Prototype of Autonomous Line-Following Vehicle

1st Rubayet Kamal
Electronic Engineering
Hochschule Hamm-Lippstadt
Lippstadt, Germany
rubayet.kamal@stud.hshl.de

2nd Yuming Wang
Electronic Engineering
Hochschule Hamm-Lippstadt
Lippstadt, Germany
yuming.wang@stud.hshl.de

3rd Mert Yavas
Electronic Engineering
Hochschule Hamm-Lippstadt
Lippstadt, Germany
mert.yavas@stud.hshl.de

4th Joseph Asare
Electronic Engineering
Hochschule Hamm-Lippstadt
Lippstadt, Germany
joseph-asare.owusu@stud.hshl.de

***Abstract*—This project report presents the design and implementation of a line-following autonomous vehicle prototype. It is designed via Solidworks and Fusion 360 which were either laser cut or 3D printed. The vehicle utilizes line sensors to adhere to a predefined path. Integrated ultrasonic sensors provide real-time obstacle detection, allowing the car to evaluate its surroundings and make informed decisions to avoid collisions. Additionally, the system incorporates color sensors to identify and respond to various obstacle colors, enhancing its adaptability and decision-making processes. This approach demonstrates the vehicle's ability to operate autonomously in dynamic environments. The project highlights the integration of multiple sensor technologies to create a robust and intelligent navigation system capable of both line-following and obstacle management.**

***Index Terms*—System Modelling Language (SysML)**

I. Introduction

This paper focuses on the team building a prototype using knowledge gained from previous semesters. The main objective is to build a self-driving car with the ability to detect and avoid obstacles based on colors detected. After the color is detected the car should be able to decide what to do.

Section II details the formal requirements of the line-following robot. In Section III, the design process is outlined, explaining how the project evolved from concept to execution. Section IV describes the components used, listing and explaining the roles of the sensors and actuators integral to the design, with an emphasis on their interdependencies.

Once the design and wiring or soldering of components were completed, programming commenced. This is covered in Section V, which explains the algorithms and includes code snippets demonstrating the execution of various functions. Following a thorough analysis, Section VI introduces the use of System Modeling Language (SysML) to model the system.

Section VII describes the simulation of the prototype using UPPAAL-5.0.0, while Section VIII presents the results obtained in the real world. The final section, Section IX, concludes the paper, summarizing the project's outcomes and potential future work.

II. Requirements

To develop an autonomous vehicle capable of driving autonomously on a specified track using line detection, several critical requirements were established and visualized through a requirement diagram using Visual Paradigm Online. The primary objective was to create a prototype of an autonomous vehicle, which encompasses three main areas: Obstacle Management, Track Management, and Direction Management.

Within Obstacle Management, an additional requirement, Colour Management, was derived, emphasizing the vehicle's need to detect obstacles and perform functions based on their color. This area also included a refinement that specifies the distance at which the vehicle must stop and begin evaluating obstacles, ensuring timely and accurate obstacle handling.

Track Management necessitated the derivation of a requirement focused on Speed Optimization. This requirement was crucial because the vehicle needs to adjust its speed dynamically whenever it makes right or left turns, ensuring smooth and efficient navigation on the track.

Direction Management led to the development of a requirement indicating that the vehicle should be capable of executing different routing strategies. This requirement was further detailed into four specific types of routing that the vehicle should manage, showcasing its versatility in navigation.

By integrating these requirements, the vehicle prototype aims to be a sophisticated autonomous system, proficient in obstacle detection and response, track navigation with speed adjustments, and versatile routing capabilities.

