Farhad Hossain

AUTOTECH: Embedded Systems for Dynamic Route Navigation and Obstacle Avoidance

Thesis
CENTRIA UNIVERSITY OF APPLIED SCIENCES
Bachelor of Engineering, Information Technology
May 2024

ABSTRACT

Centria University of Applied Sciences	**Date** 19.03.2024	**Author** Farhad Hossain
Degree programme Bachelor of Engineering, Information Technology		
Name of thesis AUTOTECH: Embedded Systems for Dynamic Route Navigation & Obstacle Avoidance.		
Centria supervisor Henry Paananen		**Pages** 30 + 3
Instructor representing commissioning institution or company Henry Paananen		

The progress in automated driving technology has opened doors to enhance smart city mobility. Automated cars are becoming increasingly popular as the smart city agenda gains momentum. Incorporating autonomous driving systems through embedded technology further transforms urban mobility, providing outstanding comfort and effectiveness. Autonomous vehicles with embedded systems are commonly used to test a variety of algorithms and imitate real-world driving behaviour. These powerful computer systems are easily incorporated into automobiles, resulting in improvements in significance, safety, and performance. This technological integration not only improves the driving experience but also opens the door for new technologies like autonomous driving, connectivity, and advanced driver assistance systems (ADAS). As embedded systems grow more prevalent, they are changing the face of modern transportation, helping in a new era of smart and connected automobiles. This thesis analyses the enormous influence of embedded systems on the automobile industry and the creative solutions they provide to fulfil changing customer needs, improve vehicle performance, radically change vehicle safety, efficiency, and connection, and change the future of transportation. We explore their influence on these important areas using Tinkercad, a virtual prototyping environment. The research project explores the implementation of autonomous vehicle embedded systems for obstacle avoidance using Tinkercad. In addition, the integration of a car GPS navigation system with the capability to modify destinations or routes is investigated. By modelling a variety of automotive applications, from sensor networks to control systems, this study proves the feasibility and usefulness of embedded systems in real-world automotive settings, providing the groundwork for future research and development efforts. Embedded systems power Advanced Driver Assistance Systems (ADAS) and other features, resulting in considerable reductions in road fatalities. They optimize engine efficiency, regulate fuel consumption, and lower emissions, helping to create a more sustainable vehicle industry. This research demonstrates the potential and effectiveness of embedded technologies in improving vehicle safety and driver assistance. The thesis evaluates the performance of obstacle avoidance algorithms using Tinkercad simulation, and an investigation of an automobile GPS navigation system with dynamic route modification capabilities reveals the possibility for enhanced navigation experiences. Consequently, this thesis states that embedded systems are an essential part of automotive innovation, propelling advances in vehicle design, performance, and functionality. With the assistance of AI, ideas have been derived for this research project.

Keywords: ADAS, Automotive Industry, Dynamic GPS Navigation System, Embedded Systems, Tinkercad

CONCEPT DEFINITIONS

Embedded System

Integrated hardware and software for specific characteristics

Automotive Industry

Designing, manufacturing, and selling vehicles

ADAS

Advanced Driver Assistance Systems

Tinkercad

Web-based tool for virtual simulation

Obstacle Avoidance

Sensor-based collision control for automatic braking

Dynamic GPS Navigation System

Adjusts the route based on real-time conditions

ROS

Robot operating system

ABSTRACT
CONCEPT DEFINITIONS
CONTENTS

FIGURES

TABLES

1 INTRODUCTION

Technological advancements are accelerating the fast-evolving sector of self-driving cars, which offers a wide range of applications. Obstacle-avoiding autonomous vehicles are particularly notable in the automobile industry and have received a lot of attention. Autonomous vehicles have a remarkable capability of evaluating their environment, preventing collisions and nothing seems to stand in their way. Telematics features are integrated in their systems with dynamic GPS navigation, making them react fast to different type of situations. The aim is to develop an autonomous car that can move properly in diverse settings and make perfect choices. They evolve by making environment appraisal, hazard recognition, and assuring collisions prevention. The three principles of exploring places safely feature several methods to avoid obstacles. This feature ensures that the autonomous mobility is safe. In its essence, this technology is expected to provide hassle-free, reliable, and efficient travel, thereby taking the roads to the next level. Safety will always be the primary theme of the revolution, as it is aimed at reducing the number of accidents and traffic congestions.

This project intends to design an obstacle-avoidance car system with the help of online simulator Tinkercad. Supporting the basic development, the project aims to develop an obstacle detection system considerably reducing traffic accidents on highways. The detection of obstructions is performed in real-time by using the ultrasonic sensors and reacting with active obstacles avoidance algorithms. The specific goals of the project are to design and develop this obstacle detection system prototype, test it on Tinkercad, and properly evaluate its performance. The feature will probably significantly cut risks of highway accidents by avoiding collisions with objects, pedestrians, and other obstacles. Another idea is dynamic route navigation system that lets passengers choose what the best route is. It does not only serve the passengers comfortably but also reduces road traffic. With the help of this technology, the right routes are identified while traffic flow is improved and congestion decreases. Hence road safety is considerably enhanced. Rigorous testing in industrial and automotive contexts highlights the car's competence in obstacle avoidance, demonstrating its capacity to navigate safely across a variety of scenarios.

This paper researches the following key findings:

1. In the field of autonomous vehicle technology, how can embedded systems be enhanced to seamlessly combine dynamic route navigation and obstacle avoidance, allowing for better real-time flexibility in response to changing traffic conditions and environmental factors?

2. How can autonomous cars navigate and avoid obstructions in highly populated metropolitan regions where people walking/running, cycling, parked cars, and construction zones are continuously changing locations and configurations?

2 TECHNICAL BACKGROUND

Embedded systems, which are essential in modern navigation solutions, enable dynamic route planning and obstacle avoidance across a wide range of applications, including autonomous vehicles such as automated cars. This section explores deeply into the fundamental principles and methodologies that underpin dynamic route navigation and obstacle avoidance, as well as an examination of the integration of online simulation tools like Tinkercad and mobile application development framework like Flutter.

2.1 Dynamic Route Navigation

Dynamic route navigation in automobile navigation systems that use Flutter application entails integrating multiple technologies and algorithms to allow for adaptive route planning and alteration depending on real-time data and changing environmental circumstances. Enhanced A* algorithms are widely utilized for both global and local path planning due to their fast calculation speeds, path optimization, and other advantages in autonomous vehicle (Dang & Bui 2023).

2.1.1 Fundamental principles

Dynamic route planning is influenced by using real-time data and environmental conditions in the decision-making process. The main objective is to find the best or the near-optimal route from a starting point to a destination, while factoring in traffic conditions, road closures and user preferences.

a. Digital Maps: These are the fundamental parts demonstrating diagrams of highways, landmarks, and other geographical elements.

b. Path Planning Algorithms: The A* algorithms are essential tools implemented for finding the best possible route that connects the present position to the target location. By assigning a variable weight to the heuristic term of the evaluation function of the original A* algorithm, it enhances the influence of heuristic information and improves the computational efficiency and the A* algorithm speeds up the search process by bidirectional search (Islam, Narayanan &

Likhachev, 2016). Having in mind several factors, such as obstacle avoidance, vehicle dynamics, and environmental limits, the algorithms select the optimal path.

c. Sensor Integration: Dynamic route navigations are data-driven, making sure the decision-making process is based on real-time sensor information. Sensors such as GPS offers data on the environment around the vehicle, thus helping in making smart decisions pertaining to route changes.

2.1.2 Implementation in apps

Flutter apps for automobile navigation systems often have the following components and functionalities:

Map Display: Flutter, besides the presenting of digital maps in the app UI, utilizes the widgets and frameworks accordingly. Now users can smoothly see their current position, target the destination and possible paths of the route on the interactive map.

Real-time Data Integration: Flutter apps collaborate with different sources to enable users to receive on-time helpful data such as traffic updates, road closures, weather conditions and others, in real time too, which help users choose the best route.

Path Planning and Optimizations: The Flutter software is powered by algorithms of path planning which select the best possible routes according to the specified input data and current real-world conditions. The modules encompass items like traffic lights, road status, and user behaviors.

2.2 Obstacle avoidance

The ultrasonic sensors are important for obstacle avoidance to detect a nearby obstacle. These sensors reverberate sound waves and use the strength and time delay of the reflected signal to find the actual obstacle distance. Processed data detect obstacles and identifies potential collisions. Taking account of the vehicle’s limitations, obstacles' distance, destination and any real-time changes, the path planning algorithms generate a safe trajectory. This technique offers a simple and cost-effective solution for obstacle avoidance in controlled environments (Russell 2019).

2.2.1 Enhanced embedded systems for obstacle avoidance

Optimising Sensor Fusion Techniques: The systems of on-board senses of autonomous vehicles can be improved through sensor fusion optimisation. It involves gathering information from multiple sources including ultrasonic sensors into one platform that will give a clear understanding of the vehicle situation. Synergizing data, the car can properly identify obstructions and their location in real time; thus, improving the probability of crashes to be avoided by this car.

Computational Efficiency: Real-time obstacle avoidance requires considering the computational efficiency of the embedded systems. The objective can be achieved through hardware optimization, various parallel approaches and by advancements in algorithms. Faster performance grants the car the ability to make quick decisions and performs neat manoeuvres even in high traffic intensity like when the car needs to make avoidance movements.

2.2.2 Obstacle avoidance in highly populated metropolitan

Autonomous vehicles to work in wisely- built cities, their vision systems must have strong power to perceive risks such as things under motion and put their cleverness of treating all these things in a dynamic way. Advanced perception algorithms, which include embedded systems with ultrasonic sensors and ability to adapt to change of a situation. Such as, a pedestrian, a cyclist, parked vehicles, construction work sites are continuously changing their positions and combinations. Ultrasonic sensors detect the distances to objects around it by transmitting high-frequency sound waves and then analyzing their reflection. Autonomous car will be able to detect any obstacle on its way in dynamic urban area, and plan its decision of the route, thus, guarantee its safe driving. Figure 1 below shows how autonomous vehicles collects data from sensor and decide.

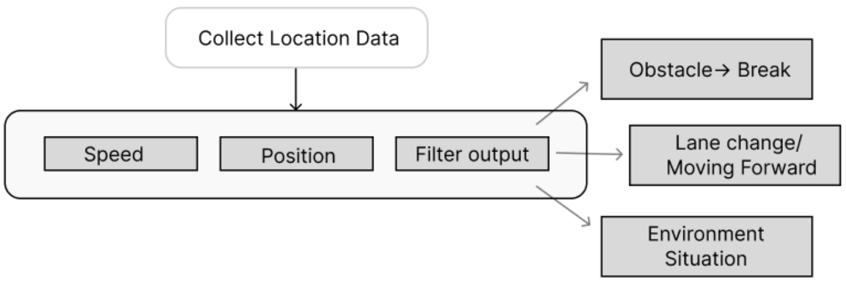


FIGURE 1: Data flow for autonomous vehicle navigation and obstacle avoidance

3 DESIGN AND IMPLEMENTATION

In designing embedded systems for obstacle avoidance and dynamic route navigation in autonomous vehicles, multiple important factors should be taken into consideration to ensure a smooth integration and proper performance in terms of real-world.

