AUTONOMOUS LINE-FOLLOWING CAR

MIR SHARJIL HASAN
Electronic Engineering
Hamm-Lippstadt
Lippstadt, Germany
mir-sharjil.hasan@stud.hshl.de

TALHA KHAN

Electronic Engineering
Hochschule Hamm-Lippstadt
Lippstadt, Germany

MUHAMMAD HAMAS
Electronic Engineering
Hochschule Hamm-Lippstadt
Lippstadt, Germany

Abstract

This project undertakes the design and development of an intelligent line follower vehicle that can detect obstacles and respond to them in an intelligent manner. The chassis of the vehicle was designed using SolidWorks and Fusion 360, modelled with laser cut parts and 3D printed components. Line sensors are used for the predefined path, while ultrasonic sensors detect obstacles around the vehicle for dynamic avoidance. Colour sensors enable the detection of the colours of obstacles which then trigger specific behaviours by the vehicle. These different types of sensing give the vehicle smartness in decision-making and adaptability to its environment. Therefore, a robust, reliable, autonomous navigation system able to work in dynamic environments is achieved.

Key Topics: SysML, Line-Following Robot, PLA, Detecting Obstacles, Merging Sensor Data

Index Terms—SysML, Line-Following Robot, PLA, Obstacle Detection, Sensor Fusion

I. Introduction

The goal of this project was to design and build a line-following bot that can not just follow a marked path but also see obstacles and act on their colour. This joins traditional way-following methods with real-time object spotting and smart choice-making. The work shown here uses ideas from embedded systems, sensor joining, and mechanical design to make a real self-driving car.

The report is organised as follows:

- **Section II** discusses how SysML was used to model and analyze the system.
- **Section III** walks through the design and development process.
- **Section IV** outlines the electronic components and explains their roles.

II. System Analysis with SysML

In order to properly scope and manage the project, we adopted SysML (Systems Modelling Language). This allowed us to maintain support throughout the entire development process— from system requirements definition all the way through design and testing. We captured desired functions of the robot using requirement diagrams, and how interactions between different components happen over time with sequence diagrams. These two artefacts have been critical in making sure all subsystems are both aligned and working cohesively.

Requirements

To guide the design and development of our autonomous vehicle prototype, we used a step-by-step requirement diagram, as shown in Figure 1. The diagram outlines all the significant functional requirements the system must fulfil, which serves to offer clear and consistent division of project objectives.

At the top level, the general goal is to design an autonomous vehicle (ID=001) to independently drive a track without any human interaction. This is the overall requirement broken down into three primary functional areas:

1.Track Management (ID=002)
This requirement blames the vehicle for keeping up with the track and reacting to change, such as curves or sharp turns. It was complemented by a derived requirement, titled Speed Optimisation (ID=006). It asks the vehicle to be able to accelerate or decelerate based on track complexity to make stable and optimal motion.

2.Obstacles Management (ID=003)
The vehicle should be able to perceive the obstacle ahead and make decisions about it. There are two additional layers in this section:

