AN-Najah National University

Faculty of Engineering & Information Technology

Computer Engineering Department

Second semester 2025/2026

Course: Graduation project I

Cambot

Written by:

Abdulkreem Abuzahra (ID: 12112886)

Ahmad Tubaila (ID: 12028317)

Under the supervision of: **Dr. Hanal Abuzant**

"Presented in partial fulfillment of the requirements for Bachelor degree in Computer Engineering."

June 2025-2026'

Disclaimer

This report was written by student(s) at the Computer Engineering Department, Faculty of Engineering, An-Najah National University. It has not been altered or corrected, other than editorial corrections, as a result of assessment and it may contain language as well as content errors. The views expressed in it together with any outcomes and recommendations are solely those of the student(s). An-Najah National University accepts no responsibility or liability for the consequences of this report being used for a purpose other than the purpose for which it was commissioned.

Acknowledgment

We would like to express our deepest gratitude to our project supervisor, **Dr. Hanal Abuzant**, for his continued guidance, support, and technical insight throughout the development of this project. His encouragement and expertise played a crucial role in shaping our approach and helped us overcome several engineering and design challenges.

We also extend our sincere thanks to the **Computer Engineering Department** at An-Najah National University for providing the facilities, tools, and environment necessary for completing this project. The lab access and technical resources made it possible for us to prototype and test our ideas effectively.

Special thanks go to our families and friends, whose patience and support helped us stay focused and motivated during the most intensive stages of the project.

Finally, we would like to acknowledge our fellow classmates and professors who provided helpful suggestions, feedback, and moral support throughout the project timeline.

This report and the system we built would not have been possible without the contributions of all the people mentioned above.

Abstract

The increasing need for hands-free video recording in dynamic environments has motivated the development of robotic camera platforms. This project presents Cambot, a prototype mobile robot designed to autonomously track and film a moving subject using onboard image processing while also supporting manual control via a wireless controller.

The system architecture integrates multiple microcontrollers for distributed task management: an Arduino Nano for controller input and NRF24 communication, an Arduino Uno for decoding NRF24 signals, and an Arduino Mega for motor control, ultrasonic obstacle avoidance, and LCD display output. A Raspberry Pi 5 with an OAK-D camera performs real-time image recognition and sends movement commands to the Mega. Mobility is achieved using four Omni-wheels driven by dual L298N motor drivers, while a 12 V/9 A battery with XL4015 buck converters ensures stable power distribution.

Key features include autonomous person-tracking, manual override through NRF24 wireless control, and security access using an RFID system. Safety is enhanced by ultrasonic sensors on all four sides that halt movement if an obstacle is detected within 20 cm. Power management is improved by splitting loads into two switchable subsystems—one dedicated to the Raspberry Pi + OAK-D, and the other to the motor control system. System performance was evaluated through video testing, power consumption measurements, and real-time LCD feedback.

The project demonstrates a scalable and energy-aware approach to robotic filming, providing a prototype suitable for hobbyist and semi-professional use in mobile videography.

Table of Contents

Figure

Chapter one

Introductory

Chapter one: Introductory

1.1 Problem Statement

In modern videography, one of the most significant challenges lies in capturing smooth, continuous footage of moving subjects without requiring a dedicated camera operator. Traditional filming setups depend on handheld stabilizers or human-operated tripods, which can introduce fatigue, shaking, or inconsistent framing. This becomes especially limiting in scenarios such as sports, live events, or personal content creation, where the subject is constantly on the move.

Likewise, available robotic camera solutions are both highly specialized and expensive or too limited in functionality for flexible use. Many lack modularity, making them unsuitable for prototype development or educational environments where experimentation with sensors, wireless control, and computer vision is essential.

From a technical perspective, integrating real-time image processing with reliable robotic mobility introduces further challenges. These include managing power distribution between high-demand components (such as the Raspberry Pi 5 and OAK-D camera), synchronizing multiple microcontrollers for distributed control, and ensuring stable motion with omni-directional wheels. Wireless manual override, obstacle avoidance, and security access add additional layers of complexity that are often not addressed in low-cost robotic filming systems.

The challenge is further extended when adding gesture-based control and intelligent tracking behaviors. Cambot activates tracking only when a person gives a thumbs-up gesture in front of the camera for one second, ensuring intentional engagement. Similarly, tracking stops when a stop sign gesture is held for one second, giving the subject precise control over the filming process. In addition, Cambot incorporates face recognition, allowing it to lock onto a specific person and ignore others, ensuring consistent subject focus. If the tracked person leaves the frame, the system can determine whether they exited to the left or right, then rotate accordingly until the subject reappears, at which point it verifies their identity through facial recognition before resuming tracking.

In many student or prototype-level robotics projects, issues such as multi-microcontroller communication, motor driver synchronization, and wiring management are encountered but rarely implemented in a fully functional system that performs under real-world filming conditions. Thus, there exists a gap between theoretical robotics solutions and practical, user-friendly robotic cameramen capable of delivering reliable filming performance.

This project aims to address these challenges by developing Cambot, a robotic filming assistant that combines real-time person tracking via image processing, gesture-based control, and face recognition with wireless manual override. The system integrates distributed microcontrollers for efficient load balancing, ensures safe navigation through ultrasonic-based obstacle avoidance, and incorporates RFID-based access for security. By focusing on modularity, scalability, and energy efficiency.

1.2 Objectives

The primary goal of this project is to design and implement Cambot, a robotic filming assistant capable of autonomously tracking a person, recording smooth footage, and offering manual override through wireless control. The system is developed to provide reliable subject tracking, modular control, and efficient power management while maintaining flexibility for prototype-level experimentation. The detailed objectives are:

- **Implement autonomous person tracking** using the OAK-D camera and Raspberry Pi 5 with real-time image processing.
- **Enable gesture-based control**, activating tracking when a thumbs-up gesture is held for 1 second and deactivating it when a stop sign gesture is shown for 1 second.
- **Integrate face recognition** to ensure the system tracks only the designated person, ignoring others in the frame.
- **Develop re-entry detection logic**, where the robot identifies if the tracked person left to the left or right side of the frame and rotates accordingly until the subject reappears, and verifying identity before resuming tracking.
- **Provide manual control mode** via a custom-built NRF24-based controller with Arduino Nano, allowing the user to move the robot when desired.
- **Distribute system tasks across multiple microcontrollers**, with Arduino Nano handling controller transmission, Arduino Uno decoding NRF24 signals, and Arduino Mega managing movement, ultrasonic sensors, and LCD output.
- **Ensure stable omni-directional mobility** using four omni wheels driven by dual L298N motor drivers, enabling smooth and flexible movement in all directions.
- **Integrate obstacle avoidance** through ultrasonic sensors on four sides, stopping the robot and rerouting when obstacles are detected within 20 cm.
- **Enhance system security** using RFID-based access, ensuring the robot only operates when the correct card is scanned.