The following sections discuss system architecture and design considerations, integration of dynamic route navigation and obstacle avoidance strategy, and selection and implementation of proper and actuator sensor for obstacle avoidance and dynamic route navigation in autonomous vehicles.

3.1 System Architecture and Design Consideration

The efficiency of the system architecture is key to the autonomous vehicles' capabilities in a complicated environment. Real time processing of sensor information, decision-making algorithms, and actuator control should be enabled by the system architecture and design while allowing the system to scale up and down without any data inconsistency. A hierarchical architecture with perception, planning, and control levels can promote efficient processing architecture and decision-making process for obstacle avoidance and dynamic route navigation. To explore deeper into the system design characteristics, investigate the function of real- ROS is a framework which is very flexible and can be used to develop and maintain software modules and other parts of an autonomous system, good for integration and communication between all components (Quigley, Conley, Gerkey, Faust, Foote, Leibs & Ng 2009).

Along with ROS middleware, the architect can design distributed software that divides tasks between different compute nodes and hence, improving the system scalability and reliability. Ultrasonic sensors are exceptionally useful for obstacle detection and proximity sensing in autonomous vehicles, offering affordable and reliable options especially in dynamic conditions.

Using ROS middleware, the developer can design distributed architectures that split computing tasks over numerous processors, increasing system scalability and reprove tolerance. Ultrasonic sensors play an essential role in allowing obstacle identification and proximity sensing in autonomous vehicles, providing dependable and cost-effective options for navigating dynamic situations. An ultrasonic sensor works by emitting high-frequency sound waves, measuring the time it takes for the sound waves to bounce back from an object and the time delay between the emission of the sound wave and the detection of the echo is used to calculate the distance to the object (Dubai Sensor 2023). Figure 2 shows how ultrasonic sensors detect obstacles in the autonomous vehicle. Ultrasonic sensors have limited range and field of view and need to carefully put many sensors around the vehicle to guarantee appropriate coverage of potential barriers in the front, back, and sides (Yusoff 2017). The system must efficiently gather data from the ultrasonic sensors in any conditions. This entails connecting the sensors to the microcontroller and analysing the obtained data to estimate the distance to obstacles. Libraries like Arduino's "NewPing" help make this procedure easier (Arduino NewPing Library).

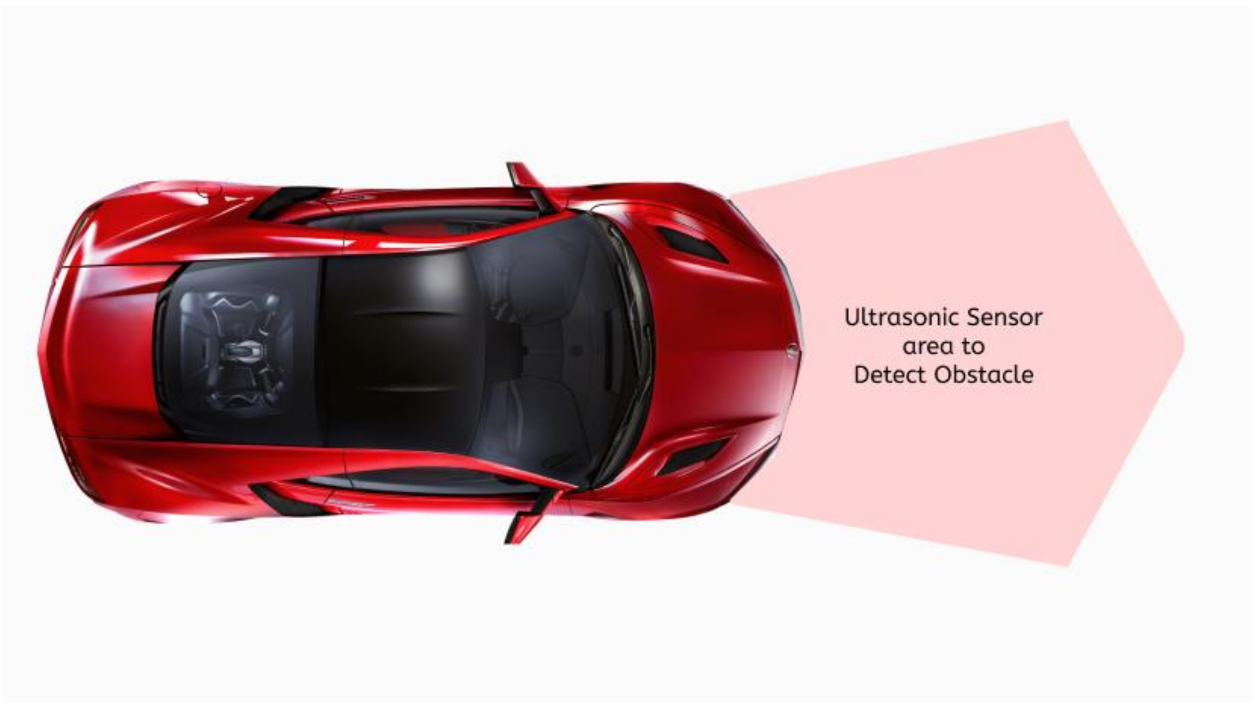
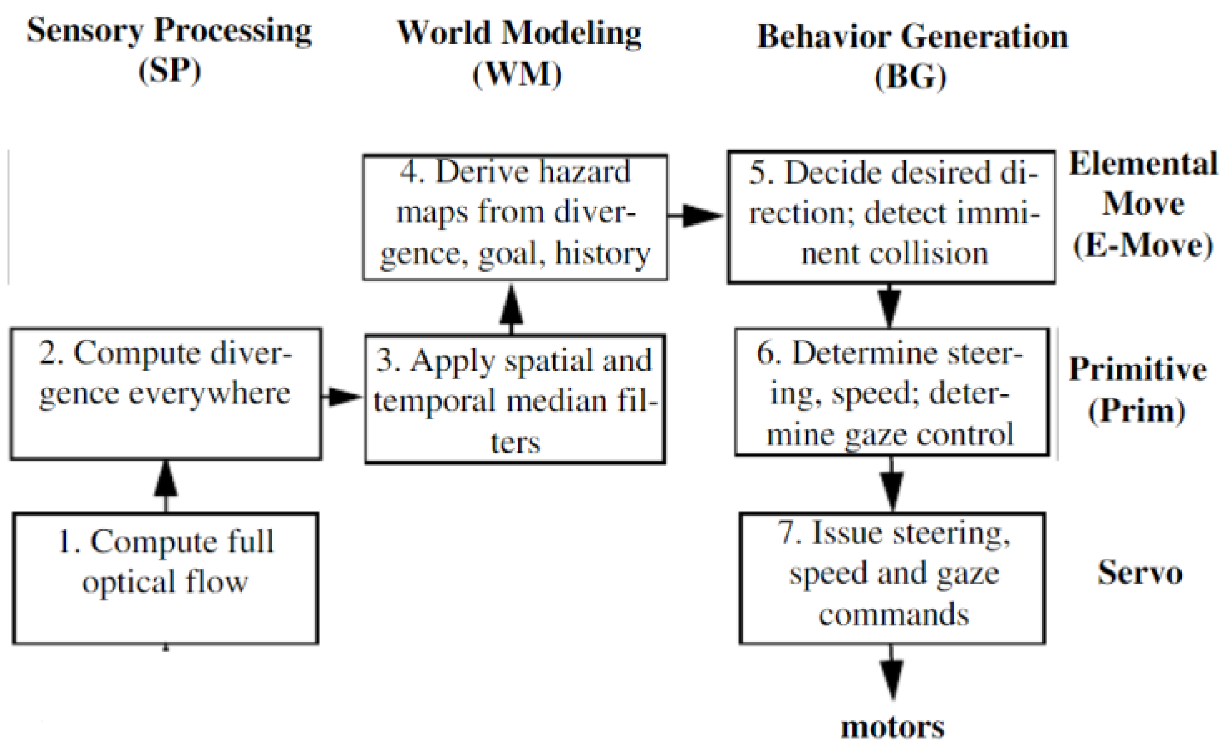


FIGURE 2: Ultrasonic sensor to detect obstacle for avoidance

The decision-making algorithm in autonomous vehicles addresses the difficulty of dynamic route navigation and obstacle avoidance in unpredictable situations. This feature uses multiple techniques to adjust for these uncertainties. Path planning approaches go beyond static routes, taking into detail elements such as moving obstacles and unexpected occurrences (for example, sudden stopping by a car ahead) and allowing the vehicle to modify its path or flow in real time, providing safe navigation in dynamic situations (Wang & Lin 2023). Figure 3 below shows the System Architecture of the obstacle avoidance.



Sensory Processing (SP)
World Modeling (WM)
Behavior Generation (BG)
4. Derive hazard maps from divergence, goal, history
5. Decide desired direction; detect imminent collision
Elemental Move (E-Move)
2. Compute divergence everywhere
3. Apply spatial and temporal median filters
6. Determine steering, speed; determine gaze control
Primitive (Prim)
1. Compute full optical flow
7. Issue steering, speed and gaze commands
Servo
motors

FIGURE 3: System architecture for obstacle avoidance (Camus, Coombs, Herman & Hong 1996)

3.2 Integration of dynamic route navigation and obstacle avoidance strategies

This section focuses on how to integrate dynamic route navigation and obstacle avoidance algorithms so that autonomous vehicles may handle complicated landscapes effortlessly. The design and implementation phases include using Tinkercad simulation for obstacle avoidance with ultrasonic sensors and Flutter-based application for adaptive route changes.

3.2.1 Tinkercad simulation for obstacle avoidance

Tinkercad is an online platform for building and simulating virtual circuits. Virtual circuit can be built on Tinkercad breadboard by placing an Arduino Uno, ultrasonic sensors, and any extra components, then connecting the components using virtual jumper wires according to the chosen circuit diagram (available online for Arduino and ultrasonic sensor connections) and coding the project. Tinkercad offers user-friendly online simulation tools for integrating the interplay of an embedded system and ultrasonic sensors for obstacle avoidance in autonomous cars. Obstacle avoidance system will be developed following the block diagram below using Tinkercad.