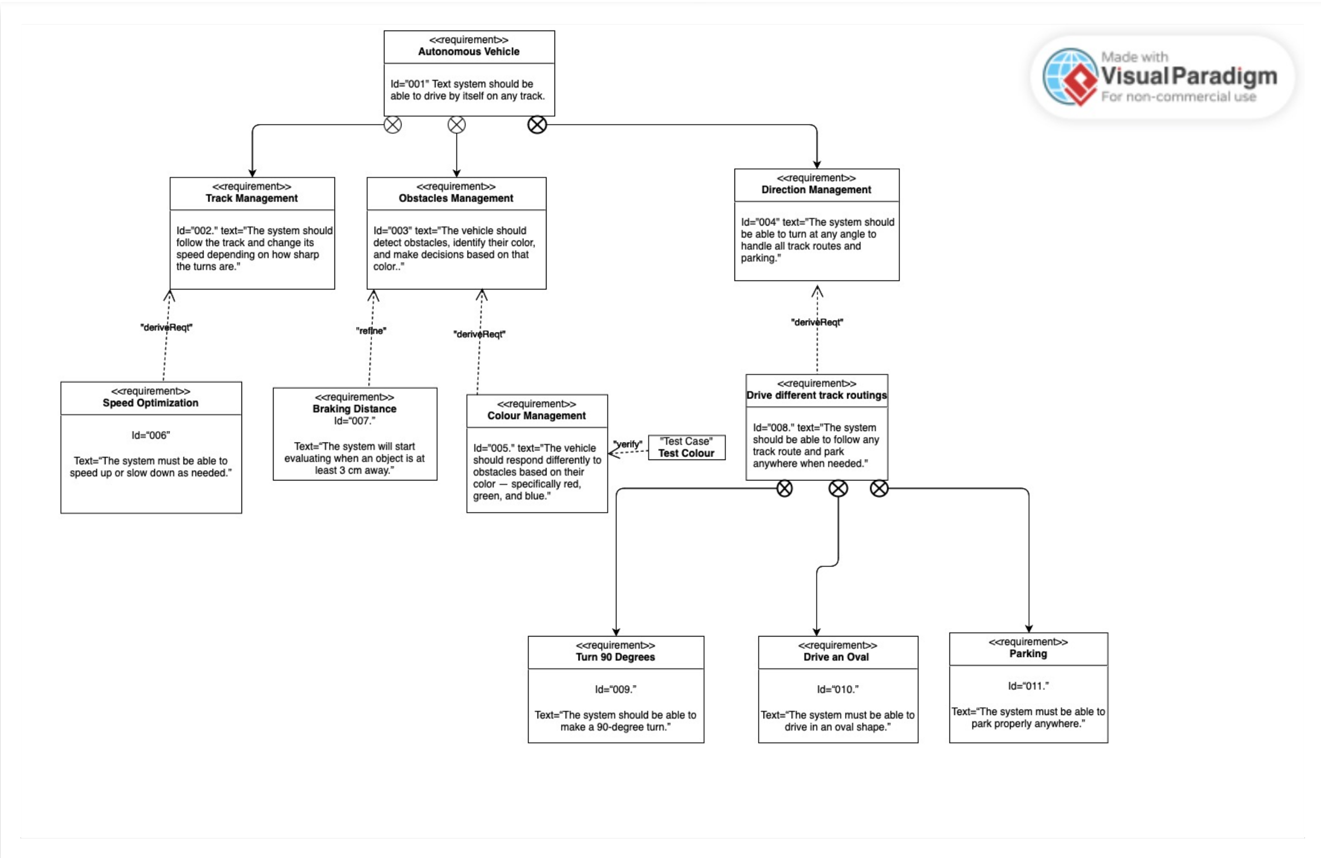
Braking Distance (ID=007): The system must begin responding when it perceives an object at least 3 cm in front.
Colour Management (ID=005): The system must perceive the color of the obstacle—red, green, or blue—and respond differently in turn. Test case ("Test Colour") was used to test this functionality while developing the application.

3.Direction Management (ID=004)
This is the turn and manoeuvring criterion while driving on various paths, including parking. Out of this, we drew out a more specific goal:
Drive Different Track Routings (ID=008): This means that the car must be capable enough to drive along complex paths and park when required.

It is also divided into three specific routing capabilities:

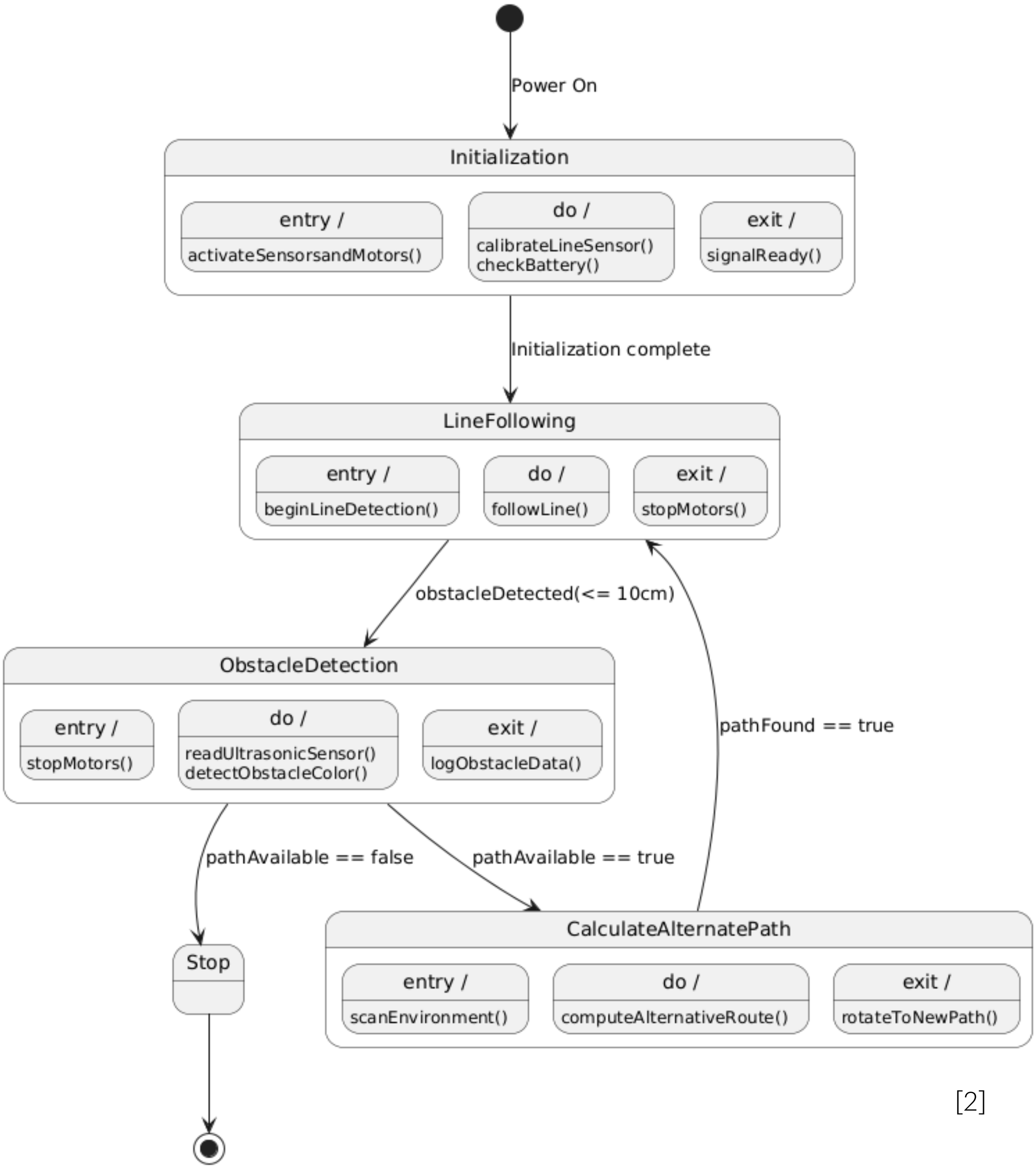
Turn 90 Degrees (ID=009)
Drive in an Oval (ID=010)
Parking (ID=011)

