UNIVERSITY OF APPLIED SCIENCES, HAMM-LIPPSTADT, GERMANY

SYSTEMS ENGINEERING AND PROTOTYPING

DESIGN AND IMPLEMENTATION OF A LINE FOLLOWING AND OBSTACLE AVOIDANCE ROBOT

A COURSE WORK PROJECT FOR SUMMER SEMESTER 2025:

A PREREQUISITE FOR COMPLETION OF PROTOTYPING AND SYSTEMS ENGINEERING COURSE AT THE UNIVERSITY OF APPLIED SCIENCES, HAMM-LIPPSTADT

PRESENTED TO:

PROFESSOR STEFAN HENKLER
DR. FAEZEH PASANDIDEH
MR. GIDO WAHRMANN

BY:

GROUP B TEAM 4

CHIMEZIE, DANIEL CHIDI / MAT.NR. 1230515 /
DANIEL-CHIDI.CHIMEZIE@STUD.HSHL.DE

CHO, BERTRAND MUNGU / MAT.NR. 2220052 /
BERTRAND-MUNGU.CHO@STUD.HSHL.DE

GHIMIRE, RIWAJ / MAT.NR. 1232171 /
RIWAJ.GHIMIRE@STUD.HSHL.DE

JULY 8, 2025

1.0 EXECUTIVE SUMMARY

The rapid advancement of automation and embedded systems have paved the way for the development of intelligent robotic systems that can perform tasks with minimal human intervention. This project focuses on designing and implementing an autonomous robotic vehicle that can follow a defined path and avoid obstacles using a combination of infrared (IR), ultrasonic and color sensors. The goal is to develop a functional prototype that simulates intelligent pathfinding and decision-making mechanisms suitable for real-world environments such as automated delivery systems or industrial navigation.

The robotic system is equipped with IR sensors to detect black lines on a contrasting surface, enabling it to follow a designated route. Ultrasonic sensors serve as proximity detectors, allowing the robot to recognize and react to obstacles within its path. A color sensor is also integrated to enhance decision-making by identifying the color characteristics of encountered objects. Based on predefined color criteria, the robot can classify objects such as `Color A` and `Color B`, influencing its navigation behavior accordingly [2].

A key highlight of the system is its modularity and low cost. The robot is built on a simple three-wheel chassis and powered by an Arduino Uno R4 Wi-Fi microcontroller. Peripheral components, including an L298N motor driver, a 12 V LiPo battery, and a power switch, provide essential control and energy management. This architecture supports expandability and makes the robot suitable for educational and research-oriented applications[4].

The software architecture is implemented using the Arduino IDE [1] in embedded C++, with a focus on real-time performance. The line-following algorithm relies on digital feedback from the IR sensors, while obstacle avoidance is triggered by distance thresholds defined for the ultrasonic sensor [2]. When an obstacle is detected, the robot either pauses or takes a detour, depending on the color of the object. For example, if the color sensor identifies a specific hue within a known RGB ratio range (Color A or B), the robot modifies its path accordingly, by either rerouting, stopping or re-centering on the line after bypassing the obstacle.

The development process was iterative, involving extensive hardware calibration and software testing under different lighting and surface conditions. The IR sensors were tuned for high contrast detection, and ultrasonic sensor readings were smoothed to minimize false positives. Color detection was verified using known samples to ensure consistent classification accuracy [2].

The robot we developed reliably demonstrates autonomous line-following behavior, detects and avoids obstacles within a 25 cm range, and reacts differently based on object color. This project serves as a practical proof-of-concept for sensor integration and autonomous decision-making in embedded systems, laying the foundation for more advanced robotics projects in the future.

2.0 PROJECT OBJECTIVES AND SCOPE

2.1 Objective:

The primary objective of this project is to design and develop an autonomous robotic vehicle capable of navigating dynamic environments using a sensor driven control system.

The following key goals guide the design and implementation phases:

a) **Autonomous Lane Following**: Develop a reliable lane-following mechanism using Infrared (IR) sensors. The system should be capable of detecting and tracking pre-defined lanes with precision, including handling curves and intersections effectively.

b) **Obstacle Detection and Avoidance:** Integrate ultrasonic sensors to facilitate real-time obstacle detection. The robotic vehicle should be able to measure distance to nearby objects and initiate avoidance manoeuvres autonomously to prevent collisions[2].

c) **Colour-Based Decision Making:** Implement a colour recognition system to classify and respond to different types of obstacles. For example, red objects may signal a full stop, while green may indicate a clear path or require an alternate route. This enables dynamic and context-aware navigation.

d) **Modular and Power-Efficient Design:** Design the hardware to be modular, allowing easy replacement and testing of individual components. Power efficiency will be emphasized to prolong operational time, utilizing optimized power management strategies across all modules.

2.2 Project scope:

This project encompasses both hardware and embedded software development, with a strong emphasis on reliability, sensor fusion, and system integration. Key aspects of the scope include:

a) **Hardware Integration and Mechanical Design:** Selection, configuration and interfacing of sensors (IR, Colour and ultrasonic), motor drivers, microcontrollers, and power supply units. Custom mechanical components such as sensor holders, battery enclosures, and structural frames were designed using SolidWorks. This CAD-based approach allowed for accurate measurements, modularity, and efficient assembly of all physical parts[4].

b) **Software Development:** Development of embedded firmware to process sensor data, control actuators, and execute decision-making algorithms. Emphasis will be placed on efficient code structure, interrupt-driven design and real-time responsiveness.

c) **System Testing and Validation:** Rigorous testing will be conducted to evaluate the system's performance under different environmental conditions. This includes assessing the accuracy of lane-following, obstacle detection range, decision logic correctness, and overall system robustness.

d) **Scalability and Extensibility:** While the initial prototype will demonstrate core functionalities, the system will be designed with future enhancements in mind, including potential wireless communication, machine learning integration, or advanced navigation algorithms.
In summary, this project aims to deliver a functional, intelligent robotic vehicle by effectively integrating sensing, decision-making, mechanical design, and control into a unified, modular, and energy-efficient system.

3.0 PROJECT METHODOLOGY

The primary objective of this project is to design and develop an autonomous robotic vehicle capable of navigating dynamic environments using a sensor-driven control system.

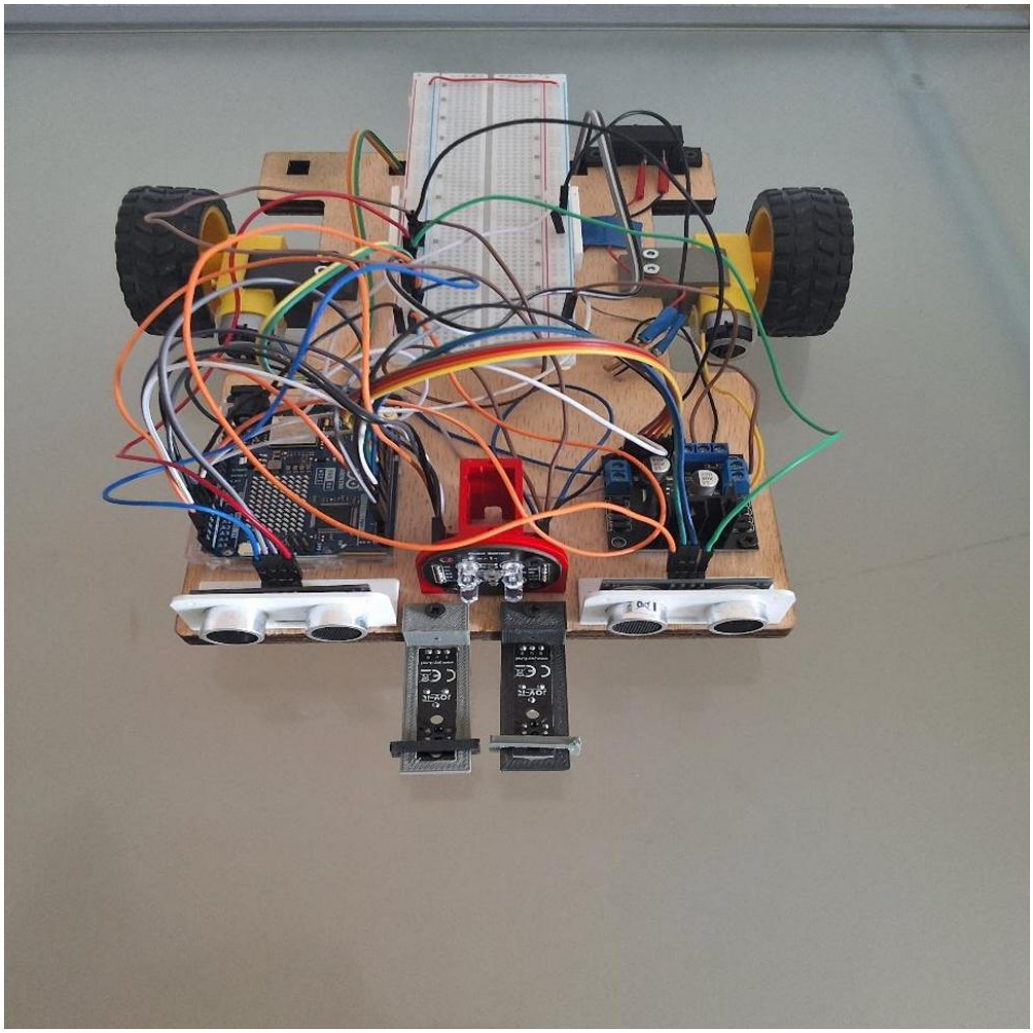
The following key goals guide the design and implementation phases:

a) **Autonomous Lane Following:** To develop a reliable lane-following mechanism using Infrared (IR) sensors. The system should be capable of detecting and tracking pre-defined lanes with precision, including handling curves and intersections effectively.

b) **Obstacle Detection and Avoidance:** To integrate ultrasonic sensors to facilitate real-time obstacle detection. The robotic vehicle should be able to measure distance to nearby objects and initiate avoidance manoeuvres autonomously to prevent collisions [2].