- **Provide user feedback and monitoring** through an LCD with I2C, displaying status information and system behavior in real time.
- **Optimize power management** by dividing the system into two switchable subsystems: one dedicated to the Raspberry Pi + OAK-D, and the other for the motor drivers and microcontrollers, improving efficiency and runtime on the 12V/9A battery.

These objectives aim to deliver a modular, scalable, and user-friendly prototype suitable for mobile videography, bridging the gap between educational robotics projects and semi-professional robotic camera systems.

1.3 Importance of work

This project demonstrates how embedded systems, robotics, and computer vision can be combined to solve real-world challenges in mobile videography. Specifically, it highlights how low-cost, modular technologies can create an autonomous robotic filming assistant capable of intelligent subject tracking and manual override. The importance of this work lies in several key areas:

1. Automation of Filming Processes

Traditional filming requires a dedicated operator to hold or adjust the camera, which introduces fatigue and inconsistency. By automating subject tracking through image processing, gesture recognition, and face identification, Cambot provides smooth and reliable footage without requiring manual intervention.

2. Integration of Real Hardware Systems

The project bridges the gap between academic robotics concepts and functional hardware deployment. It integrates omni-directional wheel mechanics, dual L298N motor drivers, distributed microcontrollers, ultrasonic sensors, and an OAK-D camera, all coordinated into a fully functional robotic platform.

3. Real-Time Monitoring and Control

Cambot provides real-time feedback through an LCD display and supports two modes of control: autonomous image-based tracking and manual override using an NRF24-based wireless controller. Gesture inputs (thumbs-up and stop signs) enhance usability by giving the subject direct control over when filming starts or stops.

4. Modular and Distributed Design

By employing three microcontrollers (Arduino Nano, Uno, and Mega) alongside the Raspberry Pi 5, the system distributes workloads across devices. This modular design improves efficiency, prevents overload on a single board, and makes the system scalable and adaptable for future upgrades in robotics or vision-based applications.

5. Engineering and Design Approach

The project goes beyond trial-and-error by addressing real engineering challenges such as power distribution, current supply for high-demand components, synchronization of multiple processors, and obstacle detection in dynamic environments. These considerations reflect practical engineering thinking suitable for real-world robotic deployments.

6. Educational and Research Relevance

Cambot serves as an academic demonstration of interdisciplinary engineering, combining robotics, embedded systems, wireless communication, power electronics, and artificial intelligence. It provides students and researchers with a working prototype that showcases how theory can be translated into practical solutions.

7. Real-World Impact and Innovation

The system shows how cost-effective, student-developed technology can provide innovative tools for content creators, event recording, or prototype development. It highlights how robotics can enhance everyday tasks such as filming, making it accessible even for individuals or startups that cannot afford high-end robotic camera systems.

1.4 Organization of the Report

This report is organized into several chapters that outline the complete lifecycle of the Cambot project, from initial concept to implementation and testing. Each chapter serves a distinct purpose and builds upon the previous to provide a comprehensive view of the work carried out.

Chapter One: Introduction

Provides the project background, defines the problem being addressed, outlines the project's objectives, and highlights its significance in academic, research, and real-world videography contexts.

Chapter Two: Constraints, Standards, and Component Selection

Discusses the design constraints encountered during development, such as power management, communication load balancing, and motion stability. It also describes safety considerations, industry standards relevant to robotics and embedded systems, and the rationale for selecting specific components including microcontrollers, motor drivers, sensors, and the OAK-D camera.

Chapter Three: Literature Review

Examines previous work in the fields of robotic tracking systems, computer vision-based human detection, omni-wheel mobility, and distributed microcontroller architectures. It positions Cambot in the context of current robotic filming and automation technologies.

Chapter Four: System Methodology and Design

Details the overall system design, including hardware architecture, microcontroller roles, sensor integration, gesture and face recognition, and user interface development. It also presents system-level block diagrams, communication flow between subsystems, and the logic behind person-tracking and obstacle avoidance.

Chapter Five: Implementation and Testing

Covers the hardware assembly, software development, and individual subsystem testing. It discusses results from integrating the Raspberry Pi with the Arduino-based controllers, evaluates tracking accuracy and response time, and presents the performance of the full system under real use-case scenarios.

Chapter Six: Conclusion and Future Work

Summarizes the project outcomes and contributions, and outlines potential improvements such as advanced AI-based gesture detection, enhanced obstacle navigation, improved wireless connectivity, or integration with cloud-based video streaming platforms.

References and Appendices

Provides a list of all academic, technical, and industry sources referenced throughout the report, along with appendices containing full circuit schematics, source code, and supplementary data supporting the project.

Chapter Two

Constraints, Standards/ Codes and Earlier course work

Chapter Two：Constraints, Standards/ Codes and Earlier course work

2.1 Constraints

During the design and implementation of the Cambot system, several constraints shaped the engineering decisions, component selection, and overall architecture. Despite these challenges, the project was successfully developed as a functioning robotic camera assistant capable of face tracking and autonomous movement.

1. Budget Constraints

As a student project, the budget was limited, which required careful selection of components. Low-cost yet reliable solutions such as Arduino microcontrollers, an NRF24L01 wireless communication module, and a repurposed 12V battery were used. Open-source libraries and pre-trained AI models for the OAK-D camera helped minimize development costs without sacrificing performance.

2. Integration of High-Performance Components

The system integrates the OAK-D depth camera, a Raspberry Pi 5 for AI processing, and multiple microcontrollers for motor control. Managing data flow between high-computation vision tasks and low-level motor control required careful hardware-software partitioning to balance workload and ensure real-time responsiveness.

3. Prototype Space and Layout Constraints

The robot base with four omni-wheels, dual L298N drivers, and battery pack had to be compact and modular to allow stable mobility while keeping wiring manageable. This required a precise physical layout to prevent interference, overheating, and mechanical imbalance.

4. Power Management Constraints

The system relies on a 12V 9A battery to power the motors and electronics. Ensuring stable power distribution across motor drivers, microcontrollers, and the Pi while preventing brownouts was a key challenge. Voltage regulation and isolation between logic and high-current motor circuits were necessary to ensure reliability.

5. Multi-Controller Synchronization

The architecture uses a Raspberry Pi for vision and decision-making, an Arduino Mega for movement control, and additional microcontrollers for input handling. Coordinating tasks across these controllers required well-structured communication protocols and careful timing to prevent data loss or command delays.

6. Relative Real-Time Tracking Accuracy

Face recognition and gesture detection needed to work in real-time for smooth operation. Processing delays from the vision pipeline (OAK-D + PI) combined with wireless communication latency to the Arduino required optimization of frame resolution, algorithm selection, and serial communication speed.

