficiency that autonomous operations can provide. The identified gap between the high demand for automation and the inherent difficulties of deploying purely autonomous systems in complex, real-world settings necessitates alternative or hybrid solutions. A hybrid approach, which combines the strengths of both paradigms, can bridge this divide by allowing the system to leverage human intelligence for complex, unpredictable situations via remote control and switch to autonomous operation for repetitive, well-defined tasks. This integration enhances overall system robustness, safety, and efficiency, aligning with research on hybrid teleoperation architectures designed for safe and flexible remote operation [6] and hybrid locomotion for increased agility and reconfigurability [7].

Furthermore, the accessibility and cost-effectiveness of robotics development platforms have revolutionized the field, making advanced technology attainable for a broader audience. Arduino, with its user-friendly Integrated Development Environment (IDE), low-cost microcontrollers, and an extensive range of sensors and actuators, has become a foundational technology for designing and implementing innovative solutions [8]. This open-source nature fosters collaboration and creativity, enabling enthusiasts and professionals alike to develop interactive objects and automated systems without prohibitive investment [8][9]. Similarly, ultrasonic sensors, specifically the HC-SR04, are widely adopted in robotics due to their reliability, cost-effectiveness, and ability to perform well in diverse conditions, including low-light and dusty environments [10][11]. These sensors are significantly cheaper than laser sensors while offering good accuracy for distance measurement [10]. The availability of such low-cost components democratizes robotics research and development, enabling practical experimentation, rapid prototyping, and fostering innovation within the community [8].

II. Research Motivation

The motivation for this project comes from a applicable need for flexibility in mobile robotic systems. In many real-world scenarios, tasks aren't exclusively autonomous or purely human-controlled; instead, they often demand a blend [6]. This means combining precise human intervention—like navigating tight spaces, handling delicate objects, or responding to unpredictable events—with efficient autonomous execution, such as long-distance travel, repetitive scanning, or operating in hazardous environments. This dual approach is also especially crucial for the current situation in Vietnam, especially in transportation and logistics. Here, a hybrid system is absolutely essential to maximize the benefits of both: an automatic mode for convenience and a manual driving mode for easier navigation of unpredictable circumstances, given the persistently chaotic traffic and incomplete infrastructure. After all, a robotic system confined to a single mode inherently limits its adaptability and utility across diverse applications.

A hybrid robot offers a practical solution to this challenge by providing flexibility to switch between human-controlled and automated navigation. This approach enhances the adaptability and utility of the robot, much like hybrid legged-wheeled locomotion increases speed, agility, and reconfigurability for traversing multiple environments [7]. The schematic design for this project, which clearly depicts both a Bluetooth module for remote control and an ultrasonic sensor for autonomous navigation, signifies a deliberate design choice to enable a human operator to intervene or guide the robot when autonomous capabilities are insufficient, or, conversely, for the robot to operate independently when conditions are suitable, thereby improving overall effectiveness, safety, intuitiveness, and usability.

Furthermore, the choice of Arduino as the central microcontroller is motivated by its open-source nature, user-friendliness, and cost-effectiveness [8][9]. This makes the project highly accessible for students and researchers, fostering practical learning in robotics, electronics, and programming [9]. Robotics is a multidisciplinary field where practical experience is crucial for understanding theoretical concepts. Arduino’s user-friendly design, cost-effective solutions, and strong community support make it an ideal platform for rapid prototyping and for students to acquire and update programming, mechanical, and electronics knowledge and technical skills [8][12]. Similarly, the HC-SR04 ultrasonic sensor, being a low-cost and widely available component, enables the implementation of fundamental autonomous navigation principles without requiring expensive, complex sensor arrays [10][14]. By building a hybrid vehicle with these components, the project serves as an excellent educational tool, bridging theory and practice for emerging engineers. The motivation extends beyond merely building a functional robot; it aims to provide a tangible, hands-on learning experience that demonstrates core robotics principles, problem-solving skills [9], and the integration of hardware and software within an embedded system context, thereby contributing to the advancement of robotics technology by training the next generation of engineers [12].

Finally, this research is motivated by the desire to contribute to the broader field of mobile robotics by exploring the integration of remote control and autonomous navigation on a single, accessible platform. While the development of fully autonomous mobile robots remains challenging [5] and multi-algorithm fusion for path planning often outperforms single-algorithm approaches [3], this project focuses on a more fundamental level of integration: combining two distinct control paradigms rather than complex multi-algorithm path planning. This represents a valuable incremental step in making robots more versatile. The project aims to provide a concrete, experimentally validated demonstration of a hybrid system, offering insights into the practical challenges and benefits of such an architecture, particularly for applications where cost and simplicity are key considerations.

III. Research Statement

This research aims to design and develop a cost-effective hybrid navigation system for mobile robot. Specifically, this involves designing the integrated hardware system and developing the control software that enables the vehicle to operate in two distinct modes: Bluetooth-controlled teleoperation and autonomous obstacle avoidance, which will leverage an HC-SR04 ultrasonic sensor for environmental perception and be built upon an Arduino Uno platform for processing and control.

The project seeks to demonstrate a versatile robotic solution that effectively bridges the gap between direct human control and limited autonomous functionality, providing adaptability for diverse operational environments. Purely autonomous systems face significant challenges in unstructured environments, including difficulties with scene understanding, dynamic adaptability, and real-time processing [4][5]. In such complex scenarios, their performance can fall short of human capabilities [5]. By combining the precision and cognitive adaptability of human teleoperation with the efficiency of automated obstacle avoidance, this work contributes to the development of more adaptable and accessible mobile robotic platforms, particularly for educational, prototyping, and small-scale application contexts. This approach directly responds to the identified challenges of pure autonomy in complex environments and the limitations of pure remote control. The use of Arduino and HC-SR04 emphasizes the cost-effective and accessible nature of the solution, making it practical. The project's significance lies in its pragmatic approach to versatility, offering a viable alternative or complement to highly complex, expensive fully autonomous systems, thereby making advanced robotic capabilities more attainable for a wider range of users and applications. This enhances operational flexibility and safety, aligning with the broader goal of optimizing task execution efficiency and improving operational safety in robotics [3].

IV. Research Aims and Objectives

The primary aim of this research is to design, develop, and comprehensively evaluate a cost-effective and versatile hybrid mobile robotic platform capable of reliable remote control via Bluetooth communication and autonomous navigation utilizing an HC-SR04 ultrasonic sensor. This project fundamentally seeks to demonstrate the feasibility and effectiveness of integrating readily available, microcontroller-based components to achieve functional mobility and real-time environmental awareness in a robotic system. This overarching aim is to be achieved through a set of four closely related and interconnected objectives, meticulously guiding the research and development process.

The first objective is centered on designing and implementing a robust modular hardware and software architecture. This initial phase involves the systematic selection and meticulous integration of key components, specifically an Arduino Uno microcontroller serving as the central processing unit, an L298N motor driver for efficient differential drive control, an HC-SR04 ultrasonic sensor mounted on a servo motor for comprehensive environmental scanning, and an HC-05 Bluetooth module to facilitate wireless communication. Establishing this robust and modular foundation is crucial for ensuring proper power distribution, logical inter-component connectivity, and the physical integrity of the entire robotic system.

Following the architectural design, the second objective transitions into the precise implementation of the physical system and its core functionalities. This involves precisely wiring all chosen hardware components to the Arduino Uno, detailing specific pin assignments. Concurrently, this objective encompasses the comprehensive development and programming of the software functionalities on the Arduino platform. This includes implementing precise motor control algorithms to enable fundamental vehicle movements (such as forward, backward, turns, and diagonal maneuvers) under manual control based on received Bluetooth commands. Furthermore, it involves developing a reactive obstacle avoidance algorithm for autonomous navigation, which meticulously processes real-time sensor data from the HC-SR04 to make flexible decisions on movement and direction, ensuring reliable command reception used the HC-05 Bluetooth module.

The third objective is specifically dedicated to optimizing the overall system performance and validating the reliability of its individual components. This will be accomplished through several key actions, including the fine-tuning of motor control parameters to ensure smooth, responsive, and precise movements in both manual and autonomous modes. Efforts will also be made to minimize latency in both command processing for manual control and reactive decision-making within the autonomous navigation system. Reliability will be further enhanced through rigorous testing of individual components, including verifying the consistent signal reception of the HC-05 Bluetooth module, assessing the accurate rotational performance of the servo motor, and critically, quantitatively evaluating the accuracy of the HC-SR04 ultrasonic sensor across its effective measurement range to confirm its dependability for precise obstacle detection.

Finally, the fourth objective centers on rigorously evaluating the developed system's overall performance under various operational conditions. This comprehensive assessment will involve extensive and systematic testing of both the manual control and autonomous navigation modes across diverse scenarios. For the manual control mode, key performance indicators such as success rates and response delays for all implemented commands will be meticulously measured and analyzed. For the autonomous navigation mode, the system's ability to detect and successfully bypass obstacles in different environmental configurations (e.g., frontal obstacles, side-blocked passages, and complex U-shape blocked scenarios) will be thoroughly tested and quantified. This final evaluation aims to provide empirical evidence of the system's practical value, its technical efficacy, and its tangible contribution to fundamental robotic navigation capabilities.

V. Research Scopes

To ensure a focused and achievable research outcome, the scope of this project is clearly defined, encompassing specific inclusions and acknowledging inherent exclusions and limitations. The project will involve the design and implementation of a four-wheeled ground mobile robot platform, with a primary focus on developing two distinct control modes: Bluetooth remote control for direct human operation and basic autonomous navigation for obstacle avoidance. The core components utilized will exclusively be the Arduino Uno as the microcontroller, an L298N motor driver, an HC-SR04 ultrasonic sensor, and an HC-05 Bluetooth module, all as depicted in the provided schematic. The autonomous navigation capabilities will primarily concentrate on reactive obstacle detection and avoidance within a structured or semi-structured indoor environment. Basic power supply integration for the chosen components will also be included, and the vehicle's performance evaluation will be conducted in controlled indoor settings.

Conversely, this project will not delve into advanced autonomous navigation techniques such as Simultaneous Localization and Mapping (SLAM), global path planning [3], dynamic obstacle prediction [3], or sophisticated AI/machine learning algorithms for complex scene understanding [5]. The focus is specifically on reactive, local obstacle avoidance. The system will not perform semantic scene understanding, meaning it will not distinguish between different types of objects or terrain features [5][14]. The HC-SR04 sensor, while effective for distance measurement, cannot differentiate between shapes or sizes of objects [14]. Furthermore, the autonomous capabilities are not designed for highly unstructured outdoor environments , which present significant challenges due to unpredictable lighting, shadows, occlusions, and varied terrain [5].

Specific limitations related to the chosen sensors and platform include those inherent to the HC-SR04 ultrasonic sensor. This sensor has a limited effective range, typically up to 4 meters [10]. Its performance can be significantly affected by environmental factors such as temperature, humidity, and air pressure, or external interferences like rain, fog, or loud noises [10][14]. Additionally, it may struggle with soft, sound-absorbing materials like fabric or carpet, or angled surfaces that deflect sound waves away from the sensor, or susceptibility to environmental factors leading to inaccurate readings [11][14]. The Bluetooth communication, facilitated by the HC-05 module, also has a limited range, and its performance can be affected by interference from other 2.4 GHz wireless devices. Lastly, the Arduino Uno, while versatile and cost-effective, possesses limited processing power compared to more advanced microcontrollers or computer systems [9]. This constraint restricts the complexity of algorithms that can be implemented and the speed at which data can be processed for real-time advanced navigation tasks. The vehicle is designed solely for mobility and navigation; it does not include capabilities for carrying significant payloads or performing manipulation tasks. While power consumption will be observed during testing, detailed energy optimization strategies are beyond the primary scope of this research. By explicitly stating these limitations, the research demonstrates academic integrity and sets realistic expectations for the project's capabilities, acknowledging that while accessible platforms enable rapid prototyping and fundamental learning, they also define the boundaries of what can be achieved without more advanced hardware and algorithms. This reinforces the cost-effective and educational aspects of the project while clearly delineating its academic contribution within a defined scope.

VI. Research Contents

The hybrid navigation system for mobile robots is comprised of carefully selected hardware components and a robust software architecture, integrated to achieve continuous operation in both remote-controlled and autonomous modes. The synergy of these components is crucial for the functional versatility of the system. The hardware architecture is centered around the Arduino Uno, which serves as the Microcontroller Unit (MCU) and acts as the orchestrator for the entire system. It is responsible for interpreting commands received via Bluetooth, processing data from the ultrasonic sensor, and generating precise control signals for the motors [8][9]. For locomotion, an L298N motor driver module interfaces the Arduino with the four DC motors that provide the vehicle's movement. This driver enables bidirectional movement and precise speed control through Pulse Width Modulation (PWM) signals from the Arduino [9]. The schematic clearly illustrates the connection of four motors to the L298N. Proximity sensing is handled by the HC-SR04 ultrasonic sensor, integrated to detect obstacles in the vehicle's path by emitting high-frequency sound waves and measuring the time it takes for the echoes to return, thereby calculating the distance to objects [10][14]. Wireless communication is facilitated by an HC-05 Bluetooth module, which enables the teleoperation mode by allowing interaction between the vehicle and a remote control device, such as a smartphone or computer. A suitable power source, typically a 9V battery as indicated in the schematic, provides electrical energy to all components, with appropriate voltage regulation for the Arduino and other modules. Finally, a basic mobile robot chassis provides the structural foundation, housing all electronic components and supporting the vehicle's movement.

The software for the vehicle will be developed using the Arduino Integrated Development Environment (IDE) and programmed in C/C++ [9]. Key software modules include a Bluetooth Communication Protocol, which will establish a serial communication link via the HC-05 Bluetooth module, parsing incoming commands (e.g., forward, backward, turn left, turn right, stop, switch mode) and translating them into actionable motor control signals. Motor Control Logic will implement algorithms for precise control of motor speed and direction, utilizing the Pulse Width Modulation (PWM) capabilities of the Arduino [9]. For environmental awareness, Ultrasonic Sensor Data Acquisition and Processing routines will manage the triggering of the HC-SR04 sensor, measure the time-of-flight of the sound waves, and accurately calculate the distance to detected obstacles [14]. An Autonomous navigation algorithm, specifically a reactive algorithm for obstacle avoidance, will be implemented to execute pre-programmed maneuvers (such as stopping, turning, and then resuming forward motion) upon detecting an obstacle within a predefined threshold. Above all, a dedicated Mode Switching Logic module will manage the transition between the Bluetooth remote control and autonomous navigation modes, ensuring a safe and smooth handover of control without unexpected behavior.

The project contents will detail the systematic integration of these hardware and software components, explaining how they interact to form a cohesive hybrid robotic system. This includes comprehensive wiring diagrams, communication protocols between modules, and the overall control flow. The success of the project hinges on this integration, where the Arduino acts as the central orchestrator, translating Bluetooth commands into motor actions and HC-SR04 readings into autonomous decisions. This integration aspect represents a core challenge and a significant learning outcome, demonstrating how disparate hardware modules are unified by embedded software to create versatile and intelligent robotic behavior.

VII. Research Method

The research will adopt an applied, experimental, and iterative design methodology, characteristic of embedded systems development [10]. This approach involves a phased progression from conceptualization to validation, allowing for continuous refinement and optimization of the system. The development of embedded systems, especially in robotics, is rarely a linear process. The importance of testing and refining [12], and the challenges of real-time performance and dynamic defects [14], indicate that a robust methodology must include iterative cycles of testing, analysis, and improvement. This ensures that the project is not merely a one-off build but a systematically developed and validated system.

The phased methodology begins with system design and system architecture, where detailed functional and non-functional requirements for both Bluetooth remote control, such as responsiveness, control range, and autonomous navigation such as obstacle detection range, avoidance strategy will be meticulously defined. Based on these requirements, the hardware architecture, the system block diagram, component layout, and the software architecture, overall program flowchart and algorithm will be designed. The selection of Arduino is justified by its cost-effectiveness and ease of use, which are ideal for rapid prototyping [8][9].

Following the design phase, Hardware prototyping and assembly will commence. The physical vehicle chassis will be constructed, and all selected electronic components—including the Arduino, L298N motor driver, HC-SR04 ultrasonic sensor, HC-05 Bluetooth module, motors, and power supply—will be assembled and wired according to the designed schematic. Emphasis will be placed on proper power distribution and robust physical integration to ensure system stability.

The next critical phase is Software Development and Algorithm Implementation. Individual software modules for Bluetooth communication, motor control, and ultrasonic sensor data processing will be developed using C/C++ within the Arduino IDE. The core algorithms for interpreting remote commands, accurately calculating distances from sensor readings [14], and implementing a basic reactive obstacle avoidance strategy will be coded. Crucially, the logic for mode switching between remote and autonomous operations will be developed and integrated.

This will be followed by Unit and Integration Testing, a phase involving rigorous testing at multiple levels. Unit testing will be performed on each hardware component (e.g., motors, sensors, Bluetooth module) and its corresponding software module independently to verify proper functionality. Subsequently, integration testing will be conducted on the assembled system to verify the interaction between all hardware and software components. This includes assessing the responsiveness of remote control, the accuracy of obstacle detection, and the effectiveness of the autonomous avoidance mechanism. The transition between remote and autonomous modes will be rigorously tested for stability and reliability.

The final phase is Performance Evaluation and Iterative Refinement. The vehicle's performance in both modes will be quantitatively evaluated against the defined objectives. This will involve measuring key performance indicators such as control latency, obstacle avoidance success rate, and operational range. Observed limitations and potential issues, such as environmental interference affecting the HC-SR04 sensor's readings [10], will be thoroughly documented. Based on these testing results, an iterative refinement process will be employed to debug, optimize, and improve both hardware and software aspects of the system, thereby enhancing its robustness and overall functionality [12]. This iterative nature allows for addressing unforeseen challenges inherent in integrating hardware and software, leading to a more reliable and optimized final product, and provides a clear framework for reporting the development process in the thesis.

VIII. Thesis layout

This report mirrors that structure:

- **Chapter 1: Introduction and Overview** – This chapter introduces the project, outlining its motivation, the research statement, its specific aims and objectives, and the research scope.
- **Chapter 2: Literature Review and Background** – This chapter reviews relevant existing technologies and foundational concepts pertinent to the project, including robotics, microcontrollers, sensors, motor drivers, and communication protocols.

- **Chapter 3: Methodology** – This chapter details the system's design, including its overall architecture, the specifications of hardware components, and the high-level software design for both manual control and autonomous navigation modes.
- **Chapter 4: Implementation and programming of functions** – This chapter describes the practical implementation of the system, including detailed hardware interfacing and the programming logic for all functional modules, such as motor control, sensor data processing, Bluetooth communication, and the autonomous navigation algorithm.
- **Chapter 5: Testing and Evaluation of results** – This chapter presents the methodology, experimental setup, and the quantitative results obtained from testing both the manual control and autonomous navigation modes, along with an evaluation of sensor accuracy and a discussion of the findings.
- **Chapter 6: Conclusion and Future developments** – This chapter summarizes the project's achievements, concludes whether the objectives were met, and proposes areas for future work and potential enhancements to the system.

Chapter 2: Literature Review and Background

I. Background

1. Concepts of Autonomous and Remote-controlled robots

In recent years, mobile robotics has become a rapidly growing field with widespread applications across various sectors, including industrial automation, healthcare, agriculture, security, and disaster response [14][15][16][17][18]. As these robots are increasingly deployed in complex, dynamic, and often unstructured environments, the ability to navigate safely and efficiently has emerged as a critical functionality [19]. Navigation in robotics involves intricate processes such as path planning, localization, and real-time decision-making, all of which directly impact the performance and usability of mobile robotic systems.

Traditionally, navigation in mobile robots has been achieved through either manual control or fully autonomous systems. Manual control, often implemented through teleoperation using joysticks or mobile applications, offers precise human oversight and adaptability in challenging environments. However, this approach necessitates continuous human involvement and is susceptible to issues like communication latency or operational errors [20]. On the other hand, autonomous navigation leverages sensors and algorithms to detect obstacles and make navigation decisions without direct human intervention, thereby reducing human workload and enabling scalability. Despite its inherent advantages, full autonomy may encounter failures in unpredictable scenarios, particularly when real-time decision-making is constrained by sensor inaccuracies or incomplete environmental models.

To address these limitations and bridge the gap between purely manual and fully autonomous systems, hybrid navigation systems have been introduced as a middle ground. These systems integrate real-time human control with autonomous obstacle detection and avoidance capabilities, offering enhanced flexibility and robustness. For instance, a robot might navigate autonomously in an open area but can switch to manual control when entering cluttered or sensitive zones. This hybrid approach allows mobile robots to operate more effectively in diverse real-world environments, where conditions can change rapidly and neither manual nor autonomous navigation alone proves sufficient.