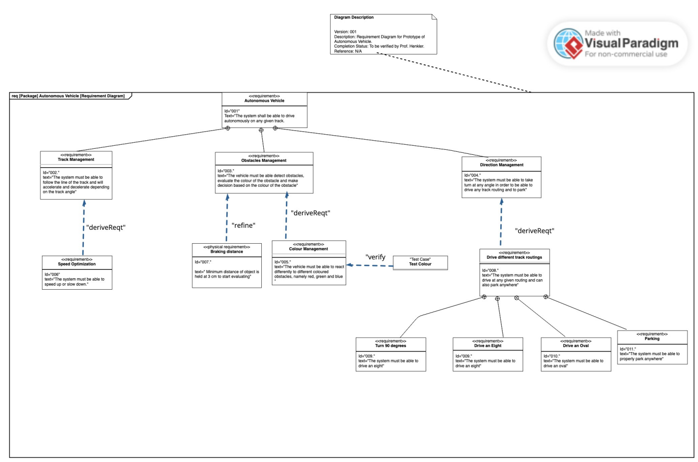

Fig. 1. Requirement Diagram

III. Design

SolidWorks and Fusion 360 were employed to design various essential components of the prototype. The designs were then exported as .dxf files for laser cutting, with wood serving as the material, and as .stl files for 3D printing using PLA. The figures below present the comprehensive drawings and 3D views from different perspectives of the designed prototype.

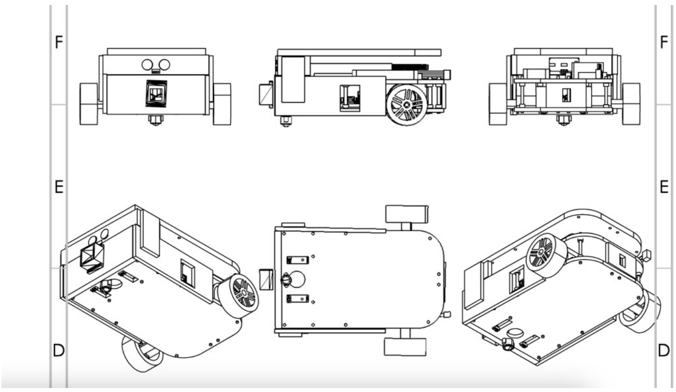

Fig. 2. Drawing

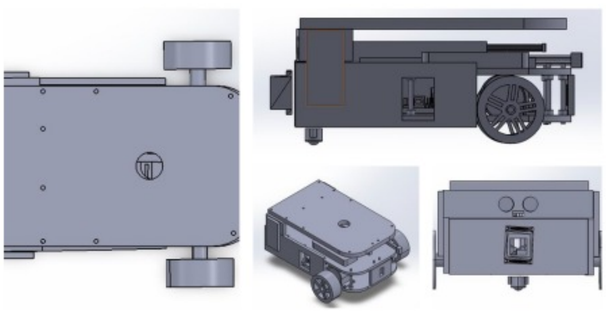

Fig. 3. 3D View

A number of different parts were designed separately to make this possible. These parts will be discussed in the upcoming subsection, detailing how the shapes of these parts played a crucial role in the prototype. The figures below present the overall drawings and 3D views from different perspectives of the designed prototype.

A. Base, Body and Cover

The design process began with the creation of the base, a critical component due to several key considerations. Firstly, the ST 1140 sensor operates optimally within a specific range, necessitating precise placement for stable performance. To achieve this, we crafted two holes tailored to fit our line sensor accurately. Additionally, we needed to accommodate the front wheel, a ball-bearing wheel with particular radius and height specifications. To ensure the wheel's proper fit and maximize its height, we designed a hole with a radius slightly smaller than the wheel's, ensuring a secure fit. All components were screwed in place, providing stability while allowing for easy modifications as needed.

The body and cover were designed with identical shapes using SolidWorks and then laser cut from a .dxf file. The base and cover were constructed from 6 mm thick wood, while the body, which supports more weight, was made from 10 mm thick wood. This distinction in thickness ensured structural integrity where it was most needed. For flexibility in adjustments, the base and body are connected using stick supports, detailed further in the next subsection. These supports fit into holes positioned at various corners, allowing for stable yet adjustable assembly. Figure 1 below illustrates these adjustments and components, showcasing the careful consideration given to each design element.

Fig. 4. 3D view of base, body or cover

B. Stick support

The stick support is crucial for ensuring that changes to wiring or other components can be made easily. This is because the sticks function as finger joints, allowing for simple connection and disconnection between the base and the body. A total of four sticks were used, with one of them shown in the figure below. Each stick was 3D printed using PLA, and the production time for each was approximately 10 minutes.