Collectively, these specifications guarantee the system is able to navigate lines accurately, deal with obstacles smartly, and accomplish various navigation tasks. The diagram was used as a basis for both design and testing, making it possible to construct the prototype incrementally while verifying all essential behaviours were included.



[1]

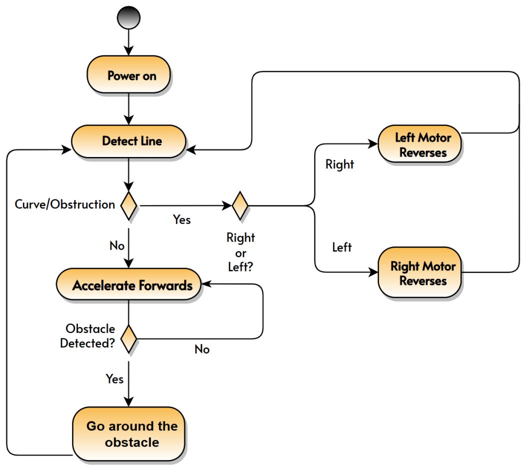
AUTONOMOUS LINE-FOLLOWING CAR



[2]

The state machine diagram represents the behaviour of an autonomous vehicle starting from the **Power On** state. It first enters the **Initialisation** state, where sensors and motors are activated, the line sensor is calibrated, and the battery is checked. Once initialisation is complete, the vehicle transitions to the **LineFollowing** state, where it continuously follows a line on the track. If an obstacle is detected within 10 cm, it enters the **ObstacleDetection** state, stopping the motors and using sensors to detect the obstacle and its colour. If no path is available, the vehicle moves to a **Stop** state. If an alternate path is available, it transitions to **CalculateAlternatePath**, scans the environment, computes a new route, and rotates to the new path. Once a valid path is found, it resumes line following.

ACTIVITY DIAGRAM



[3]

The Activity Diagram depicts the behaviour of the autonomous vehicle from a power on state to an avoiding obstacle state. Upon turning on the system the vehicle's behaviour starts with the detection of the line. The vehicle is programmed to detect curves or obstacles. When this occurs, the system determines which direction to turn. If a curve is detected, then the system will activate either the left motor or right motor in reverse to be able to adjust. If the vehicle does not detect a change in the line, it will continue to accelerate moving perpendicularly to the right line while moving forward. While in motion, the system actively checks for obstacle detection. When an object is detected, the vehicle runs the avoid routine and loops back to following the line. The Activity Diagram shows control flow pertinent to decision making and navigation logic.

Internal Block Diagram Overview

Internal Block Diagram is the physical structure of the autonomous vehicle's subsystems and how they interact. The four main units of the system are Power System, Sensor System, Controller Unit, and Motor System.