The effectiveness of such a hybrid system critically depends on the continous integration of its components and the robust mechanism for switching between control modes. Bluetooth communication, exemplified by modules like the HC-05, provides a reliable and user-friendly interface for manual input. Concurrently, ultrasonic sensors, such as the HC-SR04, offer a cost-effective and responsive solution for proximity-based navigation and obstacle detection. By combining these systems, the mobile robot gains significant versatility, capable of adapting to different use cases and user preferences. This hybrid approach is particularly valuable for educational projects, indoor exploration, or prototype development, where both user control and autonomous behavior are essential functionalities.

2. HC-SR04 Ultrasonic sensor

The HC-SR04 ultrasonic sensor is a widely adopted component in robotics due to its reliability, cost-effectiveness, and ability to perform well in diverse conditions, including low-light and dusty environments [11]. Its fundamental principle of operation involves emitting high-frequency sound waves from a transmitter and measuring the time it takes for the echoes to return to a receiver. By calculating the time difference between the transmitted burst and the received echo, and knowing the speed of sound, the sensor can accurately determine the distance to an object. This process is crucial for robots to make informed decisions and respond appropriately to their surroundings. The HC-SR04 can efficiently measure distances up to 400 cm with a slight tolerance of 3 mm. Despite its advantages, the HC-SR04 has certain limitations. Its performance can be significantly affected by environmental factors such as temperature, humidity, and air pressure, or external interferences like rain, fog, or loud noises. Moreover, it may struggle with soft, sound-absorbing materials like fabric or carpet, or angled surfaces that deflect sound waves away from the sensor, leading to inaccurate readings. The sensor also cannot distinguish between different shapes and sizes of objects, although this limitation can sometimes be mitigated by using multiple sensors [21].

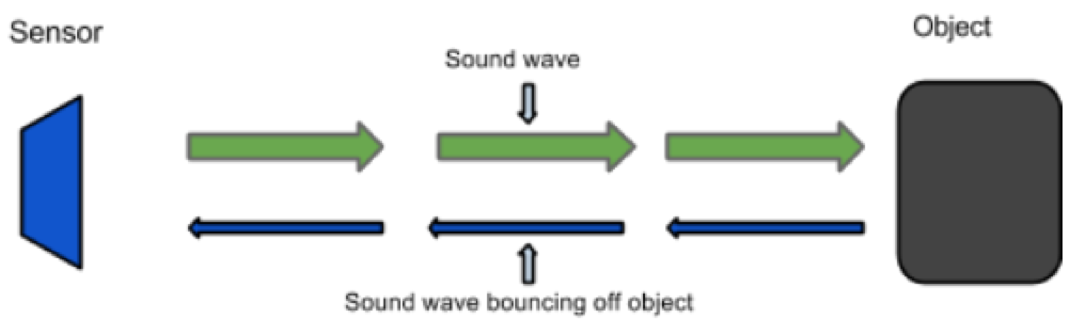

Figure 1: *Working principle of HC-SR04*

The sensor sends out a sound wave at a specific frequency. It then listens for that specific sound wave to bounce off an object and come back. The sensor keeps track of the time between sending the sound wave and the sound wave returning, with the distance as this following equation:

$$\mathbf{d = v \times t}$$

With d is the distance, v stands for velocity, t is time the sound wave bounces off of an object and comes back.

The speed of sound can be calculated based on a variety of atmospheric conditions, including temperature, humidity and pressure. Actually, calculating the distance will be shown later on in this document.

It should be noted that ultrasonic sensors have a cone of detection, the angle of this cone varies with distance, Figure 5 show this relation. The ability of a sensor to detect an object also depends on the objects orientation to the sensor. If an object doesn't present a flat surface to the sensor then it is possible the sound wave will bounce off the object in a way that it does not return to the sensor.

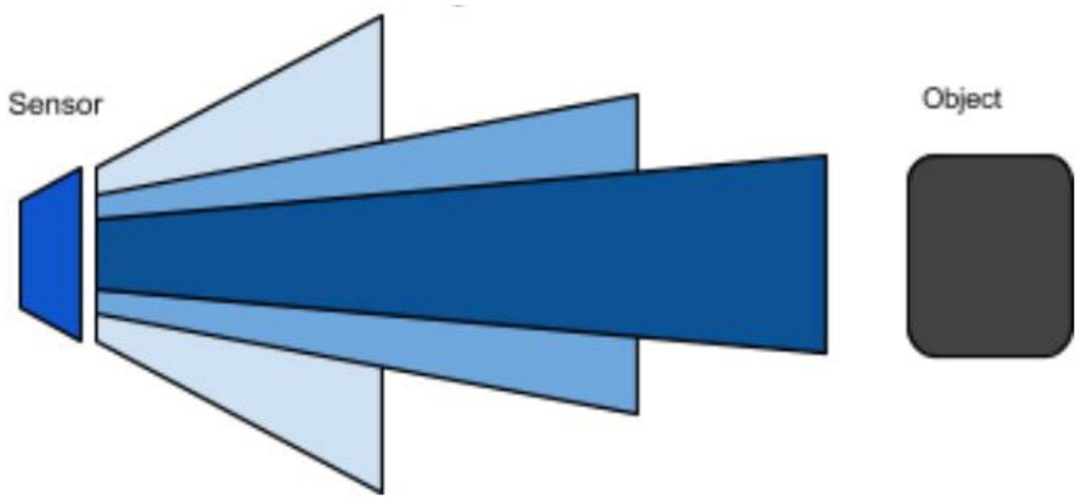

Figure 2: *Ultrasonic sensors' detection cone, the angle of this cone varies with distance*

3. Bluetooth Technology and the HC-05 Module

Bluetooth is a short-range wireless technology standard that facilitates data exchange over short distances, making it highly suitable for various applications, including embedded systems and robotics. It operates in the 2.4 GHz ISM band and is widely used for its convenience and low power consumption. The HC-05 Bluetooth module is a popular choice for Arduino-based projects due to its versatility and ease of integration. It typically operates in a Master/Slave configuration, allowing it to either initiate connections (Master mode) or accept connections from other devices (Slave mode). Communication with the HC-05 module is primarily achieved via Universal Asynchronous Receiver-Transmitter (UART) serial communication, simplifying its interface with microcontrollers like Arduino. The benefits of utilizing Bluetooth for remote control in robotics are substantial, including its widespread compatibility with smartphones and other devices, relatively low cost, and sufficient data rates for transmitting control commands. This makes it an ideal solution for creating user-friendly and accessible remote-controlled robotic systems [22].

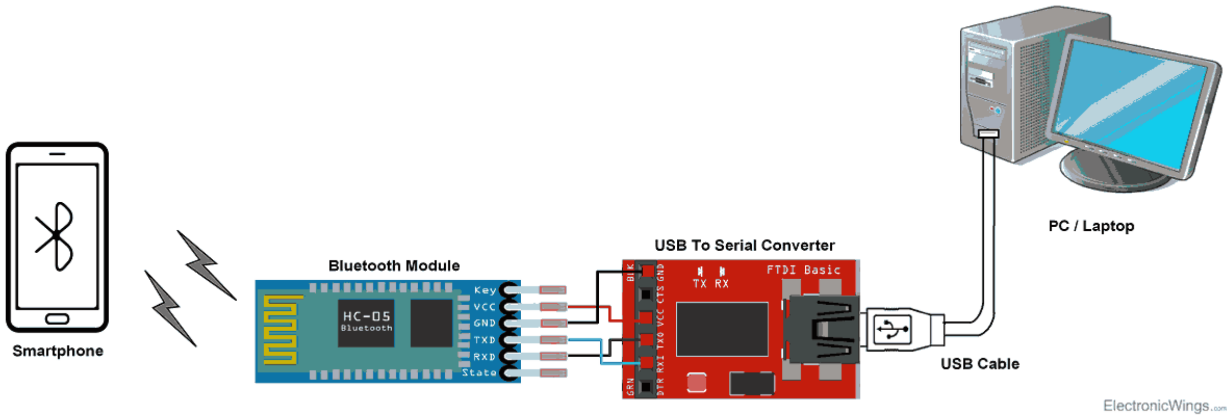

Figure 3: *HC-05 Bluetooth module working principle*

4. Arduino Uno microcontroller platform

Arduino has revolutionized the fields of robotics and automation by making advanced technology accessible to a broad audience, including enthusiasts, hobbyists, and professional [8]. The Arduino Uno, a specific board within the Arduino ecosystem, serves as a foundational technology for designing and implementing innovative solutions due to its user-friendly Integrated Development Environment (IDE), low-cost microcontrollers, and an extensive range of compatible sensors and actuators [8][9]. Its open-source nature fosters collaboration and creativity, enabling users to develop interactive objects and automated systems without prohibitive investment [8]. Arduino's advantages in robotics are numerous, encompassing its user-friendly design, cost-effective solutions, and the strong support provided by a vibrant community, which offers a wealth of online resources, tutorials, and forums [8][9]. This platform provides flexibility in project creation, allowing users to experiment with various components while reducing development costs compared to proprietary systems. Furthermore, Arduino's capability for real-time control and user-friendly programming enables the creation of intricate robotic systems capable of performing a multitude of tasks, from simple movements to complex data acquisition and analysis [8][9].

The Arduino Uno was chosen for this project for its easy-to-use function, real-time control, reliability, and wide compatibility with various electronic components. It can easily integrate with Bluetooth modules, ultrasonic sensors, motor drivers, and relay modules, making it an ideal choice for this hybrid navigation system for mobile robot. Additionally, it is cost-effective, easy to program using the Arduino IDE, and widely supported by the open-source community [8][9].

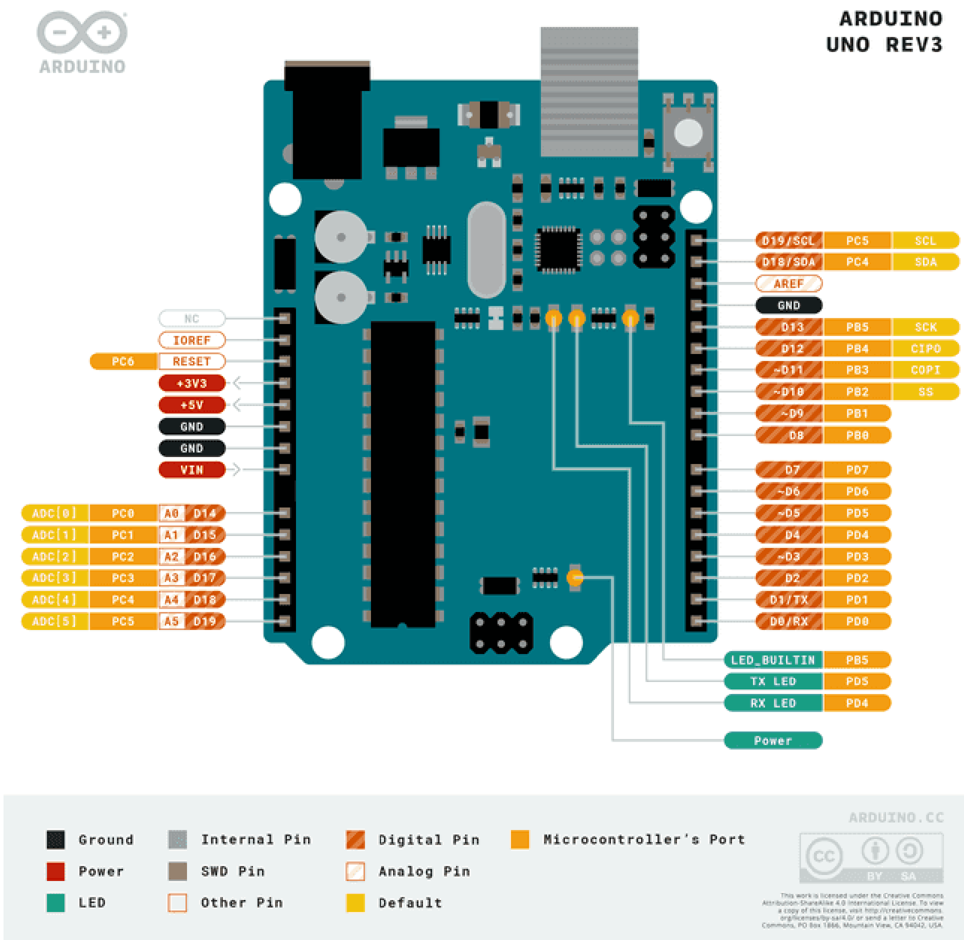

__Figure 4:__ Pinout and connection of Arduino Uno

5. L298N Motor Driver Module

The L298N motor driver module is an essential component for interfacing microcontrollers with DC motors, as microcontrollers typically cannot provide the high current required to drive motors directly. The L298N is based on the L298 integrated circuit, which is a dual full-bridge driver. This configuration, known as an H-bridge, allows for bidirectional control of two independent DC motors or a single stepper motor. The principle of an H-bridge involves a specific arrangement of transistors that enables current to flow in either direction through the motor, thereby controlling its rotation direction. The L298N module accepts low-current control signals from a microcontroller, such as the Arduino, and provides the necessary higher current and voltage to power the motors. It also typically includes enable pins (ENA, ENB) that can be connected to Pulse Width Modulation (PWM) pins on the Arduino, allowing for precise control over the motor speed. This capability makes the L298N a robust and widely used solution for robotic platforms requiring versatile motor control [23].

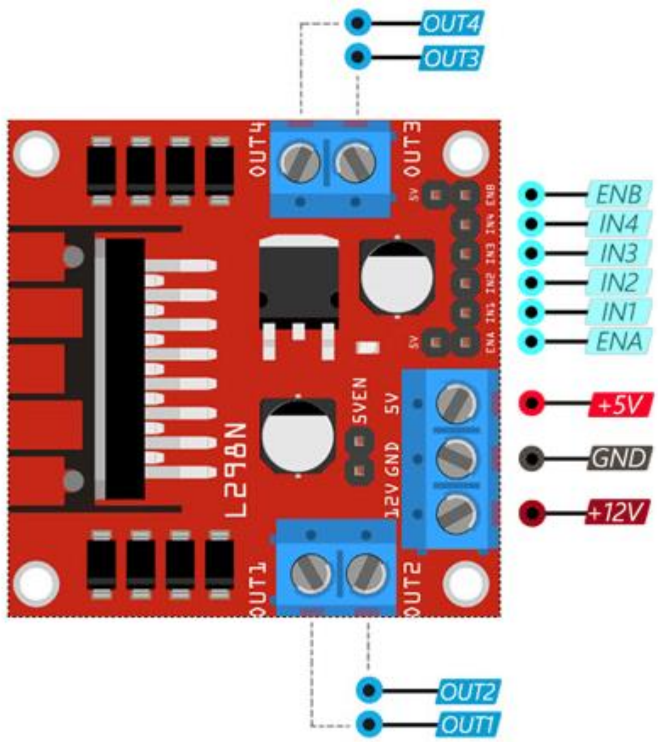

Figure 5: *L298N motor driver module*

II. Literature Review

1. Studies on Autonomous mobile robots and Obstacle avoidance

The field of autonomous mobile robotics has seen extensive research focused on enabling robots to navigate complex environments and avoid collisions [24][25]. Various methods and algorithms have been proposed for obstacle avoidance, broadly categorized into global and local planning strategies. Common reactive obstacle avoidance algorithms include the Potential Field Method [35], which guides robots by constructing virtual attractive and repulsive fields around targets and obstacles respectively. Other approaches include Bug Algorithms [36] and various learning-based methods that may replace or augment stages of the classical navigation pipeline. Many projects have utilized ultrasonic sensors, such as the HC-SR04, for obstacle detection due to their cost-effectiveness and ease of implementation. However, these simpler sensor systems face inherent limitations, particularly in unstructured environments, where challenges such as weak scene understanding, dynamic adaptability, and real-time processing issues persist. The performance of purely autonomous systems in complex scenarios can still fall below that of an average human operator, especially in off-road or highly unpredictable areas. Despite advancements, sophisticated scene understanding remains a significant hurdle for fully autonomous mobile robots in open-world scenarios.

2. Bluetooth-Based Remote control systems

Bluetooth technology has been widely adopted in various applications for remote control, particularly in robotics and Internet of Things (IoT) devices, owing to its wireless capabilities and ease of integration. Numerous studies and projects have demonstrated the use of Bluetooth modules, such as the HC-05, to establish wireless communication links between robotic platforms and control devices like smartphones or dedicated controllers [30]. These systems typically involve developing custom mobile applications or utilizing existing ones (like Dabble, as indicated in the provided code) to send commands to the robot, which are then interpreted by a microcontroller. When compared to other wireless technologies, Bluetooth offers advantages in terms of low power consumption and simplicity for short-range communication, making it suitable for personal area networks and direct device-to-device control [31]. While Wi-Fi offers higher bandwidth and longer range, it often comes with increased power consumption and network complexity. Zigbee, another wireless standard, is known for its mesh networking capabilities and low power, but may be less universally compatible with consumer devices than Bluetooth. Various Arduino libraries, such as SoftwareSerial and Dabble, have been developed to simplify the implementation of Bluetooth communication protocols, enabling developers to easily integrate wireless control into their projects.

3. Hybrid Navigation System for Mobile Robots Project

Hybrid navigation systems, which integrate both manual and autonomous control modes, are increasingly recognized as a robust solution to enhance flexibility and ensure that robots can operate effectively in a variety of environments. Research indicates that integrating both manual and autonomous control modes can lead to more suitable and accurate performance across different operational scenarios [25][26]. These systems are designed to strategically combine the advantages of human intuition and decision-making, which excel in handling unforeseen complexities and nuanced tasks, with the efficiency and consistency of autonomous systems. Several studies have explored the design and implementation of such hybrid systems, demonstrating their effectiveness, versatility, and flexibility in diverse applications [29][32].

While full automation robots are widely utilized in various fields, such as industrial manufacturing [25], their reliability can sometimes be compromised due to the inherent complexity and variability of real-world processes [28]. For instance, in some medical applications, while robotic procedures may offer enhanced precision, they can also be associated with longer operative times or higher complication rates compared to manual methods. To address these limitations, manual control methods remain highly relevant, particularly in situations demanding direct human oversight, real-time adaptability, or intervention in unpredictable circumstances. Among these, Bluetooth-based manual control has been extensively researched for its practicality and user-friendliness in real-time robot navigation. The HC-05 Bluetooth module, in particular, has gained widespread adoption in educational and prototype-level mobile robots due to its low power consumption, simple serial communication interface, and compatibility with Android-based applications [9][39]. Studies such as those by Patil et al. [31], Kraus [32], and Sudrajat and Hidayat [33] have successfully implemented Bluetooth remote control to guide mobile robots in structured environments, underscoring the module's ease of integration and responsiveness for real-time control tasks.

Conversely, purely manual robots are not entirely superior to autonomous robots, as they come with their own set of limitations, including operator fatigue, potential for human error over extended periods, and inefficiency in repetitive tasks. To overcome these drawbacks and enhance flexibility and adaptability, especially for navigating narrow or complex environments, autonomous modes are integrated. For autonomous obstacle detection, ultrasonic sensors like the HC-SR04 are among the most popular solutions due to their affordability, ease of use, and effective performance in short-range detection. The sensor measures distance based on the time-of-flight of sound waves, enabling robots to perceive and avoid obstacles in real-time [27][28]. Prior work by Venkatesh et al. [34] has demonstrated how HC-SR04 could be effectively utilized in autonomous robots to implement basic obstacle-avoidance strategies, particularly in indoor environments [27].

To synthesize the strengths and mitigate the weaknesses of both manual and autonomous control modes, the combination of these two paradigms is implemented in hybrid systems. These systems typically allow the user to drive the robot manually, providing human intuition for general navigation, but also can switch to autonomous mode when the robot encounters complex or narrow places that require precise, automated obstacle avoidance. However, many existing implementations of such hybrid systems may still lack smooth transitions between control modes or suffer from limited decision-making capabilities in highly dynamic scenarios. This observation highlights significant opportunities for further improvement through the development of more intelligent mode switching mechanisms, enhanced sensor fusion techniques, or adaptive behaviors that can better handle dynamic and unpredictable environments. Overall, the project supports the feasibility and effectiveness of hybrid control in mobile robots using low-cost components such as HC-05 and HC-SR04. Nevertheless, further research is needed to refine the integration of control logic, improve robustness in sensor data handling, and optimize the user experience in switching between manual and autonomous modes. The proposed system builds upon these foundations by offering a practical, hybrid mobile robot platform that can be further extended for educational, research, or real-world navigation tasks.

This topic was chosen because it is a topic that few people have done research on, so it could be a breakthrough for semi-automatic robot research. Further research into semi-automatic robots won't just optimize work efficiency across various industries; it could also reshape how we interact with technology. It promises to deliver innovative solutions to problems that fully autonomous or manually controlled robots struggle with, thereby ushering in a new era for robot development in the future.