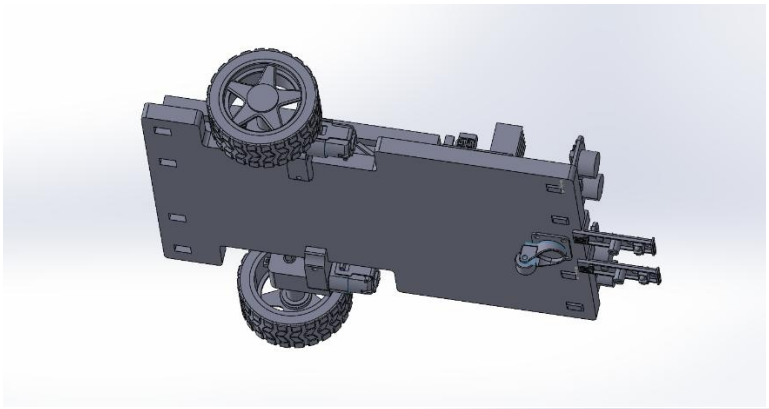
c) **Colour-Based Decision Making:** To implement a colour recognition system to classify and respond to different types of obstacles. For example, red objects may signal a full stop, while green may indicate a clear path or require an alternate route. This enables dynamic and context-aware navigation.

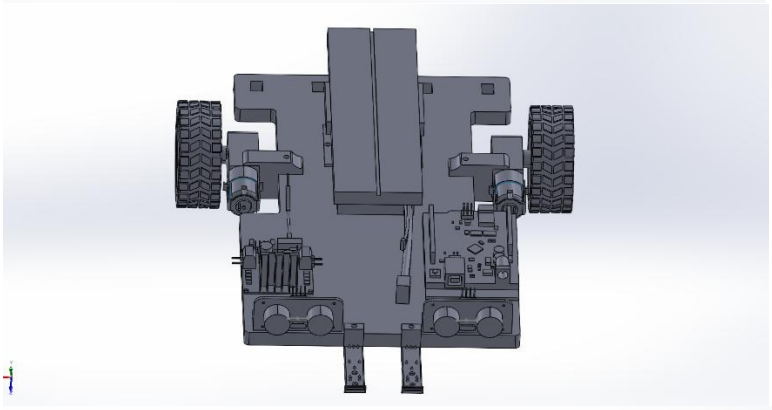
d) **Modular and Power-Efficient Design:** Design the hardware to be modular, allowing easy replacement and testing of individual components. Power efficiency will be emphasized to prolong operational time, utilizing optimized power management strategies across all modules.