Power System

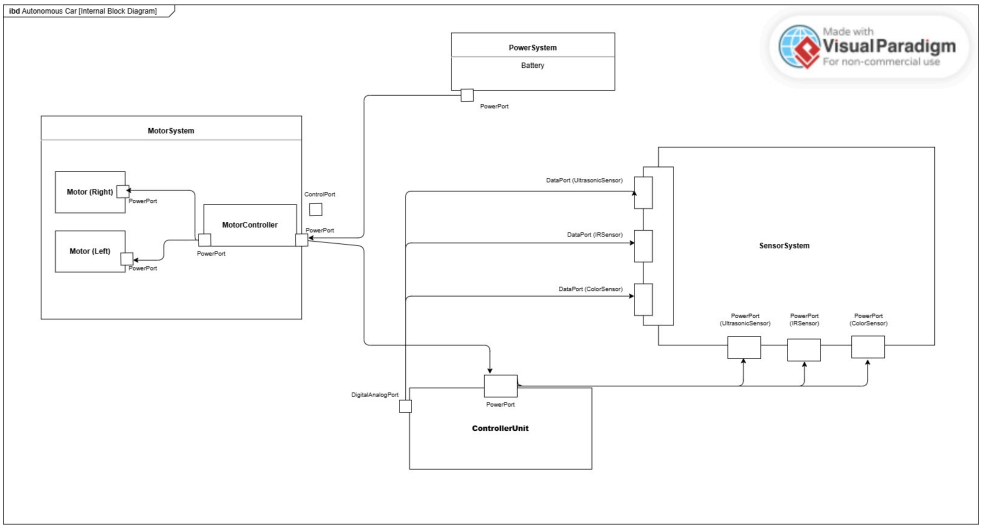
The battery supplies power to all the main components via separate PowerPorts, ensuring a continuous power supply to sensors, controller, and motors.

Sensor System

Includes an Ultrasonic Sensor, IR Sensor, and Colour Sensor. These are all powered and send data to the Controller Unit through DataPorts to enable detection of the environment and identification of obstacles.

Controller Unit

Acts as the primary processor. It takes in sensor inputs, processes data, and sends instructions through a DigitalAnalogPort to the Motor Controller. It is also directly powered from the battery.