7. Mechanical Stability and Precision

Omni-wheels enable movement in any direction but are sensitive to alignment and friction. Mechanical tolerances, driver current limits, and wheel slippage introduced constraints on stability, requiring calibration of motor speeds and fine-tuning of movement algorithms.

8. Time Constraints

The project was completed within one academic semester, which imposed strict deadlines across design, coding, testing, and integration phases. To meet these constraints, modular development was adopted, allowing subsystems (vision, movement, communication) to be developed and tested independently before full integration.

2.2 Problems and Solutions

During the development and testing of Cambot, several hardware and software challenges were encountered. The following outlines the key problems and the solutions implemented to ensure reliable autonomous tracking and control.

1. Power Distribution and Voltage Stability

Problem: The Raspberry Pi 5 and OAK-D camera require a stable high-current supply. Initially, powering all components from a single source caused voltage drops, instability, and occasional resets of the Pi. The order of turning on components also affected performance.

Solution: Two XL4015 buck converters were used: one dedicated to the Raspberry Pi 5 and OAK-D camera, and another to power the remaining microcontrollers, motor drivers, and sensors. Careful sequencing was implemented when powering up the system, turning on the non-critical components first and then the Raspberry Pi and camera to avoid voltage drops.

2. NRF24 Communication Challenges

Problem: Initially, the system was designed with just an Arduino Mega and a Nano for remote control and motor/sensor management. The NRF24 module did not function reliably on the Mega due to communication conflicts.

Solution: An Arduino Uno was introduced to handle NRF24 communication exclusively. The Uno reads commands from the controller and sends simplified 3-bit signals to the Mega, which handles movement and sensor management. This separation ensured reliable wireless communication and reduced load on the Mega.

3. Gesture Recognition Errors

Problem: The system occasionally misinterpreted gestures, reading unintended movements as a thumbs-up or stop sign, causing false activation or deactivation of tracking.

Solution: Software filters were implemented to require the gesture to be held consistently for 1 second before being recognized. This significantly reduced false triggers and ensured intentional activation or stopping of tracking.

4. Tracking and Movement Lag

Problem: The robot could not keep up with a person moving laterally (right or left) because omni-wheel movement was too slow to match the subject's speed. This caused tracking loss when the subject moved quickly across the frame.

Solution: The movement strategy was revised: instead of lateral translation, Cambot performs a rotation (left or right) to re-align with the subject, then measures distance using ultrasonic sensors before moving forward or backward. This approach ensures the robot stays aligned with the tracked person while avoiding obstacles, compensating for its slower lateral speed.

5. Obstacle Avoidance during Image-Based Movement

Problem: Coordinating obstacle avoidance with movement based on image processing introduced delays in response, especially when the robot needed to navigate around obstacles while tracking.

Solution: Movement commands were split into small incremental steps with continuous distance checks from ultrasonic sensors. The robot only advances when the path is clear, reducing collisions and ensuring smooth integration between vision-based tracking and obstacle avoidance.

6. Mechanical and Sensor Calibration

Problem: Initial omni-wheel setup and sensor placement caused occasional misalignment and inaccurate distance readings.

Solution: Wheel angles and motor speeds were carefully calibrated. Ultrasonic sensors were mounted at optimal positions on all four sides to provide consistent obstacle detection, ensuring smooth navigation and accurate tracking.

2.3 Standards and Codes

In the development of Cambot, adherence to technical standards, safe practices, and reliable design principles was essential. While the project is primarily educational, real-world considerations were applied to ensure stable operation, user safety, and system reliability.

Electrical Safety and Power Management

- Power for the Raspberry Pi 5 and OAK-D camera is stabilized using a dedicated XL4015 buck converter with an output capacitor, ensuring smooth voltage supply and preventing resets.
- Motor drivers, sensors, and microcontrollers are powered separately using a second XL4015, isolating high-current loads from sensitive electronics.
- Connectors and insulated wiring were used to prevent short circuits, and motor outputs are automatically set to low when no commands are received to avoid sudden, unsafe movement.

Microcontroller Communication and Logic

- All microcontrollers (Arduino Mega, Uno, Nano) share a common 12V battery supply, with the Nano powered separately at 6V for controller communication.
- Commands from the wireless NRF24 controller are encoded as single characters (e.g., F, B, L, R, P, Q, T) representing forward, backward, left, right, rotation left, rotation right, and AI mode activation.
- Serial communication and logic levels were carefully matched to prevent damage to microcontrollers and ensure reliable command transmission.

Mechanical Components and Motor Safety

- Omni-wheels, motors, and drivers are securely mounted, with protective housings for wiring.
- Mechanical safeguards and code logic prevent motors from moving unexpectedly when no input is present.

Camera and Vision System Standards

- The OAK-D camera is mounted on a stable stand to prevent damage from accidental drops.
- Python-based image processing code ensures safe operation, using depth sensing and face detection features of the camera to manage tracking and obstacle assessment.
- A capacitor is connected to the power supply feeding the Raspberry Pi and camera to stabilize voltage and prevent resets, ensuring smooth and safe operation.

User Interface and Feedback

- An I2C LCD displays the system’s state, including whether AI tracking is active, gestures are recognized, or manual control is engaged.
- Feedback design follows usability principles, providing clear, concise, and real-time updates to the operator.

This combination of safe power distribution, reliable communication, secure mechanical setup, and clear user feedback ensures that Cambot operates effectively and safely, aligning with best practices for robotics prototyping and embedded system design.

2.4 Hardware Components

This section describes the electronic and mechanical components used in the Cambot system. Each component was selected based on functionality, compatibility, and contribution to the overall objective of creating a mobile robot capable of both remote and AI-driven control.

1. **Raspberry Pi 5**

Figure 1:Raspberry Pi 5 4 GB Ram

The Raspberry Pi 5 serves as the primary processing unit for computer vision tasks. It runs the image-processing algorithms that enable person detection, face recognition, and gesture-based activation or deactivation of tracking. Its enhanced CPU and GPU performance compared to previous versions make it suitable for real-time AI applications. The Pi also communicates with the Arduino Mega when AI mode is active, sending navigation commands to drive the robot autonomously.

2. **Arduino Nano**

Figure 2: Arduino Nano

The Arduino Nano is embedded in the handheld remote controller. Powered by a 4 × 1.5V battery pack, it is responsible for reading push-button inputs and transmitting corresponding movement commands via the NRF24L01 wireless module. The Nano provides options for manual navigation such as forward, backward, left, right, and rotation, in addition to a dedicated button to activate AI mode.

3. Arduino Uno

Figure 3:Arduino Uno

The Arduino Uno functions as the receiver unit on the robot side. Equipped with an NRF24L01 module, it collects the commands sent from the remote and encodes them into a 3-bit binary representation. These binary signals are then forwarded through three pins to the Arduino Mega, which executes the actual motor control actions.