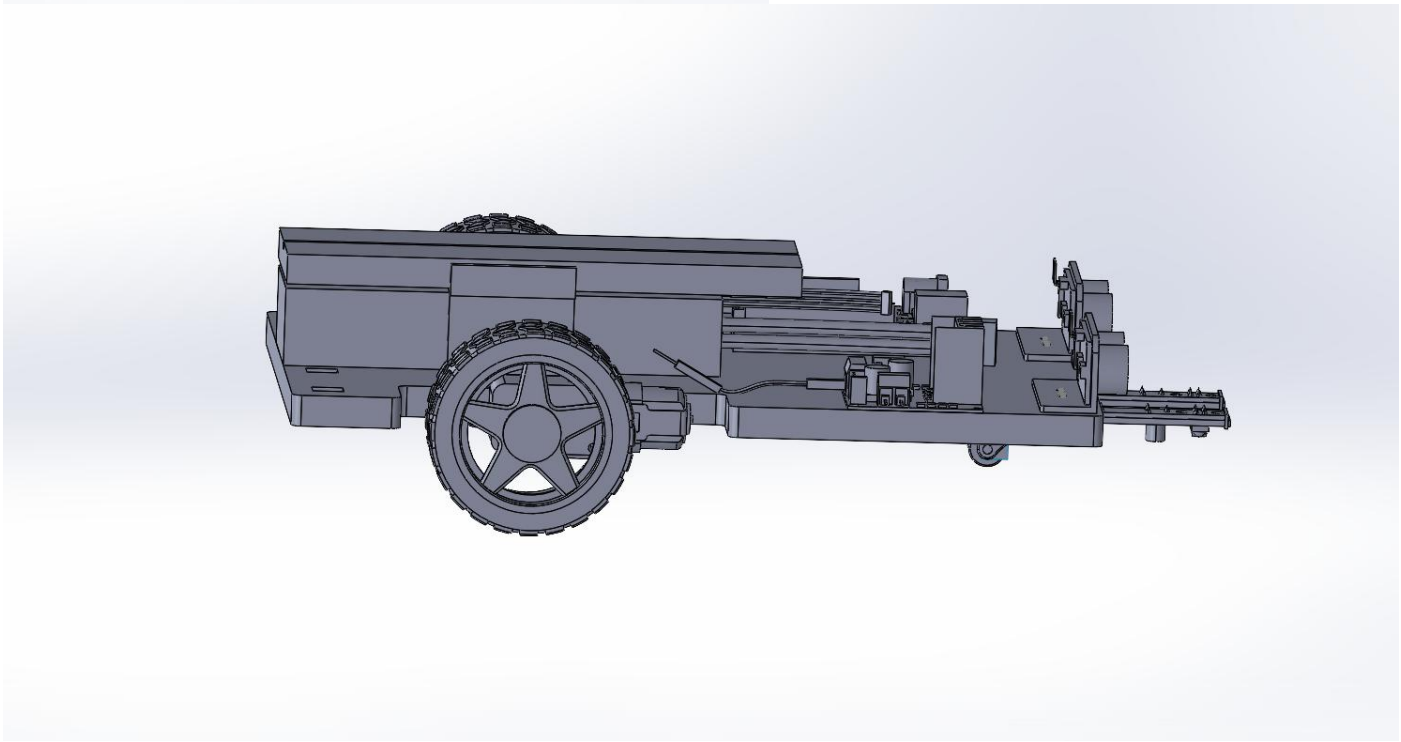


3.1 Mechanical Designs:

CAD Designs of spare parts using SolidWorks Software

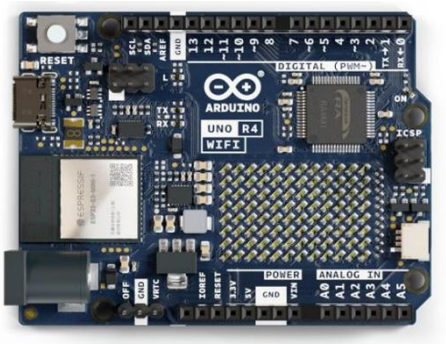




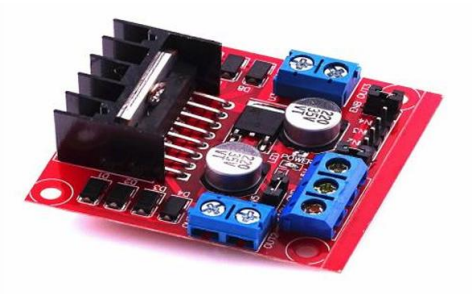


3.2 Electrical Components

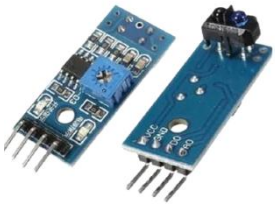
Arduino Uno R4 WiFi



SBC-Motor Driver 2



IR Sensor

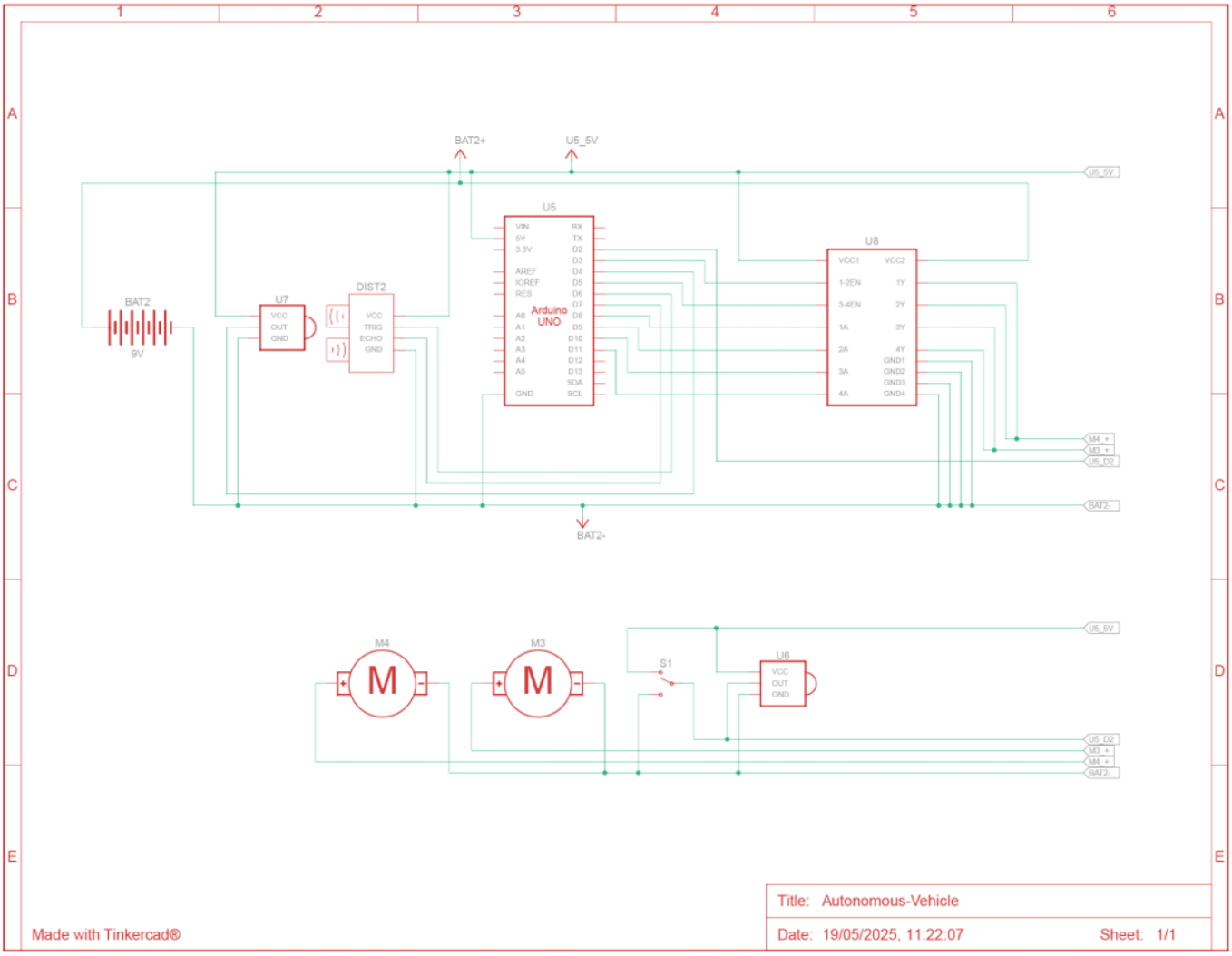


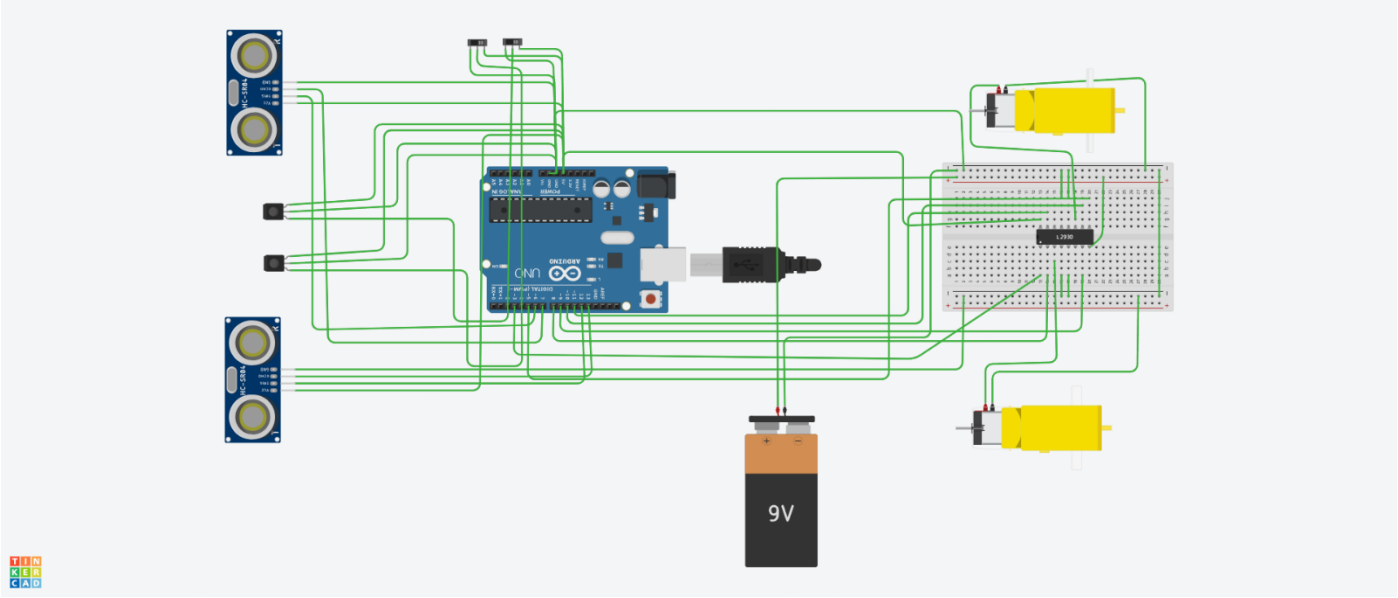
HC-SR04P Ultrasonic Sensor

TCS3200 Colour Sensor

DC Motor

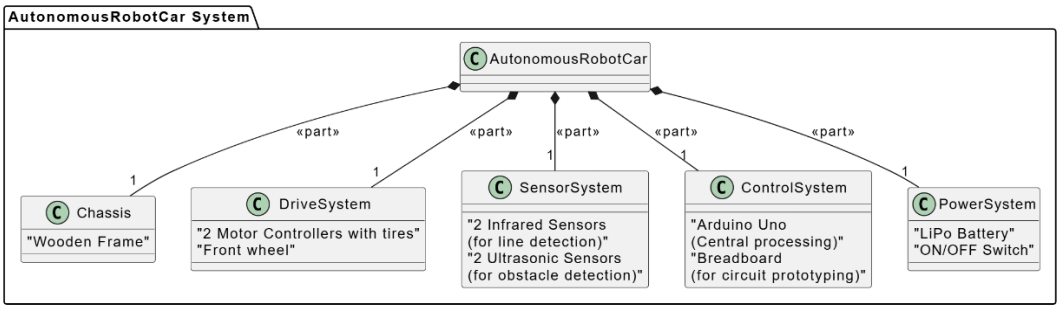
3.2.1 Electrical Schematics and Arrangements:



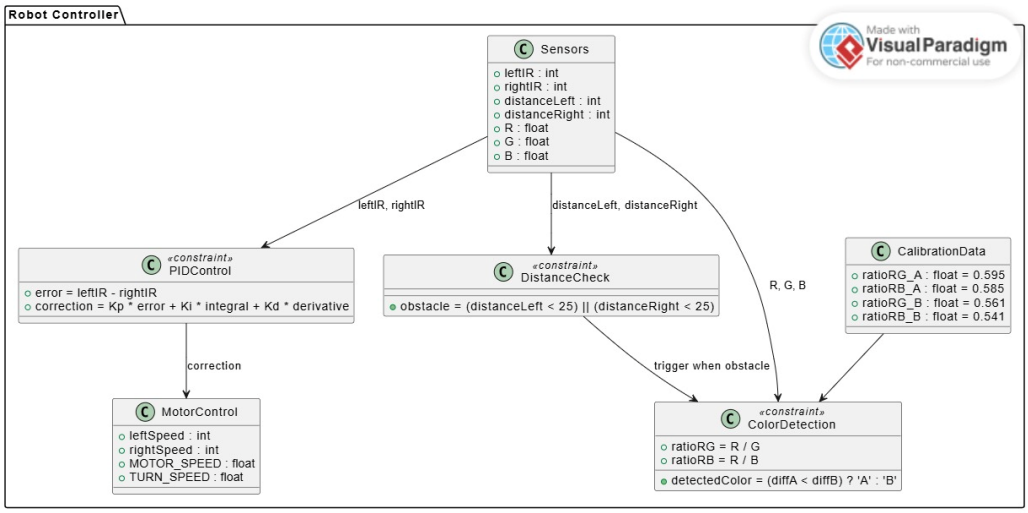


3.3 System Engineering Architecture

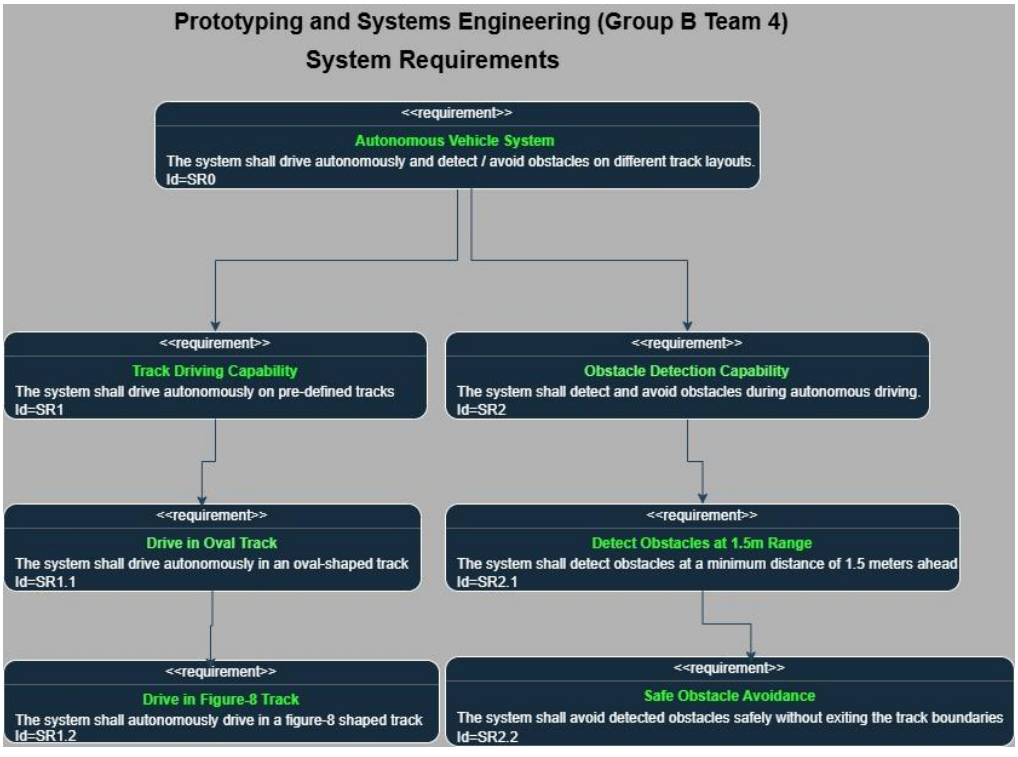
3.3.1 Block Definition Diagram:



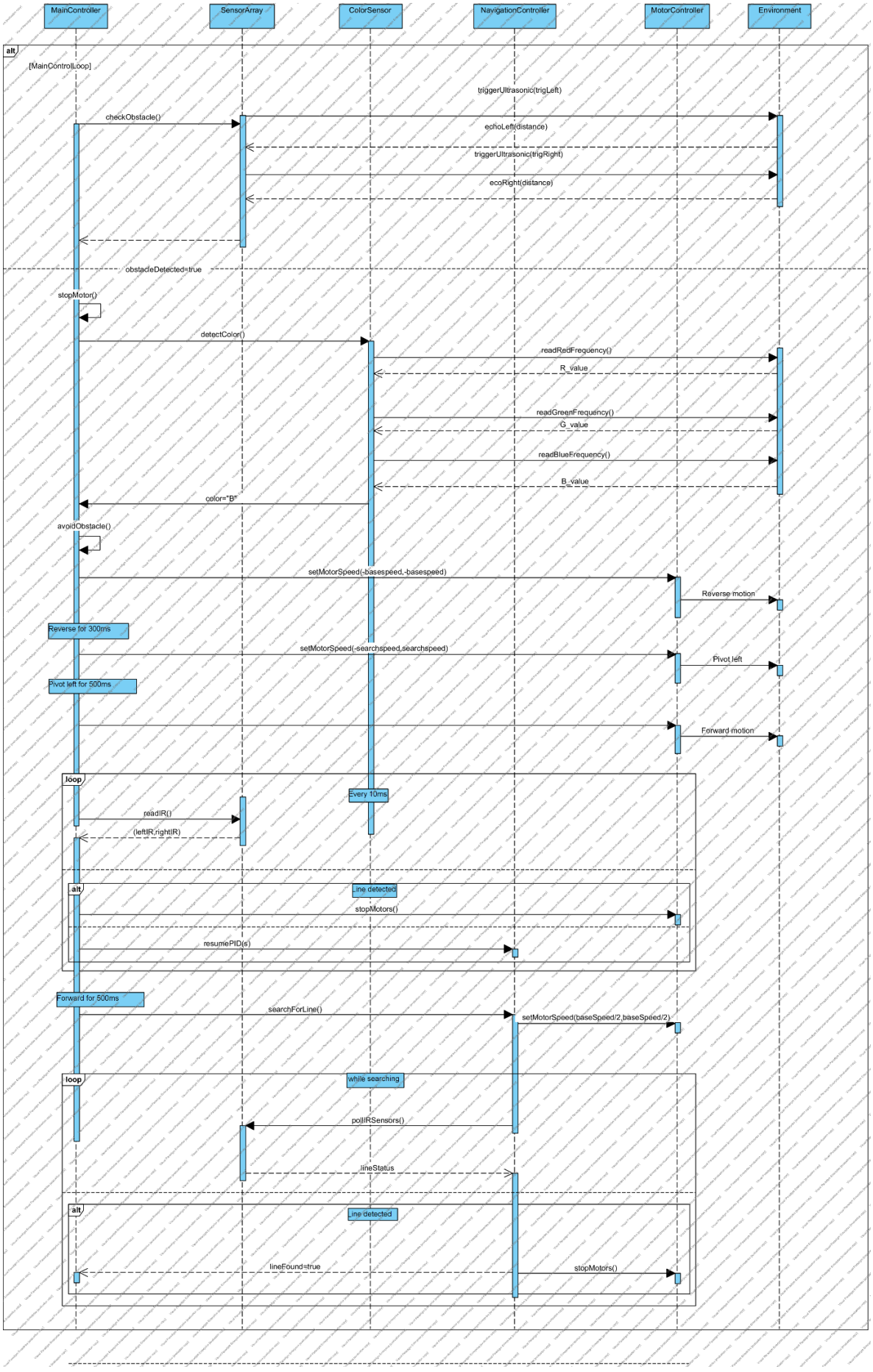
3.3.2 Parametric_Diagram:



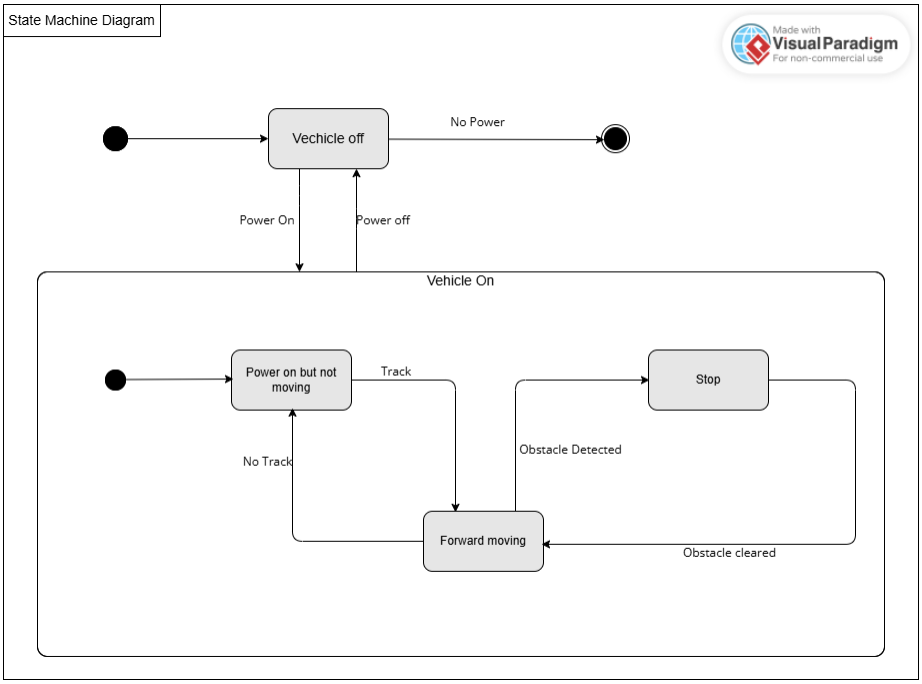
3.3.3 Requirements Diagram



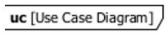
3.3.4 Sequence Diagram:

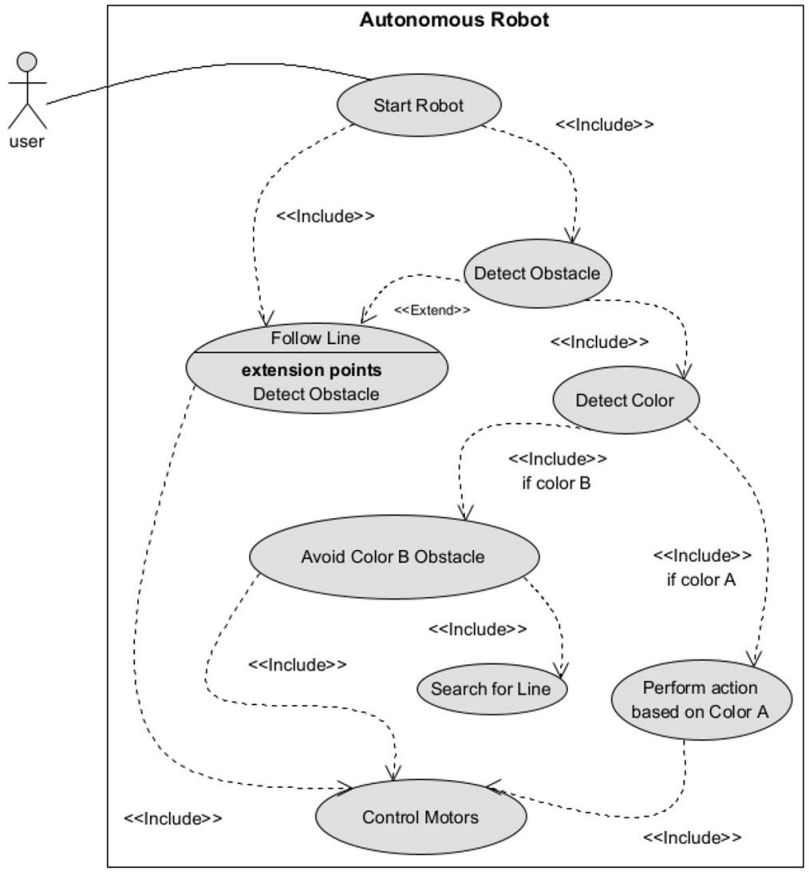


3.3.5 State Machine Diagram:

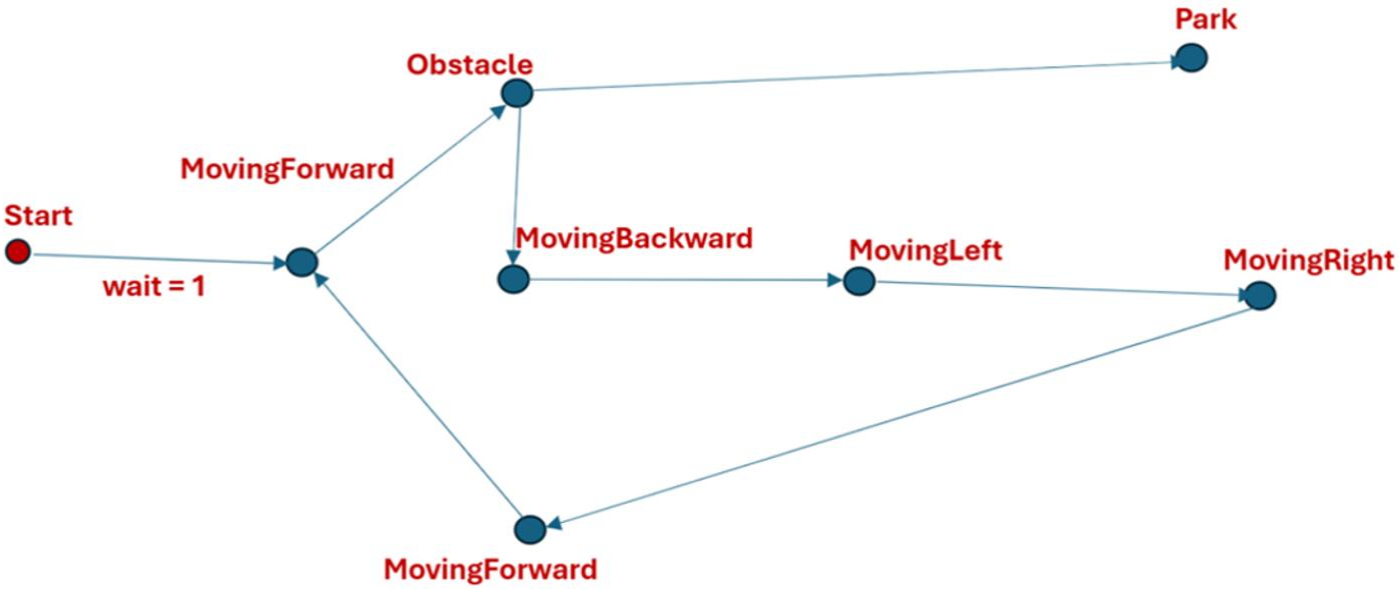


3.3.6 Use Case Diagram





3.3.7 UPAAL Model Diagram



4.0 PROGRESS SUMMARY

The project successfully reached completion after progressing through several structured development stages:

a) **Mechanical Design:**
The project began with systems engineering training and receipt of the project requirement from the professor. Thereafter, the mechanical parts were received in the lab. Custom holders for sensors and the battery were designed by us using SolidWorks and fabricated using a 3D printer. These additional parts provided a stable and modular foundation for effective assembly and efficient performance of our vehicle [10].

b) **Component Integration:**
Once the frame and holders were ready, each electronic component was tested individually. The IR sensors, ultrasonic sensors, TCS3200 colour sensor, motor driver (SBC Motor Driver 2), and 12V LiPo battery were all verified for functionality before being integrated into the autonomous vehicle system[2],[4].

c) **Core Functionality Development:**
With hardware in place, development moved to software. The robot was programmed to follow a predefined lane using IR sensors, followed by ultrasonic-based obstacle detection within a 25 cm range. The TCS3200 colour sensor was calibrated to distinguish between obstacle types (Colour A and Colour B), enabling the robot to take appropriate actions based on object colour [3].

d) **Advanced Behaviour Implementation:**
The robot was enhanced with advanced obstacle avoidance manoeuvres. It can now bypass obstacles autonomously and return to the original lane without external assistance. PID control logic was used to ensure smooth lane following and stable turning behaviour.

e) **Testing, Challenges, and Finalization:**
The system was tested extensively under different lighting and surface conditions. The main challenges included colour sensor calibration in varied lighting and PID tuning for smooth motion. These were overcome through iterative testing and adjustments. All planned objectives were achieved [3].

f) **Final Status:**
The robotic vehicle demonstrates all core capabilities lane following, real-time obstacle avoidance, colour-based decision making, and autonomous path correction. The final system is stable, efficient, and ready for deployment.