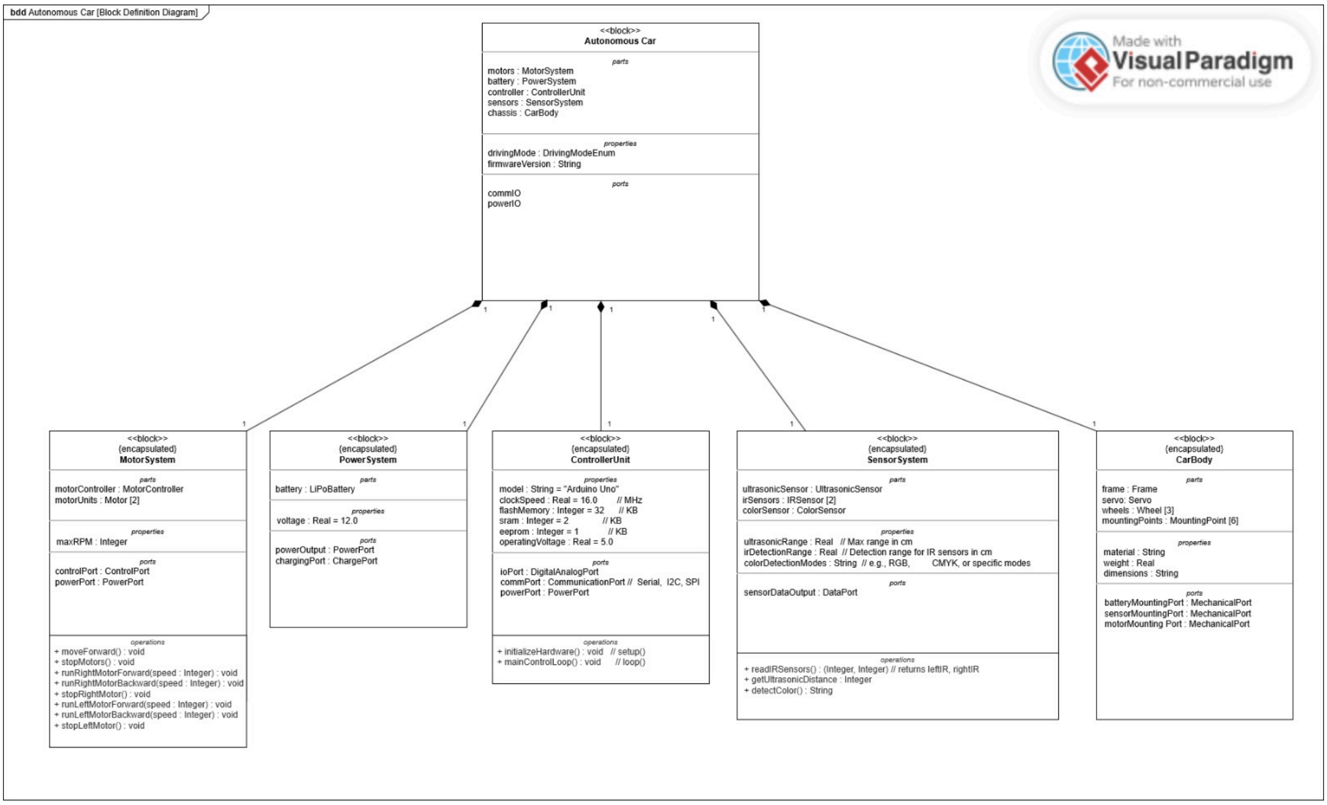
Motor System

Comprises Left and Right Motors controlled by a Motor Controller. The controller divides power and executes movement commands based on signals from the Controller Unit.

This modular structure ensures stable communication and control across all components, forming a complete autonomous vehicle system.

[4]

Figure [5] shows a **Block Definition Diagram (BDD**) that provides post-level structure and decomposition of the autonomous vehicle system using five former subsystems. These are the contained subsystems of the autonomous vehicle system as shown in the BDD: Motor System, Power System, Controller Unit, Sensor System, and Car Body. Each subsystem contains functions, activities, or components needed to accomplish vehicle autonomy.



[5]

Motor System

This subsystem contains two motors (left and right) and a motor controller. The motor system will define the movement of the vehicle for backwards, forwards, and directional. The system also defines controlled operations for control of the individual motors and defines important parameters such as maximum RPM.

Power System

This subsystem provides electrical power to all of the subsystems of the vehicle via a LiPo battery. The system defines the system-wide voltage and includes ports for charging and power distribution.

Controller Unit

This subsystem represents the main logic unit (Arduino Uno). This is the central logic unit that takes inputs from the sensors and executes control algorithms. It defines properties including clock speed, memory, and voltage, and defines ports for I/O and communication. The controller also defines operations, such as hardware initialization and where the main-loop command is executed.

Sensor System

The sensor system contains multiple sensors including two IR sensors to track the line, a color sensor to see the color of obstacles, and an ultrasonic sensor to measure distances. It provides sensor data to the controller through a single data output port, and implements functionality to read sensor values.

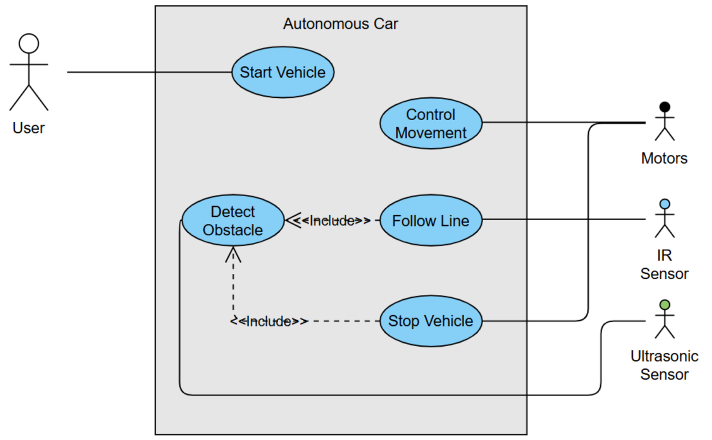
Car Body

The car body represents the mechanical nature of the vehicle, including the frame, wheels, servo motor, and mounting positions. It captures attributes associated with the car body including weight, size, and material, and it provides standard ports to connect the motors, sensors, and power system.

The modular design offers a clear separation of functions, and reduces the complexity of integrating the system while improving ease of maintaining and scaling the autonomous vehicle design.