4. Arduino Mega 2560

Figure 4: Arduino Mega 2560

The Arduino Mega is the core controller for robot motion. It receives binary commands from the Arduino Uno for remote mode operation and direct movement commands from the Raspberry Pi 5 when AI mode is enabled. Priority is always given to the remote input if a button is pressed, overriding AI control to ensure safe manual intervention. The Mega interfaces with the L298N motor drivers, ultrasonic sensors, and LCD display.

5. NRF24L01 Modules (×2)

Figure 5: NRF24L01

A pair of NRF24L01 wireless transceivers enable communication between the remote (Nano) and the robot (Uno). These modules operate at 2.4 GHz and provide a reliable, low-power communication link with sufficient range for real-time remote operation of the robot.

6. L298N Motor Driver Modules (×2)

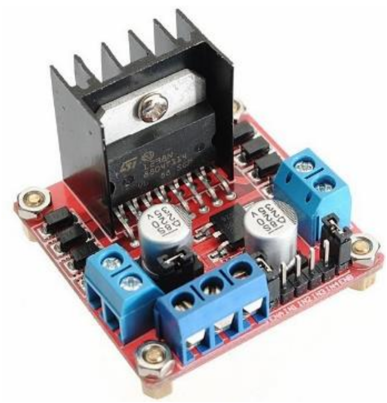

Figure 6: L298N Motor Driver

Two L298N dual H-bridge motor drivers are used to control the four DC motors. Each driver can control two motors, allowing full forward, reverse, and speed modulation capabilities.

The drivers are connected directly to the Arduino Mega for both AI-controlled and remote-controlled navigation.

7. DC Motors (12V ×4)

Figure 7: DC Motors 12V 30RPM

Four 12V brushed DC motors provide the driving force for the Cambot. They are selected for their torque output and compatibility with the L298N drivers, enabling smooth operation under load.

8. Aluminum Omni Wheels x4

Figure 8: Aluminum Omni Wheels (58 mm ×4)

Each motor is fitted with a 58 mm aluminum omni wheel. Unlike standard wheels, omni wheels allow movement in multiple directions without requiring rotation of the entire chassis. This design significantly enhances maneuverability, enabling the Cambot to strafe sideways, rotate on the spot, and follow smooth curved paths.

9. Ultrasonic Sensors (x4)

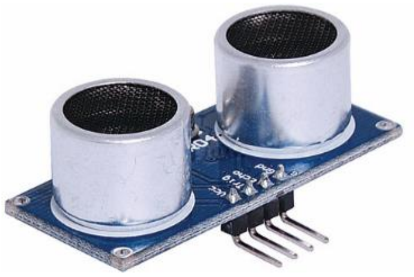

Figure 9: Ultrasonic Sensors (HC-SR04)

Four HC-SR04 ultrasonic distance sensors are mounted on different sides of the Cambot. They provide obstacle detection and avoidance by continuously measuring the distance to nearby objects. The sensors help prevent collisions during AI navigation.

10. Xl4015 DC-DC Buck Converter (×2)

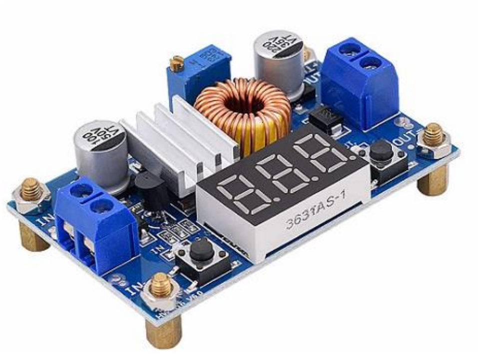

Figure 10: Xl4015 Adjustable DC-DC Buck Converter With 3-Digit 7-Segments display

Two XL4015 DC-DC buck converters regulate the supply voltage from the main 12V battery. One converter exclusively powers the Raspberry Pi 5 and OAK-D camera, while the second powers the Arduino Uno, Arduino Mega, motor drivers, and motors. This division minimizes noise interference between the processing and actuation subsystems.

11. OAK-D Camera

Figure 11: OAK-D Camera

The OAK-D stereo depth camera provides advanced AI capabilities for vision processing. It enables face recognition, and depth-aware person tracking. The camera communicates directly with the Raspberry Pi 5 for real-time analysis. A capacitor is placed across its power supply line to stabilize input voltage and protect against current fluctuations during peak operation.

12. LCD Display (16×2)

Figure 12: LCD Display (16x2)

A 16×2 character LCD display is mounted on the robot to provide real-time system feedback. It shows operation status such as “Remote Mode,” “AI Mode,” or direction of motion. This display assists both debugging and user awareness during operation.

13. I2C Module for LCD

Figure 13:I2C Module

An I2C backpack is used with the LCD display to reduce wiring complexity. This allows the LCD to be controlled using only two data pins from the Arduino Mega, freeing up GPIO resources for other sensors and drivers.

14. Jumper Wires

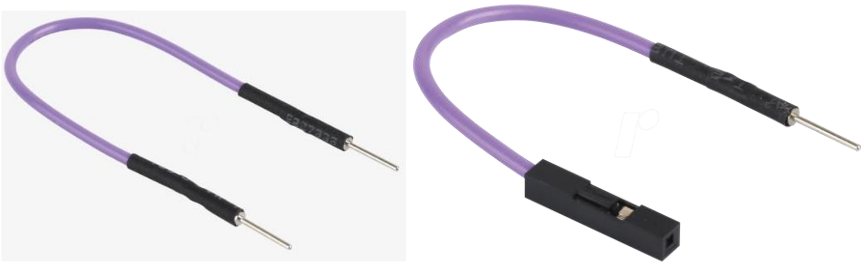

Figure 14:Jumper Wires (Male-Male/Male-Female)

A combination of male-to-male and female-to-male jumper wires are used for prototyping and interconnecting modules. These wires simplify debugging and make the assembly process more modular.

15. PCB Mount Screw Terminal Block Connectors

Figure 15: 5 mm pitch 2-pin PCB mount screw terminal block connectors

Ten pieces of 2-pin, 5 mm pitch PCB mount screw terminal block connectors were used to safely distribute power across the system. Five terminals were dedicated to the +V supply and five to the ground line. Within each group, the pins were permanently joined using metal soldering wire melted across the terminals, ensuring a strong and conductive bus connection.

16. Switches (×2)

Figure 16: ON/OFF Switch KCD1 2 Pins

Two manual switches are installed to control the outputs of the buck converters. Each switch isolates its respective converter, allowing independent shutdown of either the processing side (Raspberry Pi and OAK-D) or the control side (Arduinos, motors, and drivers).

17. Gel battery 12V/9A Along with 4*1.5V Batteries

Figure 17: 12V/9A gel battery & 1.5V batteries

The system is powered by a main 12V rechargeable battery pack, which feeds the buck converters and motor drivers. Additionally, the handheld remote uses a 4 × 1.5V battery pack to power the Arduino Nano and NRF24L01 module.