Fig. 5. Stick Support

C. Colour Sensor Cover

The TCS3200 color sensor is highly sensitive to ambient light, making calibration challenging as the values fluctuate with changes in surrounding light. To address this issue, a color sensor cover was designed to ensure that the photoresistor only receives light from the target source, effectively blocking out ambient light. The cover has a length of 3 cm, which is the optimal distance for the color sensor to provide accurate results according to the manufacturer's specifications. This cover was mounted on the front face of the prototype, parallel to the ultrasonic sensor, ensuring that the vehicle stops at the exact distance where the color sensor can accurately evaluate the color. The figure below illustrates the discussed part.

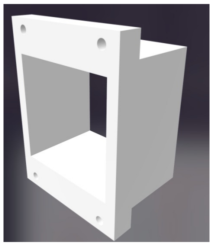

Fig. 6. Colour Sensor Cover

D. Ultrasonic Holder

The Ultrasonic sensor, HC SR-04, can detect objects within a 45-degree range from its face. Therefore, it is crucial to ensure the stability of the ultrasonic sensor. An unstable sensor can result in the prototype failing to stop at the desired distance. Additionally, when the sensor is moving, it sometimes detects noise, even without any object in front of it. To address this issue, a stabilizing part was designed in Fusion 360, and the resulting .stl file was used to 3D print the part using PLA. The figure below shows the designed stabilizing part.

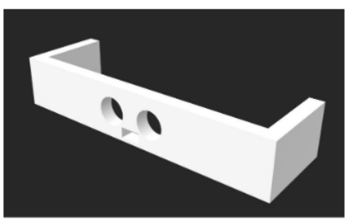

Fig. 7. Ultrasonic Holder

*E. 2 * Motor Holder*

Two 9V DC motors were used to drive the wheels of the prototype. Stability of the wheels is crucial to ensure that the height of the wheels relative to the ground remains constant, preventing the vehicle from becoming misaligned or bent. To secure the motors and ensure wheel stability, a specific part was designed in SolidWorks and then 3D printed. These parts feature 2.5 mm screw holes, making it easy to drill in screws and securely attach them to the base. The figure below shows the designed part.

IV. Components

Various hardware components were used in this project with each having distinct properties and performing specific functions. Components incorporated are mentioned below along with some of their schematic. The figure below shows the complete schematic including all components.

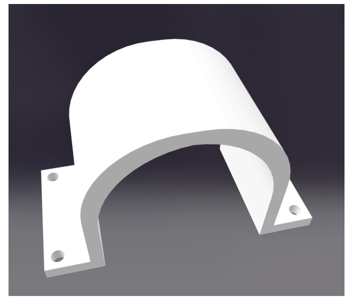

Fig. 8. Motor Holder

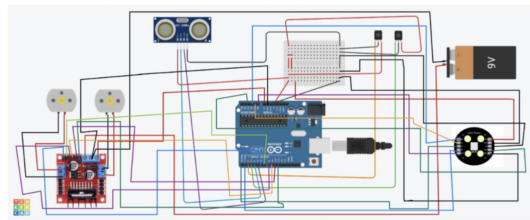

Fig. 9. Complete Schematic

A. Arduino UNO

The Arduino Uno serves as the brain of our prototype, responsible for gathering data from sensors, processing it, and initiating appropriate actions through actuators for navigation purposes. This development board incorporates the Atmega328 microcontroller from the Atmel microcontroller family, featuring an 8-bit RISC processor core. The board is equipped with 14 digital GPIO pins and 6 analog pins, as shown in Figure 8. Communication with the PC is established through a USB connection.

Key features of the Arduino Uno include the Atmega328 microcontroller, an 8-bit RISC processor core, 14 digital GPIO pins, and 6 analog pins. The USB connection facilitates communication with the PC. These features enable the Arduino Uno to effectively manage sensor data and control actuators, making it an integral part of the prototype's operation. The figure below illustrates the layout of the Arduino Uno board.

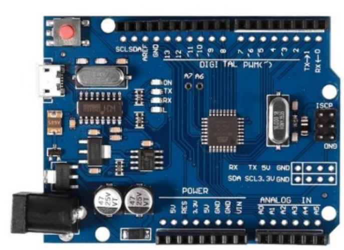

Fig. 10. Arduino Uno

B. Battery

Conrad 7.4V 3000 mAh 20C Eco-Line LiPo battery was used to power L298N. The battery has a weight of 210 g according to the manufacturer.

Fig. 11. LiPo Battery

C. 9V DC Motors

Two 9 V DC Motors as shown in the figure below were the primary tool to run back wheels. These motors were powered from the battery as the Arduino does not have enough powering capability. This is where L298N came into use and will be discussed in the next subsection.