4. Role of Servo motors in expanding sensor range

The limited field of view of static distance sensors, such as the HC-SR04 ultrasonic sensor, can be a significant constraint for autonomous navigation in complex environments. To overcome this limitation and expand the environmental perception capabilities of mobile robots, servo motors are frequently employed to dynamically orient the sensors. By attaching an ultrasonic sensor to a servo motor, the robot can "scan" its surroundings by rotating the sensor to different angles (e.g., 0°, 90°, 180°), thereby gathering distance data from multiple directions. This active scanning mechanism provides a more comprehensive understanding of the environment, allowing the robot to detect obstacles not directly in its forward path. However, the impact of servo movement on the autonomous system's performance must be carefully considered. Factors such as the time required for the servo to move and stabilize at each angle, the frequency of distance measurements, and the surface of the objects being scanned can affect the real-time responsiveness and the accuracy of the obstacle avoidance algorithm, which are important for maintaining the reliability and efficiency of the autonomous system.

Chapter 3: Methodology

In this chapter, we will dig into the methodology used to develop the hybrid navigation system for mobile robot. This section will describe detail the hardware design and software implementation, from the selection of the main components to the development of the control logic and management of the operating modes. The goal is to provide a comprehensive view of how the system was built, the rationale for the design choices, and the steps taken to achieve the desired functionality of a vehicle that can be both manually controlled via Bluetooth and autonomously driven using ultrasonic sensors.

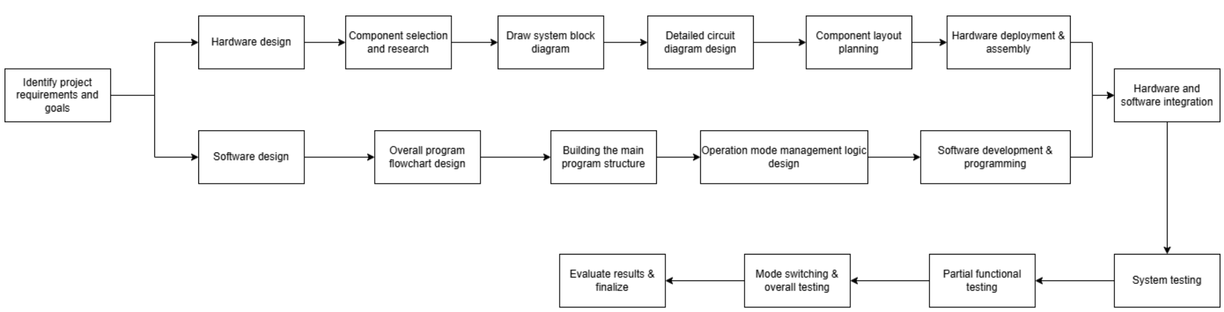

Figure 6: *Methodology steps*

I. Hardware design and Implementation

1. System block diagram

The hybrid navigation robot’s control system is designed around a centralized architecture, with the Arduino Uno microcontroller serving as the primary processing unit. This architecture enables continuous integration of various input sources and control mechanisms. The system's core functionality revolves around two distinct operational modes: manual control via Bluetooth and autonomous navigation using an ultrasonic sensor.

The overall architecture is structured to receive commands from either a human operator via a Bluetooth-enabled device (using the Dabble application or a generic serial Bluetooth terminal) or environmental data from an ultrasonic sensor. These inputs are processed by the Arduino, which then translates them into appropriate control signals for the motor driver, dictating the vehicle's movement. A servo motor facilitates the scanning capability of the ultrasonic sensor, enhancing the autonomous mode's environmental awareness.

The main functional blocks include:

- Input Interface: Consisting of the Bluetooth Module (for remote commands) and the Ultrasonic Sensor (for environmental data).
- Processing Unit: The Arduino Uno microcontroller, responsible for interpreting inputs, executing control logic, and generating output signals.
- Actuation System: Comprising the L298N Motor Driver and the DC Motors, responsible for the vehicle's movement.
- Sensing Mechanism: The HC-SR04 Ultrasonic Sensor mounted on a Servo Motor for dynamic distance measurement.

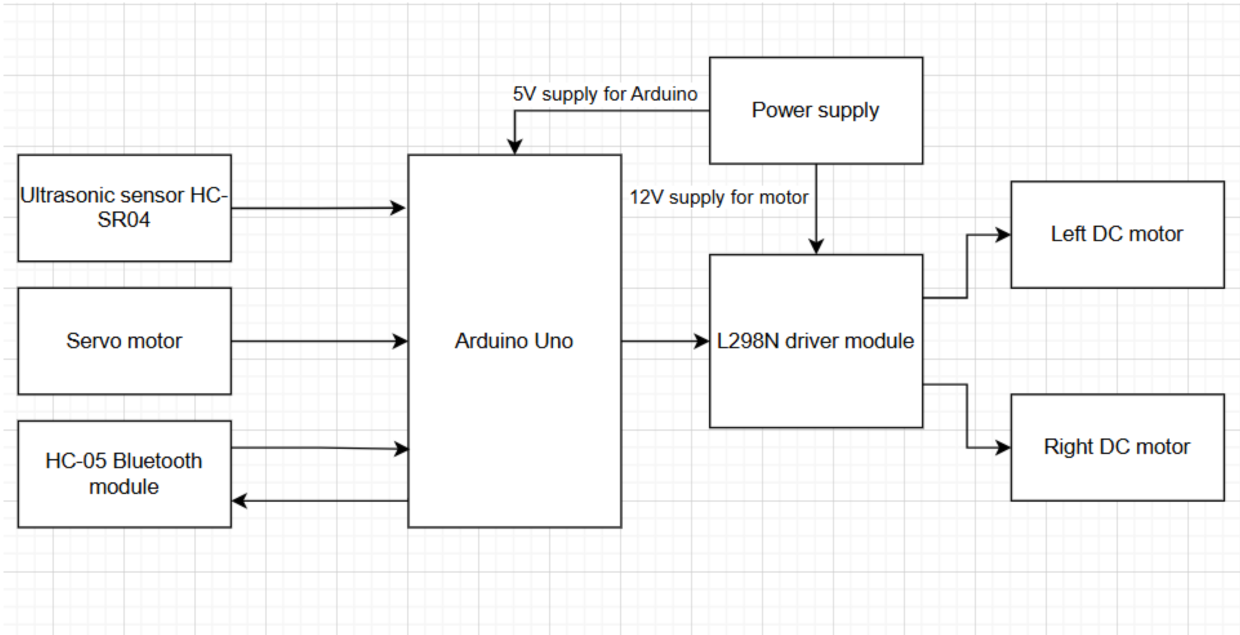

Figure 7: *System block diagram*

2. Detailed circuit schematic diagram

The precise electrical interconnections of the vehicle's control system are thoroughly documented in the detailed circuit schematic diagram, provided as Figure 3.3. This diagram outlines how each component is wired to the Arduino Uno, ensuring proper signal routing and power distribution. The Arduino Uno's digital and analog pins are strategically utilized to interface with various modules. The HC-05 Bluetooth module is connected to Arduino Digital Pins 2 (RX) and 3 (TX). This choice leverages the SoftwareSerial library, a crucial design decision that frees up the hardware serial pins (0 and 1) for simultaneous use in code uploading and debugging via the Serial Monitor. The Bluetooth module operates reliably at a standard baud rate of 9600.

For motor control, the L298N motor driver module is interfaced with Arduino Digital Pins 5, 6, 9, and 10. Specifically, pins D5 (motorA1) and D6 (motorA2) manage one pair of motors (e.g., the left side), while pins D9 (motorB1) and D10 (motorB2) control the other pair (e.g., the right side). A notable design aspect is that the ENA and ENB pins of the L298N are permanently tied to +5V, as indicated in the schematic. This configuration simplifies the control logic by allowing speed modulation to be directly applied to the INx pins (D5, D6, D9, D10) using Arduino's analogWrite() function, assuming the

L298N can interpret PWM signals on these input pins when its enable pins are continuously high.

The HC-SR04 ultrasonic sensor is connected with its Trig pin to Arduino Digital Pin 11 (ULTRASONIC_SENSOR_TRIG) and its Echo pin to Digital Pin 13 (ULTRASONIC_SENSOR_ECHO). The servo motor, integral for dynamic sensor scanning, is connected to Arduino Digital Pin 7 (SERVO_PIN). Additionally, Digital Pin 12 (relay) is connected to a relay module, providing the capability to remotely control an auxiliary device, such as a light or horn, via Bluetooth commands. Digital Pin 8 (BTState) serves as an input, though its explicit use for connection status monitoring is dependent on external circuitry or further code implementation. Power distribution is managed by an external 12V supply for the L298N and motors, while the Arduino provides regulated 5V to the Bluetooth module, ultrasonic sensor, and servo. All components is connected to a common ground to maintain electrical stability and prevent signal integrity issues.

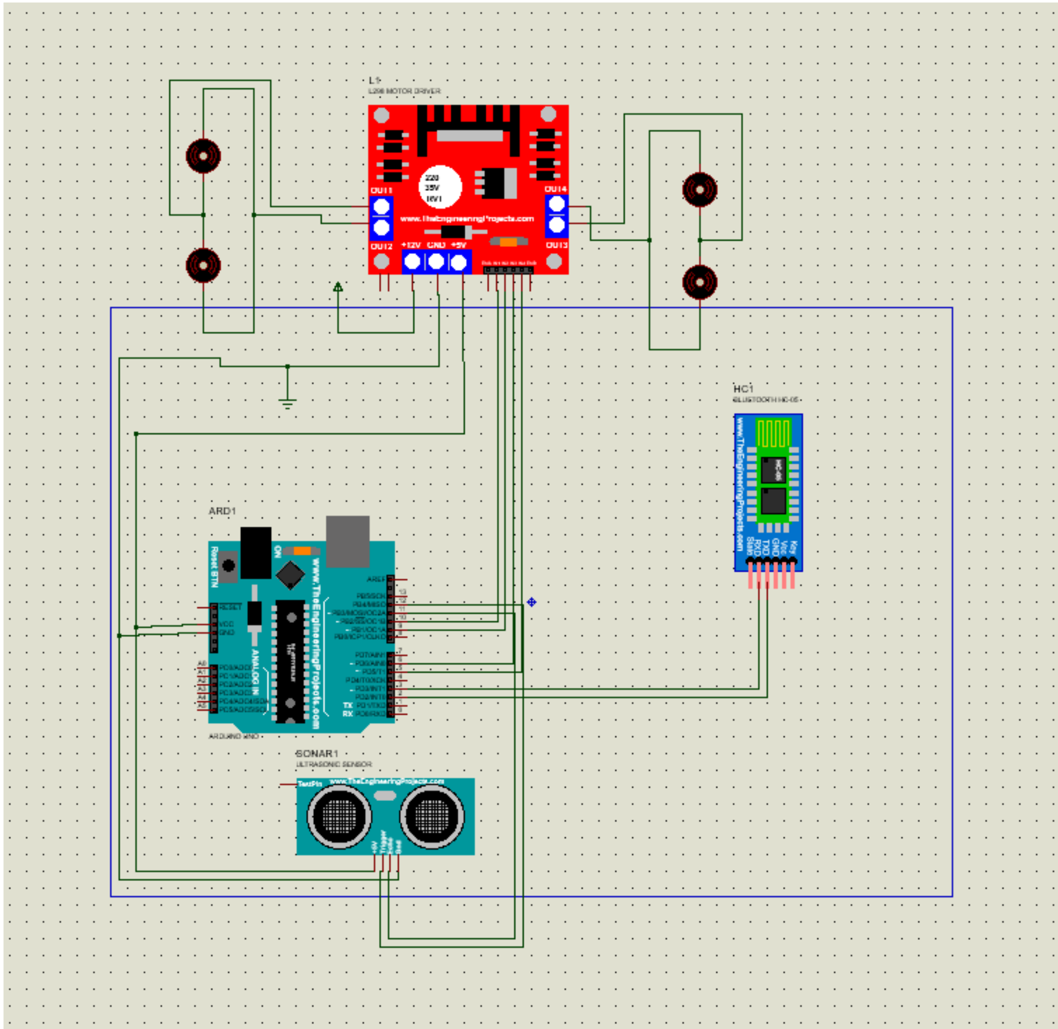

Figure 8: *System design for Hybrid Navigation System for Mobile Robots: Manual Control and Autonomous Obstacle Detection*

3. Component layout on the Vehicle chassis

The physical arrangement of components on the vehicle's chassis is carefully planned to optimize stability, functionality, and ease of maintenance. The vehicle utilizes a robust four-wheel mobile platform, providing a stable base for all integrated systems. The Arduino Uno is strategically positioned at the center of the chassis, facilitating efficient wire routing to various peripherals and contributing to the overall weight balance. The L298N motor driver module is mounted in close proximity to the DC motors, typically at the rear or side, to minimize cable length for high-current paths, thereby reducing voltage drop and potential signal degradation. The DC motors themselves are integrated with the wheels at the four corners of the chassis, providing balanced propulsion. The battery pack, serving as the primary power source, is placed at the bottom-center of the chassis. This arrangement ensures a low center of gravity, which is essential for enhancing vehicle stability and preventing accidental tipping during maneuvers. The HC-SR04 ultrasonic sensor and its accompanying servo motor are co-located and prominently mounted at the front of the vehicle. This forward-facing placement is crucial for effective obstacle detection and allows the servo to sweep the sensor across a wide field of view (0°, 90°, 180° for left, front and right side, respectively), providing comprehensive environmental awareness during autonomous operation. The Bluetooth module is positioned in an easily accessible area, ideally away from potential sources of electromagnetic interference, to guarantee reliable wireless communication. Finally, all wiring is meticulously bundled and secured to prevent entanglement with moving parts, reduce clutter, and maintain a clean, professional appearance of the robot.

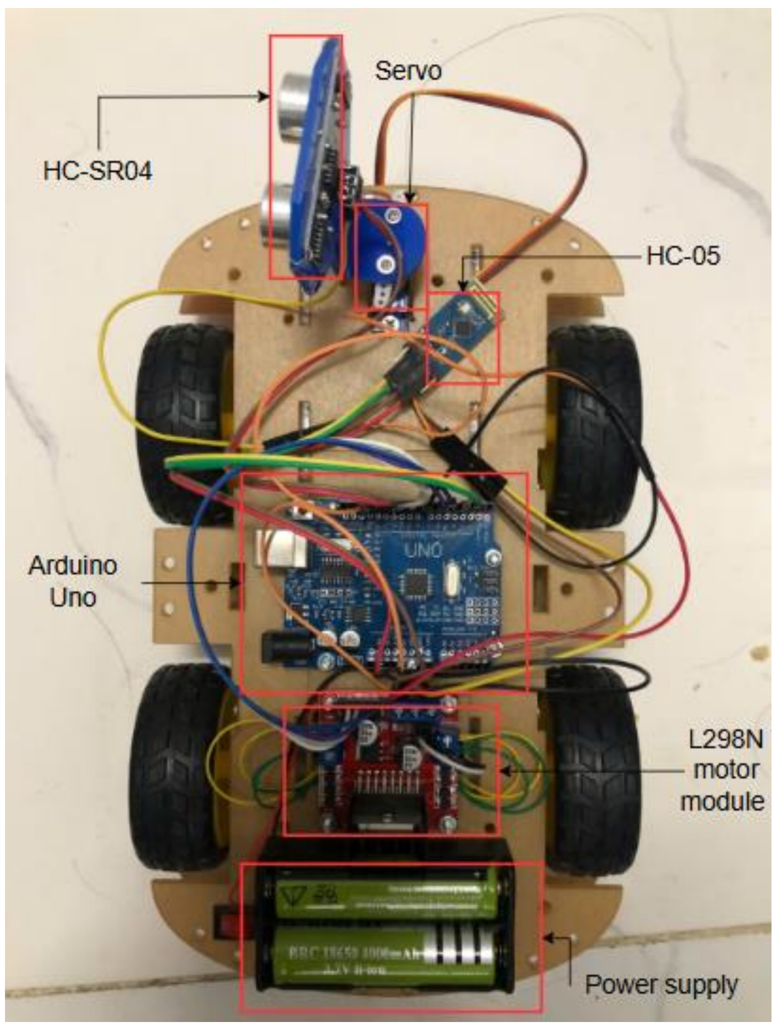

Figure 9: *Component layout on the Vehicle chassis*

4. Recommendation and discussion

The selection of hardware components was guided by a balance of functionality, cost-effectiveness, and ease of integration, ensuring robust performance for the hybrid vehicle.

The Arduino Uno microcontroller was chosen as the central processing unit due to its widespread availability, extensive community support, and straightforward programming environment. Its ample digital and analog pins, coupled with a vast ecosystem of libraries, significantly streamlined the development process and simplified power management with its 5V operating voltage. While more powerful alternatives like ESP32 or Raspberry Pi were considered, their increased complexity and higher cost were deemed unnecessary for the project's scope, as the Arduino Uno provided sufficient capabilities for real-time control without the overhead of an operating system or integrated wireless features beyond what the HC-05 could offer.

For wireless communication, the HC-05 Bluetooth module was selected for its reliability, cost-effectiveness, and seamless integration with Arduino. Supporting the Serial Port Profile (SPP), it effectively handles the transmission of simple text commands from a smartphone, which is crucial for manual control.

The L298N motor driver was chosen to control the DC motors due to its capacity to independently drive two motors with both direction and speed control. Its ability to handle currents up to 2A per channel and a wide input voltage range (up to 35V) made it fully compatible with our 12V motor setup. The L298N's straightforward interface with Arduino simplified the motor control logic, offering a superior balance of power handling, cost, and ease of use.

For obstacle detection, the HC-SR04 ultrasonic sensor was the preferred choice due to its accuracy, reliability, and cost-effectiveness. It provides a non-contact method for distance measurement, essential for autonomous navigation, and its simple two-pin interface with Arduino, complemented by robust libraries like NewPing, ensured seamless integration and reliable performance.

Finally, a small servo motor was incorporated to enable dynamic scanning of the surroundings by the ultrasonic sensor. This allowed the vehicle to detect obstacles across a wider field of view (left, front, and right), significantly enhancing its environmental awareness during autonomous operation. The ease of control offered by the Arduino Servo library for precise angular positioning made it an ideal solution, effectively balancing functionality and simplicity over static sensor mounting or the added complexity of multiple sensors.

II. Software design and Implementation

1. Overall program flowchart

The program's execution sequence begins in the setup() function, where all necessary hardware pins are initialized, serial communication channels (both hardware for debugging and software for Bluetooth) are established at 9600 baud, and the servo motor is attached and set to its initial centered position. Following this initialization phase, the program enters the continuous loop() function, which acts as the central decision-making and control hub.

Within the loop(), the program prioritizes input from the SoftwareSerial port connected to the Bluetooth module. If incoming data is detected on this port, it signifies a direct command from a Bluetooth terminal, and the handleBluetooth() function is invoked to process this input. This design ensures immediate response to explicit Bluetooth commands. If no direct Bluetooth data is available, the program then proceeds to process input from the Dabble application's GamePad module via Dabble.processInput(). Based on specific button presses from the GamePad (e.g., the "Triangle" button to enable autonomous mode, the "Cross" button to disable it), the autoMode boolean flag is updated. Finally, the program evaluates the state of the autoMode flag. If autoMode is true, the runAutonomousMode() function is executed to manage autonomous navigation; otherwise, the handleManualControl() function is called to process manual control commands from the GamePad. This structured approach guarantees that only one operational mode is active at any given time, preventing potential conflicts in control signals.