5.0 FINDINGS AND ANALYSIS

Findings and Analysis:

The following section presents the key findings obtained during the project, along with a detailed analysis. These results are based on the data collected, observations made, and methods applied throughout the project.

5.1 Sensor Behaviour and Accuracy:

a) **Ultrasonic Sensors**: Successfully detected obstacles within a range of 18cm with an average error margin of ±1cm[2].
b) **Infrared Sensors:** Accurately detected line paths, but performance was affected by lighting and surface texture.
c) **Colour Sensor:** Could reliably distinguish between Red and Green surfaces under good lighting conditions.

5.2 PID Controller Tuning:

a) Initial PID values: Kp = 100, Ki = 100, Kd = 100
b) Observed that:
    i. High Kp caused the robot to oscillate.
    ii. Ki helped reduce long term drift but too much made it unstable.
    iii. Kd improved sharp turns and reduced overshoot.
c) Final tuned values: Kp = 10, Ki = 0, Kd = 10 provided smooth line following with minimal overshooting [8].

5.3 State Machine Performance:

a) The state transitions worked correctly between:
    i. Line following
    ii. Obstacle avoidance
    iii. Turning logic
b) Edge cases (obstacles, tight corners) were handled with a delay of 200ms on average.

5.4 System Limitations:

a) Sensor noise occasionally caused false obstacle detection.
b) Bright ambient light interfered with IR accuracy.
c) Tuning PID required trial and error and was environment dependent.

5.5 Performance Metrics:

a) Average lap time on test track: 29 seconds
b) Success rate for completing track without collision is 90%.

6.0 KEY LEARNINGS AND TAKEAWAYS

Key Learnings and Takeaways

Working on the autonomous vehicle gave us valuable insight into how complex it can be to design a functional system in real life. Some of the major takeaways we had include;

6.1 Understanding Autonomous Systems:

a) Gained practical insight into how autonomous vehicles perceive their environment using sensors like ultrasonic, Infrared and color sensors.
b) Learned how decision making is handled through state machines and sensor inputs.

6.2 Sensor Integration Challenges:

a) Realized that real world sensor data is noisy and can be affected by environmental conditions (for example: lighting and surface reflectivity).

6.3 PID Control in Real Life:

a) Learned how a PID controller is used to achieve smooth line following.
b) Understood the role of tuning kp, ki and kd for optimal movement balancing responsiveness and stability.

6.4 System Thinking:

a) Understood the importance of combining hardware, software and control logic into a cohesive system.
b) Experienced how small issues in one component (a sensor for instance) can impact overall system behavior.

6.5 Programming and Logic Development:

a) Improved skills in C/C++ programming using Arduino IDE [1].
b) Learned to design modular code with functions, state machines and helper routines.
c) Developed debugging skills by using serial output for troubleshooting sensor behavior.

6.6 Project Planning and Iteration:

a) Learned the value of testing in stages. Firstly, the motors were tested, then sensors and other components before the integration of the system.
b) Iteratively refined the system through testing, debugging and improvements.

7.0 SIGNIFICANT CHALLENGES AND RESOLUTIONS

Throughout the development and testing of the autonomous vehicle, several technical and practical challenges were encountered. These challenges provided valuable learning opportunities and required iterative problem-solving, adjustments to the hardware and software, and fine-tuning of system performance. Below is a summary of the key challenges faced during the project and the solutions implemented to overcome them.

7.1 Sensor Inaccuracy and Interference:

a) Challenge: The IR and ultrasonic sensors sometimes gave inconsistent or false readings, especially in bright lighting or when objects were at awkward angles [2].
b) Resolution: Implemented multiple readings and averaging (for color sensor), timing intervals for ultrasonic checks and added logic to ignore clearly invalid values.

7.2 Obstacle Avoidance Complexity:

a) **Challenge:** Designing an effective obstacle avoidance routine that didn't cause the robot to get lost or stuck after detouring.
b) **Resolution:** Created a step-by-step avoidance sequence including reverse, pivot, bypass, and line recovery with fail-safe search patterns -ensured the robot could return to the line reliably.

7.3 Line Loss and Path Recovery:

a) **Challenge:** When both IR sensors lost the line (e.g., at junctions or sharp curves), the robot didn't always know which way to turn.
b) **Resolution:** Tracked the last seen position of the line using lastSeenLine logic and implemented directional searches (pivot left/right) to regain the path.

7.4 Color Detection Reliability:

a) **Challenge:** Colours were sometimes misclassified due to ambient lighting or variations in surface tone.
b) **Resolution:** Collected calibrated colour ratios (RG and RB) and added a differential comparison strategy instead of relying on raw RGB values - improved classification accuracy between Colour A and B.

7.5 Tuning PID Controller:

a) **Challenge:** The robot either over-corrected or didn't respond well to line deviations during movement.
b) **Resolution:** Tuned the PID constants (Kp, Ki, Kd) through trial and error to achieve smoother turns and better line-following stability.

7.6 Code Integration and Timing Conflicts:

a) **Challenge:** Running multiple sensor routines (IR, ultrasonic, colour) caused timing conflicts and delayed responses.
b) **Resolution:** Used timing intervals as millis() instead of delay() for obstacle checks and optimized sensor polling frequency to balance responsiveness and stability.

8.0 BUDGET AND FINANCES

Budget and Finances:

While working on the project, we realized that extensive testing on the autonomous vehicle was necessary to achieve optimal results. To facilitate this, we purchased a few additional items to support our efforts. Below is a list of the items we acquired:

Cardboards and coloured tapes:

Due to limited lab time for testing, we bought cardboards and coloured tapes to create a temporary track at home. This allowed us to conduct more extensive testing of the autonomous vehicle.

9V Battery:

Since students were not allowed to take the 12V LiPo battery home due to the risk of explosion, we purchased a 9V battery to power the autonomous vehicle during home testing.

3*25mm Screws:

The mounts we designed for the DC motors required 3×25mm screws, which were not available in the lab. We therefore bought the screws specifically for this purpose

HSHL Group B Team 4 Expense Report (For Team Project Only)

General Information

Purpose: To invest more time in practicing at home before final competition

Date	Description	Quantity	Unit Price	Amount	Total
25/05/2025	**9V Battery**	**1**	**3€**	**3€**	**3€**
	White Cardboard	**2**	**1€**	**2€**	**2€**
	Coloured Tape Pack	**1**	**3,29€**	**3,29€**	**3,29€**
	3*25mm Screw Pack	**1**	**4,59€**	**4,59€**	**4,59€**
Total					**12,88€**
					12,88€

9.0 TIMELINE AND SCHEDULE

This is a very compact schedule stated below. The comprehensive detail is expressed better in the attached excel spreadsheet in GitHub.

Group B Team 4:
Project Planner for Prototyping and Systems Engineering.
Focus: Autonomous Vehicle Design and Testing

Project Contributors:
Chimezie, Daniel Chidi (Mat. Nr. 1230515) - Member / Coordinator
Cho, Bertrand Mungu (Mat. Nr. 2220052) - Member
Ghimire,Riwaj (Mat. Nr. 1232171) - Member

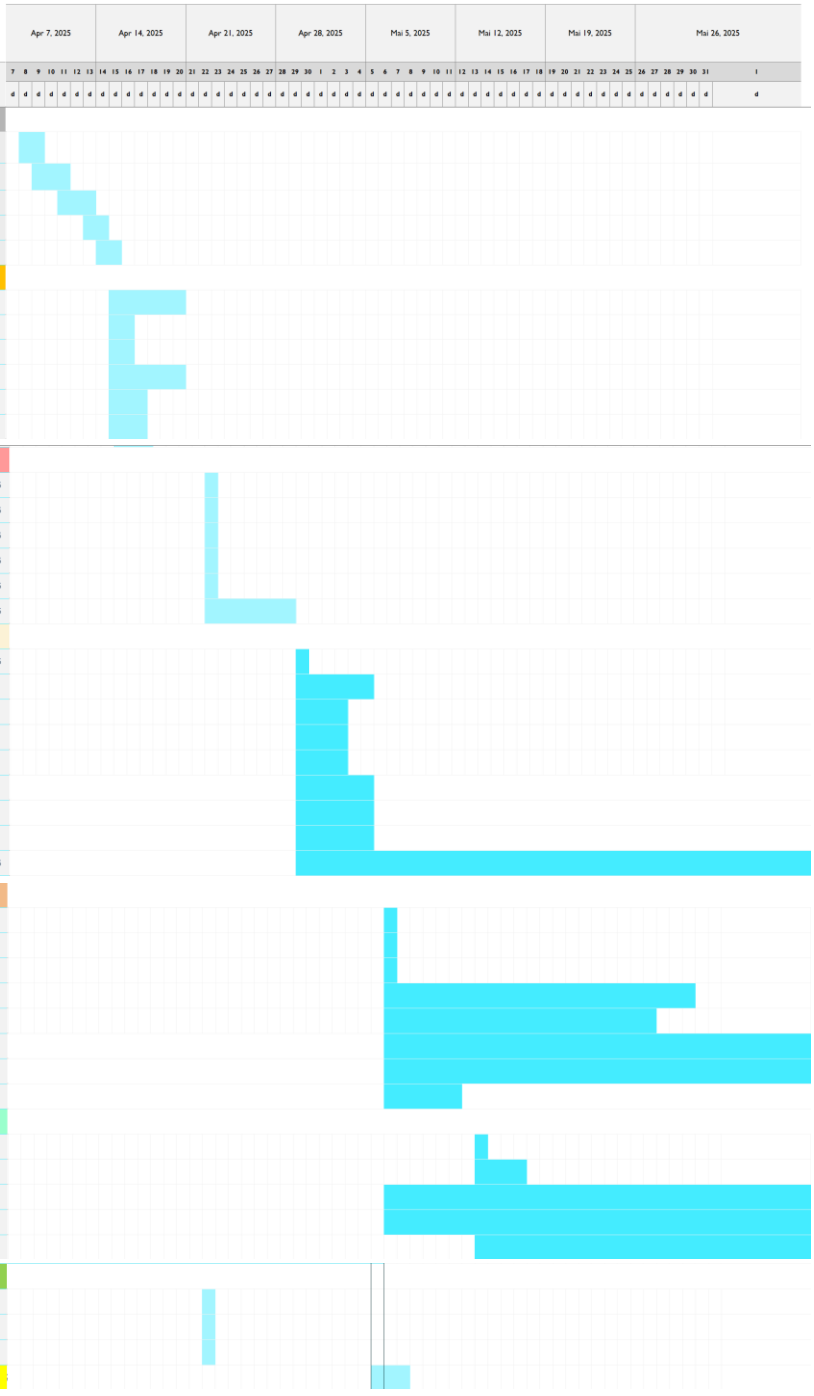
Project supervision and directions:
Professor Stefan Henkler / Gido Wahrmann / Faezeh Pasandideh