Fig. 12. DC Motor

D. L298N

L298N Motor Controller was used to power the DC Motors. It also was used to power the Arduino Uno. As Arduino UNO cannot provide more than 5 Volts, it became necessary to use the controller to run the motors. The Motor Controller has 3 Inputs. The first one is the Voltage that it receives from the battery. The second one is to the ground of the Arduino and the battery. Finally, the third one is the 5V input for the Arduino UNO.

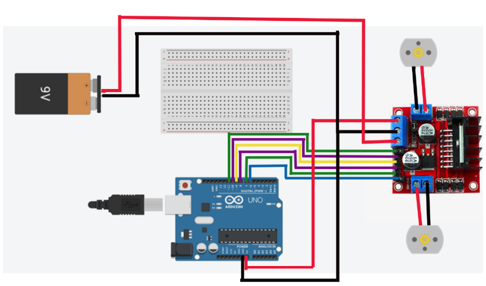

Fig. 13. L298N Motor Controller Schematic

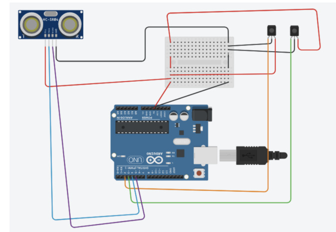

Fig. 14. Schematic of HC SR-04 and ST1140 with Arduino

E. HC SR-04

This module is ideally suited for distance measurement in a range between 2 cm and 3 m. With a resolution of about 3 mm, distances can thus be measured by ultrasonic signal. If a signal (falling edge) is applied to the trigger input, a distance measurement is performed and output at the echo output as a PWM TTL signal. The ultrasonic distance sensor is particularly suitable for obstacle detection, distance measurement, as a level indicator and for industrial applications.

When triggered, the ultrasonic loudspeaker (transducer) emits an ultrasonic noise of maximum 200µs. The ultrasonic loudspeaker emits a 40 kHz signal. This means that 8 periods (edge changes) are emitted within the 200µs in which the sensor emits its ultrasonic noise. In order to arrive at these 8 periods of the 40 kHz signal mathematically.

Transmission of the ultrasonic signal is started when a 10µs long start signal (ActiveHigh) is received at the ”Trigger input pin”. After transmission, the signal is activated at the ”Echo output signal pin” (ActiveHigh). If the reflected signal is now picked up again at the microphone, the echo signal is deactivated again after detection. The time between activation and deactivation of the echo signal can be measured and converted to distance, as this also corresponds to how long it takes the ultrasonic signal to cover the distance between loudspeaker-¿ reflecting wall -¿ microphone in the air. The conversion is then made by approximating a constant air velocity - the distance is then half the distance traveled.

The figure below shows the schematic of this component with Arduino UNO.

F. TCS3200

The TCS3200 chip is designed to detect the color of the light falling on it. It has an array of photodiodes (a matrix of 8x8, for a total of 64 sensors). These photodiodes are covered with four types of filters. Sixteen sensors have a RED filter and thus can only measure the red component in the incident light. Another sixteen have a GREEN filter and sixteen have a BLUE filter. Each visible color can be divided into three basic colors. So these three types of filtered sensors help measure the weighting of each of the primary colors in the incident light. The remaining 16 sensors have a clear filter. The TCS3200 converts the intensity of the incident lighting into a frequency. The output waveform is a square wave with 50% duty cycle. You can use the timer of a MCU to measure the period of the pulse to determine the frequency. The output

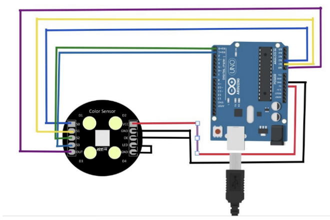

Fig. 15. Colour Sensor Schematic

of the TCS3200 is available on one line, which means that only 1 pin is available as output. The intensity of the red, green, blue and clear channels can be determined by using the inputs, S2 and S3, which are used to select the sensors, to provide their output on the output line. That means, the inputs S2 and S3 are used to control what kind of sensors give their signal to the output line. There are also two other inputs S0 and S1, which are used to regulate the output frequency. That means that with these inputs the intensity of the frequency of the channels RED, GREEN, BLUE and CLEAR can be regulated. The figure below shows a schematic of the colour sensor TCS3200 when connected to Arduino UNO.