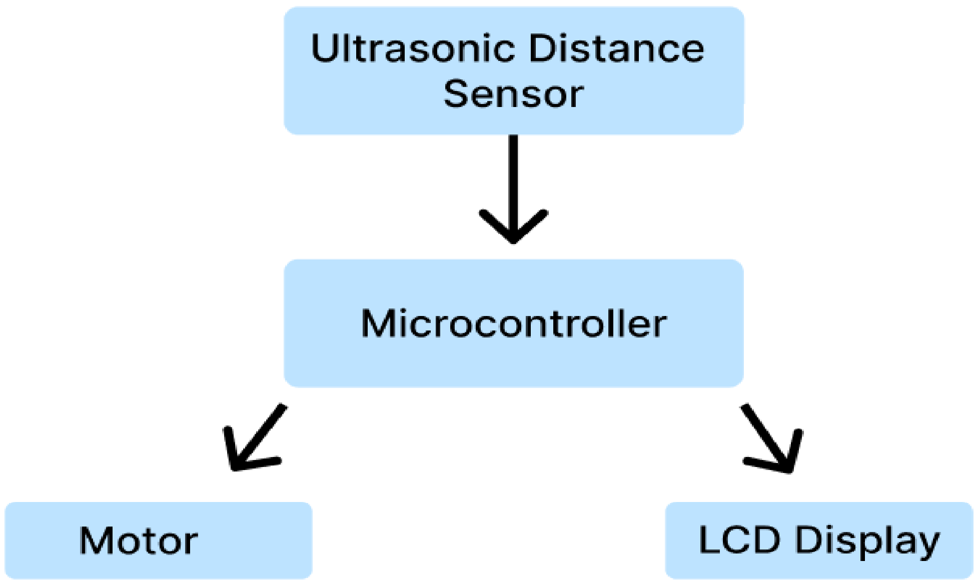


FIGURE 4: Block diagram for obstacle avoidance autonomous vehicle

3.2.2 Flutter-based application for dynamic navigated route

In affiliation with the obstacle avoidance component explored in Tinkercad, a Flutter application can be designed to manage dynamic route navigation for the autonomous vehicle. The application should display a digital map (e. g., utilizing Google Maps API) displaying the vehicle's current location and the planned route to the destination. The feature should communicate with a path planning algorithm (e. g., A*) running on an additional machine or onboard system. This communication could involve, receiving the primary route plan from the path planning algorithm (e. g. A* algorithm) and sending updates about detected obstacles (from the obstacle avoidance system) to the path planning feature. After receiving an updated route command containing obstacle avoidance techniques.

3.2.3 Implementation steps

Implementation of flutter-based application for dynamic route can be done in the following steps:

Mount ultrasonic sensors on the autonomous vehicle to ensure complete coverage of its surroundings and environments and set up communication between the ultrasonic sensors and dynamic route navigation system. Develop an algorithm to analyse sensor data for obstacle detection and use classification methods to differentiate static and moving obstacles in the path. Utilize a Flutter-based application to continuously monitor sensor data to change route. Implement dynamic route planning techniques that use real-time obstacle detection, providing real-time updates on route changes and obstacle avoidance techniques. Maintain coherent transitions between route alterations and obstacle avoidance activities.

3.3 Selection and implementation of suitable and actuator sensor

The effectiveness of an autonomous vehicle to observe its surroundings and respond accordingly determines how efficient its obstacle avoidance and dynamic route navigation are. To achieve highest efficiency, it is required to select appropriate sensors and devices.

3.3.1 Obstacle detection by embedded system

The autonomous cars' primary goal is to detect objects in the proposed path. The autonomous obstacle-avoiding car system with Arduino and Ultrasonic sensor that can autonomously detect objects in its path and navigate accordingly. The system integrates multiple components to achieve its functionality (Sunway International Business School 2023).

Arduino: Arduino is an open-source microcontroller platform that acts as the brain of the autonomous car. It acts as an interface for many modules and sensors, allowing them to efficiently interact and collaborate.

Arduino Motor Shield: The Arduino Motor Shield is an add-on board that allows for the simultaneous control of several motors. It provides precise control over the car's movements, such as speed, direction, and rotation.

Ultrasonic Sensor: Obstacle detection relies heavily on the ultrasonic sensor. By releasing ultrasonic waves and measuring the time it takes for the waves to bounce back, the sensor can measure the distance between the automobile and any obstacle in its path (Sunway International Business School 2023). Ultrasonic sensors are a frequently utilized and affordable option for detecting obstacles at close distances in autonomous vehicles. The ultrasonic sensor worked by generating high-frequency sound waves and measuring the time it takes for the echoes to return. The system can find out the distance to an obstacle by computing the duration of round-trip time and the speed of sound. Ultrasonic sensors are well-suited for detecting immovable objects and as well as rotating objects within a restricted distance, which makes them perfect for activities like parking and navigating in narrow areas.

The method of implementing ultrasonic sensors into system:

Sensor Placement: Ultrasonic sensors should be properly positioned at key locations around the vehicle, specifically the front, back and side. The precise quantity and positioning will vary based on the specific criteria of a project.

Signal processing and decision making: The embedded system must process the reflected ultrasonic signals to find out the distance and direction of obstacles. Typically, this process entails removing irrelevant information and analysing the time-of-flight (Tof) data.

Integration with Obstacle Avoidance Algorithms: The processed data from the ultrasonic sensors is then input into obstacle avoidance algorithms that determine the proper course of action for the vehicle, such as braking, swerving, stopping, or moving.

3.3.2 Dynamic route navigation using Flutter

For dynamic route changes depending on real-time traffic circumstances, a software framework like Flutter can be deployed. Flutter is an open-source framework from Google that facilitates the construction of high-performance mobile applications for many platforms, including Android and iOS (Flutter Documentation). Its features can be absorbed to develop a user-friendly interface that displays real-time traffic data and enables for on board modifications to the dynamic navigation path.

Here is a proposed implementation approach using Flutter:

Real-Time Traffic Data Acquisition: The Flutter application can interact with traffic data APIs or services (e. g., OpenStreetMap) to retrieve real-time traffic statistics (OpenStreetMap). OpenStreetMap is a freely modifiable map of the world, offering open access to transportation data.
Route Visualization and Adjustment: The application may display the intended route on a map and overlay it with real-time traffic data. Users can then dynamically change the path to avoid busy regions.
Communication with Embedded System: The Flutter application can communicate with the embedded system executing the obstacle avoidance algorithms to guarantee the chosen route is feasible and obstacle-free. This connectivity can be performed through many means including Bluetooth or Wi-Fi.

4 ALGORITHM DEVELOPMENT

This part discusses the algorithms for embedded systems. It focuses on dynamic route planning, real time obstacle detection and avoidance and optimization strategies for the effective navigation. As the navigation process uses dynamic route planning and obstacle avoidance appropriately, the development of algorithms for dynamic route planning should be the fundamental part and the starting point when it comes to blending them together. The algorithms incorporated here must be developed to respond to the real-time modification in traffic scenarios and environmental factors. One of the most popular decisions for this task is the A* algorithm because it's known for its capability in navigating. A* algorithm is used alongside flutter apps for finding the shortest and efficient path. The embedded technologies may dynamically change the vehicle's route, ensuring optimal navigation even amongst variable conditions.

A popular choice due to its efficiency is finding the shortest path (MIT Center for Bits and Atoms). An extension of A* algorithm that allows for real-time preplanning as new information becomes available from sensors (Xing, Yu, Liu, Tan, Sun & Li 2023). It is more ideal for dynamic contexts but can be computationally expensive. While A* is efficient, it assumes a static environment where the cost of moving between nodes remains constant. In dynamic environments with obstacles appearing or disappearing, the pre-computed path might become invalid. Re-running the A* algorithms from scratch to find a new path can be computationally expensive, especially for complex environments (Cormen, Leiserson, Rivest, & Stein 2009). Using an optimal search strategy ensures that the path has the lowest cost and optimizes the efficiency of the operation. However, it relies more on heuristic functions. Once the heuristic functions are complex or invalid, it causes poor smoothness and continuity of the pathways, which are not hazardous to the navigation of the vessel (Sang, You, Sun, Zhou, & Liu 2021). The traditional A* algorithm is a heuristic search algorithm that can realize path planning in a global static environment. The evaluation function of the A* algorithm is:

F(n) = G(n) + H(n). (Xiang, Lin, & Ouyang 2022)

Therefore, the evaluation function is improved as follows:

$F(n)$=$G(n)$+$H(n)$+$o(n)$ (Xiang, Lin, & Ouyang 2022)

Where,

$$\begin{cases} o(n) = -\alpha C(n) + \beta \mathrm{I}(n) \\ C(n) = \frac{1}{50}\frac{R-r}{R}H(n) \\ I(n) = \frac{1}{20}\sum_{i=1}^{M}\frac{1}{\sqrt{(x_i - x_s)^2 + (y_i - y_s)^2}} \\ \alpha + \beta = 1 \end{cases}$$

C(n) is the information weight sloped towards the target point and is the information value of the next node around the current node to the target point. The closer the next node is to the destination point, the less the total H(n). r is the distance from the nearby coordinate point of the current position to the destination point. R is the distance from the start point to the goal point (Xiang, Lin, & Ouyang 2022).

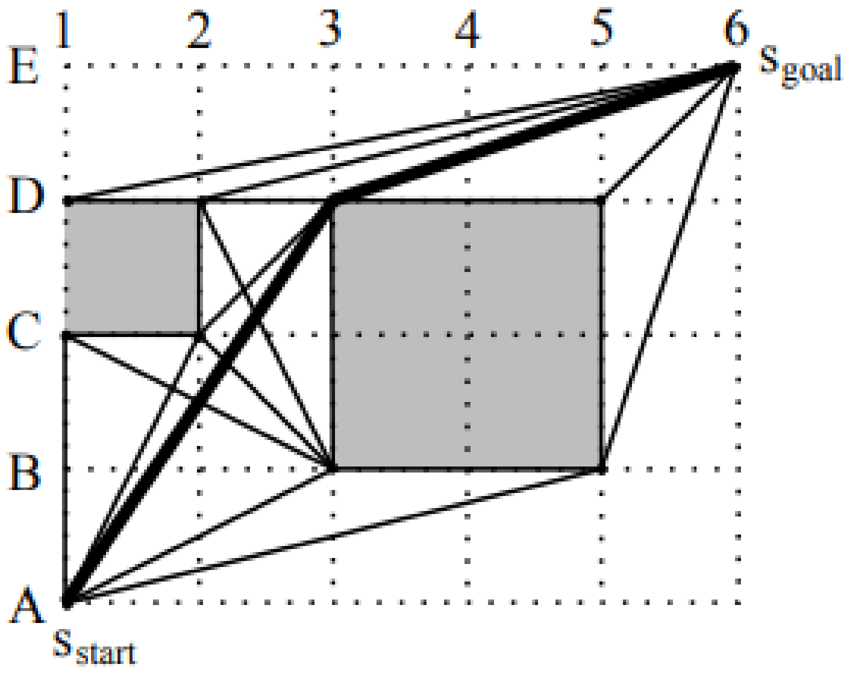


Figure 5: Visibility graph (Sišlák, Volf, & Pěchouček 2009)

The bold solid line is the shortest path; solid lines are edges in the visibility graph in the above picture.

5 HARDWARE IMPLEMENTATION AND SOFTWARE DEVELOPMENT

The foundation of any autonomous vehicle lies in its embedded system, the brains responsible for real-time decision making and control. This section dives into the core of this research, Here the crucial hardware and software components that enable dynamic route navigation and obstacle avoidance is discussed thoroughly.