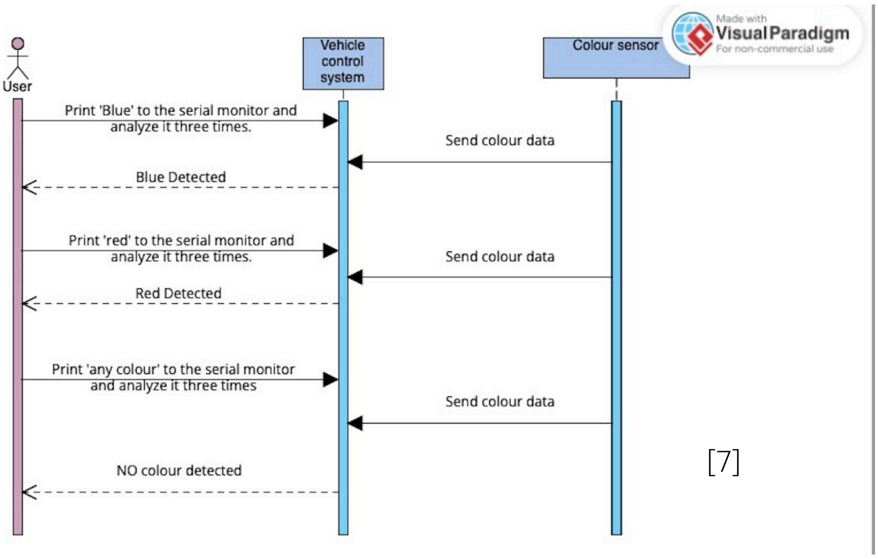
Use Case Diagram

The Use Case Diagram shows the way the user interacts with the system of the autonomous car. The user instantiates the Start Vehicle use case, and the core actions like Control Movement and Follow Line can occur with coordination of the motor and IR sensor. The Detect Obstacle use case, enabled by the ultrasonic sensor, will be invoked by both routine, line following or stopping. Once the obstacle is detected the system goes into the Stop Vehicle use case to avoid collision. This diagram covers the key functions of the system and their dependencies on sensors and actuators.



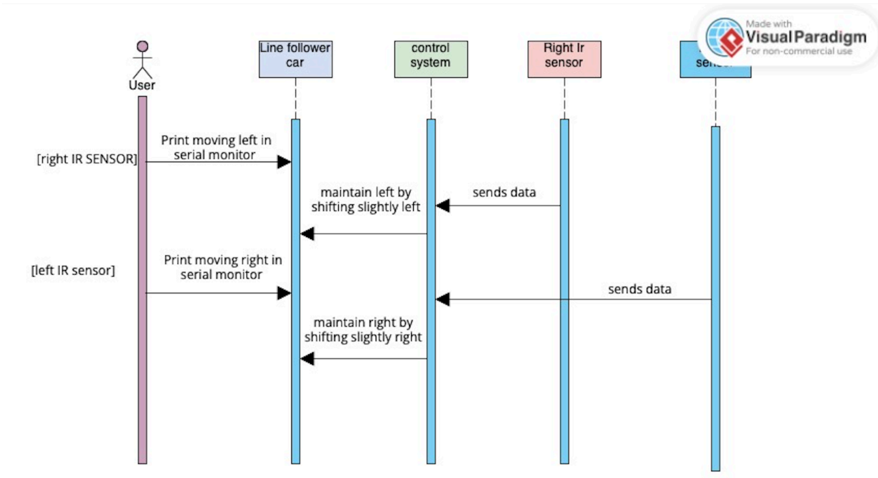
[6]

Sequence Diagram



[7]

The sequence diagram illustrates the interaction between a user, the vehicle control system, and the colour sensor. The process begins with the colour sensor sending colour data to the vehicle control system. Based on the received data, the system performs three checks and prints the result to the serial monitor. If the detected colour is blue or red, it prints "Blue" or "Red" respectively and confirms detection. If no specific colour is detected, it prints "any colour" and concludes with "NO colour detected." This looped analysis helps ensure reliable colour identification and response within the autonomous system.



[8]

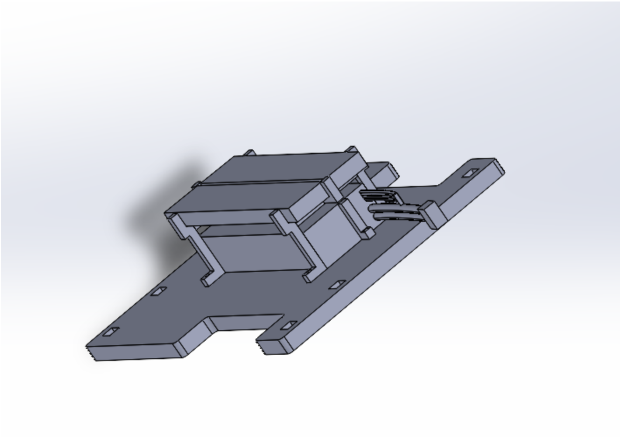
This sequence diagram illustrates the interaction between a user, a line-following car, its control system, and IR sensors. The system uses left and right IR sensors to detect the vehicle's alignment on the track. If the right sensor detects deviation, the system prints "moving left" on the serial monitor and adjusts the car to the left to maintain alignment. Similarly, if the left sensor is triggered, the system prints "moving right" and shifts the vehicle slightly to the right. This constant data exchange between sensors and the vehicle control system enables smooth and continuous path correction for effective line following.