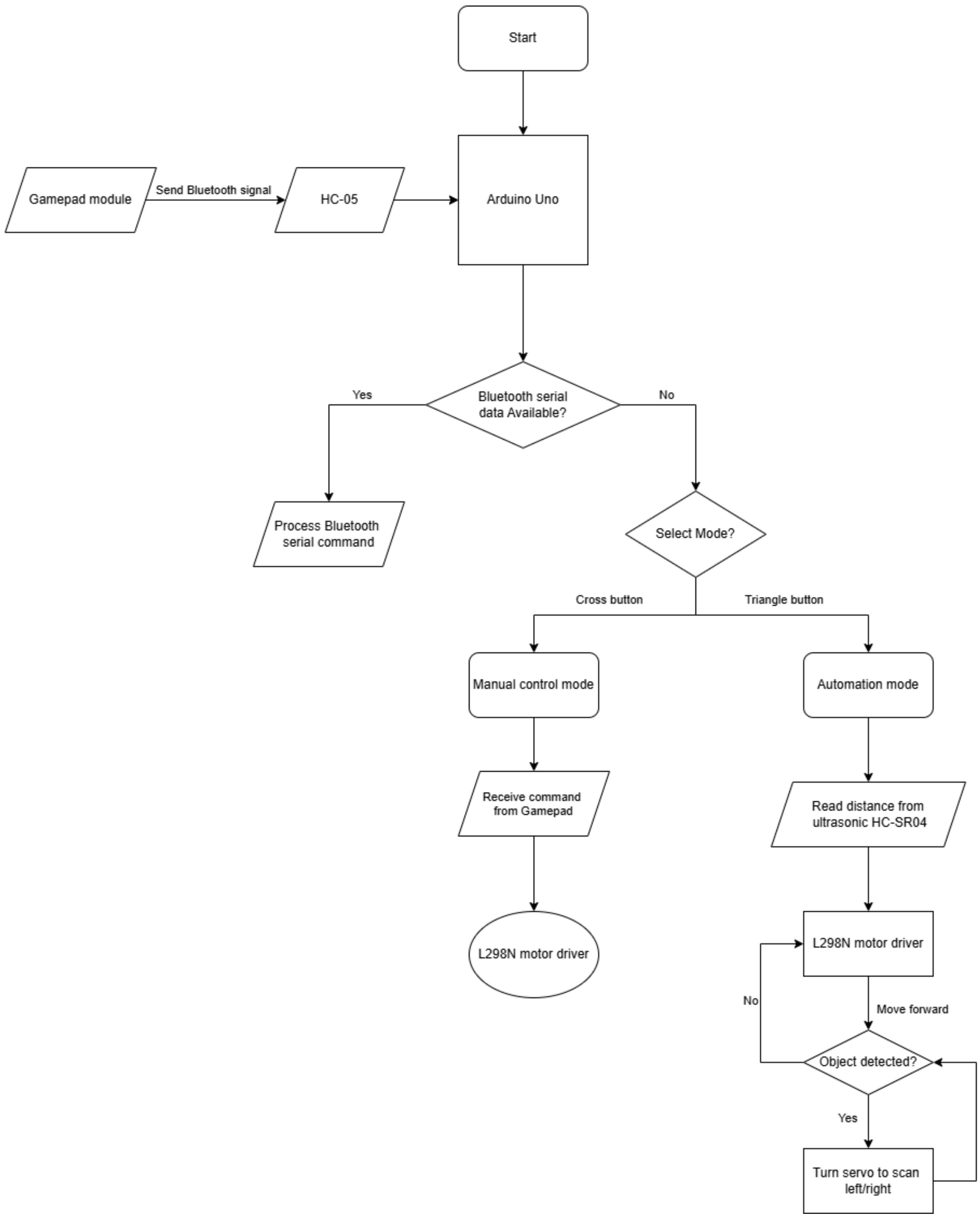

Figure 10: *Program flowchart*

2. Main program structure

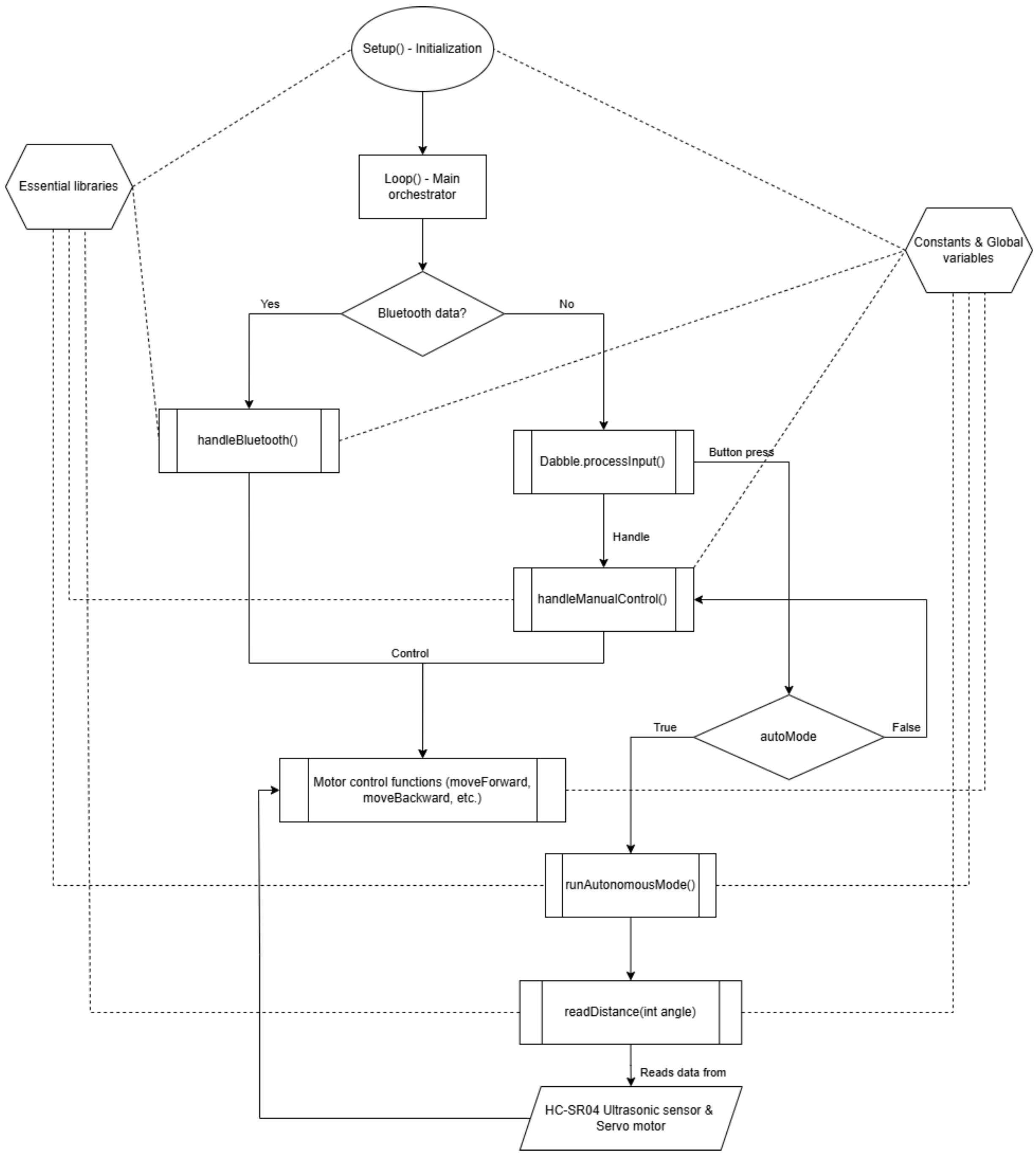

Figure 11: *Main program structure*

This diagram illustrates the interconnections and data flow between the primary functions and modules of the Arduino code. The Arduino code is organized into a modular and hierarchical structure, enhancing readability, maintainability, and reusability of functional blocks. The program starts by including essential libraries:

Dabble.h for simplified GamePad communication with the Dabble app, SoftwareSerial.h to enable serial communication on custom digital pins for the Bluetooth module, NewPing.h for efficient and accurate readings from the HC-SR04 ultrasonic sensor, and Servo.h for precise control of the servo motor. Global constants are defined for all hardware pin assignments, including those for the ultrasonic sensor (ULTRASONIC_SENSOR_TRIG, ULTRASONIC_SENSOR_ECHO), the servo (SERVO_PIN), the motor control pins (motorA1 through motorB2), the relay (relay), and the Bluetooth state pin (BTState). Critical operational parameters such as SAFE_DISTANCE (50 cm for obstacle avoidance) and MAX_DISTANCE (400 cm for sensor range) are also defined as constants. Key global variables, including speed (default 200 for motor PWM), state (to store incoming Bluetooth commands), and autoMode (a boolean flag to track the current operating mode), are declared to facilitate data sharing across different functions.

The setup() function is responsible for all initial configurations, including setting pin modes for inputs and outputs, initializing Serial communication at 9600 baud for debugging, and mySerial (SoftwareSerial) at 9600 baud for Bluetooth. It also initiates the Dabble communication, attaches the servo motor to its designated pin, and centers it to 90 degrees, finally calling stopMotors() to ensure the vehicle is stationary upon startup. The loop() function serves as the central orchestrator, continuously checking for inputs, managing the autoMode flag, and invoking the appropriate control functions based on the current mode.

Specialized functions handle specific aspects of the vehicle's operation. runAutonomousMode() contains the core logic for self-navigation, periodically reading distances from the ultrasonic sensor at various angles (front, left, right) and making intelligent decisions for movement (forward, turn, backward) to avoid obstacles. The readDistance(int angle) helper function facilitates this by moving the servo to a specified angle, allowing a brief delay for stability, and then using the NewPing library to measure and return the distance, defaulting to MAX_DISTANCE if no object is detected. handleManualControl() processes button presses from the Dabble GamePad, translating them into specific motor commands for forward, backward, turning, and diagonal movements by adjusting PWM values. handleBluetooth(String command) parses string commands received via the SoftwareSerial port, mapping specific characters (e.g., "F", "B", "L", "R", "S") to corresponding motor actions and also controlling the auxiliary relay. A set of dedicated motor control functions (moveForward, moveBackward, turnRight, turnLeft, etc.) to utilize analogWrite() to send signals to the L298N motor driver, effectively controlling both the speed and direction of the DC motors.

3. Management of Operating modes

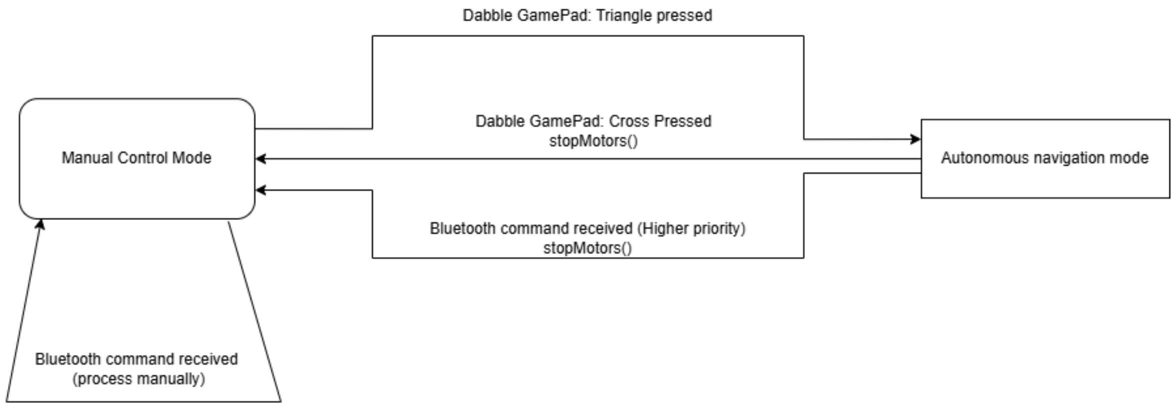

Figure 12: *The transition between manual and automatic control modes*

The reliable management of operating modes is a core of the hybrid navigation robot's software design, ensuring a uninterrupted and controlled transition between manual Bluetooth control and autonomous navigation. This key feature is primarily controlled by the autoMode boolean flag, which serves as a state indicator for the vehicle's operational status.

The autoMode flag is automatically controlled through specific inputs from the Dabble GamePad interface. For instance, pressing the "Triangle" button (GamePad.isTrianglePressed()) explicitly sets autoMode to true, thereby activating the autonomous navigation mode. Conversely, pressing the "Cross" button (GamePad.isCrossPressed()) sets autoMode to false, effectively deactivating the autonomous mode and reverting control back to the manual mode. A crucial safety measure implemented during this transition is the immediate invocation of stopMotors() when exiting autonomous mode. This ensures the vehicle comes to a complete halt before any manual commands are processed, mitigating the risk of unintended movements.

Furthermore, the system incorporates an input prioritization scheme. Commands received directly via the mySerial (Bluetooth) port are given precedence. If mySerial.available() evaluates to true, indicating incoming Bluetooth data, the handleBluetooth() function is immediately executed. This design choice ensures that explicit, direct commands from a Bluetooth terminal (or a custom Bluetooth application) can override any ongoing Dabble GamePad input processing, providing a direct and immediate control mechanism even when Dabble is active. This layered control structure, combined with clear mode activation/deactivation triggers, contributes to a highly predictable and user-friendly operational experience for the hybrid robot.

4. Recommendation and discussion

The software design prioritizes simplicity, rapid development, and intuitive control, achieved through careful selection of programming environment and libraries.

The Arduino IDE and its C-based programming language were chosen for their ease-to-access, ease-to-use and variety of references for embedded systems, and extensive library ecosystem. This environment significantly accelerated development by simplifying sensor interaction, motor control, and serial communication.

To enhance user interaction for manual control, the Dabble.h library was integrated. Dabble provides a user-friendly and customizable GamePad interface on a smartphone, offering intuitive control over vehicle movements and seamless mode switching between manual and autonomous operation. This approach significantly improved the user experience compared to relying on raw serial commands or undertaking the extensive effort of custom mobile application development.

The SoftwareSerial.h library proved crucial for enabling reliable communication with the HC-05 Bluetooth module. By creating a software serial port on dedicated digital pins, it effectively freed up the hardware serial pins for concurrent debugging via the Serial Monitor. This dual-channel capability was invaluable during development, allowing for real-time monitoring of data and system status without the inconvenience of constantly disconnecting the Bluetooth module.

For accurate obstacle detection, the NewPing.h library was selected for its efficiency, precision, and robust error handling when interacting with the HC-SR04 ultrasonic sensor. This library simplifies the process of distance measurement, including important timeout mechanisms that prevent program crashé, thereby improving the overall reliability of the autonomous navigation system.

Finally, the Servo.h library provided a simple but effective method for controlling the servo motor, allowing precise positioning of the ultrasonic sensor. This help scanning capability expanded the vehicle's environmental awareness for autonomous mode, ensured accurate angular control.

Chapter 4: Implementation and programming of functions

I. Bluetooth remote control mode

The Bluetooth remote control mode facilitates user interaction with the vehicle via a smartphone application, primarily using the Dabble application's GamePad interface or a generic serial Bluetooth terminal. Communication between the smartphone and the Arduino Uno is established through the HC-05 Bluetooth module, which is configured to communicate with the Arduino using a SoftwareSerial instance (mySerial) on digital pins D2 (RX) and D3 (TX). This setup is crucial as it preserves the hardware serial port (pins 0 and 1) for debugging and code uploads, preventing conflicts. The mySerial object is initialized at a baud rate of 9600, ensuring reliable data exchange.

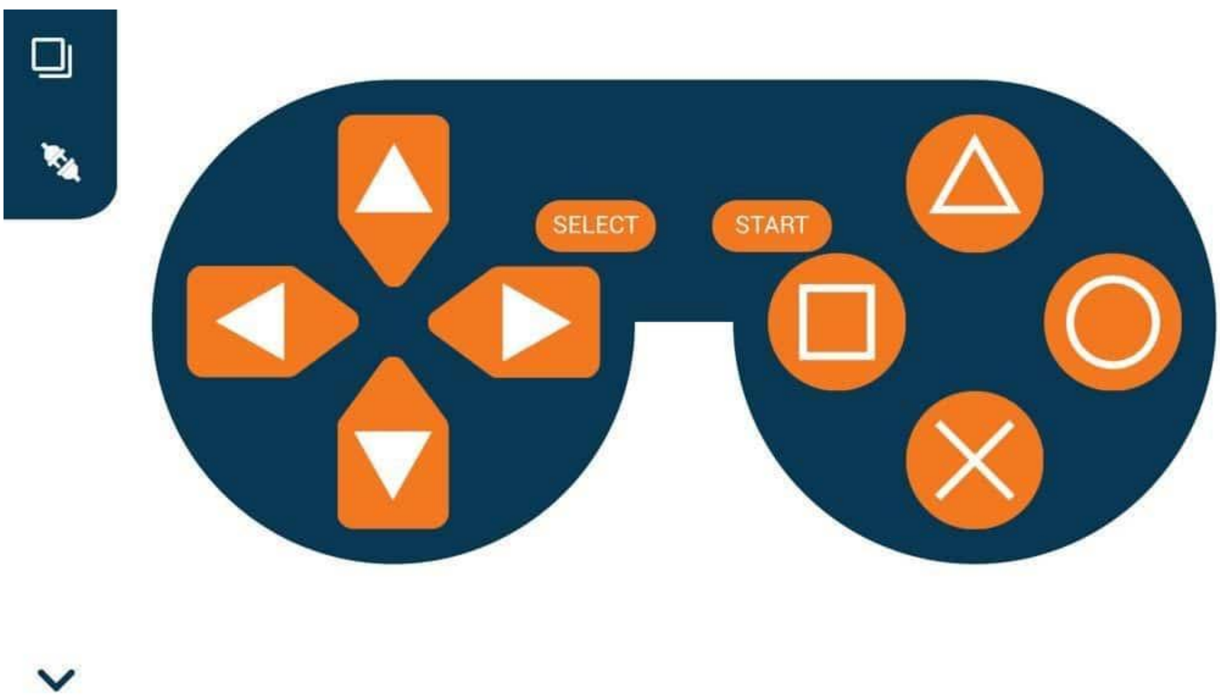

Figure 13: *Dabble gamepad interface*

The implementation handles two primary types of Bluetooth input. Firstly, raw string commands can be sent directly via a Bluetooth serial terminal. The handleBluetooth() function continuously monitors mySerial.available() for incoming data. Upon receiving a complete string command (terminated by a newline character), it is processed to identify specific single-character instructions (e.g., "F" for Forward, "B" for Backward, "L" for Turn Left, "R" for Turn Right, "S" for Stop, and diagonal movements like "FL", "FR", "BL", "BR"). These commands are then mapped directly to corresponding motor control functions. Additionally, the handleBluetooth() function includes logic to control an auxiliary relay (connected to D12) based on "X" (ON) and "x" (OFF) commands.

```
void handleBluetooth(String command) {
  if (command == "F") moveForward(speed);
  else if (command == "B") moveBackward(speed);
  else if (command == "L") turnLeft(speed);
  else if (command == "R") turnRight(speed);
  else if (command == "FL") moveForwardLeft(speed);
  else if (command == "FR") moveForwardRight(speed);
  else if (command == "BL") moveBackwardLeft(speed);
  else if (command == "BR") moveBackwardRight(speed);
  else if (command == "S") stopMotors();
  else if (command == "X") digitalWrite(relay, HIGH);
  else if (command == "x") digitalWrite(relay, LOW);
}
```

***Figure 14:** handleBluetooth() function*

Secondly, the system integrates with the Dabble application, which simplifies gamepad-style control. The Dabble.processInput() function is continuously called within the main loop to parse data received from the Dabble app. The handleManualControl() function then interprets button presses from Dabble's GamePad module (e.g., GamePad.isUpPressed(), GamePad.isLeftPressed()). Each button corresponds to a specific movement action, such as moveForward(), turnLeft(), or stopMotors(). The motor control functions themselves (moveForward, moveBackward, turnRight, turnLeft, moveForwardLeft, moveForwardRight, moveBackwardLeft, moveBackwardRight, stopMotors) are implemented using analogWrite() to control the speed and direction of the DC motors. By applying PWM to the L298N's INx pins (D5, D6, D9, D10) while its ENA/ENB pins are held high, precise speed regulation is achieved.

```
void handleManualControl() {
  if (GamePad.isUpPressed() && GamePad.isLeftPressed()) {
    moveForwardLeft(255);
  }
  else if (GamePad.isUpPressed() && GamePad.isRightPressed()) {
    moveForwardRight(255);
  }
  else if (GamePad.isDownPressed() && GamePad.isLeftPressed()) {
    moveBackwardLeft(255);
  }
  else if (GamePad.isDownPressed() && GamePad.isRightPressed()) {
    moveBackwardRight(255);
  }
  else if (GamePad.isUpPressed()) {
    moveForward(255);
  }
  else if (GamePad.isDownPressed()) {
    moveBackward(255);
  }
  else if (GamePad.isLeftPressed()) {
    turnLeft(180);
  }
  else if (GamePad.isRightPressed()) {
    turnRight(180);
  }
  else {
    stopMotors();
  }
}
```

***Figure 15**: handleManualControl() function*

II. Autonomous navigation mode with Ultrasonic sensor

Data collection from HC-SR04 Ultrasonic sensor

Environmental perception is an important component of autonomous navigation, primarily achieved through the HC-SR04 ultrasonic sensor. The implementation utilizes the NewPing library, which simplifies the interaction with the sensor and provides robust distance measurements. The HC-SR04 sensor is connected to the Arduino Uno with its Trig pin on Digital Pin 11 and Echo pin on Digital Pin 13. Within the runAutonomousMode() function, distance measurements are periodically initiated, ensuring that the system continuously updates its understanding of the surrounding environment. This periodic check is governed by a checkInterval (e.g., 200 ms), managed by the millis() function, to prevent excessive rapid readings that could lead to erratic behavior or waste processing resources. The primary measurement, distanceFront, is obtained by reading the sensor when it is directed straight ahead, providing real-time data on immediate obstacles.