5.1 Hardware implementation

The hardware foundation of the system can be contained with two main components:

Microcontroller: This programmable device functions as the brain of the system, gathering sensor data, processing it, and regulating actuators (e. g., motors, steering) depending on programmed algorithms. Popular choices for autonomous vehicle applications include:

Arduino Uno: A beginner-friendly microcontroller board suited for prototyping due to its ease of use and extensive database of resources (Arduino).

Raspberry Pi: A more capable single-board computer giving better processing capability for complex algorithms (Raspberry Pi).

Specialized Automotive Microcontrollers: For real-world applications, it may be beneficial to consider industry-standard microcontrollers designed specifically for the severe automotive environment and safety requirements. These often demand in-depth expertise in embedded systems programming.

Sensors: Sensors operate as the eyes and ears of the system, gathering data and information of the surrounding environment. As described above, these sensors emit high-frequency sound waves and detect the echo's return time to determine the distance to an obstacle. They are a good starting point because of their affordability and ease of usage. Utrasonic sensors use sonic transducers to emit sonic waves in the 40–70 kHz range. This frequency range is beyond the human audible range, making it safe for human ears (Ignatious, El-Sayed, & Khan 2022).

5.1.1 Hardware implementation with Tinkercad

Tinkercad is a free online web-based tool that allows to create and simulate circuits electronically. It provides a user-friendly interface with virtual components, making it excellent for prototyping and testing with hardware setups. In this project, Tinkercad has been used to simulate obstacle avoidance sytem. The table below shows the list of components used for simulating obstacle avoidance prototype.

	Component	**Count**
1.	Arduino Uno Microcontroller R3	1
2.	Ultrasonic Distance Sensor	1
3.	Breadboard	1
4.	LCD 16*2	1
5.	DC Motor	4
6.	9V Battery	1
7	Potentiometer	1
8.	Connecting Wires	~

TABLE 1: List of components

In Figure 6 below, the simulation of an obstacle avoidance vehicle by using Tinkercad has been exhibited. An Arduino Uno R3 microcontroller, ultrasonic distance sensor, breadboard, potentiometer, LCD 16*2, DC motor, 9V battery and connected wires have been used. The Arduino Uno transmits a short electrical pulse to the ultrasonic distance sensor's trigger pin. This activates the sensor's transmitter, which generates a burst of high-frequency sound waves. When the sound wave contacts an object within the sensor's range, it reflects towards the sensor. The receiver on the ultrasonic distance sensor catches this returning echo. The Arduino receives a signal back from the sensor's echo pin after the echo is detected. By analysing the time difference between the trigger pulse delivered and the echo received, the Arduino can compute the distance to the obstacle using the formula: distance = (speed of sound * time difference) / 2. (The speed of sound is a constant value). Based on the determined distance, the Arduino program generates an output signal. On Tinkercad, virtual component like LCD display is employed. For example, the LCD display shows the output result. When the obstruction gets closer the autonomous car will be stopped, and if the car discovers the space of its right or left side then the car will move forward with that output, otherwise the car will be waiting for a moving obstacle. If the obstruction moves from the automobile area, again the car will be going forward.

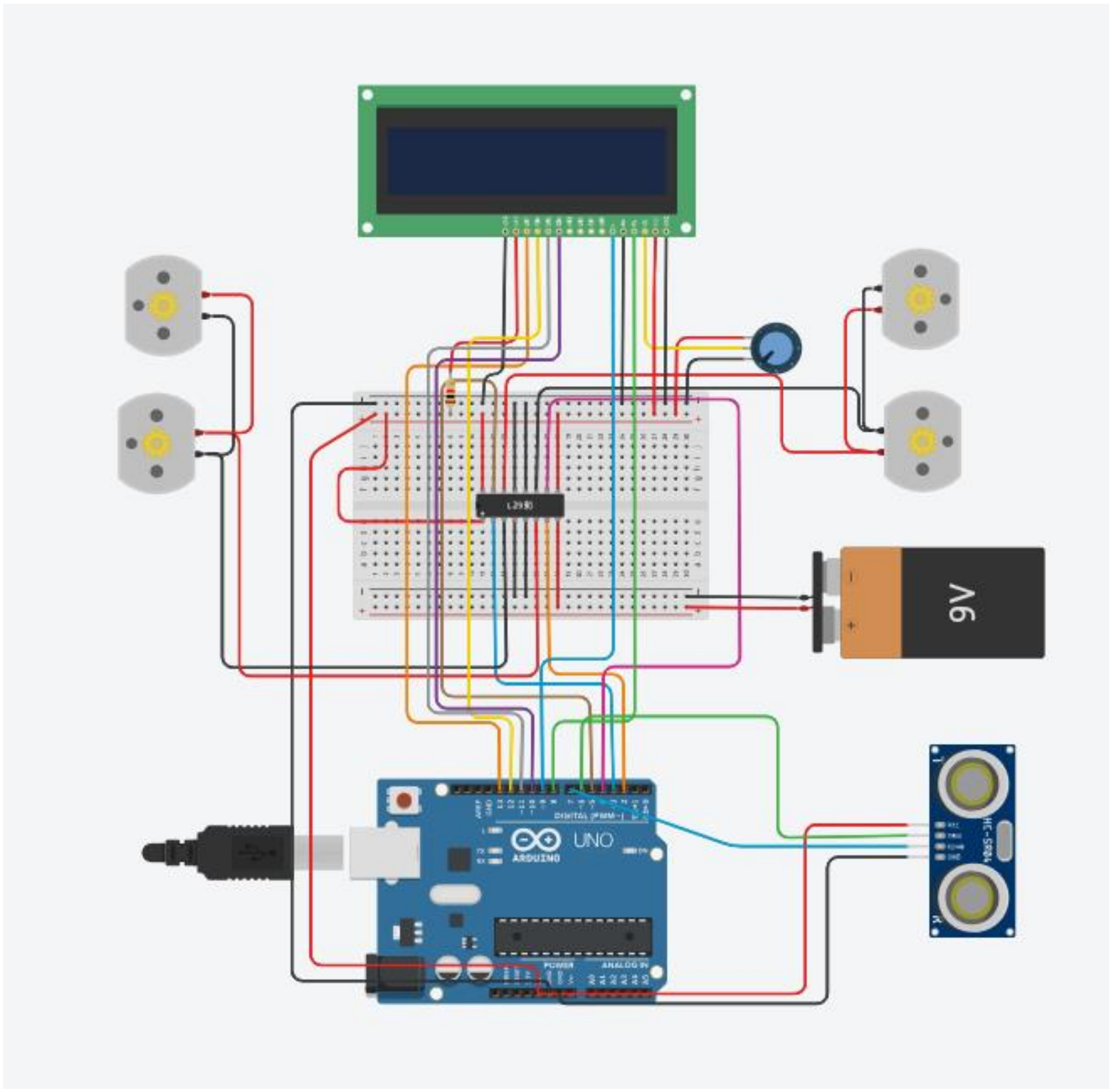


FIGURE 6: Simulation of autonomous vehicle for obstacle avoidance system

Figures 7 and 8 below show the circuit diagram and schematic view of obstacle avoidance simulated by Tinkercad.

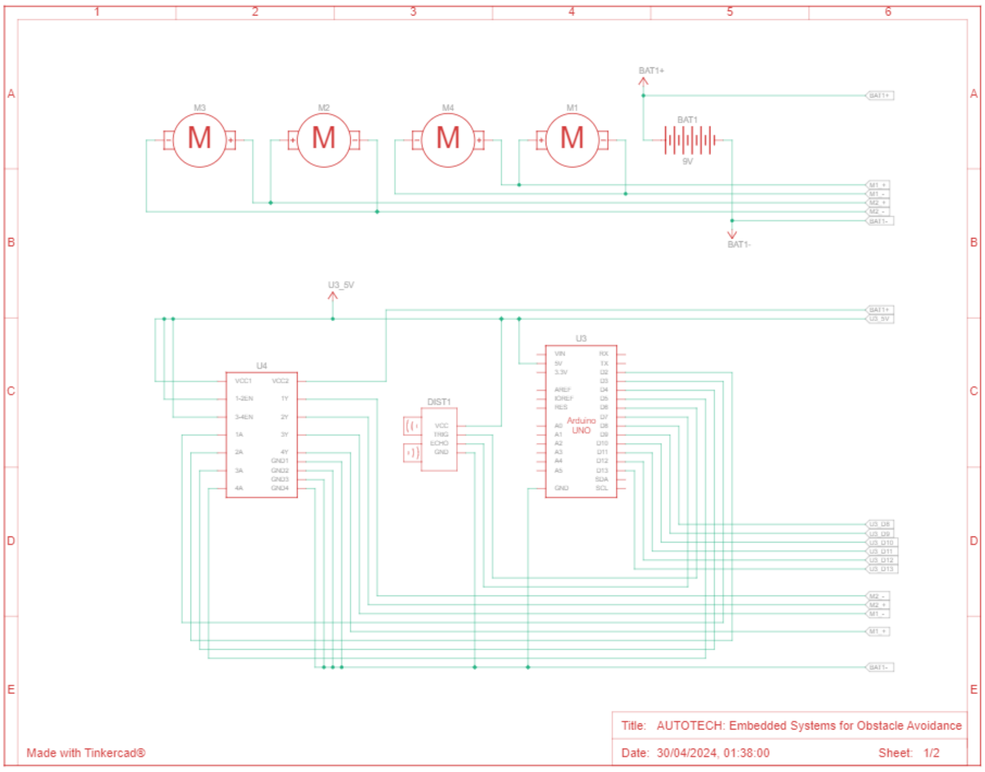

M3
M2
M4
M1
BAT1+
BAT1
9V
BAT1-
U3_5V
U4
DIST1
U3
Arduino UNO
Title: AUTOTECH: Embedded Systems for Obstacle Avoidance
Date: 30/04/2024, 01:38:00
Sheet: 1/2
Made with Tinkercad®