Project start: Di, 4.8.2025
Display week: 1,00

Project report submission timeline: Tue,07.08.2025 [MM/DD/YYYY]

HOCHSCHULE HAMM-LIPPSTADT

TASK	ASSIGNED TO	SUPPORTED BY	PROGRESS	START	END
Project Initialization and Planning					
Project goals and scope definition	Chimezie, Daniel C.	Cho, Bertrand M. / Ghimire Riwaj	100%	4.8.25	4.9.25
Research similar systems	Cho, Bertrand M.	Chimezie, Daniel C./Ghimire Riwaj	100%	4.9.25	4.11.25
Define system requirements (hardware/software)	Chimezie, Daniel C.	Cho, Bertrand M. / Ghimire Riwaj	100%	4.11.25	4.13.25
Budget and resource planning	Cho, Bertrand M.	Ghimire Riwaj / Chimezie Daniel C.	100%	4.13.25	4.14.25
Risk assessment	Ghimire Riwaj	Cho, Bertrand M. / Chimezie Daniel C.	100%	4.14.25	4.15.25
System Design					
Block diagram/system architecture	Cho, Bertrand M.	Chimezie, Daniel C./Ghimire Riwaj	100%	4.15.25	4.20.25
Sensor and actuator selection	Ghimire Riwaj	Cho, Bertrand M. / Chimezie Daniel C.	100%	4.15.25	4.16.25
Circuit schematic design	Chimezie, Daniel C.	Cho, Bertrand M. / Ghimire Riwaj	100%	4.15.25	4.17.25
Mechanical chassis design	Chimezie, Daniel C.	Cho, Bertrand M. / Ghimire Riwaj	100%	4.15.25	4.20.25
Power supply holder design	Cho, Bertrand M.	Chimezie, Daniel C./Ghimire Riwaj	100%	4.15.25	4.17.25
Pin mapping for Arduino Uno R4	Ghimire Riwaj	Cho, Bertrand M. / Chimezie Daniel C.	100%	4.15.25	4.17.25
Hardware Procurement and Assembly					
Order and replacement for components (Arduino Uno R4, sensors, motors)	Ghimire Riwaj	Cho, Bertrand M. / Chimezie Daniel C.	100%	4.22.25	4.22.25
Assemble the chassis	Cho, Bertrand M.	Chimezie, Daniel C./Ghimire Riwaj	100%	4.22.25	4.22.25
Mount motors and wheels	Chimezie, Daniel C.	Cho, Bertrand M. / Ghimire Riwaj	100%	4.22.25	4.22.25
Connect sensors (IR, Ultrasonic, Colour.)	Ghimire Riwaj	Cho, Bertrand M. / Chimezie Daniel C.	100%	4.22.25	4.22.25
Connect motor drivers and power modules	Chimezie, Daniel C.	Cho, Bertrand M. / Ghimire Riwaj	100%	4.22.25	4.22.25
Wiring and cable management	Cho, Bertrand M.	Chimezie, Daniel C./Ghimire Riwaj	100%	4.22.25	4.28.25
Software Development					
Arduino IDE setup	Ghimire Riwaj	Cho, Bertrand M. / Chimezie Daniel C.	100%	4.29.25	4.29.25
Write/test modules:	Ghimire Riwaj	Cho, Bertrand M. / Chimezie Daniel C.	100%	4.29.25	5.4.25
>> Line-following using IR sensors	Chimezie, Daniel C.	Cho, Bertrand M. / Ghimire Riwaj	100%	4.29.25	5.2.25
>> Obstacle avoidance using ultrasonic sensor	Chimezie, Daniel C.	Cho, Bertrand M. / Ghimire Riwaj	100%	4.29.25	5.2.25
>> Motor control (PWM, direction)	Cho, Bertrand M.	Chimezie, Daniel C./Ghimire Riwaj	100%	4.29.25	5.2.25
>> Colour detection	Cho, Bertrand M.	Chimezie, Daniel C./Ghimire Riwaj	100%	4.29.25	5.4.25
Integrate modules into main control loop	Chimezie, Daniel C.	Cho, Bertrand M. / Ghimire Riwaj	100%	4.29.25	5.4.25
PID and Logic tuning for line-following	Cho, Bertrand M.	Chimezie, Daniel C./Ghimire Riwaj	100%	4.29.25	5.4.25
Fail-safe and debugging (routine process)	Ghimire Riwaj	Cho, Bertrand M. / Chimezie Daniel C.	100%	4.29.25	6.24.25
Integration and Testing					
Combine hardware and software	Cho, Bertrand M.	Chimezie, Daniel C./Ghimire Riwaj	100%	5.6.25	5.6.25
Functional testing of all subsystems	Chimezie, Daniel C.	Cho, Bertrand M. / Ghimire Riwaj	100%	5.6.25	5.6.25
Field tests:	Chimezie, Daniel C.	Cho, Bertrand M. / Ghimire Riwaj	100%	5.6.25	5.6.25
>> Line-following accuracy	Ghimire Riwaj	Cho, Bertrand M. / Chimezie Daniel C.	100%	5.6.25	5.29.25
>>Obstacle detection responsiveness	Chimezie, Daniel C.	Cho, Bertrand M. / Ghimire Riwaj	100%	5.6.25	5.26.25
>>Route completion success rate	Cho, Bertrand M.	Chimezie, Daniel C./Ghimire Riwaj	100%	5.6.25	6.15.25
Debugging and iteration (routine process)	Ghimire Riwaj	Cho, Bertrand M. / Chimezie Daniel C.	100%	5.6.25	6.23.25
Power efficiency testing	Ghimire Riwaj	Cho, Bertrand M. / Chimezie Daniel C.	100%	5.6.25	5.11.25
Documentation					
Wiring diagrams and circuit explanations	Chimezie, Daniel C.	Cho, Bertrand M. / Ghimire Riwaj	100%	5.13.25	5.13.25
Code documentation (comments, flowcharts)	Ghimire Riwaj	Cho, Bertrand M. / Chimezie Daniel C.	100%	5.13.25	5.16.25
Test results and data logs	Cho, Bertrand M.	Chimezie, Daniel C./Ghimire Riwaj	100%	5.6.25	6.24.25
User manual or usage instructions	Chimezie, Daniel C.	Cho, Bertrand M. / Ghimire Riwaj	100%	5.6.25	6.5.25
Final report	Chimezie, Daniel C.	Cho, Bertrand M. / Ghimire Riwaj	100%	5.13.25	7.7.25
Presentation and Delivery					
Prepare demo video or live demo (Competition Day) * 1st position for Group A and B Contest.***	Chimezie, Daniel C.	Cho, Bertrand M. / Ghimire Riwaj	100%	6.24.25	6.24.25
Slides and presentation materials	Chimezie, Daniel C.	Cho, Bertrand M. / Ghimire Riwaj	100%	6.24.25	6.24.25
Q&A preparation for inputs and future advances	Cho, Bertrand M.	Chimezie, Daniel C./Ghimire Riwaj	100%	6.24.25	6.24.25
Final handover of project	Chimezie, Daniel C.	Cho, Bertrand M. / Ghimire Riwaj	100%	7.7.25	7.9.25



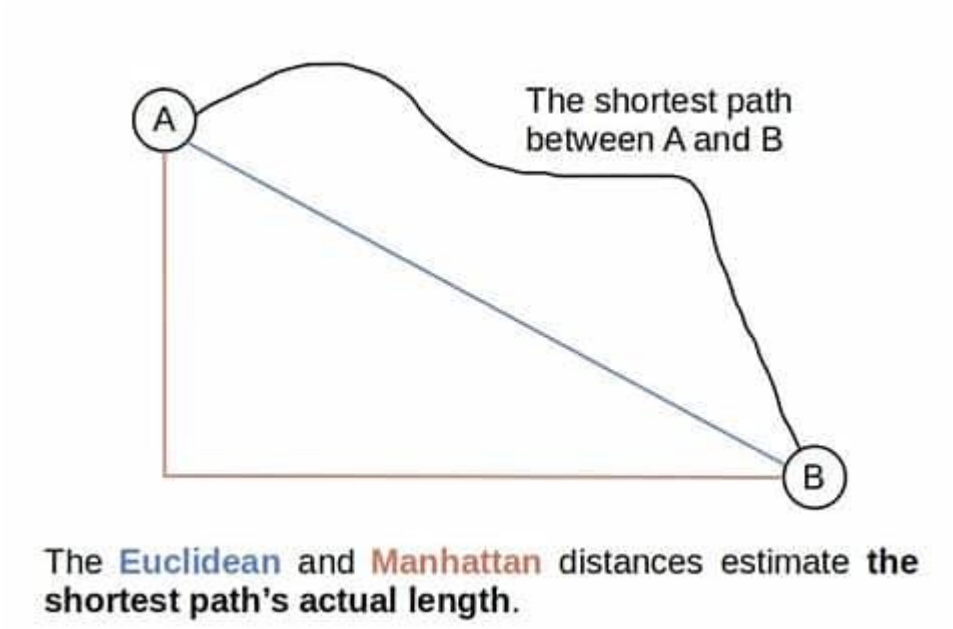
10. RECOMMENDATIONS AND NEXT STEPS

The autonomous vehicle project has successfully met its core objectives, including reliable lane following, obstacle avoidance, and colour-based decision making. To further enhance the system's robustness, scalability, and practical applicability, the following recommendations and next steps are proposed:

a) **Improve Sensor Fusion:**
Integrate data from multiple sensors (IR, ultrasonic, and camera modules) using sensor fusion techniques to enhance accuracy in complex environments.

b) **Add Camera-Based Vision:**
Introduce a camera module with basic computer vision algorithms (OpenCV) to recognize traffic signs, lane markings, and dynamic obstacles beyond the capability of IR or colour sensors [9].

c) **Enhance Navigation Algorithms:**
Implement advanced algorithms such as A* or Dijkstra for path planning, and Kalman Filter for better localization and control under uncertain conditions. [6]



d) **Develop a Scalable Software Architecture:**
Refactor the current code into modular blocks to support future upgrades, debugging, and potential collaboration with other developers [5], [6], [7].

e) **Test in Real-World Scenarios:**
Move beyond lab conditions and test the vehicle on larger, more dynamic tracks with real-world variations in lighting, surfaces, and obstacle types. [5], [6], [7].

f) **Add Remote Monitoring:**
Incorporate wireless communication (e.g., Bluetooth or Wi-Fi) for remote monitoring, telemetry, and control, allowing real-time feedback and data collection during operation.

g) **Energy Optimization:**
Analyse battery usage and optimize motor control and sensor usage to extend operating time and improve energy efficiency [5], [6], [7].