18. Radio Frequency Identification(RFID)

Figure 18:Radio Frequency Identification(RFID)

The RFID module provides secure identification and access control capabilities. It enables the system to recognize predefined RFID tags for mode activation or restricted operation. The module communicates with the Arduino over a serial interface, ensuring reliable and low-latency data transfer. Proper wiring and connectors are used to protect against short circuits, and the module operates from a regulated 5V supply for stable performance.

2.5 Earlier Coursework and Related Skills

This graduation project, Cambot: AI-Assisted Omni-Wheel Robot, integrates multiple disciplines and skills developed throughout our Computer Engineering program. The design, implementation, and testing of the robot required both theoretical knowledge and hands-on technical expertise.

1. Microcontrollers and Embedded Systems

Coursework involving Arduino (Uno, Nano, Mega) and Raspberry Pi provided the foundation for implementing distributed control. Concepts like serial communication, interrupt-driven design, and PWM-based motor control were directly applied to ensure reliable coordination between motion commands, wireless communication, and real-time AI feedback.

2. Digital and Analog Electronics

Knowledge of voltage regulation, current handling, and interfacing circuits was crucial in safely combining the Raspberry Pi 5, OAK-D camera, and multiple motor drivers with a 12V battery system. Capacitors, connectors, and regulated step-down converters (XL4015) were applied to stabilize power and protect sensitive electronics.

3. Programming and Software Design

Skills in Python (for AI and image processing on the Raspberry Pi) and C/C++ (for Arduino firmware) were essential. Modular coding, use of state machines, and efficient resource management allowed us to implement gesture recognition, wireless control, and autonomous modes without system instability.

4. Computer Vision and Artificial Intelligence

Previous exposure to machine learning and vision frameworks supported the integration of the OAK-D stereo camera. We applied face recognition, depth-based tracking, and gesture classification, directly linking academic knowledge with practical real-time robotics.

5. Networks and Wireless Communication

Experience with communication protocols and IoT concepts helped us implement wireless control using the NRF24L01 module. Each character-based command (F, B, L, R, P, Q, T) was mapped to specific motion or mode functions, ensuring efficient and low-latency command transfer between the transmitter and receiver modules.

6. Circuits and Electrical Safety

Knowledge of circuit protection, safe wiring practices, and power distribution was critical in designing the robot's electrical system. Proper grounding, current-rated connectors, and safety fallbacks in code (e.g., motors defaulting to OFF when no command is received) ensured stable and safe operation.

7. Critical Thinking, Technical Writing and Project Management

Team collaboration, milestone planning, and structured reporting built on project management courses. These skills supported efficient task division, troubleshooting, and the preparation of this final report.

8. Human-Computer Interaction and Interfaces

Coursework on UI/UX and embedded interfaces guided the development of simple but effective feedback mechanisms. The LCD screen provides real-time status updates (e.g., tracking active/inactive), while the mobile app and remote controller enable intuitive human-robot interaction.

Chapter Three

Literature Review

Chapter Three: Literature Review

This chapter explores prior work and relevant technologies that informed the design of our Cambot system. It covers existing person-tracking robots, gesture-based activation, sensor and vision integration, and multi-controller embedded design, establishing the academic and industrial context of our work.

3.1 Vision-Based Person Tracking and Recognition

Vision-guided robotics has been extensively studied in mobile and service robot applications. The use of stereo depth cameras enables robots to perceive distance, detect obstacles, and track human movement in real time. Industrial and research platforms commonly integrate cameras with AI accelerators for face recognition and depth-aware navigation.

The OAK-D stereo depth camera is a proven platform for combining depth sensing with on-board neural network inference. Previous projects often focus on either simple object tracking or simulation-based recognition. However, our implementation extends this work by incorporating both face recognition and depth-aware person tracking in a compact robotic system. The Raspberry Pi 5 processes incoming camera data for real-time analysis, bridging the gap between academic computer vision models and deployable robotic systems.

3.2 Gesture-Based Robot Activation

Gesture recognition is an emerging field in human–robot interaction (HRI). Many commercial and academic projects explore the use of hand tracking and body movements to command robots without traditional controllers. Systems such as drones and service robots increasingly rely on hand gestures for activation and basic commands.

DIY and academic projects often use MediaPipe-based solutions for gesture recognition, typically demonstrated on standard webcams. Our project builds on this by integrating gesture-based start/stop activation directly into Cambot’s control system. This enables natural and intuitive interaction, reducing dependency on physical switches or mobile applications.

3.3 Embedded Multi-Controller Architecture

Traditional robotic platforms often use a centralized microcontroller or single-board computer to manage all sensing and actuation. However, distributed architectures using multiple controllers are gaining attention due to their modularity and reduced processing load on a single device.

Our design follows this decentralized approach. The Arduino Uno reads commands received via the NRF24L01 wireless module and encodes them into a 3-bit format. These signals are then transmitted to the Arduino Mega, which manages motor control through L298N drivers. Meanwhile, the Raspberry Pi 5 handles computationally intensive tasks such as image processing and face recognition. This architecture improves parallel task handling and simplifies system development.

3.4 Sensor Integration and Real-Time Feedback

Robotic navigation systems frequently incorporate multiple sensors to enhance decision-making and safety. Ultrasonic and infrared sensors are widely used in obstacle avoidance and proximity detection, while cameras are used for high-level perception.

In Cambot, the OAK-D camera serves as the primary perception sensor, integrating both depth data and visual recognition. The multi-controller design ensures that sensory information can be processed in real time, enabling stable navigation and person-following behavior. Additionally, capacitors are placed across the power supply lines of sensitive components (such as the OAK-D) to stabilize input voltage and protect against current fluctuations, improving overall system reliability.

3.5 Wireless Control and Remote Communication

Many service robots rely on wireless communication modules for control. NRF24L01 modules are widely used in robotics research and hobbyist applications due to their low cost, low power consumption, and reliable data transmission. Previous projects often demonstrate their use for simple remote-controlled vehicles.

Our design expands on this by encoding movement commands into a 3-bit binary protocol, allowing the Arduino Mega to interpret and execute precise holonomic motion commands. This approach enhances communication efficiency while maintaining reliable remote operation of the robot. Unlike IoT-based mobile control systems, Cambot focuses on direct wireless control and on-board vision processing, aligning more closely with responsive, low-latency robotics applications.

3.6 Gaps in Existing Projects

From our survey, we found that while many robotic projects exist for mobile tracking and gesture interaction, few:

- Combine **stereo depth-based person tracking with real-time face recognition.**
- Include **a gesture-activated start/stop mechanism** in addition to wireless control.
- Employ a **multi-controller architecture** distributing tasks across Pi 5, Mega, and Uno.
- Address **power stability and protection** through capacitor integration.
- Apply **holonomic Omni-wheel mobility** in compact service robotics.