FIGURE 7: Circuit diagram of obstacle avoidance

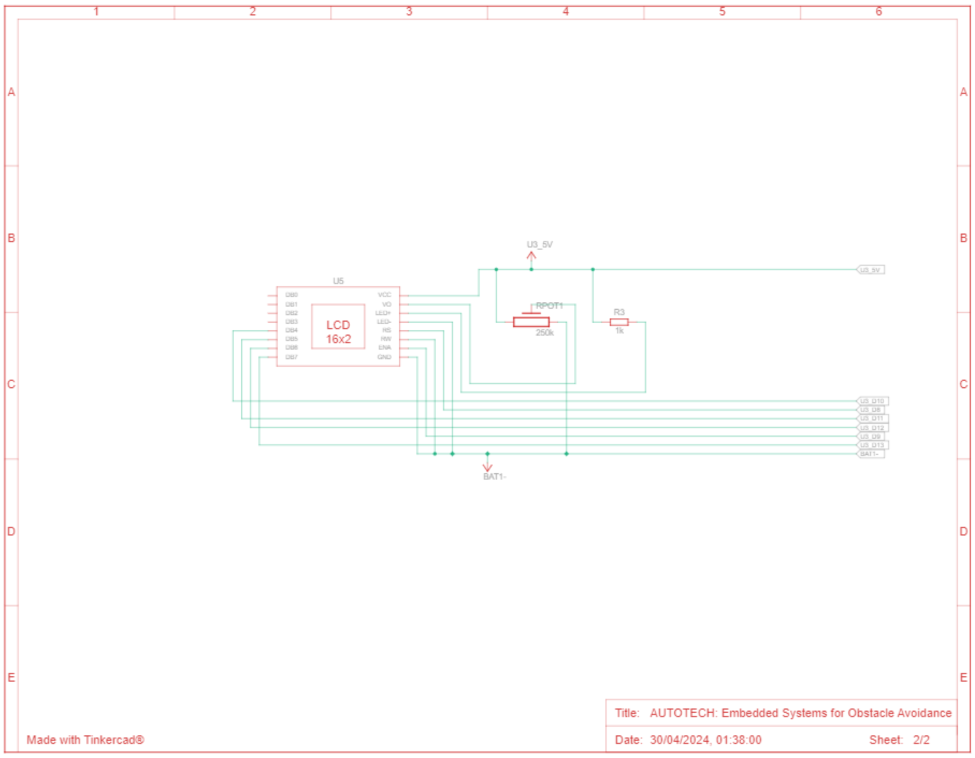

U5
LCD 16x2
U3_5V
RPOT1
250k
R3
1k
BAT1-
Title: AUTOTECH: Embedded Systems for Obstacle Avoidance
Date: 30/04/2024, 01:38:00
Sheet: 2/2
Made with Tinkercad®

FIGURE 8: Schematic view of obstacle avoidance

5.2 Software development

The software component controls the way the microcontroller engages with the sensors and actuators. The software component has a crucial function in converting sensor data from hardware setup into functional choices for the autonomous vehicle. The software development foundation for this system can be contained in two key components.

5.2.1 Software development for obstacle avoidance and adjusting path

The obstacle avoidance software method is coded in C++ utilizing the Arduino programming environment. The method utilizes a responsive control strategy, in which the vehicle modifies its path according to the closeness of obstacles identified by the ultrasonic sensors. The implementation of this method in the given C++ code within the Arduino environment ensures that the vehicle can adjust its route in real-time, taking into factor the distance of objects detected by ultrasonic sensors. The autonomous car can drive through complex situations and prevent crashes by constantly analysing its surroundings and adjusting its path according to traffic and environment conditions. Autonomous cars face distinct difficulties when navigating densely populated metropolitan areas, as they must deal with constantly moving objects such as pedestrians, bicycles, bikes, parked vehicles, and construction zones. This paper came up with a programmed solution in an Arduino project using C++ language, which can quickly detect objects within a specific sensor area and change the lanes accordingly. In dynamic critical conditions, the implementation of a responsive control technique is vital for autonomous vehicles. This technology allows the vehicle to quickly make decisions and navigate safely through areas with heavy traffic. The autonomous vehicle navigates in the complex metropolitan areas while avoiding collisions with fixed and moving objects because it continuously modifies its path through processing of sensor data and the roadway traffic situations. This technology assures the safety of pedestrians, cyclists, and other road users while preserving efficient traffic flow in metropolitan areas. So, it can be said that this approach ensures greater efficiency in autonomous driving experiences and increases real time flexibility by enhancing vehicle abilities to quickly adjust to the shifting traffic conditions and environment factors. The figure below shows the data flowchart of the system, which has been followed to develop the system to handle obstacle avoidance and navigating route path in complex situations.

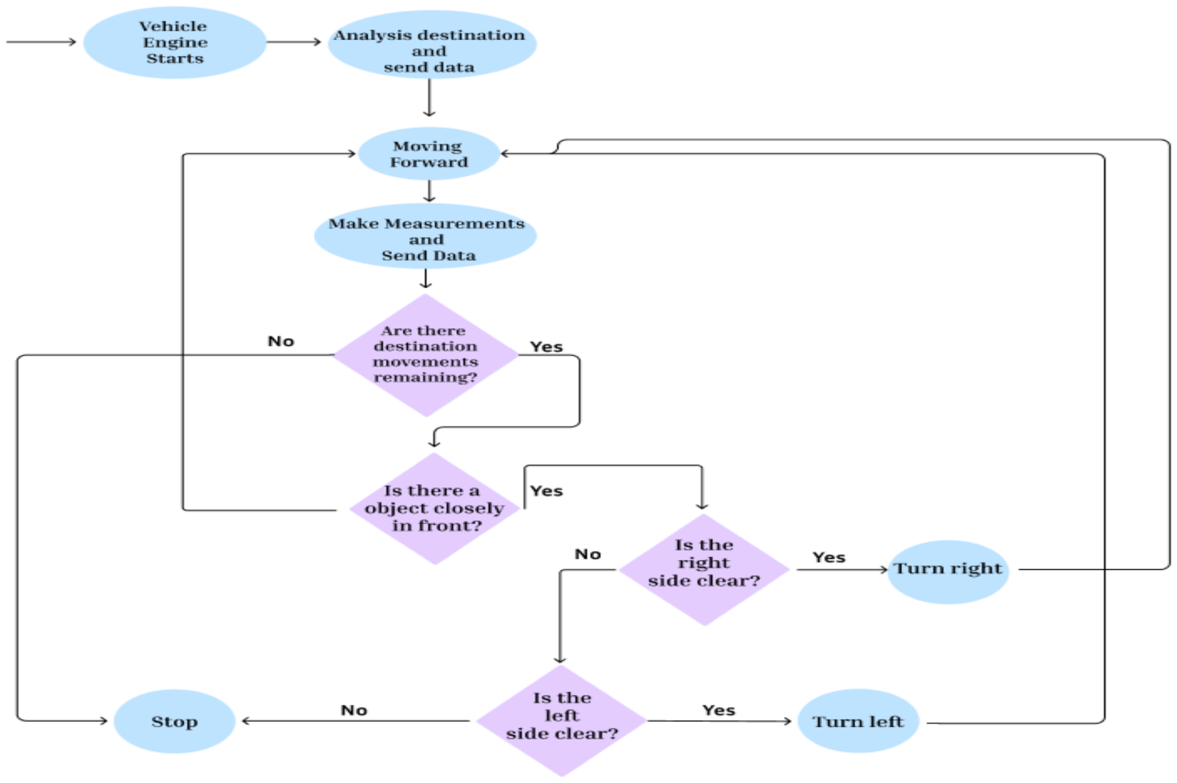

Vehicle Engine Starts
Analysis destination and send data
Moving Forward
Make Measurements and Send Data
Are there destination movements remaining?
No
Yes
Is there a object closely in front?
Yes
Is the right side clear?
No
Yes
Turn right
Is the left side clear?
No
Yes
Stop
Turn left

FIGURE 9: Data flow chart of obstacle avoidance and navigation path

5.2.2 Software development for Dynamic Route Navigation

This part of the thesis demonstrates the potential of integrating a dynamic route navigation system for identifying the efficient and shortest path by utilizing Flutter application along with A* algorithm. By developing these technologies, the proposed system offers most efficient routes by utilizing roadway conditions. A* algorithm can use in unstructured environment where obstacles are relatively cluttered and vehicle speeds are low and to avoid collision of vehicle contours, redundant spaces are set up based on vehicle size when the map is rasterized (Xiong, Min, Yu, & Wang 2020).

Flutter app's UI is designed to provide users with intuitive controls for inputting their destination and viewing the proposed shortest paths. The map display features interactive elements, allowing users to zoom in/out and pan across the map to discover different routes in that situation. When a destination is inputted from the user, the app triggers the A* algorithm to calculate the shortest route, taking into consideration real-time traffic conditions. Here A* algorithm library (Open-source resource) is used from Pub.dev (Pub.dev is official package repository for Flutter). The determined route is then displayed to the user, along with expected trip time and distance, Integrate the collected traffic data with the A* algorithm. By factoring in traffic congestion statistics, the algorithm may dynamically change the intended route to avoid traffics and minimize travel time. User can choose their suitable location and adjust the route by road condition.

FIGURE 10: Navigation System for Dynamic Route Change

6 RESULT AND DISCUSSION

Through the integration of ultrasonic sensors and the deployment of the flutter apps along with A* algorithm for path navigation, the embedded systems demonstrated a considerable enhancement in dynamic route navigation and obstacle avoidance capabilities. The system accurately recognized obstacles in real-time and computed the shortest way to go around them, ensuring seamless movement even in challenging environments.

In highly populated metropolitan areas characterized by diverse obstacles such as pedestrians, cyclists, parked cars, and construction zones, the embedded systems proved to be effective. The combination of obstacle identification with ultrasonic sensors allows autonomous vehicles to navigate heavily crowded areas with precision and caution. This enabled autonomous vehicles to dynamically adjust their routes, optimizing travel time and ensuring safe passage through congested or evolving traffic scenarios in the street. Figure 11-14 demonstrate the embedded system of obstacle avoidance by Tinkercad. On the Tinkercad simulation, it is seen that the autonomous vehicle prototype works properly, when the ultrasonic sensor identifies any object, the vehicle stops moving ahead and then the vehicle checks its right and left to turn the car if the obstacle is not moving. This approach is done by C++ then integrated with hardware simulation.

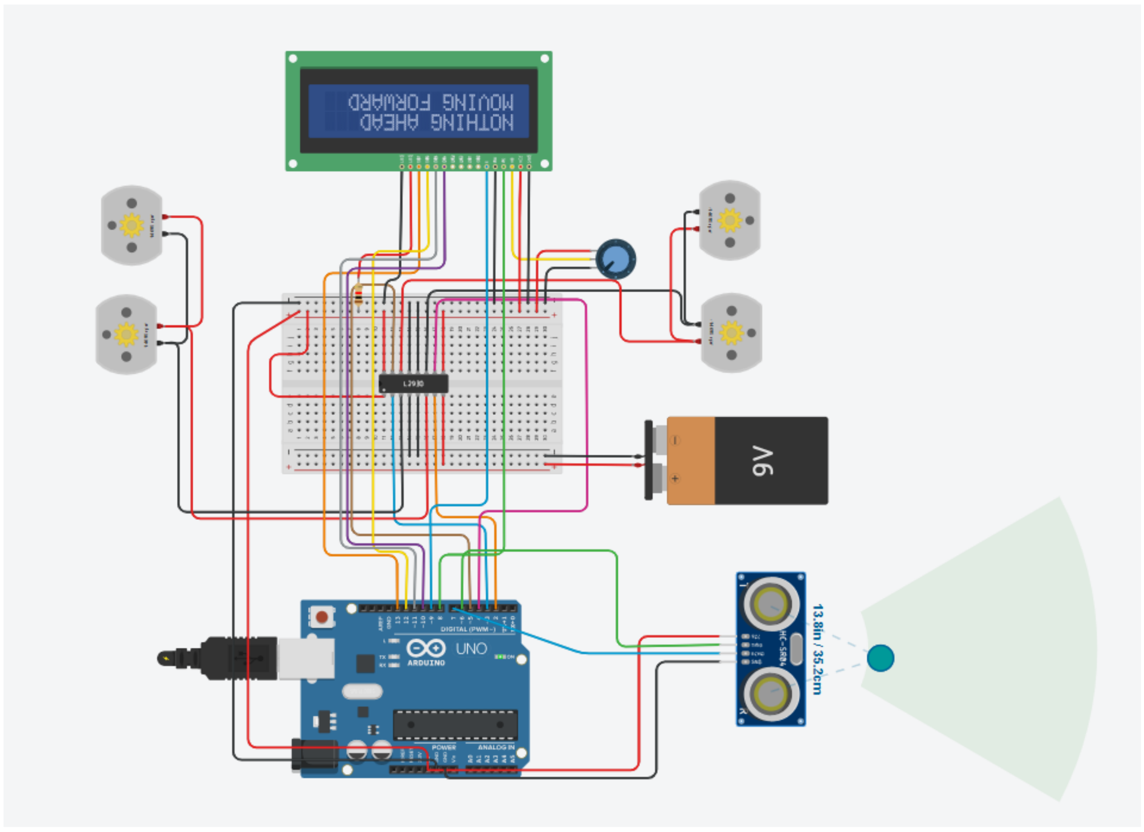


FIGURE 11: Simulation result of autonomous vehicle obstacle avoidance (1/4)