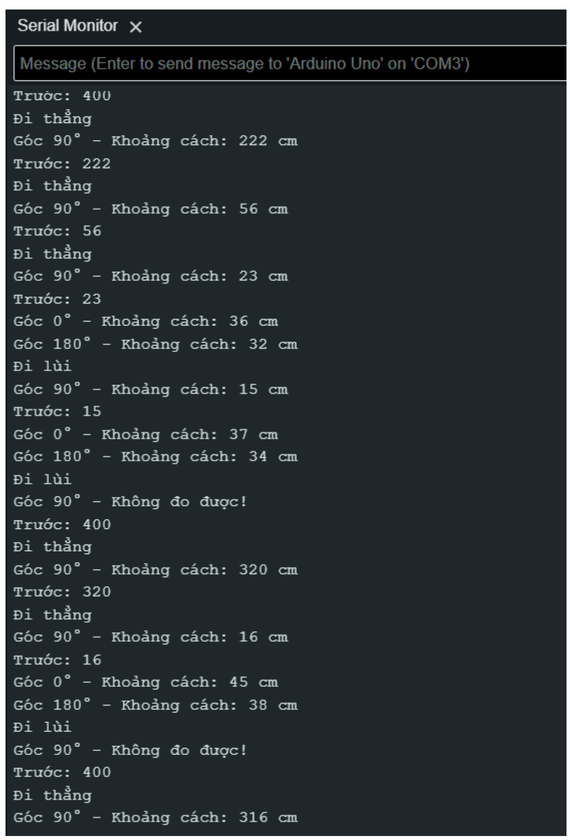

Figure 16: *Data Collected from HC-SR04 Ultrasonic Sensor*

Servo motor rotation for Sensor scanning

To enhance the vehicle's spatial awareness beyond just the frontal view, the HC-SR04 ultrasonic sensor is strategically mounted on a servo motor. This integration allows the sensor to dynamically scan the environment by rotating to different angles, providing a broader perception of potential clear paths or hidden obstacles. The servo motor is interfaced with Arduino Digital Pin 7. The readDistance(int angle) helper function is specifically implemented to manage this scanning process. When called, this function first commands the servo to rotate to the specified angle (e.g., 0° for left, 90° for front, 180° for right). A brief delay (e.g., 500 ms) is then introduced to allow the servo to physically stabilize at the new position before the ultrasonic sensor takes a distance reading. This meticulous timing ensures that the distance measurements correspond accurately to the intended scanning direction, providing reliable data for decision-making.

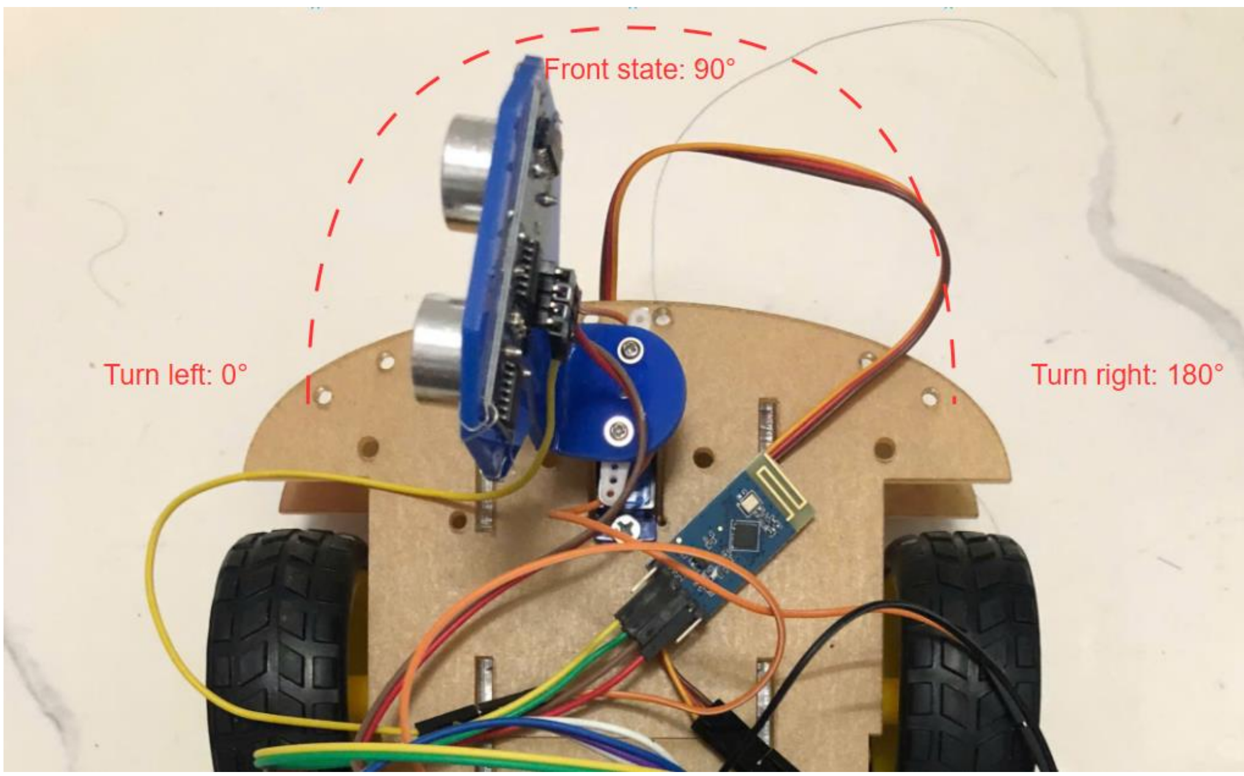

Figure 17: *Servo motor rotation for sensor scanning*

Obstacle avoidance algorithm

The efficacious operation of the autonomous navigation mode hinges upon a sophisticated obstacle avoidance algorithm, meticulously embedded within the runAutonomousMode() function to facilitate intelligent decision-making predicated on real-time sensor data. This algorithm systematically guides the vehicle through its environment, prioritizing safety and efficiency.

Initially, the vehicle endeavors to maintain a continuous forward trajectory by invoking the moveForward() function. This state of progression persists contingent upon the distanceFront measurement, obtained from the front-facing ultrasonic sensor, consistently exceeding a predefined SAFE_DISTANCE (e.g., 50 cm), thereby indicating an unimpeded path ahead.

Upon the detection of an imminent obstacle—specifically, when the distanceFront measurement registers at or below the SAFE_DISTANCE threshold—the vehicle's immediate and critical response is to engage stopMotors(), ensuring a prompt halt to avert potential collision. Subsequent to this cessation of movement, the algorithm initiates a crucial path-finding routine, involving a comprehensive environmental scan. This involves commanding the servo motor to precisely readDistance() first to the left (0°) and then to the right (180°), thereby yielding distanceLeft and distanceRight measurements, respectively. These two critical data points are fundamental for discerning the most viable escape route.

The subsequent decision-making process is hierarchically structured to optimize navigation. In an ideal scenario where both distanceLeft and distanceRight are found to unequivocally exceed the SAFE_DISTANCE (indicating clear paths on both flanks), the algorithm employs an optimization strategy. It performs a comparative analysis between distanceLeft and distanceRight to ascertain which side offers superior clearance. Should distanceLeft be greater than distanceRight, the vehicle executes a turnLeft() maneuver; conversely, if distanceRight is equal to or greater than distanceLeft, it performs a turnRight() action. This intelligent selection ensures the robot consistently prioritizes the wider, and thus presumptively safer, passage. Conversely, if only one side is determined to be clear (e.g., distanceLeft is sufficient while distanceRight is not), the vehicle's strategy dictates a priority for that discernible clear path; for instance, if distanceLeft allows, the vehicle will execute a turnLeft().

The most challenging scenario arises when both distanceLeft and distanceRight are determined to be at or below the SAFE_DISTANCE (indicating simultaneous blockage on both flanks), in conjunction with the front also being obstructed. In such an "all blocked" predicament, the algorithm prescribes a compensatory maneuver: the vehicle must execute a moveBackward() action for a predetermined brief duration. This strategic retreat serves to create the necessary proximate space, allowing the robot to potentially reorient itself or gain a fresh perspective on its immediate surroundings before the next iterative evaluation. Following any execution of a turning or backward movement, the runAutonomousMode() function seamlessly loops back to its initial phase, recommencing the assessment of the frontal path and continuously adapting its navigation strategy based on newly acquired sensor data. This iterative and responsive processing, governed by the checkInterval (e.g., 200 ms), underpins the dynamic and adaptive nature of the autonomous avoidance mechanism.

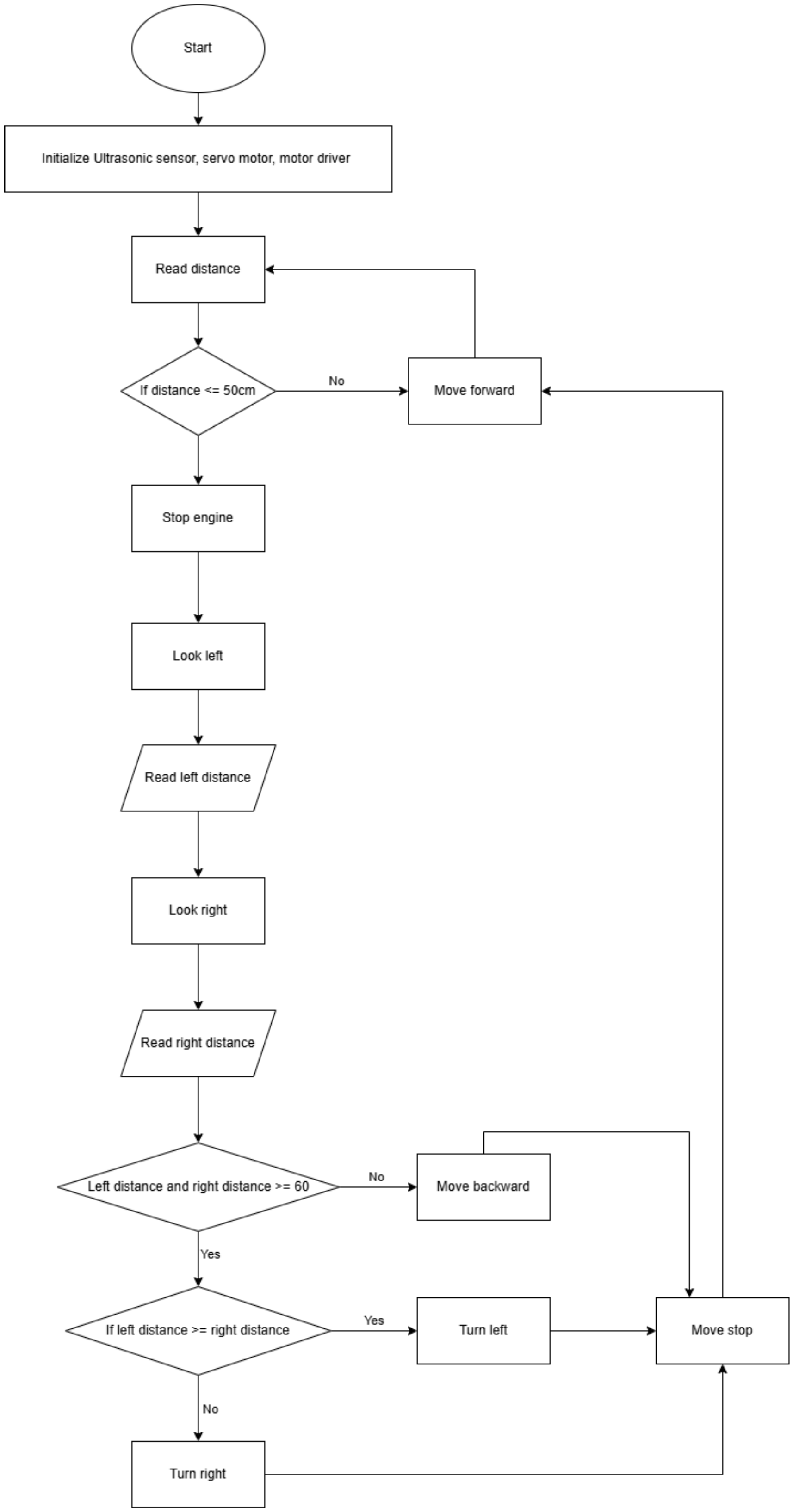

Figure 18: *Obstacle avoidance algorithm*

III. Switching between control modes (Manual/Autonomous)

The ability to switch between manual and autonomous modes is an important feature of the vehicle. This functionality is controlled by the autoMode boolean flag, which dictates the primary execution path within the loop() function.

Mode activation and deactivation are primarily managed through specific button presses on the Dabble GamePad. Pressing the "Triangle" button (GamePad.isTrianglePressed()) sets the autoMode flag to true, thereby enabling autonomous navigation. Conversely, pressing the "Cross" button (GamePad.isCrossPressed()) sets autoMode to false, deactivating the autonomous mode and reverting control back to manual operation. A critical safety measure implemented during this transition is the immediate invocation of stopMotors() whenever autoMode is set to false. This ensures the vehicle comes to a complete halt before any manual commands are processed, mitigating the risk of unintended movements during mode changes.

A key design consideration is the input priority given to direct serial commands received via the mySerial (Bluetooth) port. If mySerial.available() returns true, indicating that data is present in the Bluetooth buffer, the handleBluetooth() function is executed preferentially. This design ensures that explicit commands sent from a Bluetooth terminal can immediately override any ongoing GamePad inputs, including mode switching requests. This reliable override capability provides an additional layer of control and debugging flexibility. This layered control structure, combined with clear mode activation/deactivation triggers and input prioritization, contributes to a highly predictable and user-friendly operational experience for the hybrid robot.

```
void loop() {
  if (mySerial.available()) {
    state = mySerial.readStringUntil('\n');
    state.trim();
    Serial.println("BT: " + state);
    handleBluetooth(state);
  }
  else {
    Dabble.processInput();

    if (GamePad.isTrianglePressed()) {
      autoMode = true;
      Serial.println("Autonomous Mode ON");
    }

    if (GamePad.isCrossPressed()) {
      autoMode = false;
      stopMotors();
      Serial.println("Autonomous Mode OFF");
    }

    if (autoMode) {
      runAutonomousMode();
    }
    else {
      handleManualControl();
    }
  }
}
```

Figure 19: *Switching between two modes*

Chapter 5: Testing and Evaluation of results

I. Testing environment and measurement tools

All tests were conducted in a controlled indoor environment, primarily on a flat, even surface to eliminate external variables like rough terrain. Lighting conditions were kept consistent to avoid interference with sensor readings. Various obstacles, such as cardboard boxes and walls, were used to simulate real-world impediments for the autonomous mode.

Data collection during testing utilized a combination of direct observation and digital tools. A stopwatch and ruler were employed for quantitative measurements such as vehicle speed, turning radius, and obstacle distances. The Arduino Serial Monitor served as a important debugging and data logging tool, providing real-time output of sensor readings (e.g., distanceFront, distanceLeft, distanceRight), received Bluetooth signals and internal state variables, which allowed for detailed analysis of the algorithm's decision-making process.

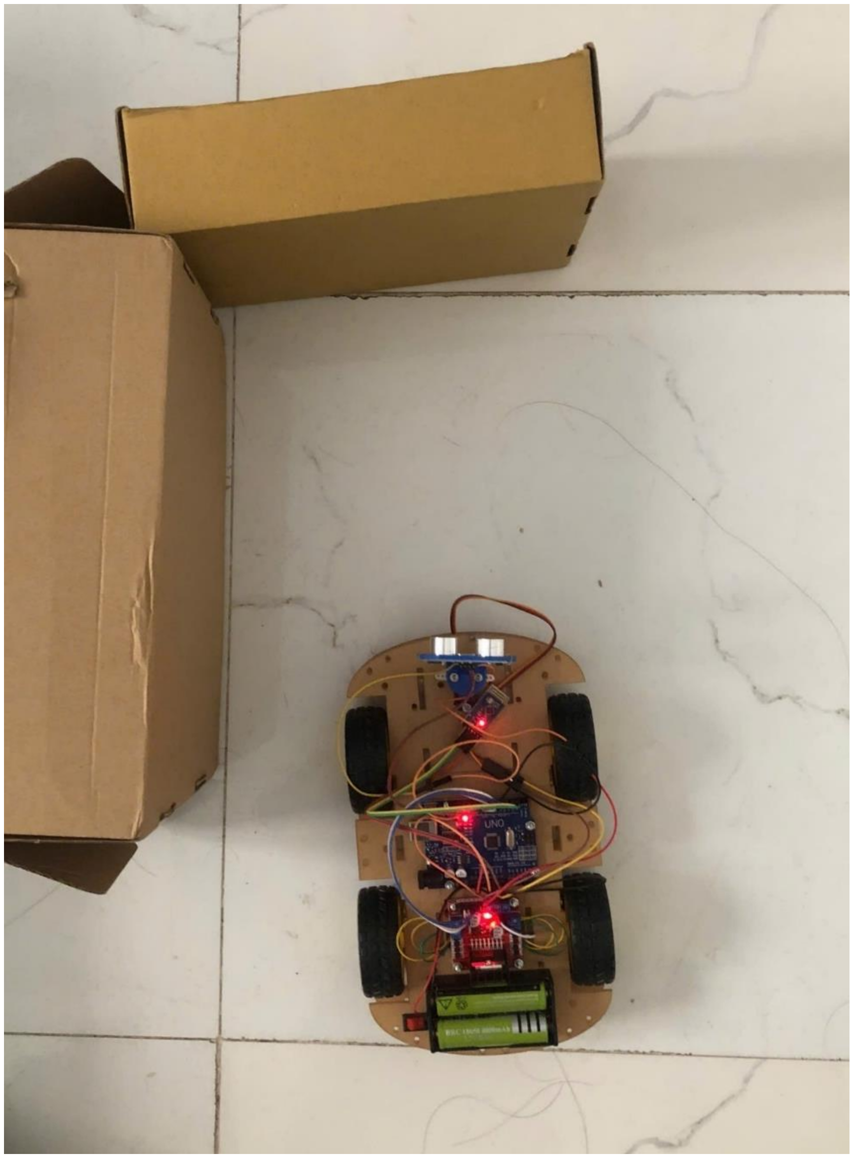

Figure 20: *Testing environment example*

II. Testing of Manual control mode

1. Test cases and procedure

Testing for manual control mode began with thorough preparation, ensuring the vehicle was fully assembled, powered, and placed on a flat, unobstructed surface. Most importantly, the Arduino Uno was made sure for stable power and secure hardware connections, while the HC-05 Bluetooth module was confirmed to be paired and connected to the controlling smartphone running the Dabble application.

For each directional command (e.g., forward, backward, turn left), the corresponding button on the Dabble GamePad was pressed and held with each command repeated 10 times to assess consistency. The vehicle's movement was carefully observed and recorded. The mode switching functionality was tested similarly by pressing the designated button 10 times, noting the transition's reliability and latency. Finally, the HC-05 module's signal reception was validated through 10 instances of command transmission while the vehicle was idle, observing its responsiveness. All real results and qualitative observations were then documented against the expected outcomes to determine the test's pass/fail status.

Number	Action	Expected results	Real results	Delay (seconds)	Result
1	Forward command	The model performs forward 10/10 times	10/10	0.5	Passed
2	Backward command	The model performs backward 10/10 times	10/10	0.5	Passed
3	Turn left command	The model performs turn left 10/10 times	10/10	0.5	Passed
4	Turn right command	The model performs turn right 10/10 times	10/10	0.5	Passed
5	Forward-right command	The model performs forward-right 10/10 times	10/10	0.5	Passed

6	Forward-left command	The model performs forward-left 10/10 times	10/10	0.5	Passed
7	Backward-right command	The model performs backward-right 10/10 times	10/10	0.5	Passed
8	Backward-left command	The model performs backward-left 10/10 times	9/10	0.5	Passed
9	Switch mode command	The model performs switch mode 8/10 times	8/10	3-4	Passed
10	HC-05 module	The model received signals 10/10 times	10/10	0.5	Passed

Table 1: *Test cases for Manual control mode*

Observation

- **Forward command:** Consistent and smooth forward movement.