III. Structural Component Design

Figure X illustrates three key structural components designed and fabricated to ensure the secure mounting of critical modules within the autonomous vehicle prototype. All designs were created using CAD tools such as SolidWorks and Tinkercad, then 3D printed using PLA or laser-cut from wood depending on the required strength and positioning.

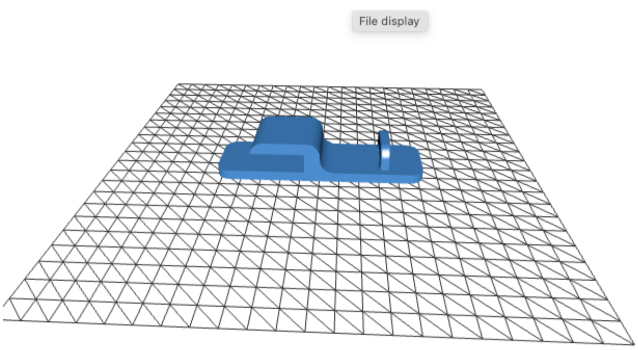
A. Battery Holder

The largest structure in the design layout serves as the **battery holder**, crafted to accommodate a standard LiPo battery. The dimensions were tailored to snugly fit the 7.4V 3200mAh unit while allowing cable clearance. Mounting holes and slots were integrated to ensure easy fastening to the vehicle base, while still allowing for efficient replacement and recharge operations. Its position on the main chassis was carefully selected to maintain center-of-gravity balance.



B. IR Sensor Holder

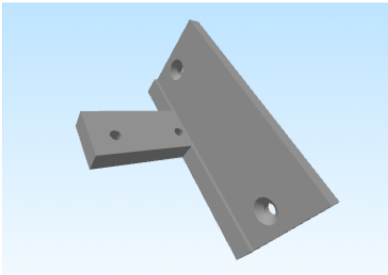
The component shown on the left (top corner) in the image is the **IR sensor holder**, designed to house the ST1140 IR module. It includes a precise cutout and screw holes to secure the sensor tightly in place. The holder tilts the IR sensor at an optimized downward angle to ensure maximum surface detection accuracy. This ensures stable performance during line following, especially when tracking black lines on white backgrounds.



[9]

C. Motor Holder

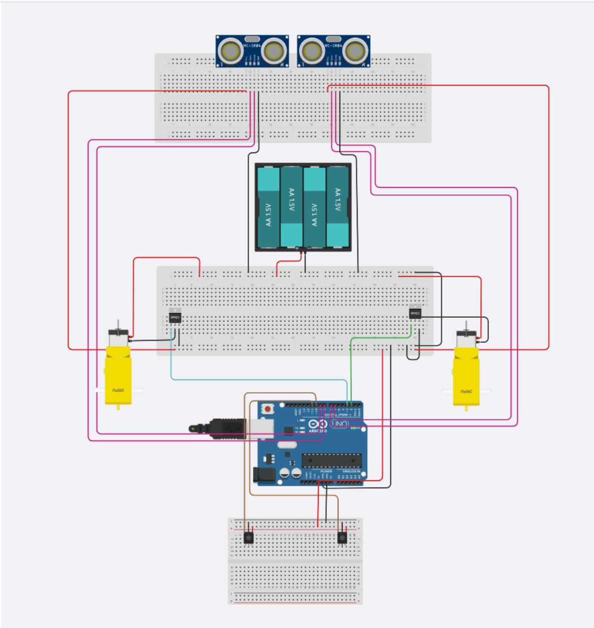
The component shown in the bottom-left corner represents the **motor holder**. It was designed to tightly fit standard 5V–9V DC motors and features side flanges with 2.5 mm screw holes for stable attachment to the wooden base. The motor shaft alignment was carefully considered in the design to avoid slippage and ensure even torque delivery to the rear wheels, enhancing driving stability.
All parts were designed for modularity and ease of assembly, contributing to a more maintainable and adjustable prototype.



[10]

IV. COMPONENTS

COMPLETE SCHEMATIC

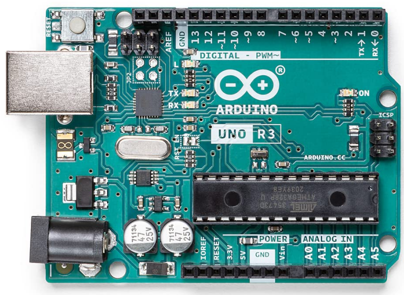


[11]

A variety of hardware components were integrated into the autonomous vehicle prototype. Each component played a specific role in ensuring the successful execution of functions such as motion control, obstacle detection, line following, and colour recognition.