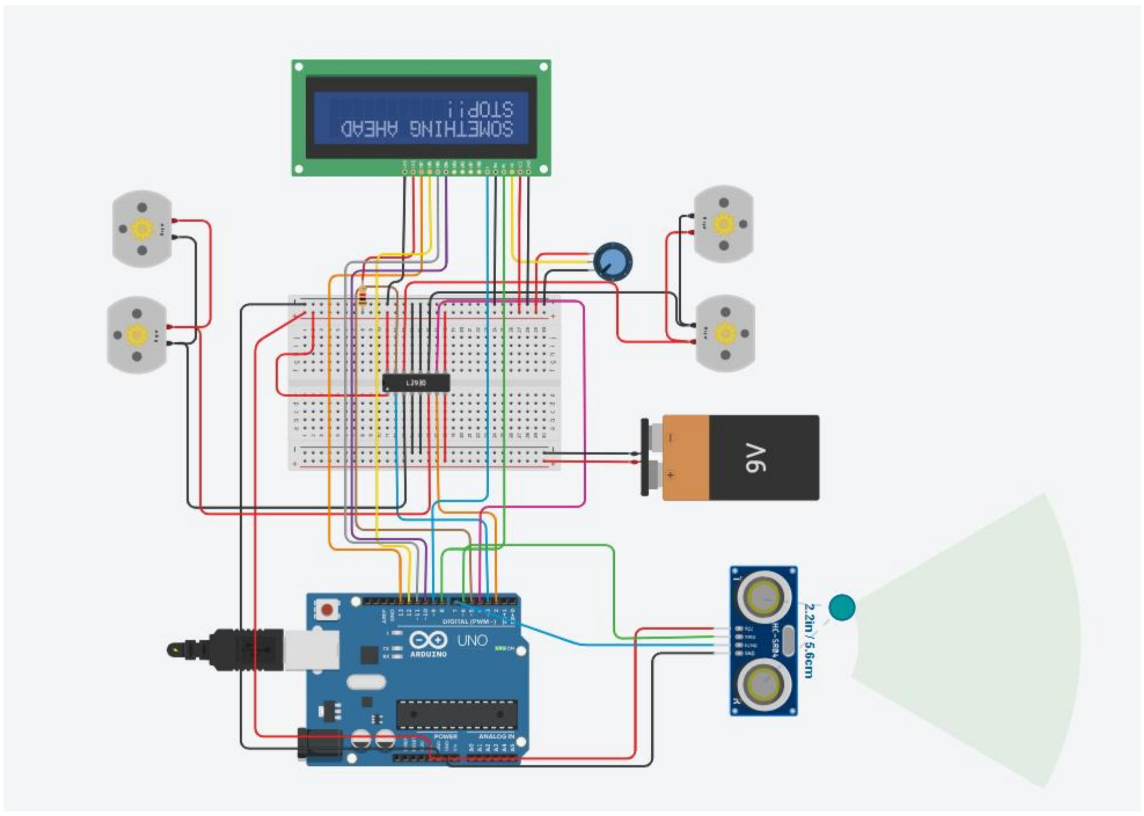


FIGURE 12: Simulation result of autonomous vehicle obstacle avoidance (2/4)

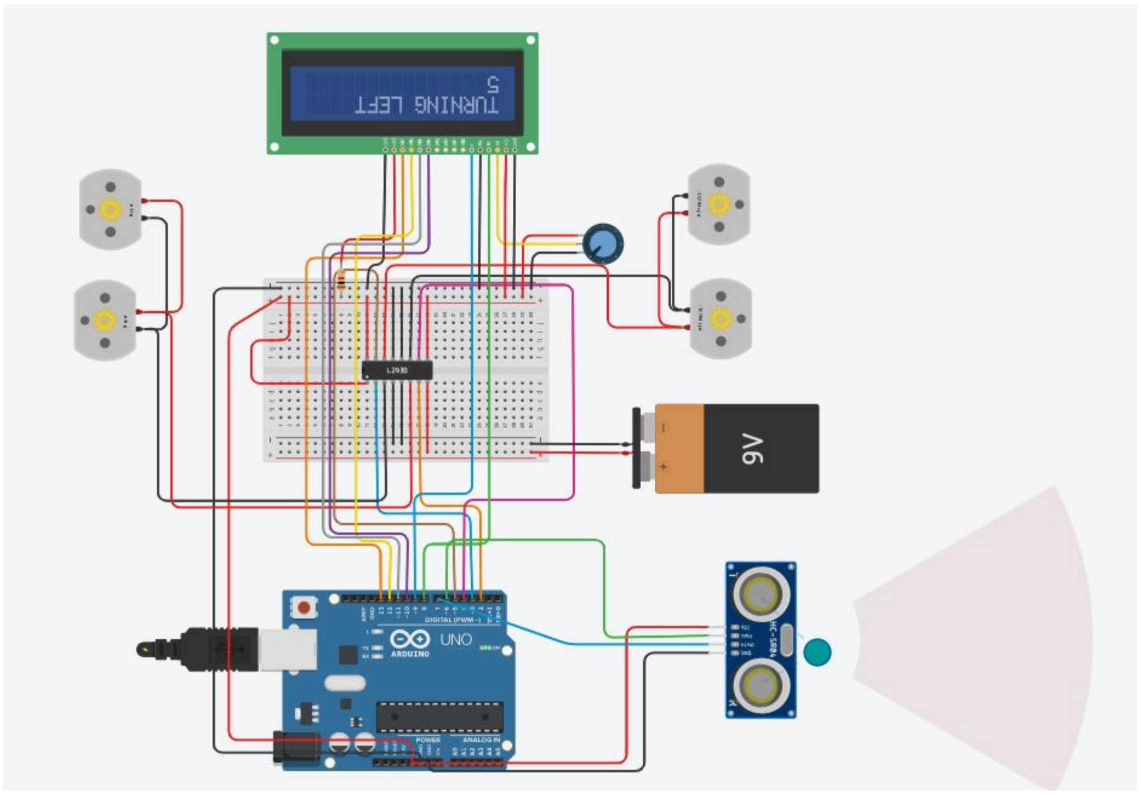


FIGURE 13: Simulation result of autonomous vehicle obstacle avoidance (3/4)

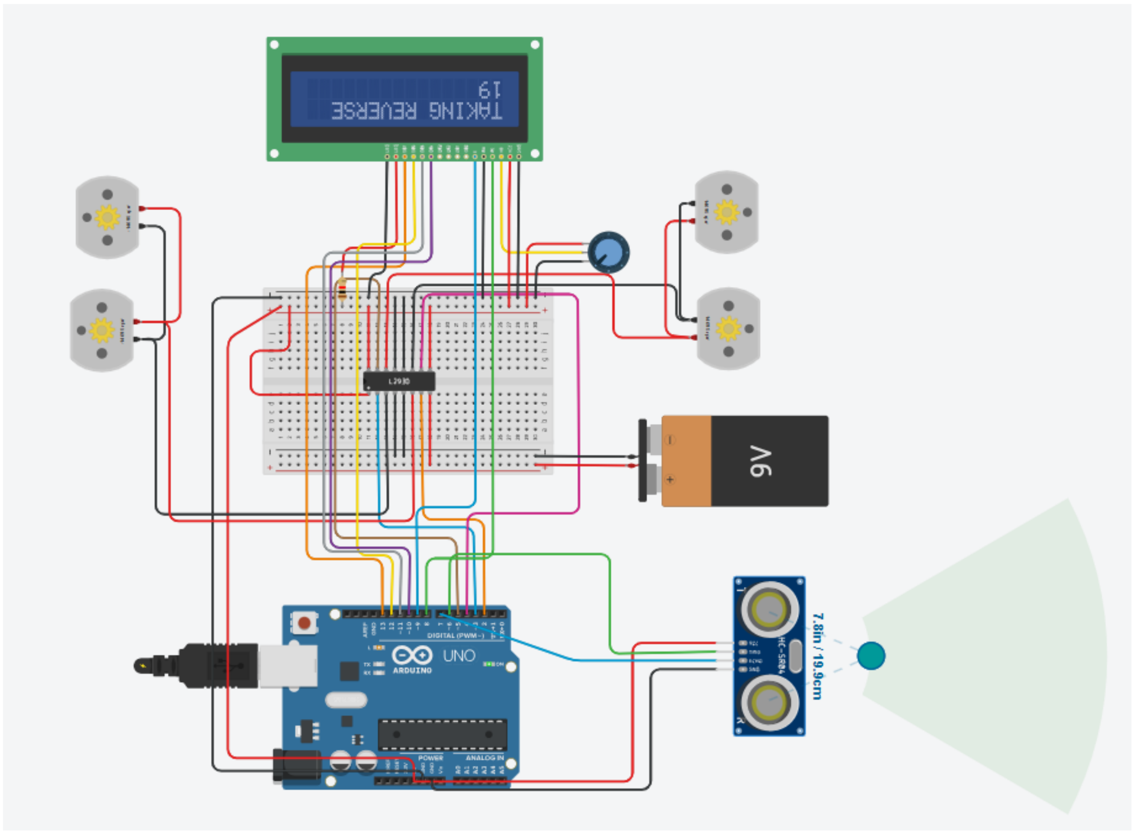


FIGURE 14: Simulation result of autonomous vehicle obstacle avoidance (4/4)

The integration of Flutter apps provided an intuitive interface for users to interact and alongside A* algorithm provided the shortest and efficient path for autonomous vehicles. Through the app, users could input destinations, receive real-time updates on travel progress, and even override autonomous navigation in certain situations. This seamless interaction between the embedded systems and user interfaces contributed to a holistic autonomous driving experience, promoting trust and acceptance among passengers and pedestrians alike. From Figure 10, it is seen that passengers can input the destination and select the suitable path according to their will.