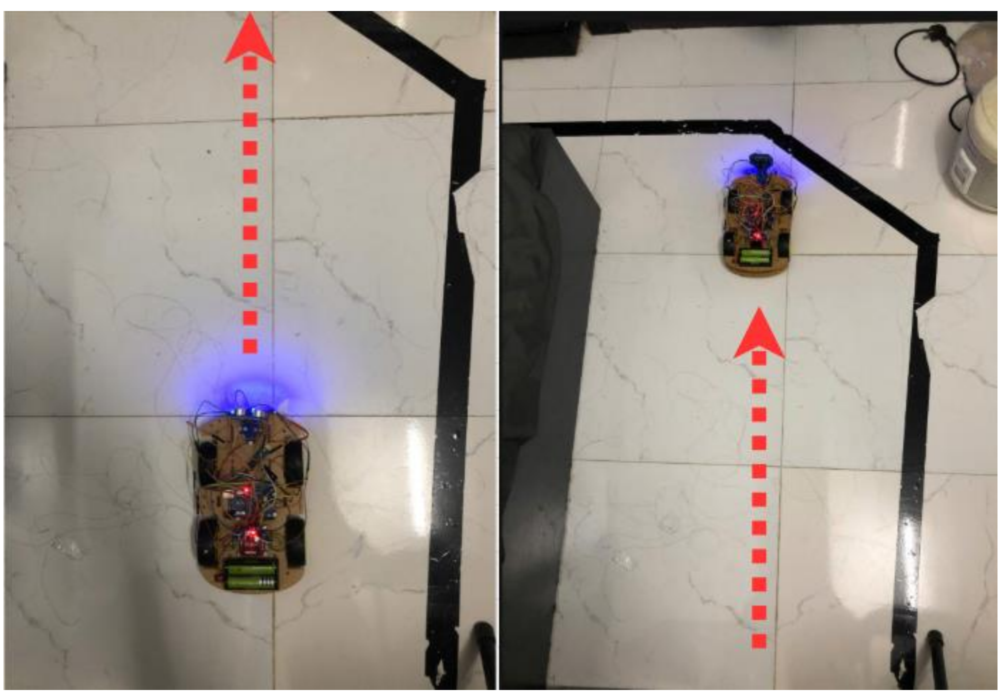

Figure II. a: *Forward command*

- **Backward command:** Consistent and smooth backward movement.

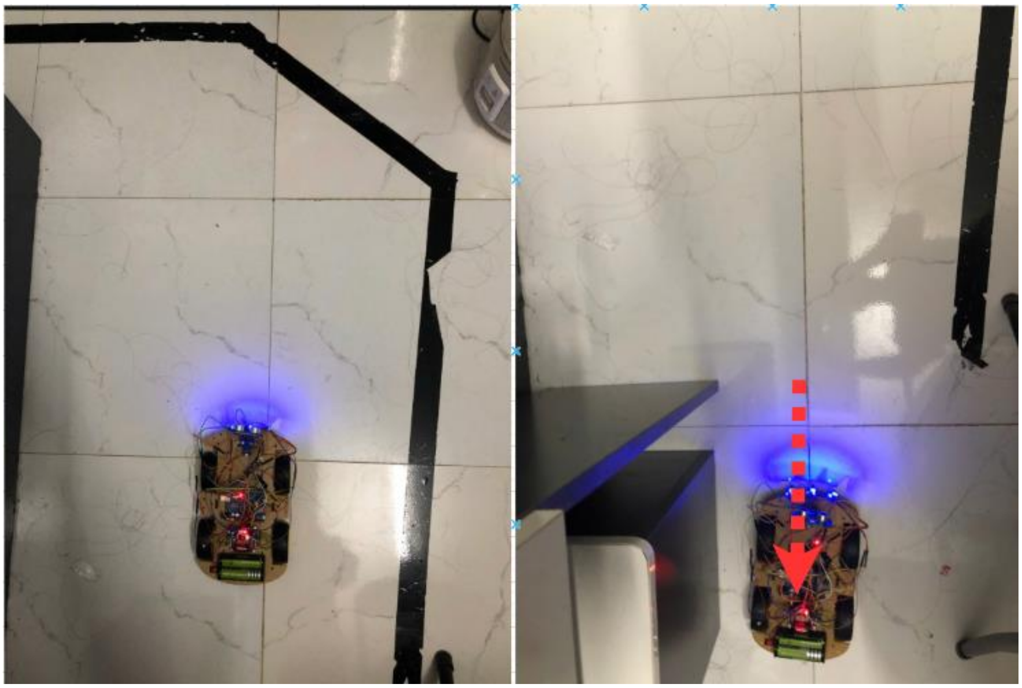

Figure II. b: *Backward command*

- **Turn left command:** Accurate and consistent left turns.

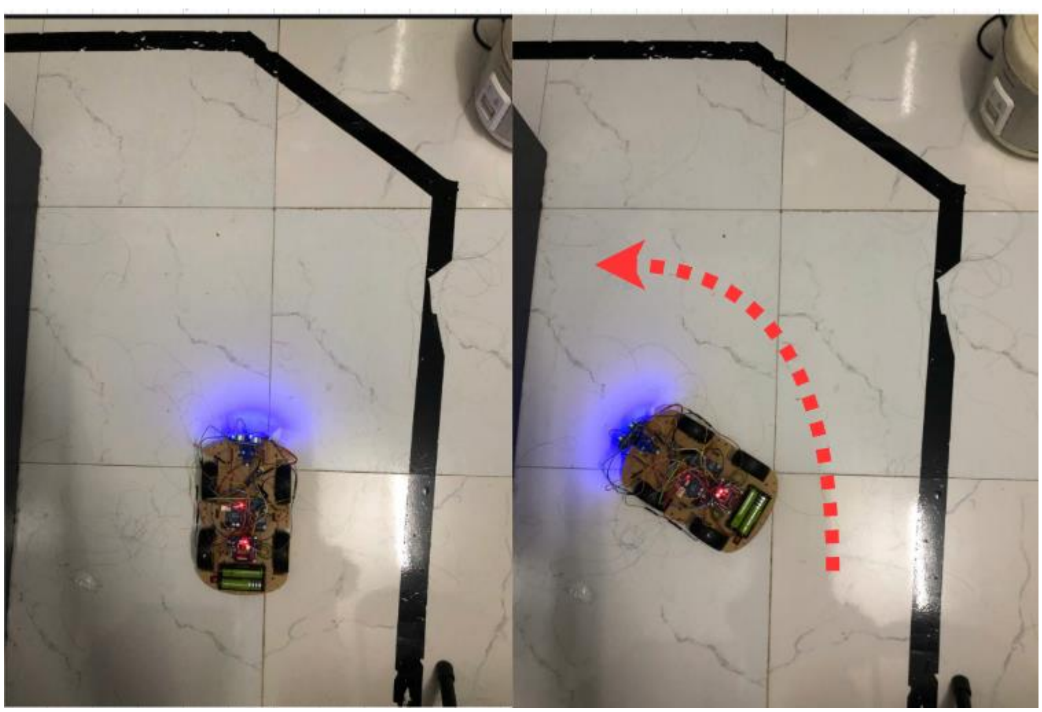

Figure II. c: *Turn left command*

- **Turn right command:** Accurate and consistent right turns.

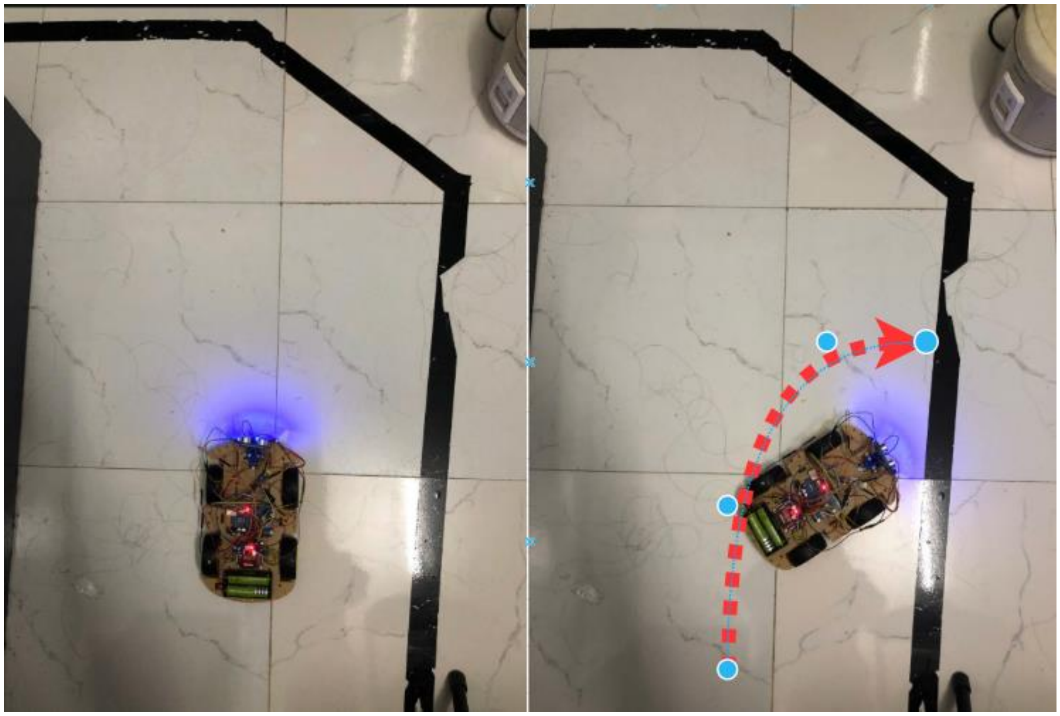

Figure II. d: *Turn right command*

- **Forward-right command**: Correct forward-right movement.

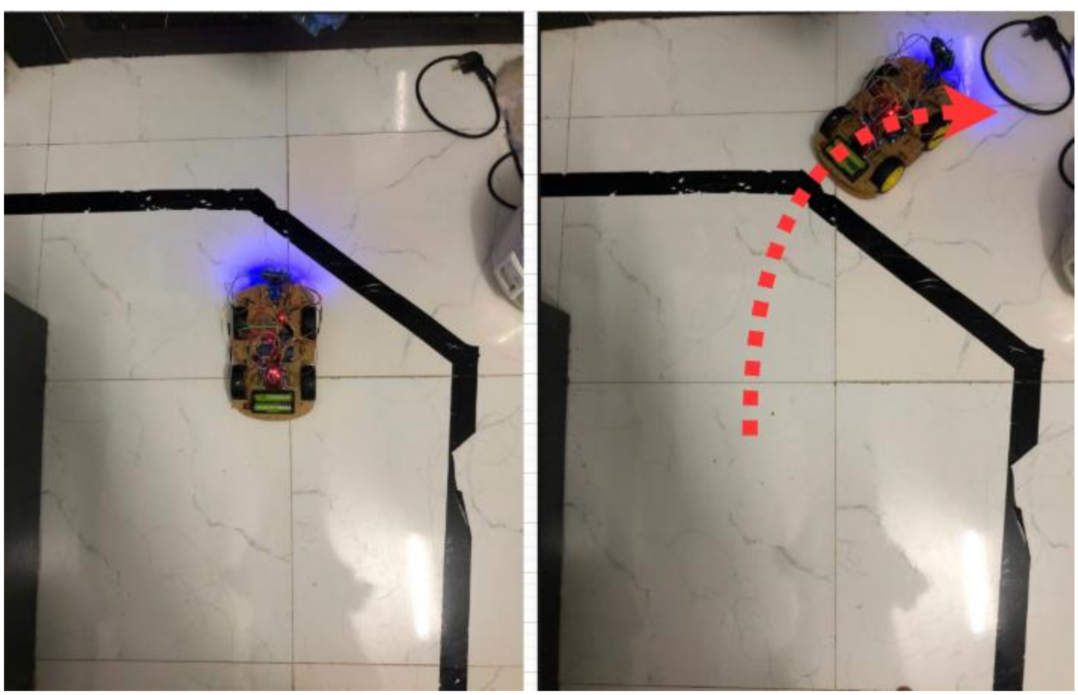

Figure II. e: *Forward-right command*

- **Forward-left command:** Correct forward-left movement.

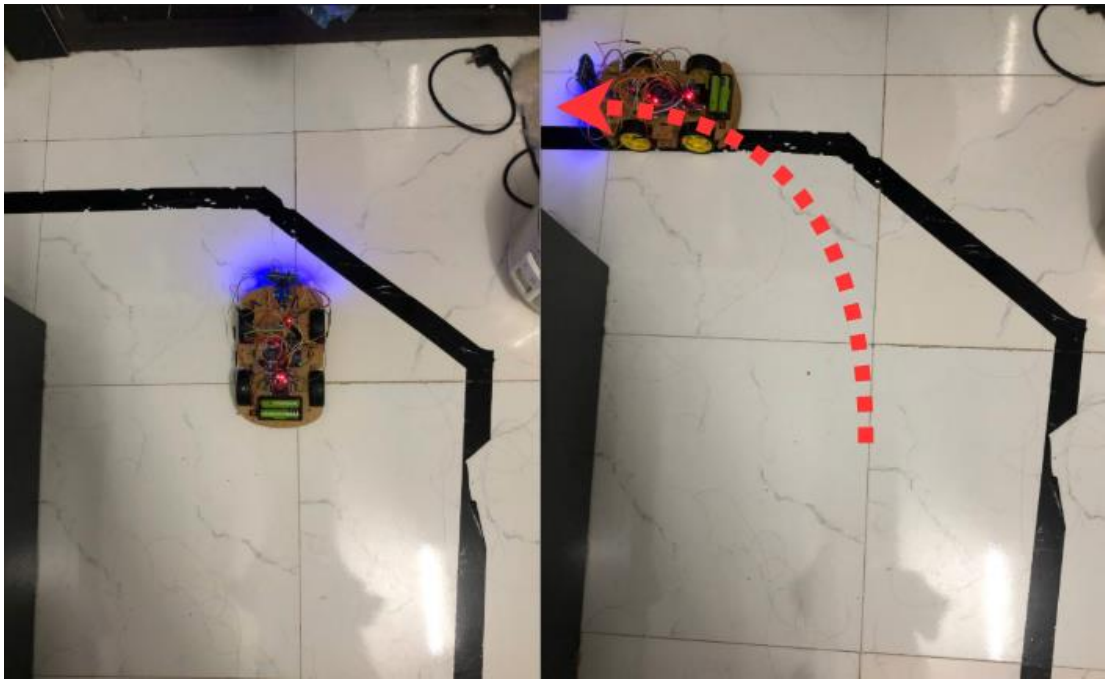

Figure II. f: *Forward-left command*

- **Backward-right command:** Correct backward-right movement.

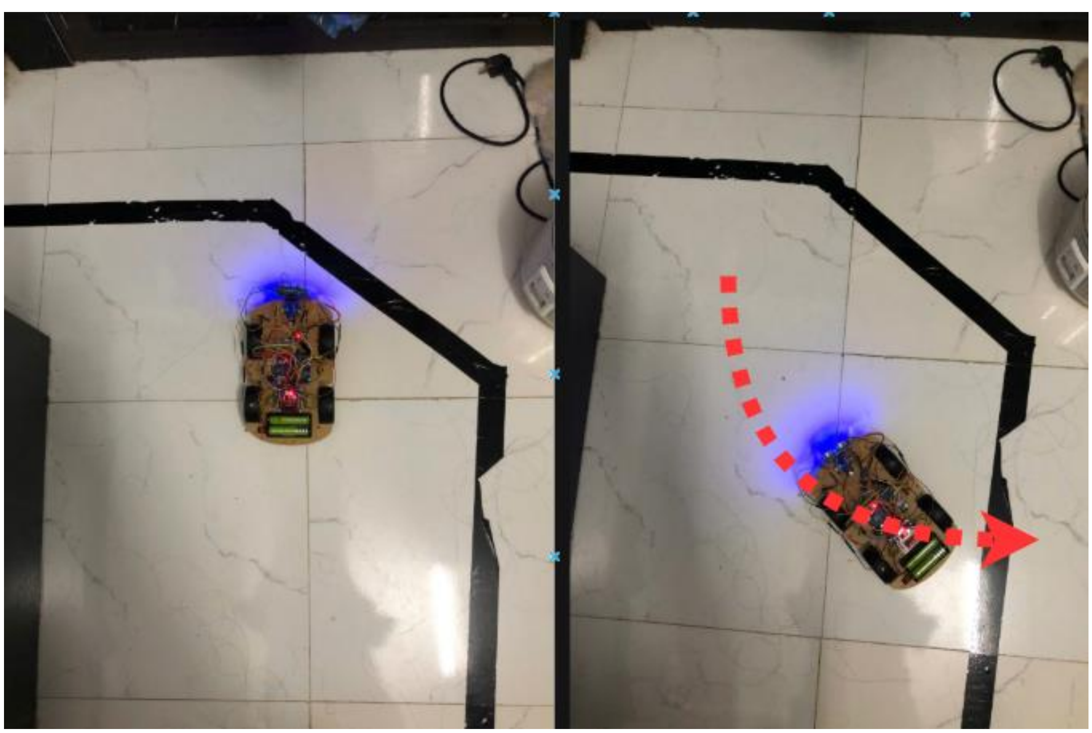

Figure II. g: *Backward-right command*

- **Backward-left command:** Correct backward-left movement.

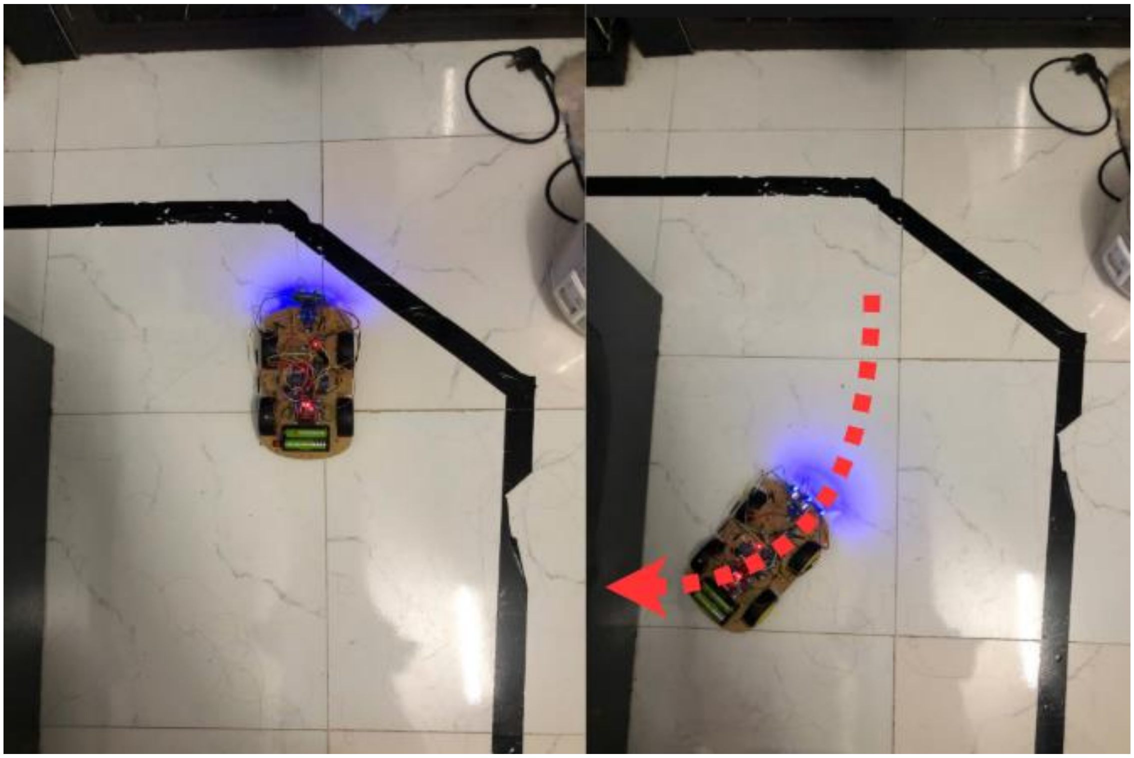

Figure II. h: *Backward-left command*

- **Switch mode command**: Signal reception was slower than expected, leading to a noticeable delay in mode transition.
- **HC-05 module:** Consistent and reliable signal reception from Bluetooth module.

2. Results and summary

Table 1 illustrates the test results of the manual control mode, which demonstrated a high degree of responsiveness and reliability for fundamental commands. The vehicle consistently executed forward, backward, left turn, and right turn commands with a 100% success rate (10/10 trials), indicating robust communication and motor control for basic movements. These actions typically exhibited a minimal delay of approximately 0.5 seconds between command transmission and execution, which is indicative of effective real-time control. Diagonal movements, while largely successful, showed a slight decrement in perfect execution; for instance, the "Backward-left command" achieved a 9/10 success rate, suggesting minor variations in complex multi-motor coordination. A notable observation pertained to the "Switch mode command," which registered an 8/10 success rate and was accompanied by a significantly longer delay of

3-4 seconds, explicitly noted as requiring "longer signal reception than expected". This prolonged transition time suggests a potential area for optimization in the command processing or Bluetooth communication protocol for mode switching. The HC-05 module's ability to consistently receive signals (10/10 times) broadly confirms the stability of the Bluetooth communication link itself for general command reception. Overall, the manual control mode is highly functional for direct navigation, with mode switching presenting the primary area for further improvement in user experience.