G. ST 1140

The sensor module detects whether there is a light-reflecting or light-absorbing surface in front of the sensor. The IR sensor emits an infrared light that hits an obstacle, it is reflected and received by the photodiode. It has a detection range between 2 to 20cm. The sensitivity of both the receiver and the strength of the transmitter can be adjusted. This makes it possible to adjust the distance at which the sensor is triggered. This is done with two different potentiometers. In this project, 2 IR sensors were used and were installed at the base of the prototype. Theyy aid in line following and object detection. When a black line is detected, the sensor sends 1 or HIGH to the microcontroller. On the other hand, when a white or any other line is detected, the sensor sends 0 or LOW to the microcontroller. This was crucial in forming an algorithm to ensure the prototype follows the line. Here a picture of the above mentioned thing should be mentioned and then pinout should be mentioned as per schematic. The schematic figure should be added. It can be seen in Fig. 12 how the Line Sensors were connected to the Arduino.

V. ALGORITHM

Our prototype is programmed in Arduino IDE.In this section, we discuss the different function of the car and the algorithm under the behavior.We separate the behaviors of car into Line Following, Object Mnagement, Colour Evaluation, Overtake, Parking, and U-Turn. It should be mentioned that the prototype can do all the following functions in tracks that has white surface and black line. However, for this paper, all results and simulations have been shown for a black surface with white line. A boolena variable called whiteLine is created that can be tuned to 'true' or 'false' in order to ensure the prototype perform the desired fucntions in the available track. The following link is where our github repository can be found: https://github.com/RubayetKamal/Autonomous-Line-Following-Vehicle-Prototype

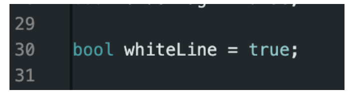

Fig. 16. Enter Caption

A. Line Follower

When one IR sensor detects black for example left sensor, that leads to a turning right behavior, to realize it, we need to make the left motor speed up and at the same time, make the right motor slow down. For turning left behavior, we realize it in the same way.

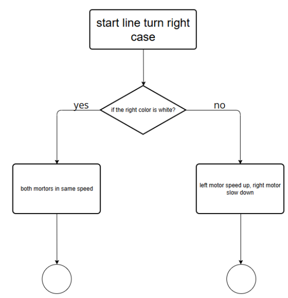

Fig. 17. Line Follower

B. Object Management

The car is supposed to stop while detecting one object with 3cm so that the color sensor can get a more accurate detecting rate. If the sensor is failed to detect one object or the distance from the object is more than 3 cm, the car will keep moving and the sensor will keep detecting at meantime.

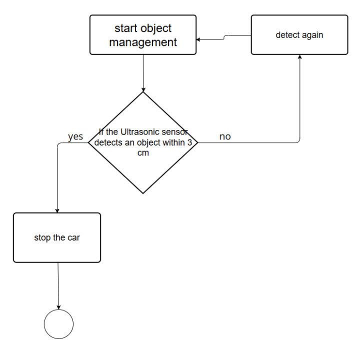

Fig. 18. Object Management

C. Colour Evaluation

As shown in figure 17, the prototype can react differently depends on the color of the objects it detects. The prototype will execurate overtaking while detecting blue, on the other hand, if the color of object is blue, the car will do parking procedure, if it is any other color, the car will turn 180 degrees.

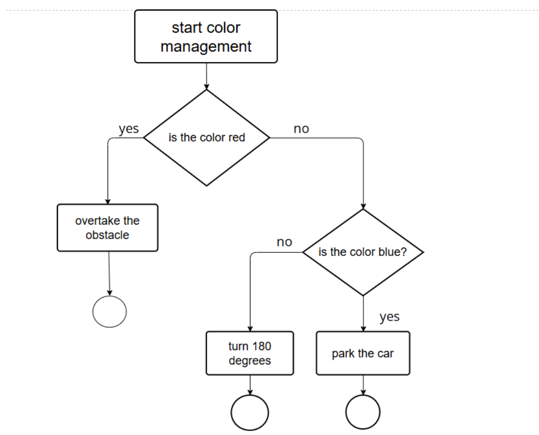

Fig. 19. Color Evaluation

D. Overtake

Once the prototype detects the red, it will go backward and then ultrasonic sensor will detect again whether there is one object within 100 cm. In this case, if it is true, the car will turn left go forward then turn right until the IR sensor detects the line again. On the other hand, if there is no object within 100 cm, the prototype will turn 180 degrees, this procedure is same as the following case for Parking behavior.