11. TEAM PERFORMANCE

We worked extremely very hard and also, invested much time in testing and re-testing to ensure that our modest design meets the core aim of the project which is:

The ability of our designed robot to do the following with success:

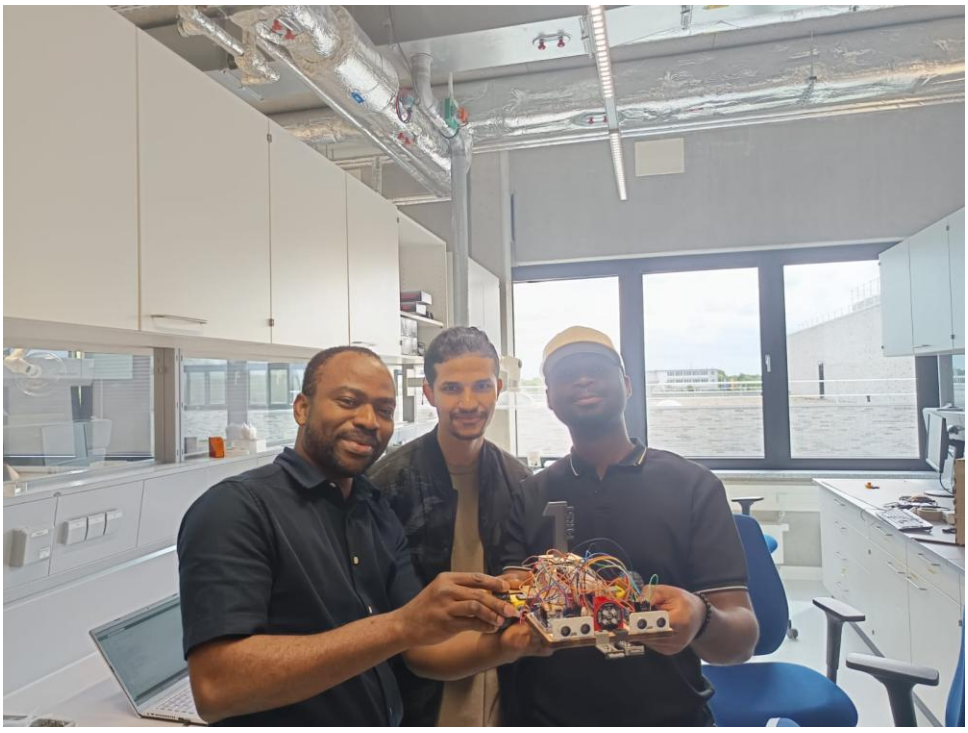
a) To follow a dark lane and travel through an oval part.
b) To sense and detect an obstacle and follow our desired instructions in Arduino IDE coding [1].
c) To avoid obstacles it sensed and return back to the lane and continue its travel via the assigned tracks.

For Group A and Group B Teams racing and obstacle avoidance contest, we won the contest with records of **less than 29 seconds** travel time across the oval.

Future trajectory:

We aim to investigate further into robotics by applying the skills acquired in this course and project in designing a car with similar capabilities which would integrate other capabilities such as artificial intelligence and real life camera sensing devices.

We recommend this course for any Electrical and Electronic Engineering student or any human who is innovative minded. There should be no single limits if we push very hard with unwavering efforts.



12. RISKS AND MITIGATION

Risk	Description	Mitigation Strategy
Sensor Malfunction or Inaccuracy	Inconsistent readings due to lighting, surface reflectivity, or hardware issues	a) Regular calibration under varying conditions. b) Use redundant sensors or sensor fusion. c) Implement error-handling in code.
Power Supply Issues	Battery drain or voltage drops causing system failure	a) Monitor battery voltage in real time. b) Optimize power usage in code - Keep spare batteries on hand.
Software Bugs and Instability	Unexpected behaviour due to bugs or incomplete testing	a) Modular programming with good documentation. b) Conduct unit and integration tests. c) Use version control.
Hardware Damage	Mechanical failure or wear and tear from testing	a) Use durable materials and proper mounting. b) Regular physical inspection. c) Keep spare parts available.
Environmental Variability	Performance drops in outdoor or non-lab settings	a) Test in diverse environments early. b) Use adaptive thresholds and calibration techniques.
Team Coordination and Time Constraints	Delays from poor communication or uneven workload	a) Hold regular team meetings. b) Use project management tools (e.g., Trello, GitHub, MS-PM). c) Allocate buffer time for issues.

13. APPENDICES

All supporting documents including technical specifications, tables, charts, and detailed data

Appendix A: Technical Specifications

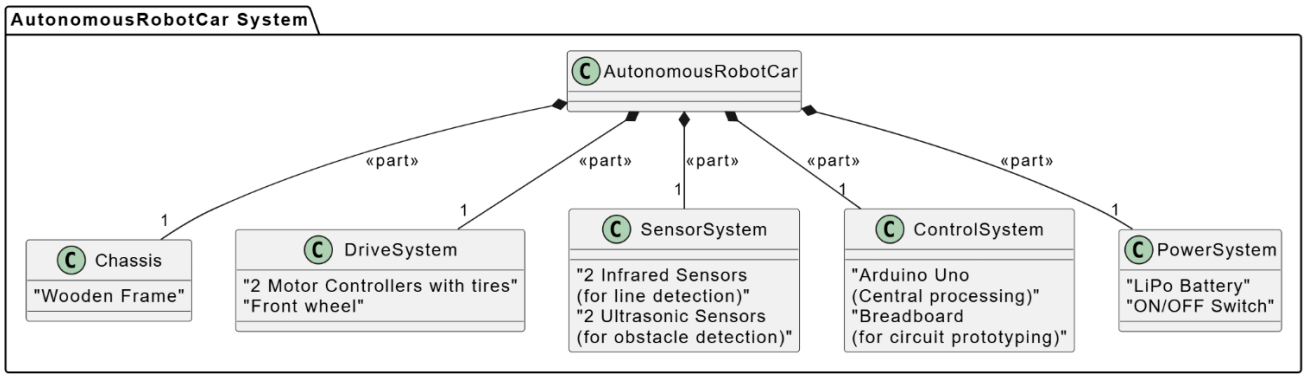
Component	Specification
IR Sensors	Digital reflectance sensors, 3–5V, 20–30 cm range
Ultrasonic Sensor	HC-SR04, 2–400 cm range, 5V operating voltage
Color Sensor	TCS3200, RGB color detection, 3–5V input
Motor Driver	SBC Motor Driver 2, dual channel, 5–30V DC
Power Source	12V LiPo Battery, 2200 mAh
Microcontroller	Arduino Uno R4 WiFI (ATmega328P)
Chassis Material	Acrylic frame with 3D-printed sensor holders
Wheels and Motors	DC geared motors, 100 RPM, plastic wheels

[2], [4].