In addition, the above results have shown that we have completed the objectives has shown as in the introduction, that is evaluating the developed system's overall performance under various test cases and implementing precise motor control algorithms to enable fundamental vehicle movements (such as forward, backward, turns, and diagonal maneuvers) under manual control based on received Bluetooth commands.

III. Testing of Autonomous Navigation mode

1. Test cases and procedure

The autonomous navigation mode was systematically tested to evaluate its obstacle detection and avoidance capabilities across various scenarios. The testing procedure commenced with careful preparation, including full vehicle assembly, power-on, and placement within a controlled test environment configured with specific flat obstacles. All sensor and motor connections were verified, and the Arduino was confirmed to be loaded with the correct autonomous navigation logic. Autonomous mode was then activated via the Dabble application's designated switch command.

For each distinct obstacle avoidance scenario (e.g., clear path, frontal, side, or U-shape blockages), the environment was precisely set up, and the vehicle was allowed to operate autonomously. Its movement, responsiveness to obstacles, and decision-making for avoidance maneuvers were meticulously observed and recorded over 10 repetitions per scenario to ensure consistency. Following this, the mode switching back to manual control was tested 10 times, noting the transition's reliability and latency. Specific obstacle distance tests were performed to assess stopping behavior and reading accuracy at varying ranges (e.g., 65cm, 55cm, 30cm). Finally, the consistent functionality of the HC-SR04 ultrasonic sensor and servo motor was verified through their performance during autonomous operations. All real results and qualitative observations were systematically documented against expected outcomes to determine the success of each test case.

Number	**Action**	**Expected results**	**Real results**	**Delay (second)**	**Result**
1	Go forward when there are no obstacle	The model perform moving forward 10/10 times	10/10	0.5	Passed
2	Frontal obstacle	Stop, scan, turn to clear side 9/10 times	10/10	0.5	Passed
3	Right side blocked	Stop, scan, turn to left side 9/10 times	9/10	0.5	Passed
4	Left side blocked	Stop, scan, turn to right side 9/10 times	10/10	0.5	Passed
5	U-shape blocked	Stop, scan, go backward then stop, rescanned 7/10 times	8/10	1	Passed
6	Switching mode back to manual control	The model performs switch mode 8/10 times	8/10	3-4	Passed
6	Obstacle 65cm, 55cm, 30cm ahead	Stop and scan at 50-60cm 8/10 times	9/10	1	Passed
7	HC-SR04 module	The model detected obstacles 10/10 times	10/10	0.5	Passed
8	Servo motor	The model rotates 0°, 180° (scan left then right side) 10/10 times	10/10	0.5	Passed

Table 2: *Test cases of Autonomous Navigation mode*

Observation

- **Go forward when there are no obstacle:** Maintained consistent forward movement.

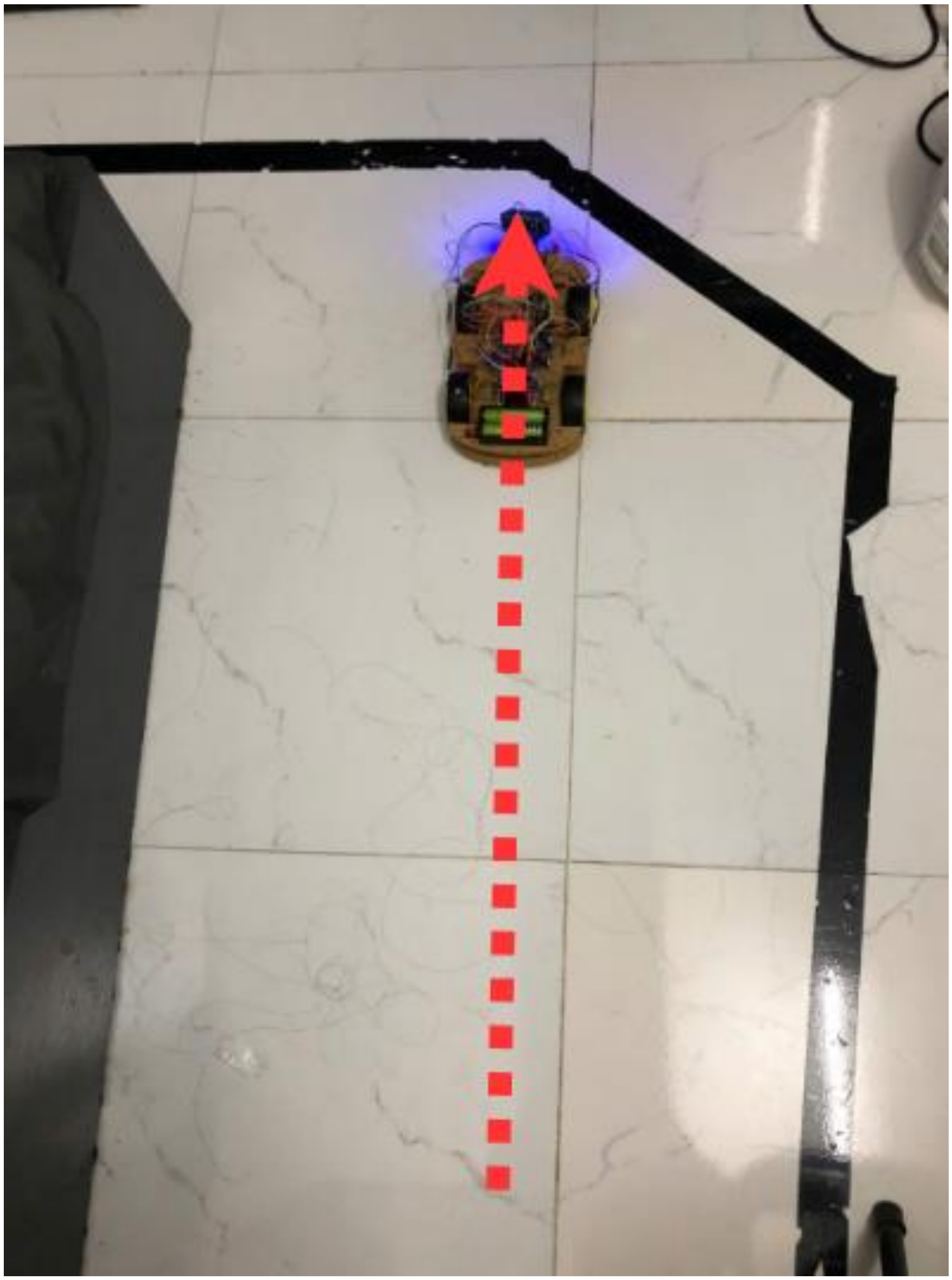

Figure III. a: *No obstacle ahead*

- **Frontal obstacle:** Consistently stopped at ~50-60cm, correctly identified clear path.

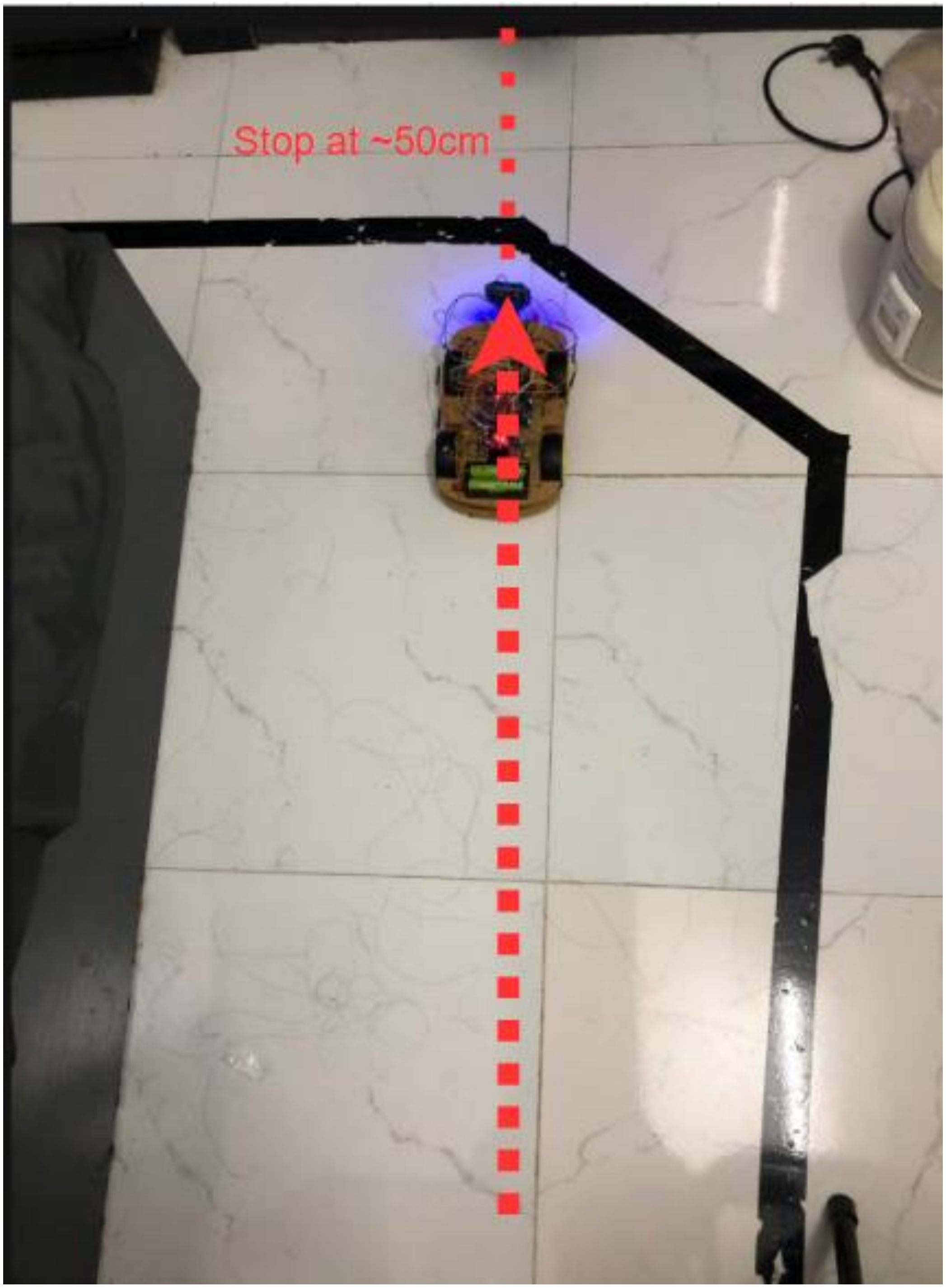

Figure III. b: *Frontal obstacle*

- **Right side blocked:** One instance where the turn was slightly delayed, leading to a minor contact, but still have consistent detection and correct turning.

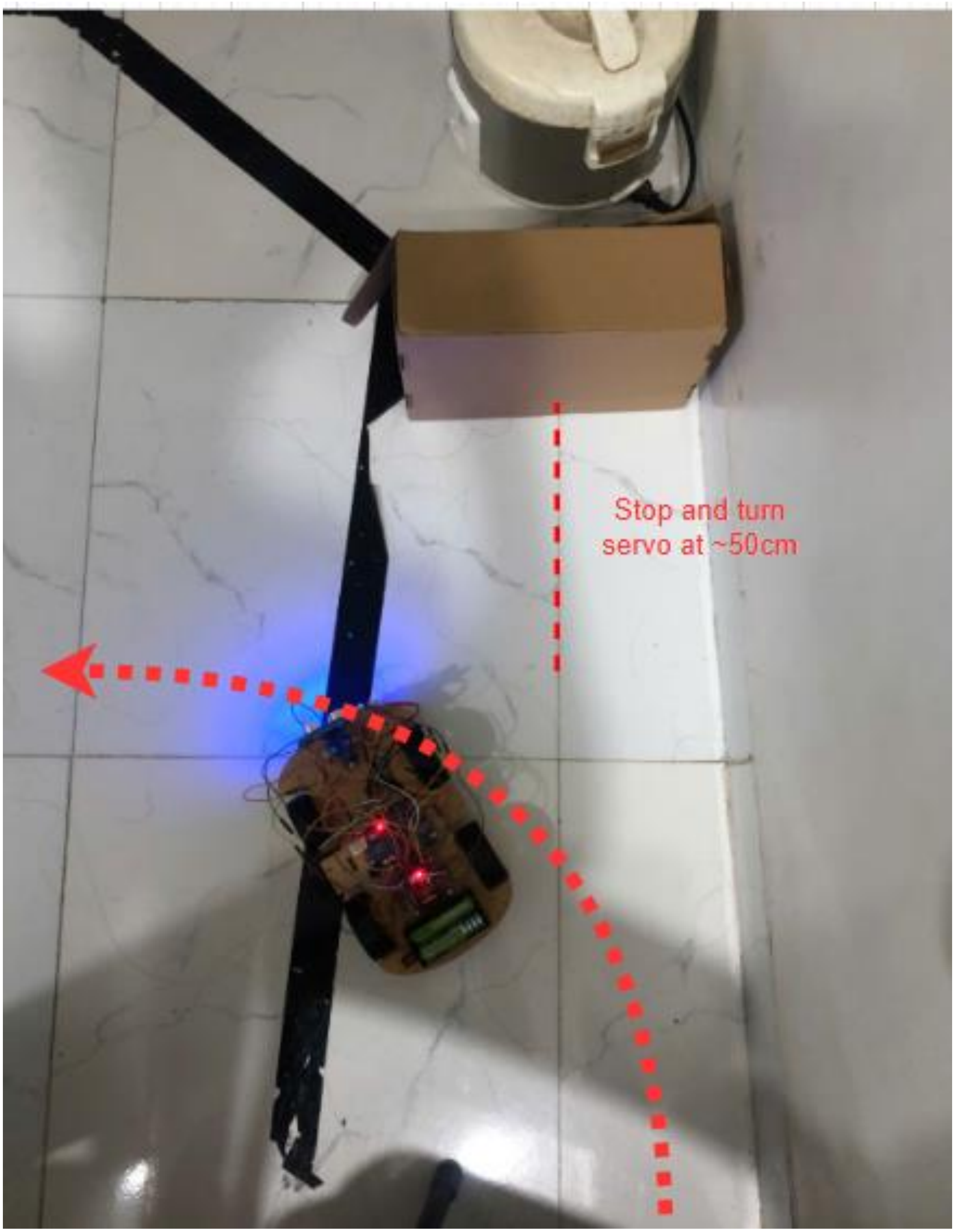

Figure III. c: *Frontal and right obstacle*

- **Left side blocked:** Consistent detection and correct turning

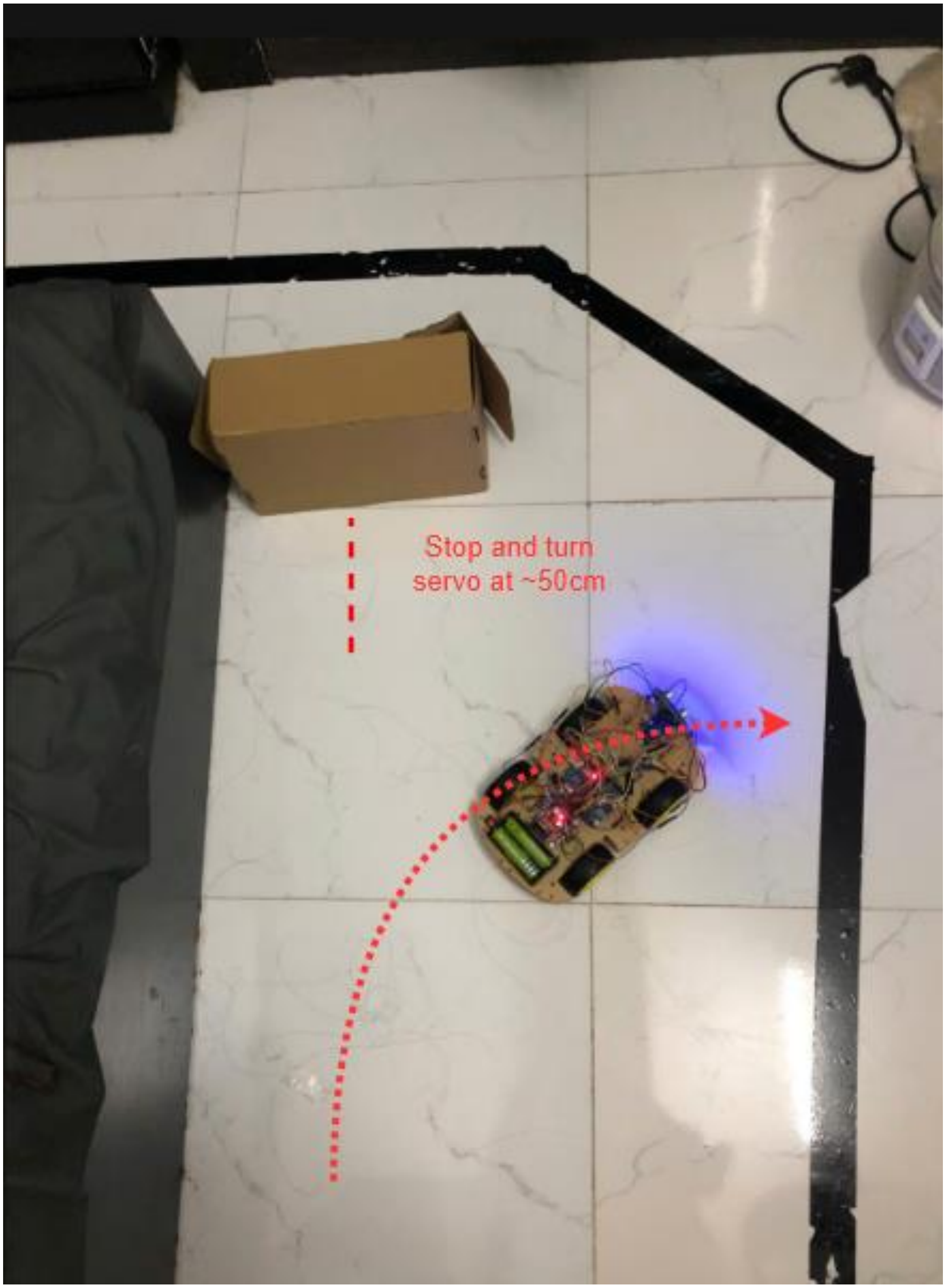

Figure III. d: *Frontal and left obstacle*

- **U-shape blocked:** Successfully moved backward to escape most times; occasionally made minor contact with obstacles during backward movement due to tight space.

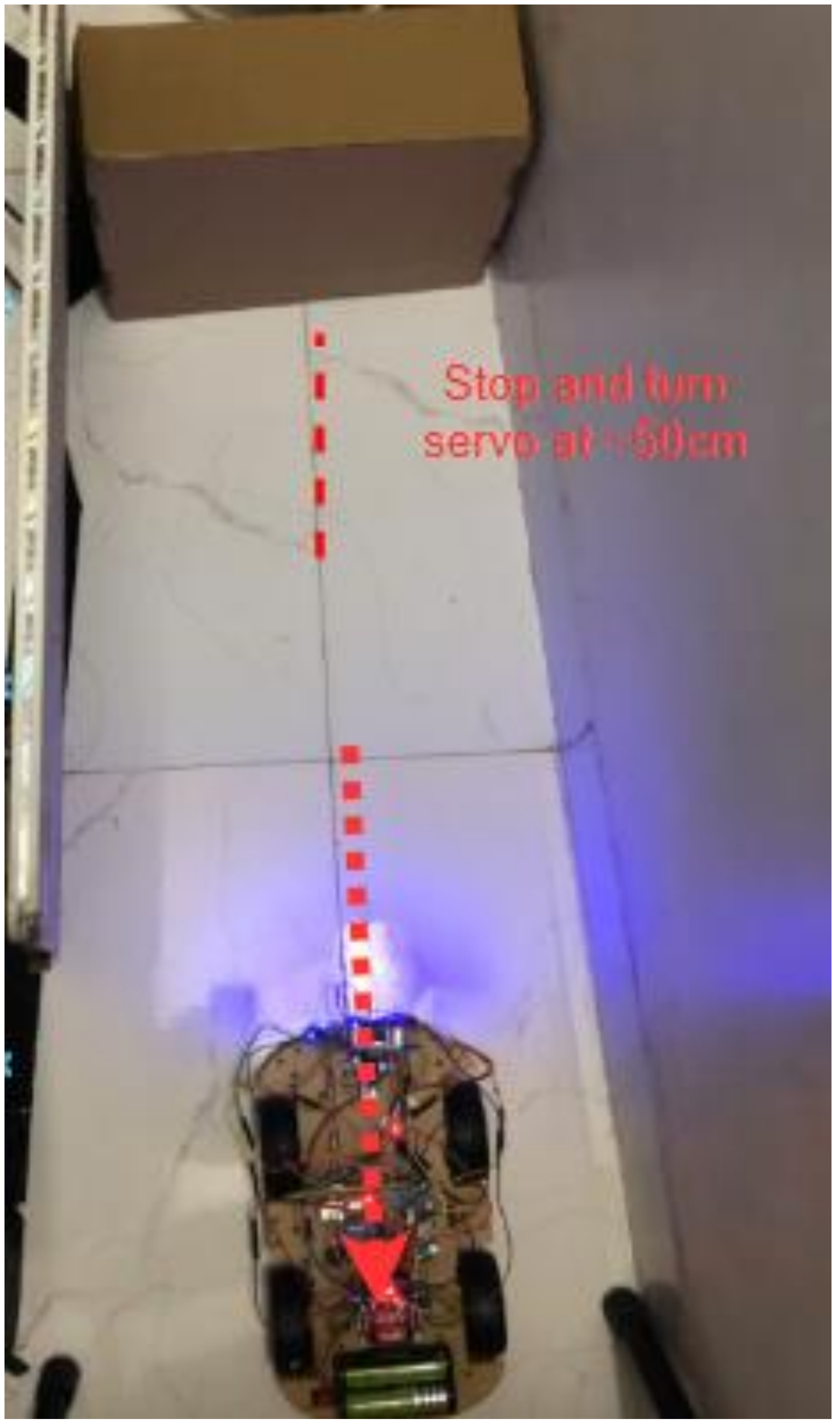

Figure III. e: *U-shape obstacle*

- **Switching mode back to manual control:** Required multiple button presses to switch mode; noticeable latency in transition.
- **Obstacle 65cm, 55cm, 30cm ahead:** Reading speed and distance are a bit slower and have a little measurement error than expected.
- **HC-SR04 module:** Sometimes showed slight deviation, but generally reliable within the range.
- **Servo motor:** Servo rotated to all specified angles (0°, 180° for left and right side respectively) correctly and consistently.