E. Parking

The prototype will first go backward, then it will turn left 45 degrees, then go behind and stop. In this case, if the ultrasonic sensor detects any object, the car will drive forward until detecting the line, otherwise the car will keep stopped.

F. U-Turn

Explained but the picture for this will be done by Leano later.

VI. Analysis using SysML

SYSML is a tool for system engineering. By using this, we are able to analyze the problems in a more efficient way, and set up goals to realize. Moreover, feedback of the system is also available while using this tool.

A. State-Machine Diagram

state machine is a tool we use to analyze different states of the prototype. In this diagram, we describe the whole state changes of our prototype through its lifetime. the car will follow the line, and do object detection, color detection.

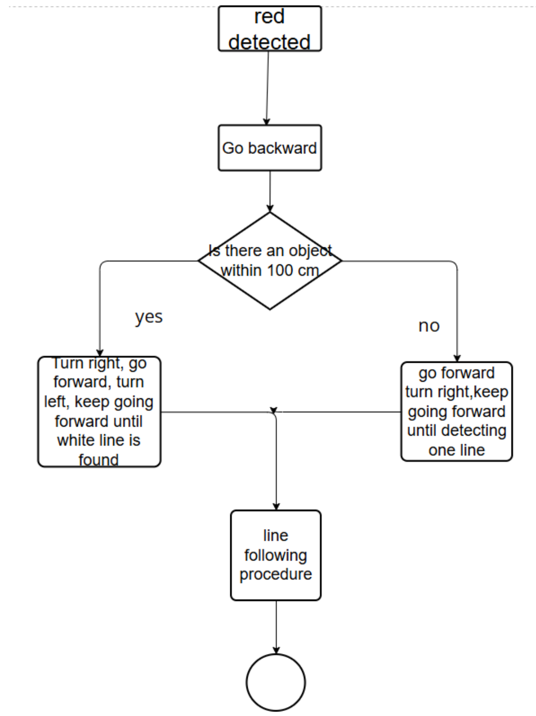

Fig. 20. Overtake

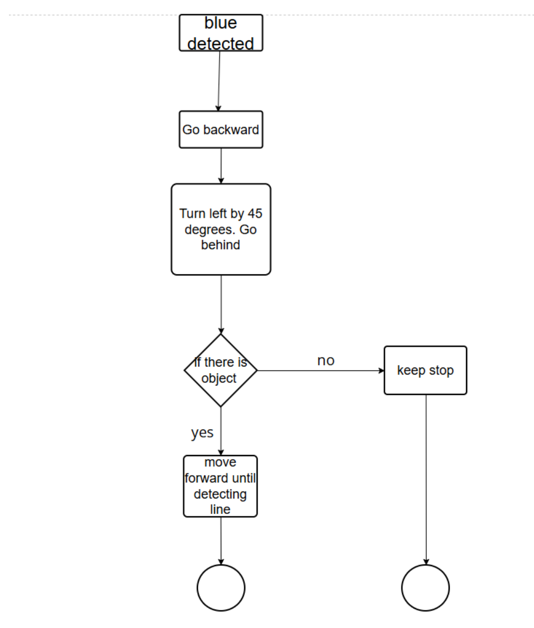

Fig. 21. Parking

B. Activity Diagram

Activity diagram descibes the behaviors of the car while running. With the power on, the car will first move forward, then it will detect the line, if it is on the line, the car will keep moving forward otherwise it will go back to line by turning

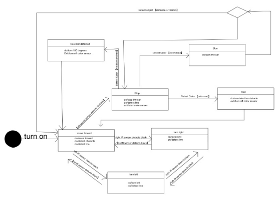

Fig. 22. state machine

right or left. At the same time, the ultrasonic sensor is keeping detecting obstacles, and if there is one object within range, the color sensor will start.