We studied several commercial and academic mobile robots to understand practical design approaches, such as:

- Intel RealSense-based autonomous robots for person-following.
- DIY projects using MediaPipe for gesture recognition.
- Research platforms demonstrating omni-wheel holonomic drive systems.

We studied several commercial and academic mobile robots to understand practical design approaches. For instance, Intel RealSense-based autonomous robots have demonstrated effective depth-guided person-following. Similarly, many DIY projects showcase MediaPipe-based gesture recognition, highlighting its accessibility for human–robot interaction. Finally, numerous research and hobbyist demonstrations explore omni-wheel holonomic drive systems, emphasizing their maneuverability advantages. These references, including publicly available video demonstrations, directly influenced our choices for Cambot's vision processing, control distribution, and mobility design.

Chapter Four

Methodology

Chapter Four: Methodology

4.1 The system consists of two major sections:

1. **Perception and Processing Unit**

   A vision-based module built around the OAK-D stereo depth camera, which performs real-time face recognition and depth-aware person tracking. The camera communicates directly with the Raspberry Pi 5, which processes vision data and manages decision-making for navigation. To ensure stable operation, a capacitor is connected across the power supply line of the OAK-D to stabilize voltage and protect against current fluctuations during peak processing loads.

2. **Robotic Mobility and Control Unit**

   A mobile robotic platform with four omni-directional wheels driven by dual L298N motor drivers. Movement commands are transmitted via an NRF24L01 wireless module. The remote control uses an Arduino Nano, which encodes joystick and button inputs into 3-bit binary signals representing movement directions and AI activation. These signals are received by an Arduino Uno, which forwards them to the Arduino Mega. The Mega executes motor control logic, while the Uno also handles system monitoring, sensor feedback, and coordination between the controllers. This setup ensures safe, reliable, and responsive navigation across multiple directions (forward, backward, lateral, and rotational).

4.2 System Block Diagram

Figure 19:System Block Diagram

4.3 Control Flow Summary

- **Startup:** Cambot powers on with all controllers and sensors initialized, but neither the remote nor the Raspberry Pi can control movement until a valid RFID card is scanned. Once the RFID is verified, the remote controller can operate the robot manually. AI mode on the Raspberry Pi remains inactive until the user activates it via a dedicated button on the remote.

- **User Input:** In AI mode, the system waits for the user to perform a gesture in front of the camera (thumbs-up to start tracking, stop sign to deactivate). Manual control commands from the remote are also accepted and will immediately override AI mode.
- **Tracking / AI Mode:** Once the gesture is detected, the Raspberry Pi performs face detection and saves the tracked person's face for safe recognition. The robot then begins autonomous tracking, sending movement commands to the Arduino Mega. The LCD provides continuous feedback:
    - **First line:** Displays AI mode status (AI:OFF/ON) and camera status (C:ON). When a person is being tracked, it updates to T:(ON/OFF).
    - **Second line:** Displays real-time movement commands sent from Raspberry Pi to Arduino Mega, such as forward, backward, or rotation commands.
- **Movement Control:** Arduino Mega receives commands from either the Raspberry Pi (AI mode) or Arduino Uno (remote control). It drives the omni-wheel motors, manages ultrasonic sensors for obstacle detection, and updates the LCD display. AI mode is automatically disabled if any manual movement button is pressed on the controller.
- **Monitoring / Feedback:** The system continuously updates the LCD with AI status, camera activation, tracking state, and the current movement commands. Ultrasonic sensors ensure safe navigation while tracking, and any loss of the tracked person triggers rotation and search commands until the person is detected again.

4.4 Power and Movement Calculations

- **Battery Load:**
    - Raspberry Pi 5 + OAK-D camera: ~3 A at 5V.
    - Four DC omni-wheel motors: ~2 A total at 12V.
    - Arduino Mega, Uno, Nano + sensors + LCD: ~0.5 A at 12V.
- **Estimated Runtime:**
    - Using a 12V 9A battery:
        - Total current ≈ 2 + 0.5 + (3×5V buck conversion) ≈ ~6 A equivalent.
        - Runtime ≈ 9 Ah / 6 A ≈ 1.5 hours.

    - Using a 6V battery for Remote Control:
        - The remote control, powered by 4 × 1.5V AA batteries (6V, 2500 mAh), drives an Arduino Nano (~50 mA average) and an NRF24 wireless module (~60 mA average transmit bursts), giving a total average current of ~110 mA.
        - Estimated runtime ≈ 2500 mAh / 110 mA ≈ 22–23 hours.
- **Movement / Tracking Timing:**
    - Rotation time to realign with person: ~1–2 seconds per 90° turn.
    - Forward/backward movement based on ultrasonic distance: 0.2 m/s average speed, suitable for tracking a walking person in confined space.

4.6 Safety and Efficiency Design

- **Power Safety with XL4015:** The Raspberry Pi 5 and OAK-D camera are powered through a dedicated XL4015 buck converter with output capacitors to stabilize voltage, ensuring safe and reliable operation without voltage drops. Other components are powered separately, preventing overload on the main power supply.

- **Motor Current Limits:** All omni-wheel DC motors are driven via LM298 motor drivers, with current limits set to prevent overcurrent situations. Sudden movements are avoided by defaulting motors to low unless a valid command is received.

- **Modular Architecture**: The system uses multiple controllers—Arduino Mega for movement, sensors, and LCD feedback; Arduino Uno for reading commands from the remote; Arduino Nano on the remote; and Raspberry Pi 5 for image processing. This separation reduces interference and prevents system crashes, ensuring safe and smooth operation.

- **Display Feedback**: LCD screens provide real-time status updates, including AI mode (on/off), camera state (C: ON or T: tracking), movement commands received from the Raspberry Pi, and any detected gestures. This allows the user to monitor system operation at all times.

- **Secure Start Procedure:** The system only activates after the RFID card is presented. The remote can control movement immediately, but AI tracking mode on the Raspberry Pi is only enabled via a dedicated button, preventing accidental activation.

Chapter Five

Implementation and Testing

Chapter Five: Implementation and Testing

This chapter details the process of building the system, connecting hardware components, writing and uploading the control code, and validating performance through systematic testing.

5.1 Hardware Assembly

The implementation phase for Cambot involved assembling the mechanical platform, connecting all electronic components, and ensuring safe power distribution and communication between controllers.