2. Results and summary

The evaluation of the autonomous navigation mode revealed strong capabilities in obstacle detection and avoidance, coupled with some identifiable challenges in complex scenarios. The vehicle successfully navigated clear paths, achieving a 10/10 success rate for continuous forward movement when no obstacles were present. Furthermore, its response to frontal obstacles was consistently accurate, demonstrating a 10/10 success rate in stopping, scanning, and turning towards a clear side. Avoidance maneuvers when only one side was blocked also performed commendably, with 9/10 success for "Right side blocked" and 10/10 for "Left side blocked" scenarios, confirming the efficacy of the side-scanning and decision-making logic.

However, more challenging environments presented notable difficulties. In the "U-shape blocked" scenario, simulating an environment where all immediate paths are obstructed, the success rate dropped to 8/10. Crucially, observations indicated that the robot "sometimes collides into the obstacles" in this condition, suggesting that while the moveBackward() strategy is employed, its execution or subsequent re-evaluation might require refinement to consistently avoid contact. The issue with mode switching persisted in this mode as well, with "Switching mode back to manual control" achieving only an 8/10 success rate and exhibiting the 3-4 second delay, often necessitating "many presses to switch mode", consistent with observations from the manual mode testing. The autonomous mode's performance, the individual component tests for the HC-SR04 ultrasonic sensor and the servo motor demonstrated perfect functionality, both achieving 10/10 success. In short, the above results have illustrates that the objectives which has shown as in the introduction are completed, that is evaluating the developed system's overall performance under various operational conditions in autonomous navigation mode using HC-SR04 ultrasonic sensor, to detect and successfully bypass obstacles in different environmental configurations (e.g., frontal obstacles, side-blocked passages, and complex U-shape blocked scenarios)

IV. Analysis of HC-SR04 measuring distances

Actual distance	Measured distance 1 (cm)	Measured distance 2 (cm)	Measured distance 3(cm)	Average measured (cm)	Absolute error (cm)
10	10	12	12	11.33	±1.33
20	21	22	19	20.67	±0.67
25	25	26	25	25.33	±0.33
30	30	30	31	30.33	±0.33
40	39	39	41	39.67	±0.33
50	50	49	50	49.67	±0.33
60	60	59	59	59.33	±0.67
70	72	69	70	70.33	±0.33

Table 3: *Analysis of HC-SR04 measuring distances*

This table provides a statiscal analysis of the HC-SR04 ultrasonic sensor's accuracy, a fundamental aspect for reliable autonomous navigation. The results demonstrate a remarkably high level of precision across the tested range from 10 cm to 70 cm. The absolute error for the majority of measurements consistently remained very low, typically within ±0.33 cm. Notably, at 10, 20 cm and 60 cm—distances that are too close or nearly for the SAFE_DISTANCE variable in the obstacle avoidance algorithm—the sensor exhibited errors, which recorded at ±1.33 cm and ±0.67 cm for both 20cm and 60cm respectively), which shows that the sensor accuracy is unstable, but it still directly contributes to the reliable obstacle detection and precise decision-making capability of the autonomous navigation algorithm.

Chapter 6: Conclusion and Future developments

I. Conclusion

The primary objective of this project was to design and implement a hybrid navigation system for robotic vehicle capable of both manual control via Bluetooth and autonomous navigation with obstacle avoidance. The developed system successfully achieved these core objectives, demonstrating functional and adaptable capabilities in various operational scenarios. The hardware architecture was effectively established through the integration of key components, including the Arduino Uno serving as the central processing unit, an L298N motor driver module enabling differential drive control, an HC-SR04 ultrasonic sensor for environmental perception, a servo motor facilitating sensor scanning, and an HC-05 Bluetooth module for wireless communication. This robust hardware foundation provided the necessary platform for the implementation of the desired functionalities.

The software implementation successfully enabled two distinct operational modes. The manual control mode proved highly responsive and reliable for fundamental movements such as forward, backward, left turn, and right turn, consistently achieving a 100% success rate across trials. These basic commands exhibited minimal latency, approximately 0.5 seconds between command transmission and execution, which contributes to an intuitive control experience. The autonomous navigation mode demonstrated effective obstacle avoidance capabilities, reliably detecting and reacting to frontal obstacles by stopping, scanning, and turning towards a clear path. The robust accuracy of the HC-SR04 ultrasonic sensor, characterized by absolute errors typically within at least ±0.33 cm, significantly contributed to the reliability of the obstacle detection system. Furthermore, the core obstacle avoidance algorithm effectively enabled the robot to manage "all blocked" scenarios by initiating a backward movement to free itself and re-evaluate its surroundings. In conclusion, this project successfully developed a functional hybrid robotic platform that can be intuitively controlled manually and efficiently navigate autonomously to avoid obstacles, with the integration of its hardware and software components meeting the defined project requirements.

II. Future Work

While the current robotic vehicle successfully fulfills its core functionalities, several areas have been identified for future enhancement and exploration to further improve its performance, robustness, and expanded capabilities. One critical area for improvement lies in the refinement of the obstacle avoidance algorithm in complex environments. The current algorithm, although effective for single obstacles, showed some limitations in highly confined or "U-shape blocked" scenarios, occasionally resulting in minor collisions. Future work could involve implementing more advanced path planning algorithms, such as Rapidly-exploring Random Tree (RRT) [37], or incorporating techniques like fuzzy logic [38] to enable more advanced decision-making and smoother navigation in narrow environments.

Another important area for development is the enhancement of mode switching reliability and speed. Both manual and autonomous mode testing consistently revealed a significant delay, ranging from 3 to 4 seconds, and occasional unresponsiveness when transitioning between modes, frequently necessitating multiple button presses. Future efforts should therefore focus on optimizing the underlying Bluetooth communication protocol, refining the command parsing logic, or exploring alternative wireless communication methods, such as Wi-Fi modules, to achieve instantaneous and reliable mode transitions.

Beyond these core refinements, integration of additional sensors would substantially enhance the vehicle's environmental perception capabilities, thereby improving its robustness and versatility. The incorporation of infrared (IR) sensors [39], for instance, could provide supplementary short-range detection, particularly beneficial for identifying objects very close to the vehicle's sides and complementing the ultrasonic sensor's broader range. Adding line-following sensors would enable the robot to navigate predefined paths, significantly expanding its application scope [40]. For more advanced autonomous navigation, the integration of a camera system could facilitate visual odometry or object recognition, allowing for more sophisticated environmental mapping and intelligent decision-making. Furthermore, power management optimization through investigating component power consumption and implementing power-saving modes (e.g., sleeping the Arduino when idle, optimizing motor PWM values) could extend the vehicle's operational battery life. Minor mechanical design enhancements, such as ensuring precise motor alignment and optimizing wheel traction, could also improve straight-line tracking and turning accuracy, which were observed to have slight deviations. Finally, the development of a custom Graphical User Interface (GUI) on a PC or smartphone, beyond the existing Dabble application, would provide a more intuitive control interface, enable real-time data visualization (e.g., sensor readings, robot status), and facilitate remote debugging, greatly enhancing both the user experience and the overall development process. These proposed future works aim to transform the current functional prototype into a more robust, intelligent, and versatile robotic platform, expanding its capabilities for a wider range of applications.

Chapter 7: References

[1] Roboticstomorrow.com. (2023). *Autonomous Mobile Robots Market Analysis: $3.49B Valuation by 2023, 15.3% CAGR Predicted to 2030 | RoboticsTomorrow*. [online] Available at: https://www.roboticstomorrow.com/news/2025/04/22/autonomous-mobile-robots-market-analysis-349b-valuation-by-2023-153-cagr-predicted-to-2030/24623/.

[2] Vietnam+ (VietnamPlus) (2024). *Ministry announces Vietnam's digital infrastructure plan for 2021-2030*. [online] Vietnam+ (VietnamPlus). Available at: https://en.vietnamplus.vn/ministry-announces-vietnams-digital-infrastructure-plan-for-2021-2030-post279986.vnp.

[3] Tang, Y., Zakaria, M.A. and Younas, M. (2025). Path Planning Trends for Autonomous Mobile Robot Navigation: A Review. *Sensors*, 25(4), p.1206. doi:https://doi.org/10.3390/s25041206.

[4] Frontiers. (2023). *Frontiers | Advancing Autonomous Robots: Challenges and Innovations in Open-World Scene Understanding*. [online] Available at: https://www.frontiersin.org/research-topics/61975/advancing-autonomous-robots-challenges-and-innovations-in-open-world-scene-understanding.

[5] Wijayathunga, L., Rassau, A. and Chai, D. (2023). Challenges and Solutions for Autonomous Ground Robot Scene Understanding and Navigation in Unstructured Outdoor Environments: A Review. *Applied Sciences*, [online] 13(17), p.9877. doi:https://doi.org/10.3390/app13179877.

[6] Martens, M.H. and van den Beukel, A.P. (2013). The road to automated driving: Dual mode and human factors considerations. *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*. doi:https://doi.org/10.1109/itsc.2013.6728564.

[7] Wang, W., Xu, X., Chen, Z., Yang, J., Ran, Y., Tan, Z. and Luo, M. (2024). Research on Autonomous Navigation of Dual-mode Wheel-legged Robot. *2021*

IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), [online] pp.9022–9029. doi:https://doi.org/10.1109/iros58592.2024.10801460.

[8] Kapoor, R. (2023). *The Benefits of Using Arduino in Robotics and Automation*. [online] kunkune.co.uk. Available at: https://kunkune.co.uk/blog/the-benefits-of-using-arduino-in-robotics-and-automation/.

[9] Prasad, R. (2023). *Is Arduino Used In Robotics? - An Overview and Future*. [online] Moonpreneur. Available at: https://mp.moonpreneur.com/blog/arduino-used-in-robotics/.

[10] *Reese, L. (2019). The working principle, applications and limitations of ultrasonic sensors. [online] Microcontrollertips.com. Available at: https://www.microcontrollertips.com/principle-applications-limitations-ultrasonic-sensors-faq/.*

[11] Sawkare, R. (2024). *Ultrasonic Sensor in Robotics - Object Detection and Beyond*. [online] Vayuyaan. Available at: https://vayuyaan.com/blog/ultrasonic-sensor-in-robotics/.

[12] Kapoor, R. (2023). *The Benefits of Using Arduino in Robotics and Automation*. [online] kunkune.co.uk. Available at: https://kunkune.co.uk/blog/the-benefits-of-using-arduino-in-robotics-and-automation/.

[13] MaxBotix (2019). *Ultrasonic Sensors: Advantages and Limitations*. [online] MaxBotix. Available at: https://maxbotix.com/blogs/blog/advantages-limitations-ultrasonic-sensors.

[14] Dzedzickis, A., Subačiūtė-Žemaitienė, J., Šutinys, E., Samukaitė-Bubnienė, U. and Bučinskas, V. (2021). Advanced Applications of Industrial Robotics: New Trends and Possibilities. *Applied Sciences*, 12(1), p.135. doi:https://doi.org/10.3390/app12010135.

[15] Luiz F. P. Oliveira, António P. Moreira, Manuel F. Silva (2021). Advances in Agriculture Robotics: A State-of-the-Art Review and Challenges Ahead. *ProQuest*, [online] 10(2), p.52. doi:https://doi.org/10.3390/robotics10020052.

[16] Nam, T.Q., Tien, H.V., Van, N.A. and Quan, N.D. (2023). Development of an Autonomous Mobile Robot System for Hospital Logistics in Quarantine Zones. *Lecture notes in networks and systems*, pp.271–281. doi:https://doi.org/10.1007/978-981-99-4725-6_34.

[17] Dieber, B., Breiling, B., Taurer, S., Kacianka, S., Rass, S. and Schartner, P. (2017). Security for the Robot Operating System. *Robotics and Autonomous Systems*, [online] 98, pp.192–203. doi:https://doi.org/10.1016/j.robot.2017.09.017.

[18] Min-Fan Ricky Lee and Chien, T.-W. (2020). Artificial Intelligence and Internet of Things for Robotic Disaster Response. doi:https://doi.org/10.1109/aris50834.2020.9205794.

[19] Urrea, C. and Kern, J. (2025). Recent Advances and Challenges in Industrial Robotics: A Systematic Review of Technological Trends and Emerging Applications. *Processes*, [online] 13(3), p.832. doi:https://doi.org/10.3390/pr13030832.

[20] Seppänen, A., Vepsäläinen, J., Ojala, R. and Tammi, K. (2022). Comparison of Semi-autonomous Mobile Robot Control Strategies in Presence of Large Delay Fluctuation. *Journal of Intelligent & Robotic Systems*, 106(1). doi:https://doi.org/10.1007/s10846-022-01711-3.

[21] Morgan, E. (2014). *HCSR04 Ultrasonic Sensor*. [online] Available at: https://www.bitmi.ro/domains/bitmi.ro/files/files/bitmi-datasheet-senzor-ultrasonic-hc-sr04-2702.pdf.

[22] George, S. (n.d.). *HC05 MCU Software-Driven Asynchronous Serial Communication Techniques Using the MC68HC705J1A*. [online] Available at: https://www.nxp.jp/docs/en/application-note/AN1240.pdf.

[23] projecthub.arduino.cc. (2021). *L298n Motor driver Arduino | Motors | Motor Driver | L298n*. [online] Available at: https://projecthub.arduino.cc/lakshyajhalani56/l298n-motor-driver-arduino-motors-motor-driver-l298n-7e1b3b.

[24] Sousa, L.C., Silva, Schettino, V.B., Tatiana, Alessandro, Gouvêa, J.A. and Pinto, M.F. (2025). Obstacle Avoidance Technique for Mobile Robots at Autonomous

Human-Robot Collaborative Warehouse Environments. *Sensors*, [online] 25(8), pp.2387–2387. doi:https://doi.org/10.3390/s25082387.

[25] Shi, Y., Huang, S. and Li, M. (2024). An Improved Global and Local Fusion Path-Planning Algorithm for Mobile Robots. *Sensors*, [online] 24(24), p.7950. doi:https://doi.org/10.3390/s24247950.

[26] McGinn, C., Sena, A. and Kelly, K. (2017). Controlling robots in the home: Factors that affect the performance of novice robot operators. *Applied Ergonomics*, 65, pp.23–32. doi:https://doi.org/10.1016/j.apergo.2017.05.005.

[27] Alqobali, R., Alshmrani, M., Alnasser, R., Rashidi, A., Alhmiedat, T. and Alia, O.M. (2024). A Survey on Robot Semantic Navigation Systems for Indoor Environments. *Applied Sciences*, [online] 14(1), p.89. doi:https://doi.org/10.3390/app14010089.

[28] Brogårdh, T. (2007). Present and future robot control development—An industrial perspective. *Annual Reviews in Control*, 31(1), pp.69–79. doi:https://doi.org/10.1016/j.arcontrol.2007.01.002.

[29] Erden, M.S. and Marić, B. (2011). Assisting manual welding with robot. *Robotics and Computer-Integrated Manufacturing*, [online] 27(4), pp.818–828. doi:https://doi.org/10.1016/j.rcim.2011.01.003.

[30] Patil, V.P., Umakant Gohatre, Matey, S., Jadhav, R. and Gandhi, S.I. (2022). International Research Journal of Modernization in Engineering Technology and Science. *Design and development of Bluetooth control car based on Arduino Uno*. [online] 4(8), pp.2582–5208. Available at: https://www.researchgate.net/publication/362889441_DESIGN_AND_DEVELOPMENT_OF_BLUETOOTH_CONTROL_CAR_BASED_ON_ARDUINO_UNO.

[31] Lee, J.-S., Su, Y.-W. and Shen, C.-C. (2007). A Comparative Study of Wireless Protocols: Bluetooth, UWB, ZigBee, and Wi-Fi. *IECON 2007 - 33rd Annual Conference of the IEEE Industrial Electronics Society*. [online] doi:https://doi.org/10.1109/iecon.2007.4460126.

[32] Johnathan Tyler Kraus (2023). *The Design and Assembly of a Bluetooth Controlled Remote Controlled Car* . [online] University of Alabama in Huntsville .

Available at: https://louis.uah.edu/cgi/viewcontent.cgi?article=1812&context=honors-capstones [Accessed 24 Apr. 2023].

[33] Sudrajat, M. and Hidayat, R. (2023) "The Remote Control Car System using Bluetooth HC-05 with Android Smartphone", *Jurnal Komputer dan Elektro Sains*, 1(2), pp. 30–34. doi: 10.58291/komets.v1i2.105.

[34] Shashank Venkatesh , Shiva Kundan , Alla Srija (2025). *Obsatacle Avoidance Robotic Vehicle Using Hc-Sr04 Ultrasonic Sensor*. [online] Available at: https://www.proquest.com/openview/df09f412b8ce50fd550156770645e898/1?cbl=2045096&pq-origsite=gscholar.

[35] Lee, J., Hager, G., Dodds, Z. and Choset, H. (n.d.). *6-735, Howie Choset, with slides from Robotic Motion Planning: Potential Functions Robotics Institute 16-735*. [online] Available at: https://www.cs.cmu.edu/~motionplanning/lecture/Chap4-Potential-Field_howie.pdf.

[36] Hager, Z., Dodds and Choset, H. (n.d.). *6-735, Howie Choset with slides from G*. [online] Available at: https://www.cs.cmu.edu/~motionplanning/lecture/Chap2-Bug-Alg_howie.pdf.

[37] Rodriguez, S., Xinyu Tang, Jyh-Ming Lien and Amato, N.M. (2019). An obstacle-based rapidly-exploring random tree. *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.* [online] doi:https://doi.org/10.1109/robot.2006.1641823.

[38] Pradhan, S.K., Parhi, D.R. and Panda, A.K. (2009). Fuzzy logic techniques for navigation of several mobile robots. *Applied Soft Computing*, 9(1), pp.290–304. doi:https://doi.org/10.1016/j.asoc.2008.04.008.

[39] Adarsh, S., Kaleemuddin, S.M., Bose, D. and Ramachandran, K.I. (2016). Performance comparison of Infrared and Ultrasonic sensors for obstacles of different materials in vehicle/ robot navigation applications. *IOP Conference Series: Materials Science and Engineering*, 149, p.012141. doi:https://doi.org/10.1088/1757-899x/149/1/012141.

[40] Hasan, K.M., Al-Nahid, A., Reza, K.J., Khatun, S. and Basar, M.R. (2013). Sensor based autonomous color line follower robot with obstacle avoidance. *2013 IEEE*

Business Engineering and Industrial Applications Colloquium (BEIAC). doi:https://doi.org/10.1109/beiac.2013.6560199.

www.electronicwings.com. (n.d.). *Bluetooth Module HC-05 | Sensors & Modules*. [online] Available at: https://www.electronicwings.com/sensors-modules/bluetooth-module-hc-05-.

Arduino.cc. (2024). *Arduino UNO Rev3 with Long Pins*. [online] Available at: https://docs.arduino.cc/retired/boards/arduino-uno-rev3-with-long-pins/.

HiBit (2023). *How to use the L298N motor driver module*. [online] www.hibit.dev. Available at: https://www.hibit.dev/posts/89/how-to-use-the-l298n-motor-driver-module.