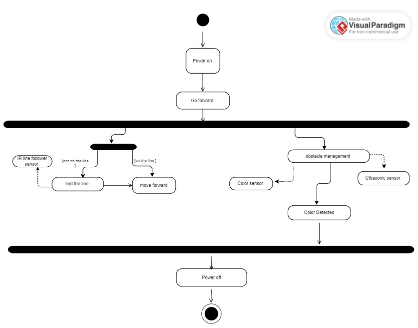

Fig. 23. activity diagram

C. Block Definition Diagram

Block Definition Diagram(BDD) is a tool we use to define the parts and values that have been used in the prototype. In our prototype, we use 3200 mah battery, powerbank, DC motor, ultrasonic sensor, color sensor, and IR sensor. More details can be checked in the diagram.

D. Interaction Structure for Colour Evaluation

Interaction Structure for COlor sensor evaluation shows the sequence of the color evaluation behavior.

E. Interaction Structure for Following Line

Interaction Structure for Following Line describes the sequence of the line following management. In this case, User, the car, control system, right IR sensor, and left IR sensor are showed. The sequence between these parts can also be checked.

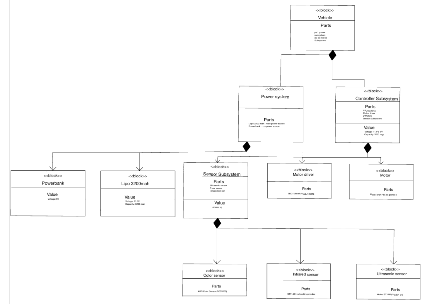

Fig. 24. BDD

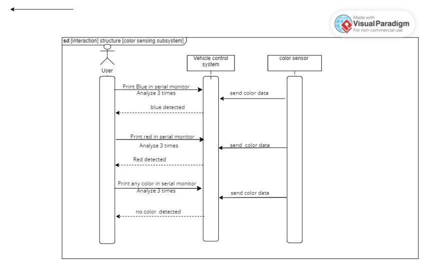

Fig. 25. sequence for color sensing

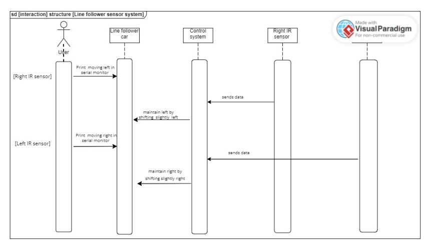

Fig. 26. sequence of Following Line

F. Interaction Structure for Obstacle Management

Interaction Structure for Obstacle management shows how we realize the function of avoiding obstacles, and the sequence of it.

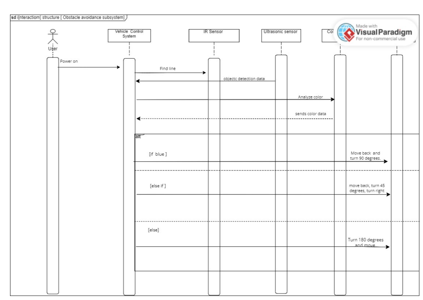

Fig. 27. sequence diagram of object avoidance

VII. Simulation

All scenarios have been simulated using UPPAAL-5.0.0. 4 templates were created: MyAuto, ObjectManagement, LeftHandle and RightHandle. MyAuto shows the overall picture of what the vehicle does. LeftHandle and RightHandle represent left and right line sensors respectively meaning the car will turn left if LeftHandle is trigerred and vice versa. Additionally, ObjectManagement is the integration of Ultrasonic Sensor with Colour Sensor. The figure below is a screenshot of a simulation scenario in UPPAAL-5.0.0.

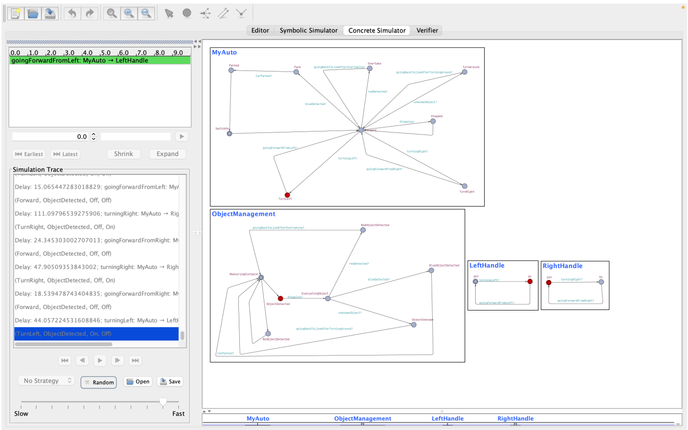

Fig. 28. Simulation in UPPAAL-5.0.0

VIII. Result

A. Complete Design

The figure below shows the final prototype after everything has been successfully connected.