Appendix B: PID Control Parameters

Parameter	Value
P (Proportional)	10
I (Integral)	0
D (Derivative)	10

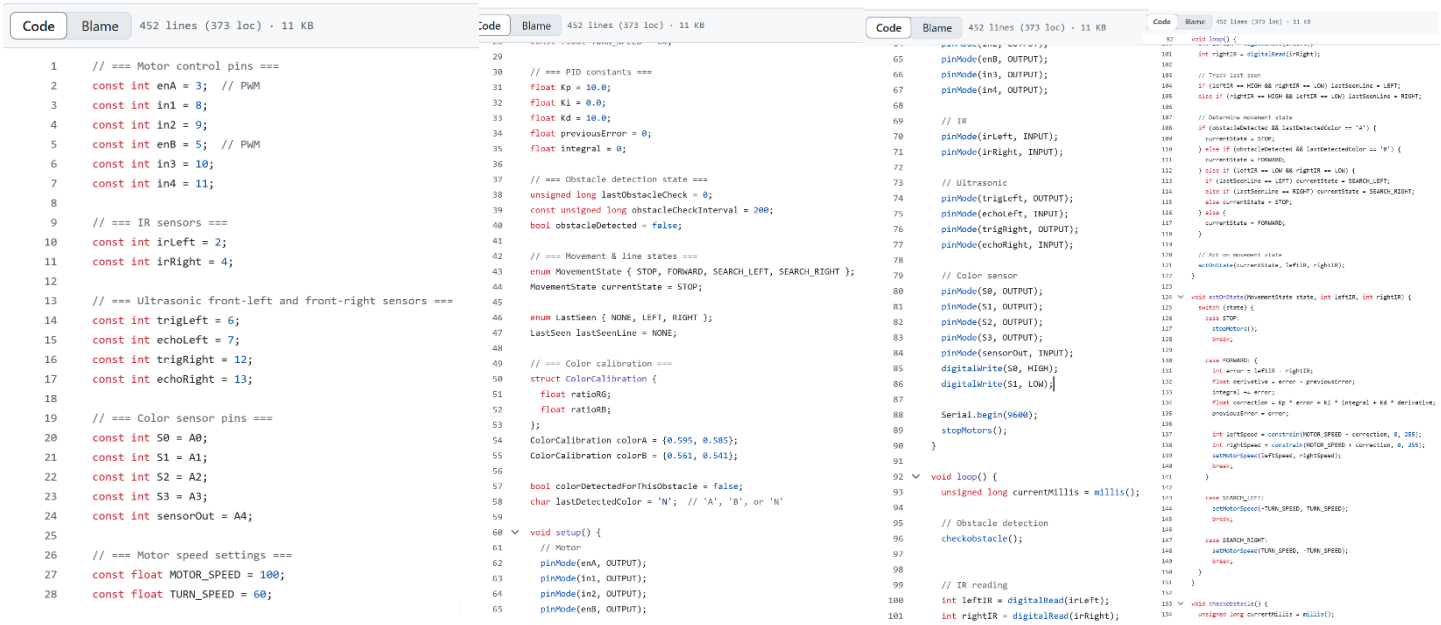
Appendix C: System Block Diagram



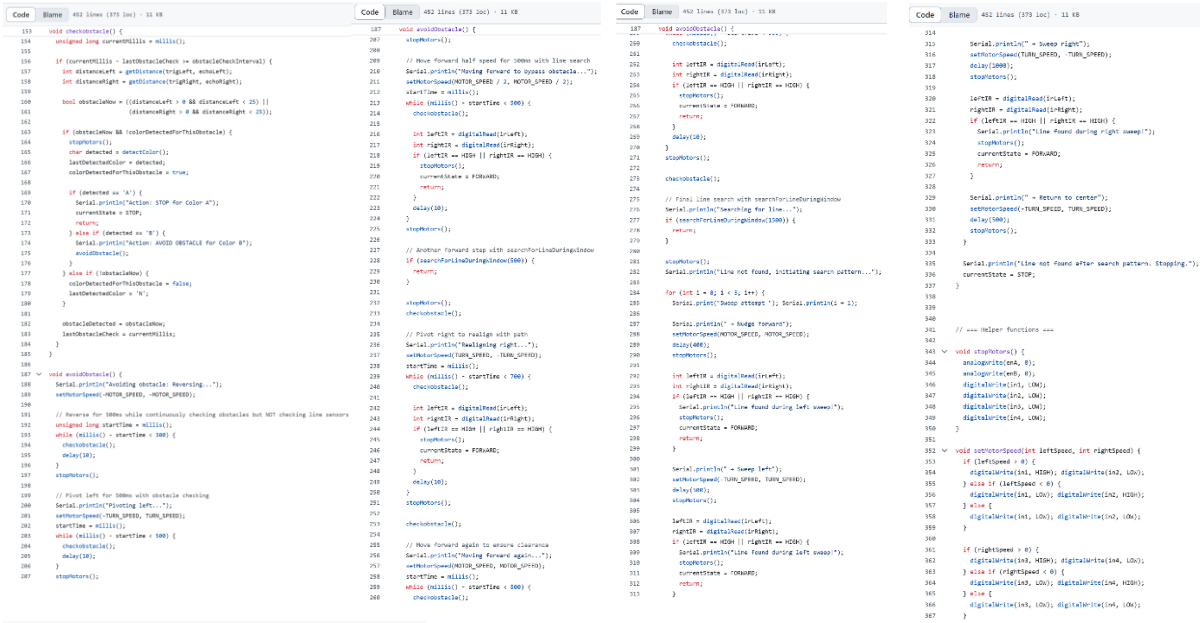
Appendix D: Test Results Summary

Test Scenario	Outcome	Notes
Lane following on white background	Successful	Minor tuning required for sharp turns
Obstacle avoidance at 25 cm	Successful	Smooth detour and return to lane
Colour detection under indoor light	Reliable (Colour A & B detected)	Requires recalibration under sunlight
Battery endurance test	2.5 hours of operation	Under typical use with full charge

Appendix E: Source Code Snapshots



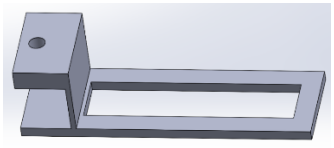
```
// === Motor control pins ===
const int enA = 3;  // PWM
const int in1 = 8;
const int in2 = 9;
const int enB = 5;  // PWM
const int in3 = 10;
const int in4 = 11;

// === IR sensors ===
const int irLeft = 2;
const int irRight = 4;

// === Ultrasonic front-left and front-right sensors ===
const int trigLeft = 6;
const int echoLeft = 7;
const int trigRight = 12;
const int echoRight = 13;

// === Color sensor pins ===
const int S0 = A0;
const int S1 = A1;
const int S2 = A2;
const int S3 = A3;
const int sensorOut = A4;

// === Motor speed settings ===
const float MOTOR_SPEED = 100;
const float TURN_SPEED = 60;

// === PID constants ===
float Kp = 10.0;
float Ki = 0.0;
float Kd = 10.0;
float previousError = 0;
float integral = 0;

// === Obstacle detection state ===
unsigned long lastObstacleCheck = 0;
const unsigned long obstacleCheckInterval = 200;
bool obstacleDetected = false;

// === Movement & line states ===
enum MovementState { STOP, FORWARD, SEARCH_LEFT, SEARCH_RIGHT };
MovementState currentState = STOP;

enum LastSeen { NONE, LEFT, RIGHT };
LastSeen lastSeenLine = NONE;

// === Color calibration ===
struct ColorCalibration {
  float ratioRG;
  float ratioRB;
};
ColorCalibration colorA = {0.595, 0.585};
ColorCalibration colorB = {0.561, 0.541};

bool colorDetectedForThisObstacle = false;
char lastDetectedColor = 'N';  // 'A', 'B', or 'N'

void setup() {
  // Motor
  pinMode(enA, OUTPUT);
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
  pinMode(enB, OUTPUT);
  pinMode(in3, OUTPUT);
  pinMode(in4, OUTPUT);

  // IR
  pinMode(irLeft, INPUT);
  pinMode(irRight, INPUT);

  // Ultrasonic
  pinMode(trigLeft, OUTPUT);
  pinMode(echoLeft, INPUT);
  pinMode(trigRight, OUTPUT);
  pinMode(echoRight, INPUT);

  // Color sensor
  pinMode(S0, OUTPUT);
  pinMode(S1, OUTPUT);
  pinMode(S2, OUTPUT);
  pinMode(S3, OUTPUT);
  pinMode(sensorOut, INPUT);
  digitalWrite(S0, HIGH);
  digitalWrite(S1, LOW);

  Serial.begin(9600);
  stopMotors();
}

void loop() {
  unsigned long currentMillis = millis();

  // Obstacle detection
  checkobstacle();


  // IR reading
  int leftIR = digitalRead(irLeft);
  int rightIR = digitalRead(irRight);

  // Track last seen
  if (leftIR == HIGH && rightIR == LOW) lastSeenLine = LEFT;
  else if (rightIR == HIGH && leftIR == LOW) lastSeenLine = RIGHT;

  // Determine movement state
  if (obstacleDetected && lastDetectedColor == 'A') {
    currentState = STOP;
  } else if (obstacleDetected && lastDetectedColor == 'B') {
    currentState = FORWARD;
  } else if (leftIR == LOW && rightIR == LOW) {
    if (lastSeenLine == LEFT) currentState = SEARCH_LEFT;
    else if (lastSeenLine == RIGHT) currentState = SEARCH_RIGHT;
    else currentState = STOP;
  } else {
    currentState = FORWARD;
  }

  // Act on movement state
  actOnState(currentState, leftIR, rightIR);
}

void actOnState(MovementState state, int leftIR, int rightIR) {
  switch (state) {
    case STOP:
      stopMotors();
      break;

    case FORWARD: {
      int error = leftIR - rightIR;
      float derivative = error - previousError;
      integral += error;
      float correction = Kp * error + Ki * integral + Kd * derivative;
      previousError = error;

      int leftSpeed = constrain(MOTOR_SPEED - correction, 0, 255);
      int rightSpeed = constrain(MOTOR_SPEED + correction, 0, 255);
      setMotorSpeed(leftSpeed, rightSpeed);
      break;
    }

    case SEARCH_LEFT:
      setMotorSpeed(-TURN_SPEED, TURN_SPEED);
      break;

    case SEARCH_RIGHT:
      setMotorSpeed(TURN_SPEED, -TURN_SPEED);
      break;
  }
}

void checkobstacle() {
  unsigned long currentMillis = millis();
```



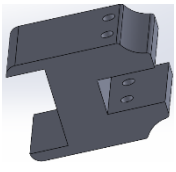
```
422     }
423
424   int readFrequency(bool s2, bool s3) {
425     digitalWrite(S2, s2);
426     digitalWrite(S3, s3);
427     delay(50);
428     return pulseIn(sensorOut, LOW);
429   }
430
431   bool searchForLineDuringWindow(unsigned long maxDuration) {
432     unsigned long startTime = millis();
433     setMotorSpeed(MOTOR_SPEED / 1.8, MOTOR_SPEED / 1.8);
434
435     while (millis() - startTime < maxDuration) {
436       int leftIR = digitalRead(irLeft);
437       int rightIR = digitalRead(irRight);
438
439       if (leftIR == HIGH || rightIR == HIGH) {
440         stopMotors();
441
442         if (leftIR == HIGH && rightIR == LOW) lastSeenLine = LEFT;
443         else if (rightIR == HIGH && leftIR == LOW) lastSeenLine = RIGHT;
444
445         currentState = FORWARD;
446         return true;
447       }
448     }
449
450     stopMotors();
451     return false;
452   }
```

Appendix F: 3D-Printed Part Designs

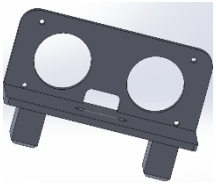
(a) IR Sensor holder



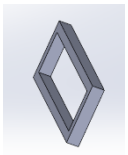
(b) Motor Holder



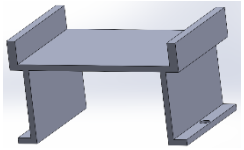
(c) Ultrasonic Sensor Holder [2].



(d) IR Sensor Clip



(d) Battery and Breadboard holder



14.0 REFERENCES (IEEE Standard)

[1] Arduino, “Arduino UNO R4 WiFi,” [Online]. Available: https://www.arduino.cc/reference/en/boards/uno-r4-wifi/. [Accessed: 07-Jul-2025].

[2] HC-SR04 Datasheet, “Ultrasonic Distance Sensor,” [Online]. Available: https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf. [Accessed: 07-Jul-2025].

[3] Adafruit, “TCS3200 Colour Sensor,” [Online]. Available: https://learn.adafruit.com/tcs34725-color-sensor. [Accessed: 07-Jul-2025].

[4] Texas Instruments, “L298N Dual H-Bridge Motor Driver Datasheet,” [Online]. Available: https://www.ti.com/lit/ds/symlink/l298.pdf. [Accessed: 07-Jul-2025].

[5] S. Skogestad and I. Postlethwaite, *Multivariable Feedback Control: Analysis and Design*, 2nd ed. Chichester, U.K.: Wiley, 2005.

[6] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, *Introduction to Autonomous Mobile Robots*, 2nd ed. Cambridge, MA: MIT Press, 2011.

[7] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Reading, MA: Addison-Wesley, 1989.

[8] MathWorks, “PID Controller Tuning,” [Online]. Available: https://www.mathworks.com/help/control/ug/pid-controller.html. [Accessed: 07-Jul-2025].

[9] OpenCV, “Open Source Computer Vision Library,” [Online]. Available: https://opencv.org/. [Accessed: 07-Jul-2025].

[10] SolidWorks, “SOLIDWORKS 3D CAD Software,” Dassault Systèmes, [Online]. Available: https://www.solidworks.com/. [Accessed: 07-Jul-2025].

15.0 DECLARATION OF ORIGINALITY

We hereby declare that this report is our own work and that

we have acknowledged all sources used.

Bremen, July 7, 2025, CHIMEZIE DANIEL CHIDI

Hamm, July 7, 2025, BERTRAND MUNGU CHO

Soest, July 7, 2025, GHIMIRE RIWAJ