Main Robot:

- **Mobility:** Four omni-wheel DC motors mounted on a stable chassis, each driven by an LM298 motor driver. The wheels were positioned to allow smooth omnidirectional movement, with careful testing to determine optimal angles.
- **Control Boards:** An Arduino Mega 2560 manages motor control, ultrasonic sensors, LCD feedback, and RFID input. An Arduino Uno reads commands from the remote control via NRF24 and relays them to the Mega. The Raspberry Pi 5 handles image processing using the OAK-D camera.
- **Sensors and Feedback:** Ultrasonic sensors are mounted on four sides to detect obstacles and distances. A 20x4 I2C LCD displays AI mode, camera status, tracking status, and movement commands. An RFID reader ensures secure activation before any operation.
- **Power Distribution:** Two XL4015 buck converters are used—one powers the Raspberry Pi 5 and OAK-D camera, with output capacitors for voltage stabilization, and the other powers the rest of the components. Proper startup order is followed: first the general components, then the Pi and camera to prevent voltage drops.
- **Connections and Safety:** All wires are properly insulated and managed to prevent short circuits or interference. Modular controller architecture reduces the risk of system crashes or overloads, and motors default to low until a valid command is received.

Remote Control:

- An Arduino Nano interfaces with an NRF24 module powered by a 4 × 1.5V AA battery pack (6V). Buttons on the remote send commands (Forward, Backward, Left, Right, Rotate Left, Rotate Right, and Activate AI) to the Arduino Uno on the robot, which forwards the signals to the Mega. The Nano and NRF24 assembly is compact and securely housed to prevent accidental disconnections.

Power-On Sequence and Safety Measures:

- The system remains inactive until a valid RFID card is presented. Only after this is the remote control enabled, and AI mode on the Raspberry Pi is activated via a dedicated button on the remote.
- Capacitors on the XL4015 output ensure stable voltage to the Pi and camera, protecting sensitive electronics during startup.
- All high-current and high-frequency lines are physically separated from signal lines to prevent interference, and all modules are mounted on nonconductive surfaces when needed.

5.2 Software Development

The Cambot control software consists of Python code running on the Raspberry Pi 5 for image processing and C++ code for the microcontrollers. Each module is responsible for a specific set of tasks to ensure safe and efficient operation.

Raspberry Pi 5 (Image Processing and AI):

- Performs real-time image processing to detect gestures for activating (thumb up) or deactivating (stop sign) AI tracking mode.
- Uses face recognition to ensure the robot tracks only the designated person.
- Monitors if the tracked person leaves the frame from the right or left side and sends rotation commands to the Arduino Mega until the person is located again.

- Sends movement instructions and AI mode status to the Arduino Mega via serial communication.

Arduino Mega 2560:

- Controls omni-wheel motors via LM298 drivers based on commands from the Raspberry Pi or remote control.
- Reads ultrasonic sensors for obstacle detection and adjusts movement accordingly.
- Processes RFID input to ensure the system only operates after valid authorization.
- Updates the 20x4 LCD with real-time feedback, including AI mode status, camera state, tracking status, and movement commands.
- Ensures motors remain idle if no valid command is received, preventing sudden or unsafe movement.

Arduino Uno (Robot Side):

- Reads commands from the remote control using the NRF24 wireless module.
- Relays validated movement commands to the Arduino Mega, distributing workload efficiently and maintaining system stability.

Arduino Nano (Remote Control):

- Sends movement commands and AI mode activation to the Arduino Uno via NRF24 wireless communication.
- Powered by a 4 × 1.5V AA battery pack, optimized for low current draw to ensure long runtime.

Software Safety and Modularity:

- Each controller operates independently to reduce the risk of system crashes or interference.
- The robot only becomes active after a valid RFID card is presented. The remote can control movement immediately, but AI tracking mode on the Raspberry Pi must be explicitly enabled by the user.
- Communications between controllers are validated to prevent unintended commands.

- Capacitors are used on the power supply to the Raspberry Pi and camera to stabilize voltage and protect sensitive components.

This modular software architecture ensures reliable, real-time tracking, safe motor operation, and clear user feedback through both gesture control and remote commands.

5.3 Integration and Communication

To ensure smooth operation and prevent overload or interference between controllers, the Cambot system employs a modular and coordinated communication structure:

- The **Arduino Uno** operates independently to handle incoming commands from the remote control via the NRF24 module. It validates these commands and relays them to the Arduino Mega, effectively offloading the Mega from direct wireless communication and reducing workload.
- The **Arduino Mega** manages all motors, ultrasonic sensors, RFID input, and the 20x4 LCD display. It receives movement and AI mode instructions both from the Uno (remote control) and the Raspberry Pi (image processing and AI tracking).
- The **Raspberry Pi 5** communicates with the Mega over a serial connection, sending commands related to gesture detection, face recognition, and AI tracking mode. It only activates robot movement once AI mode is explicitly enabled and a valid face is recognized.
- To maintain signal integrity, all controllers share a **common ground**, and voltage levels are carefully matched between serial and wireless interfaces.
- Modular separation ensures that each controller can operate independently when necessary, minimizing the risk of crashes or interference. For example, the Mega continues to manage motor safety and sensor monitoring even if the Pi temporarily stops processing.

5.4 Testing Procedures

Each component and subsystem of Cambot was tested individually and then in integrated operation to ensure reliable performance and safe operation:

- **Gesture Recognition and AI Mode Activation:** The Raspberry Pi 5 and OAK-D camera were tested to ensure that the robot only enters AI tracking mode when a valid gesture (thumb up) is detected for one second. Similarly, the system correctly deactivates AI mode when a stop-sign gesture is held for one second. False positives were monitored and minimized during testing.
- **Face Recognition and Tracking:** After detecting the activation gesture, the system successfully identifies and stores the designated person's face. The robot maintains tracking of that person only, even if others enter the frame, and resumes tracking if the person briefly leaves and reappears from any direction.
- **Motor and Movement Testing:** All four omni-wheel motors were tested for forward, backward, and rotational movements. Movements were coordinated based on commands from both the remote control and AI mode. Obstacle avoidance using ultrasonic sensors was verified to prevent collisions and maintain smooth tracking. The rotation response time to realign with a person leaving the frame was measured to be approximately 1–2 seconds per 90° turn.
- **Remote Control Communication:** The Arduino Nano and Uno modules were validated for reliable wireless transmission of commands via NRF24. Command reception, relaying to the Mega, and execution by the motors were monitored to ensure minimal latency and no command loss.
- **Ultrasonic Sensors and LCD Feedback:** Ultrasonic distance measurements were tested to ensure correct obstacle detection and safe navigation. The 20x4 LCD continuously displayed AI mode status, camera activation, tracking status, and movement commands. Feedback accuracy was verified in both AI and manual control modes.
- **Power and Voltage Stability:** The XL4015 power modules with capacitors were tested to ensure stable voltage supply to the Raspberry Pi and OAK-D camera during full operation. Power distribution to other components was verified to prevent voltage drops or unexpected resets.