Fig. 29. Complete Look

B. Line Follower

The figure below shows the car has been successfully programmed to follow line in a black track with white line.

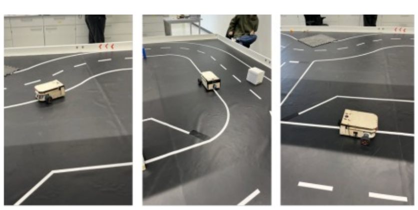

Fig. 30. Car Following Line

C. Object Management

The figure below shows that the car stops successfully when it detects an obstacle before starting to evaluate the colour.

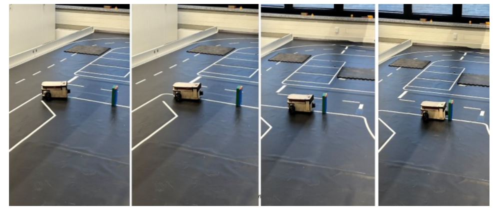

Fig. 31. Car Detecting Obstacle

D. Overtake

Fig. 11. shows the car goes behind, turns left 45 degrees, goes forward, turns right at an angle and then keeps moving forward until it finds a line again.

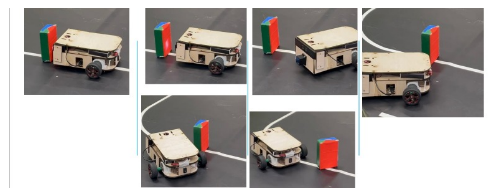

Fig. 32. Car Overtaking if Obstacle is red coloured

E. Parking

Fig. 12. below shows the vehicle parks if it sees a blue object within it's range. First it goes behind a little bit after which it turns right by aprroximately 45 degrees and then goes behind. It remains in this state until it sees an object within 100 cm from the front.

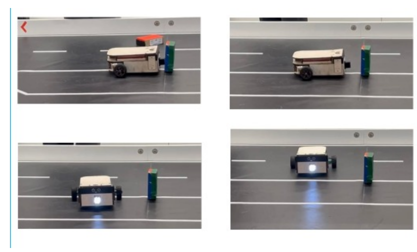

Fig. 33. Car to park if object is Blue

F. U-Turn

If the colour of the object is neither red nor blue, the vehicle will take a U-turn and start moving facing the opposite direction. It does so by rotating for as long as it sees the white line again.

IX. Conclusion

In this paper we presented a prototype of a line-following autonomous vehicle that is able to make self-decisions based on the colour of the obstacle. The results show that project has been a success in fulfilling the key requirements.

As part of future work, we plan to use two more dc motors that can be used as front wheel instead of the ball-bearing

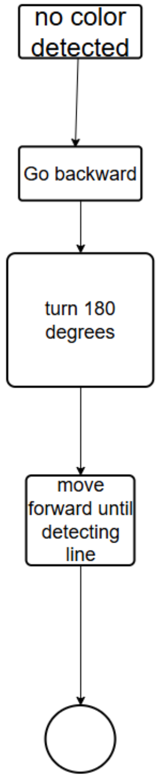

Fig. 34. Car taking U-turn for unknown coloured object

wheel. Additionally, we would like to use one more HCSR-04 at the back. Both these ultrasonic sensors are planned to be connected to a servo motor ensuring that the vehicle can monitor surroundings and make more consistent decisions. In order to add all these interesting new features, we would like to use a different micrcontroller that will have more digital and analog pins.

Acknowledgment

This project has been possible solely because of the support of tools and expertise provided by Hamm-Lippstadt University of Applied Sciences. Special mention to Prof. Dr.Stefan for continuous guidance and feedback allowing us to improve week-in week-out.

References

[1] G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955.
[2] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.
[3] I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.
[4] K. Elissa, “Title of paper if known,” unpublished.
[5] R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.
[6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].
[7] M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.