The successful execution of embedded systems combining dynamic route navigation and obstacle avoidance represents a significant development in autonomous vehicle technology. By integrating accurate sensors and algorithms, these systems offer higher levels of security, efficiency, and flexibility in navigating challenging circumstances, thereby pushing this era closer to the widespread use of autonomous transportation.

7 FUTURE WORK

This research displays the perspective of embedded systems for seamless route navigation and obstacle avoidance in autonomous vehicles. The simulation using Tinkercad to avoid obstacle and for dynamic route navigation using flutter apps alongside the A* algorithm gives an environment for further development. But the actual application can be built in the future as discussed using flutter applications and A* algorithm. Along with this, working with machine learning techniques to enable autonomous vehicles to learn from past experiences and make informed decisions in complex urban environments and the integration of LiDAR, cameras, and radar into autonomous vehicle systems to increase their perception abilities can be planned. By utilizing these advanced sensor technologies, autonomous cars can obtain a more thorough understanding of their surroundings, hence enhancing object detection and tracking under varied conditions.

To summarize, the future work indicated above presents a multidimensional strategy to increasing autonomous vehicle technology, with a focus on strengthening vision capabilities, building intelligent navigation apps along with algorithms, and improving human-machine interactions. Addressing these research challenges, we can pave the way for mainstream use of autonomous vehicles and realize their full potential to impact mobility in the world.

8 CONCLUSION

The integration of embedded systems technology into autonomous vehicles, focusing on dynamic route navigation and obstacle avoidance are reviewed in this paper. This project revealed two essential aspects: upgrading embedded systems for seamless integration of dynamic route navigation and obstacle avoidance and solving the obstacles of navigating through heavily populated metropolitan areas (e. g. people running, cycling, parked cars, and construction zones).

The research highlighted the importance of obstacle avoidance strategies in navigating through heavily metropolitan areas. Use of ultrasonic sensors to implement an obstacle avoidance system for real time application in autonomous vehicles. By integrating these sensors into the embedded systems of autonomous vehicles, vehicles are empowered with the capability of detecting objects and reacting to dynamic changes in their environment. This innovative approach is essential for ensuring the safety of pedestrians, cyclists, and other vehicles in busy metropolitan areas where risks do not come once but keep knocking at every moment.

Also, dynamic route navigation system with Flutter apps and A* algorithm has been discussed widely. The users can easily navigate through the flutter application for seamless route navigation interaction with autonomous vehicles, whereas the A* algorithm does the job of quickly calculating optimal routes on short notice due to traffic and changing surrounding. This synergy not only increases autonomous vehicles' flexibility but also has a positive impact towards making transporting a safer and more efficient solution.

In summary, the introduction of embedded systems technology into the flutter application eliminating the limitations of the previous systems presents an opportunity to develop the autonomous vehicle features particularly in the fields of obstacle avoidance and dynamic route finding. Having these powerful algorithms and sensor technology, we can permit autonomous cars to move around better even in congested and complex pathways leading to the future in which systems of transportation will be more secured, sustainable.

REFERENCES

Arduino. Available at:
https://www.arduino.cc/

Arduino NewPing Library**.** Available at:
https://www.arduino.cc/reference/en/libraries/newping/

Camus, T., Coombs, D., Herman, M. & Hong, T. 1996. Real-time single-workstation obstacle avoidance using only wide-field flow divergence. *In Proceedings of the 1996 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* Available at:
https://ieeexplore.ieee.org/document/546964

Cormen, H., Leiserson, E., Rivest, R. & Stein, C. 2009. Introduction to Algorithms (3rd ed.). The MIT Press. Available at:
https://cdn.manesht.ir/19908___Introduction%20to%20Algorithms.pdf

Dang, T. & Bui, N. 2023. Obstacle Avoidance Strategy for Mobile Robot Based on Monocular Camera. *Electronics*. Available at:
https://doi.org/10.3390/electronics12081932

Dubai Sensor. 2023. Ultrasonic Sensors: Harnessing the Power of Sound Waves for Accurate Detection and Measurement. Available at:
https://www.dubai-sensor.com/blog/ultrasonic-sensors-harnessing-the-power-of-sound-waves-for-accurate-detection-and-measurement/

Flutter Documentation. Available at:
https://docs.flutter.dev

Ignatious, A., El-Sayed, H. & Khan, A. 2022. Sensor Technology for Autonomous Vehicles. *In Advances in Smart Technologies.* Available at:
https://www.sciencedirect.com/science/article/abs/pii/B9780128225486001229

Islam, F., Narayanan, V. & Likhachev, M. 2016. A*-Connect: Bounded suboptimal bidirectional heuristic search. *In Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA)*. Available at:
https://ieeexplore.ieee.org/abstract/document/7487437

MIT Center for Bits and Atoms. Robot Path Planning. Available at:
https://fab.cba.mit.edu/classes/865.21/topics/path_planning/robotic.html.

OpenStreetMap. Available at:
https://www.openstreetmap.org/

Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J. & Ng, A. Y. 2009. ROS: an open-source Robot Operating System. *ICRA Workshop on Open Source Software*. Available at:
http://lars.mec.ua.pt/public/LAR%20Projects/BinPicking/2016_RodrigoSalgueiro/LIB/ROS/icraoss09-ROS. Pdf

Raspberry Pi. Available at:
https://www.raspberrypi.com/

Russell, S. 2019. *Artificial intelligence: A modern approach* (Third Edition). Pearson Education Limited. Available at:
https://thuvienso.hoasen.edu.vn/handle/123456789/8967

Sang, Q., You, S., Sun, J., Zhou, Y. & Liu, F. 2021. The hybrid path planning algorithm based on improved A* and artificial potential field for unmanned surface vehicle formations. *Ocean Eng*. Available at:
https://www.sciencedirect.com/science/article/abs/pii/S002980182100144X

Sišlák, D., Volf, P. & Pěchouček, M. 2009. Accelerated A* Trajectory Planning: Grid-based Path Planning Comparison. Available at:
https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=77f1306116d27fdce4b911d6b9451e6c37f81812

Sunway International Business School. 2023. Automatic Obstacle Avoiding Car with Arduino and Ultrasonic. Available at:
https://sunway.edu.np/automatic-obstacle-avoiding-car-with-arduino-and-ultrasonic/

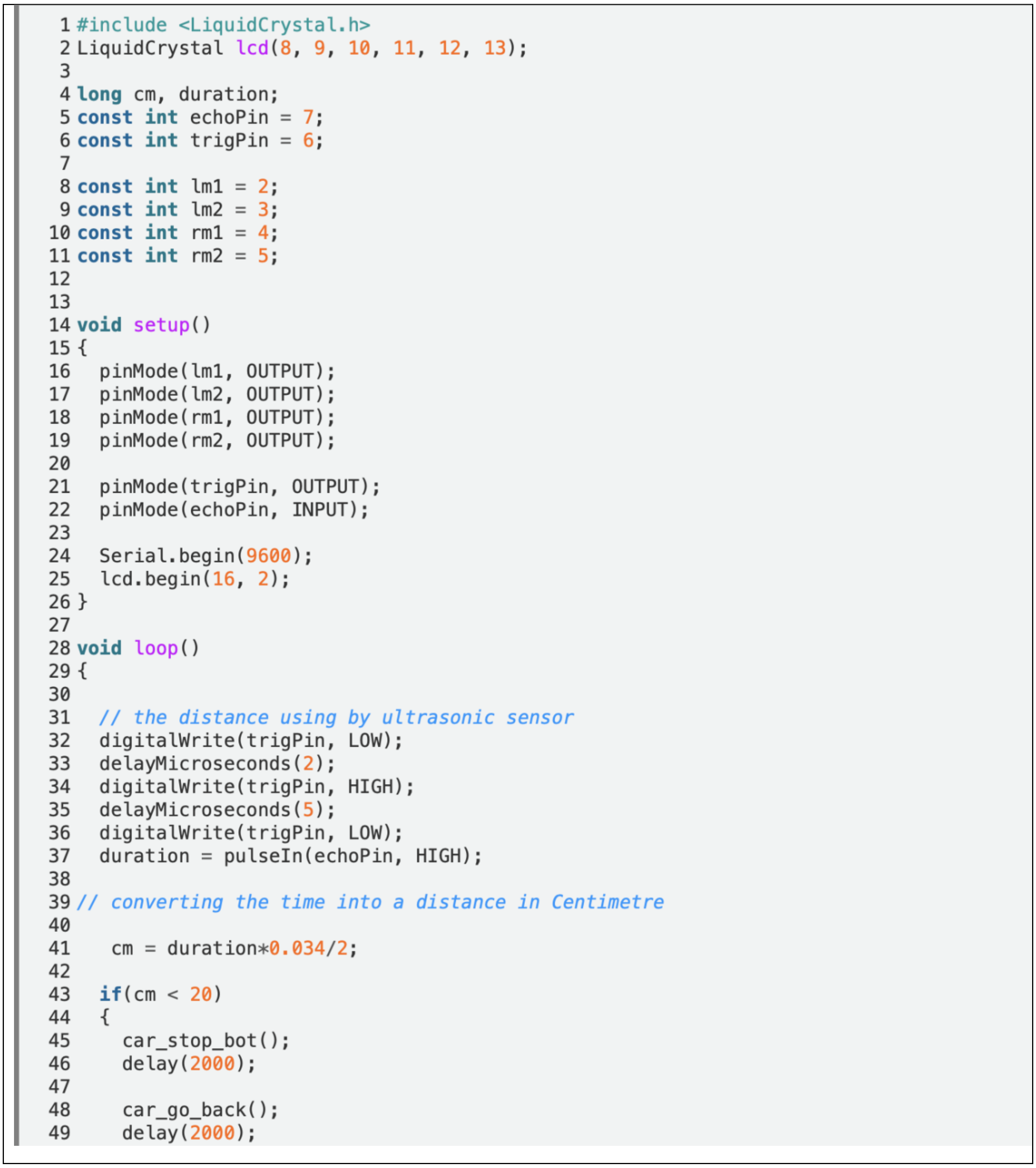
Wang, Y. & Lin, Z. 2023. Research on path planning for autonomous vehicle based on Frenet system. *Sustainable Computing: Informatics and Systems*. Available at:
https://www.sciencedirect.com/science/article/pii/S2307187723000810

Xiang, D., Lin, H. & Ouyang. 2022. Combined improved A* and greedy algorithm for path planning of multi-objective mobile robot. *Scientific Reports*. Available at:
https://doi.org/10.1038/s41598-022-17684-0