- **RFID Authorization:** The system correctly remained inactive until a valid RFID card was presented. Both AI mode and remote control operation were only allowed after proper authorization.
- **Integrated System Testing:** Finally, the complete system was tested in real-world conditions, combining gesture control, AI tracking, remote control, motor movement, obstacle avoidance, and LCD feedback. The robot successfully followed a moving person, responded to manual remote commands, and provided continuous visual feedback without errors or unsafe behavior.

5.5 Results

- **AI Tracking and Gesture Recognition:** Cambot reliably entered AI tracking mode when a thumb-up gesture was detected for one second, and successfully deactivated on a stop-sign gesture. Face recognition accurately identified and tracked the designated person, ignoring others in the frame, and resumed tracking if the person briefly left and returned from either side.
- **Movement and Obstacle Avoidance:** The Omni-wheel motors performed smooth forward, backward, and rotational movements. Ultrasonic sensors effectively prevented collisions, maintaining a safe distance of approximately 20 cm from obstacles. Rotation response to realign with a person leaving the frame averaged 1–2 seconds per 90° turn, and forward/backward speed was sufficient to follow a walking person in a confined space.
- **Remote Control Performance:** Commands sent via the NRF24 from the Arduino Nano to the Arduino Uno and relayed to the Mega were executed accurately, with minimal latency. The system allowed seamless switching between manual and AI modes.
- **LCD Feedback and System Monitoring:** The 20x4 LCD displayed real-time status, including AI mode (on/off), camera activation, tracking status, and movement commands. Feedback was clear and updated consistently without flicker.

- **Power and Runtime:** Power distribution using XL4015 modules with capacitors provided stable voltage to the Raspberry Pi and OAK-D camera. The remote control powered by 4 × 1.5V AA batteries ran for approximately 22 hours under typical use, while the main robot operated for about 1.5 hours using a 12V 9A battery.
- **Overall System Reliability:** The integrated system successfully combined AI tracking, gesture control, remote operation, obstacle avoidance, and real-time feedback, operating safely and efficiently in prototype-level testing scenarios.

Chapter Six

Conclusion and Future Work

Chapter Six: Conclusion and Future Work

6.1 Conclusion

This project demonstrates the successful design and realization of **Cambot**, a mobile robotic platform that integrates computer vision, wireless communication, and embedded control. The system combined the strengths of a **Raspberry Pi** for image processing, an **Arduino Nano** for motor actuation, and peripheral modules such as the LCD display and screw terminal power distribution to create a reliable and flexible robot.

The OAK-D camera provided depth estimation and face/gesture recognition, enabling Cambot to interpret commands beyond traditional manual control. Motor drivers, ultrasonic sensors, and rotation/translation control logic ensured stable navigation. Safety was prioritized through the inclusion of capacitors for voltage stabilization, protected screw-terminal wiring, and code safeguards to prevent unintended motor activation.

The final prototype demonstrated smooth operation under different scenarios, responsive real-time control through character-based wireless commands, and effective AI-assisted behavior. The project not only confirmed the practicality of combining off-the-shelf modules into an intelligent robot but also highlighted the educational value of multidisciplinary integration across hardware, software, and AI.

6.2 Future Work

Although the current system is functional, several areas can be extended in future iterations:

1. Improved Obstacle Detection with LiDAR

In the current design, one ultrasonic sensor is placed at the center of each side of the robot. This limits detection coverage, as objects at the edges may be missed. Moreover, the elevated camera stand poses a risk of colliding with tall obstacles such as tables. As a future improvement, the ultrasonic sensors can be replaced or complemented with a LiDAR system, offering 360° scanning, higher accuracy, and the ability to detect both edge-level and elevated obstacles.

2. Enhanced Locomotion with Multi-Motor Chain Drive System

The omni wheels occasionally slide on uneven surfaces, even with a slight inclination. To overcome this, a novel approach inspired by bicycle chain systems is proposed. The design consists of **four motors per side** in a triangular arrangement—three at the corners for normal ground movement and a fourth motor in the middle to drive the chain synchronously. This setup would enable climbing small stairs and rough surfaces while maintaining stability.

3. Integrated Touch Screen for Local Monitoring

A touchscreen display can be added to the front of the robot to provide live feedback from the OAK-D camera. This would allow the speaker to see how they are framed in the video in real time, enhancing usability in presentations or interactive scenarios.

4. Mobile Application for Remote Streaming and Storage

A dedicated mobile app can be developed to connect with the robot, enabling users to view the live video feed from the OAK-D camera and store recorded sessions directly onto their smartphones for later review.

5. High-RPM Motors for Improved Maneuverability

Currently, the 30 RPM motors limit the robot's speed and fine maneuverability. By using higher RPM motors, the system could gain better speed control, allowing smoother lateral adjustments (moving left or right without full rotation), improving precision in person-tracking scenarios.

6. System Consolidation Using a Single Arduino Mega

The existing setup requires both an Arduino Uno and Arduino Mega because the Mega alone cannot reliably interface with the NRF24L01 module. In the future, the hardware and software can be optimized to integrate all functions into a single Mega board, reducing complexity, wiring, and power consumption.

7. Battery Management System (BMS) Integration

To improve safety and efficiency, a proper BMS could be added to monitor voltage, current, and temperature of the batteries, ensuring balanced charging/discharging and extending battery life.

8. Automatic Mode Switching Between AI and Manual Control

Currently, the remote push button selects AI mode or manual mode. A smarter system could detect signal loss from the remote and automatically switch to AI mode, or revert to manual priority if human input is detected, ensuring seamless transitions.

6.3 References

6.3.1 Technical and Engineering References:

1. Arduino Documentation – https://www.arduino.cc
2. Raspberry Pi Documentation – https://www.raspberrypi.com/documentationc
3. Luxonis OAK-D DepthAI Camera Documentation – https://docs.luxonis.com
4. **IEC 60364** – Low-voltage electrical installations – International Electrotechnical Commission.
5. **IEC 61010 –** Safety Requirements for Electrical Equipment for Measurement, Control, and Laboratory Use.
6. XL4015 Buck Converter Datasheet – Texas Instruments.

   Available at: https://datasheets.maximintegrated.com/en/ds/MAX31865.pdf
7. L298N and Motor Driver Modules Reference Guides.
8. STMicroelectronics: Fundamentals of Capacitor-Based Power Filtering.

6.3.2 Video & Media References:

9. **DepthAI - Step By Step tutorial For Using OAK-D – YouTube, LearnOpenCV**

   Link: https://www.youtube.com/watch?v=e_uPEE_zlDo

10. **4WD Omni Wheel Arduino Robot Car**

    YouTube, Luo Henry.

    Link: https://www.youtube.com/watch?v=HBdZzmTGeaI

11. **Raspberry Pi Computer Vision Tutorials**

    YouTube, Core Electronics.

    Link:

    https://www.youtube.com/playlist?list=PLPK2l9Knytg7O_okVr-prI1KbZ8GJeMKz