A. Arduino UNO

The Arduino Uno serves as the brain of the system, interfacing with all sensors and actuators. It features the ATmega328P microcontroller, an 8-bit RISC core, 14 digital GPIO pins, and 6 analog input pins. It processes incoming data and controls motor drivers accordingly.



[12]

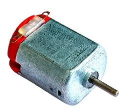
B. LiPo Battery

A 7.4V 3200 mAh 20C LiPo battery from Conrad was used as the power source for the L298N motor driver and motors. With a compact design and 210 g weight, it provides sufficient discharge current for all driving components.

[13]

C. 9V DC Motors

Two 9V brushed DC motors were installed to drive the rear wheels of the prototype. These motors offer sufficient torque and speed for small-scale

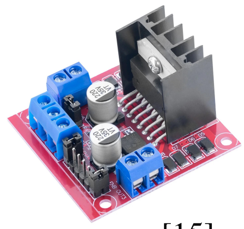


[14]

autonomous navigation. They were powered through the L298N driver instead of the Arduino to meet current demands.

D. L298N Motor Driver

The L298N module was used to drive the DC motors and supply power to the Arduino. It accepts power from the LiPo battery and provides direction and speed control via PWM signals from the Arduino. The module also ensures shared ground and voltage regulation.



[15]

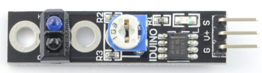
E. HC-SR04 Ultrasonic Sensor

The HC-SR04 module was used for real-time distance measurement between 2 cm and 3 m, ideal for obstacle detection. It uses 40 kHz ultrasonic pulses and reflects them back to compute the distance. The sensor provides input to the system to decide when to stop or reroute.

[16]

F. ST1140 IR Sensors

Two ST1140 IR sensors were mounted at the base to follow a black line on a light surface. Each sensor emits infrared light and detects reflections to determine if the surface is reflective (white) or absorbing (black). This forms the basis of the vehicle's path-following logic.



VIII. CONCLUSION

This paper presented the development of a line-following autonomous vehicle capable of making independent decisions based on the color of detected obstacles. The experimental results demonstrate that the prototype successfully fulfills all primary functional requirements.

For future enhancements, we plan to integrate two additional DC motors to replace the current ball-bearing front wheel, aiming to improve stability and manoeuvrability. Furthermore, we intend to mount an extra HC-SR04 ultrasonic sensor at the rear of the vehicle. Both ultrasonic sensors will be attached to a servo motor, enabling dynamic environmental scanning and more consistent decision-making. To support these hardware upgrades, a microcontroller with an increased number of digital and analog I/O pins will be adopted.

ACKNOWLEDGMENT

This project was made possible through the support and resources provided by Hamm-Lippstadt University of Applied Sciences. We would like to express our sincere gratitude to Prof. Dr. S. Henkler and Prof. G. Wahrmann for their invaluable guidance, continuous support, and constructive feedback throughout the project.

APPENDIX

All team members contributed equally to the design, development, and successful completion of this project.

REFERENCE

[1] Joy-IT, "Color sensor module TCS3200," *Joy-IT.net*, [Online]. Available: https://joy-it.net/de/products/SEN-Color. [Accessed: 21-Jun-2025].

[2] R. Santos, "Complete Guide for Ultrasonic Sensor HC-SR04 with Arduino," *Random Nerd Tutorials*, [Online]. Available: https://randomnerdtutorials.com/complete-guide-for-ultrasonic-sensor-hc-sr04/. [Accessed: 21-Jun-2025].

[3] Lextronic, "Capteur de ligne Arduino OpenST1140," *Lextronic.fr*, [Online]. Available: https://www.lextronic.fr/capteur-ligne-arduino-openst1140-51718.html. [Accessed: 21-Jun-2025].

[4] Components101, "L293D Motor Driver Module," *Components101.com*, [Online]. Available: https://components101.com/modules/l293n-motor-driver-module. [Accessed: 21-Jun-2025].

[5] Arduino, "Arduino UNO Rev3 with long pins (Retired)," *Arduino Documentation*, [Online]. Available: https://docs.arduino.cc/retired/boards/arduino-uno-rev3-with-long-pins/. [Accessed: 21-Jun-2025].

[6] Autodesk, "Tinkercad Circuits: Online Simulator for Arduino and Electronics," *Tinkercad.com*, [Online]. Available: https://www.tinkercad.com. [Accessed: 21-Jun-2025].