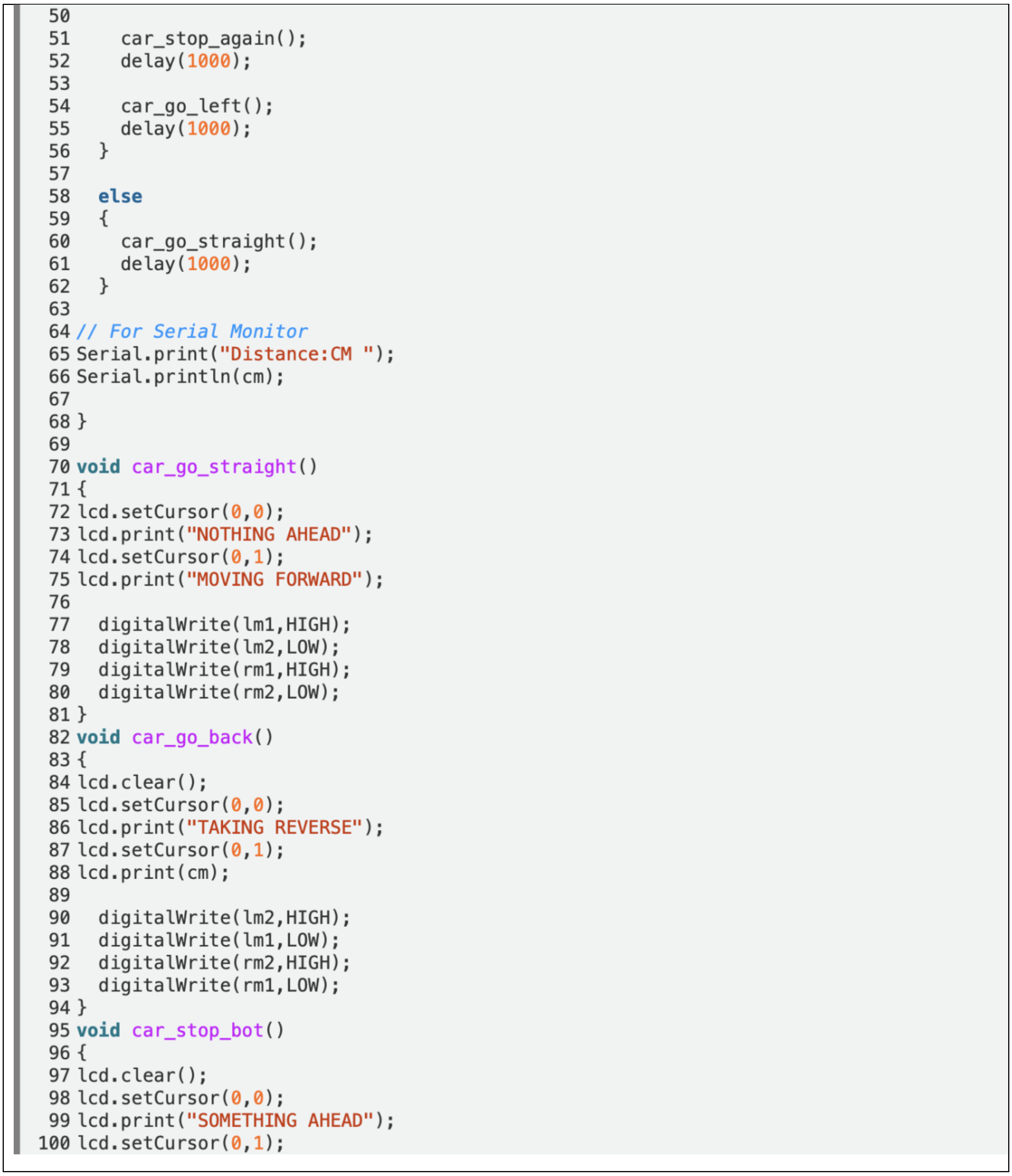
Xing, B., Yu, M., Liu, Z., Tan, Y., Sun, Y. & Li, B. 2023. A Review of Path Planning for Unmanned Surface Vehicles. Available at:
https://www.mdpi.com/2077-1312/11/8/1556

Xiong, X., Min, H., Yu, Y. & Wang, P. 2020. Application improvement of A* algorithm in intelligent vehicle trajectory planning. *Mathematical Biosciences and Engineering*. Available at:
https://www.aimspress.com/aimspress-data/mbe/2021/1/PDF/mbe-18-01-001.pdf

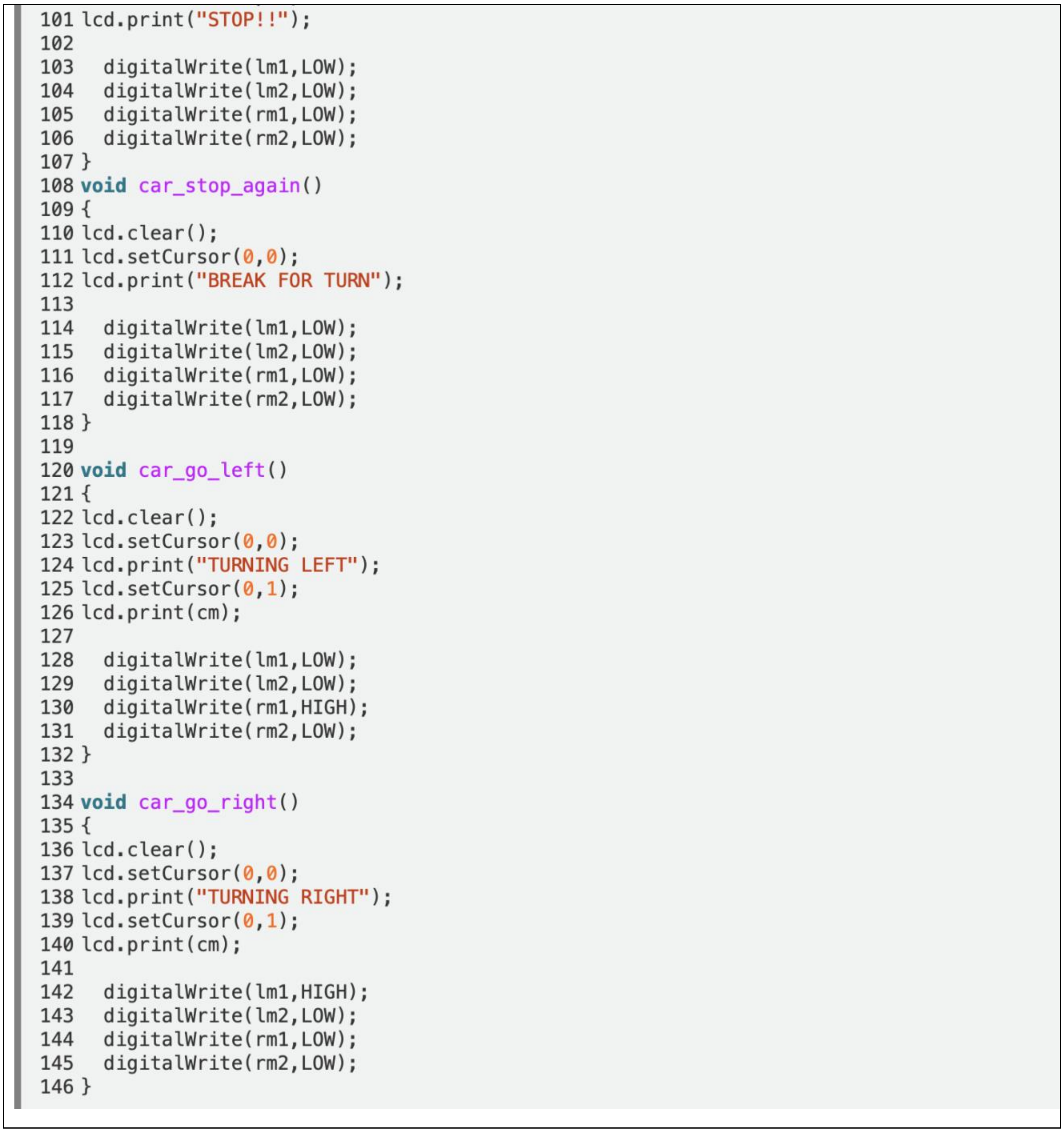
Sánchez, F., González, J.A., and Moreno, J. (2018). Autonomous robot for mapping using ultrasonic sensors. Available at:
https://www.researchgate.net/publication/323056345_Autonomous_robot_for_mapping_using_ultrasonic_sensors



```
#include <LiquidCrystal.h>
LiquidCrystal lcd(8, 9, 10, 11, 12, 13);

long cm, duration;
const int echoPin = 7;
const int trigPin = 6;

const int lm1 = 2;
const int lm2 = 3;
const int rm1 = 4;
const int rm2 = 5;


void setup()
{
  pinMode(lm1, OUTPUT);
  pinMode(lm2, OUTPUT);
  pinMode(rm1, OUTPUT);
  pinMode(rm2, OUTPUT);

  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);

  Serial.begin(9600);
  lcd.begin(16, 2);
}

void loop()
{

  // the distance using by ultrasonic sensor
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(5);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);

// converting the time into a distance in Centimetre

   cm = duration*0.034/2;

  if(cm < 20)
  {
    car_stop_bot();
    delay(2000);

    car_go_back();
    delay(2000);
```



```
    car_stop_again();
    delay(1000);

    car_go_left();
    delay(1000);
  }

  else
  {
    car_go_straight();
    delay(1000);
  }

// For Serial Monitor
Serial.print("Distance:CM ");
Serial.println(cm);

}

void car_go_straight()
{
lcd.setCursor(0,0);
lcd.print("NOTHING AHEAD");
lcd.setCursor(0,1);
lcd.print("MOVING FORWARD");

  digitalWrite(lm1,HIGH);
  digitalWrite(lm2,LOW);
  digitalWrite(rm1,HIGH);
  digitalWrite(rm2,LOW);
}
void car_go_back()
{
lcd.clear();
lcd.setCursor(0,0);
lcd.print("TAKING REVERSE");
lcd.setCursor(0,1);
lcd.print(cm);

  digitalWrite(lm2,HIGH);
  digitalWrite(lm1,LOW);
  digitalWrite(rm2,HIGH);
  digitalWrite(rm1,LOW);
}
void car_stop_bot()
{
lcd.clear();
lcd.setCursor(0,0);
lcd.print("SOMETHING AHEAD");
lcd.setCursor(0,1);
```



```
lcd.print("STOP!!");

  digitalWrite(lm1,LOW);
  digitalWrite(lm2,LOW);
  digitalWrite(rm1,LOW);
  digitalWrite(rm2,LOW);
}
void car_stop_again()
{
lcd.clear();
lcd.setCursor(0,0);
lcd.print("BREAK FOR TURN");

  digitalWrite(lm1,LOW);
  digitalWrite(lm2,LOW);
  digitalWrite(rm1,LOW);
  digitalWrite(rm2,LOW);
}

void car_go_left()
{
lcd.clear();
lcd.setCursor(0,0);
lcd.print("TURNING LEFT");
lcd.setCursor(0,1);
lcd.print(cm);

  digitalWrite(lm1,LOW);
  digitalWrite(lm2,LOW);
  digitalWrite(rm1,HIGH);
  digitalWrite(rm2,LOW);
}

void car_go_right()
{
lcd.clear();
lcd.setCursor(0,0);
lcd.print("TURNING RIGHT");
lcd.setCursor(0,1);
lcd.print(cm);

  digitalWrite(lm1,HIGH);
  digitalWrite(lm2,LOW);
  digitalWrite(rm1,LOW);
  digitalWrite(rm2,LOW);
}
```