Spring 2025

UNIVERSITY OF ILLINOIS CHICAGO

MECHANICAL AND INDUSTRIAL ENGINEERING

ROBOTICS AND MOTION LABORATORY

Project report

Autonomous Lane Driving Car

DATED: 29/04/2025

Submitted to

Professor Pranav Bhounsule

by

Mir Yamin

UIN: 670529549

Abstract

This project focused on the design, fabrication, and testing of a small-scale autonomous vehicle capable of navigating between lanes marked by white tape. The vehicle was mechanically designed in SolidWorks, 3D printed using PETG filament and assembled with custom and off-the-shelf parts. The car is made of 54 parts (41 different parts). The design used an Intel RealSense D435i depth camera, a Youyeetoo A1M8 LIDAR, and an Nvidia Jetson Orin Nano running ROS 2 Humble. Custom ROS 2 nodes were developed in Python to handle sensor data acquisition, lane detection, obstacle detection, data fusion, and motor control. A PID controller was implemented to smooth the steering response and ensure stable lane following.

Testing demonstrated that the car was able to reliably follow straight and moderately curved lanes while successfully detecting and avoiding static obstacles such as walls using LIDAR feedback. Key challenges encountered included sensor noise, environmental lighting variability, and mechanical durability, which were addressed through multiple design improvements and software filtering techniques. Although the system struggled with sharp turns and cluttered environments, it laid a strong foundation for future enhancements involving neural network-based lane detection, advanced path planning, and dynamic obstacle avoidance. The project provided valuable experience in multidisciplinary system integration for real-time autonomous robotics.

Introduction

In recent years, autonomous vehicles have emerged as one of the most transformative technologies in the fields of robotics, mechanical engineering, and artificial intelligence. The purpose of this project was to design and develop a small-scale autonomous car capable of navigating between lanes denoted by white tape, simulating the lane-following behavior seen in full-size autonomous vehicles like Tesla's. The project provided an opportunity to integrate mechanical design, electronics, and software control into a cohesive system, demonstrating fundamental principles of perception, decision-making, and actuation.

The project was done during the Spring 2025 semester. The design phase took approximately seven weeks, during which thirteen different designs were created and 3D printed, ultimately leading to a final, sturdy car design optimized for strength and integration.

The parts were 3d printed using Bambo Lab A1 mini and filament used was Bambo Lab's PETG.

The project involved both mechanical and electrical design. The mechanical design of the car chassis was created using SolidWorks to ensure proper integration of all components while maintaining a compact structure. The electronics used are Intel RealSense D435i depth camera and a Youyeetoo A1M8 360-degree LIDAR for environmental perception. The motion of the car was made possible using a PCA9685 servo driver controlling a 995MG servo and an Injora motor with integrated gearbox and ESC.

For computation and control, the Nvidia Jetson Orin Nano served as the onboard processing unit, running ROS 2 Humble. ROS 2 nodes were written in Python to process sensor data, detect lane markings, plan motion paths, and generate commands which were sent to motor. Depth and LIDAR data were fused to improve obstacle detection and lane-following accuracy.

The system aimed to achieve robust lane-following behavior under varying conditions in a laboratory environment, simulating real-world autonomous driving scenarios at a smaller scale. In addition to the autonomous navigation functionality, a major focus was placed on modular design and scalability, allowing future enhancements such as dynamic obstacle avoidance, overtaking, and real-time bird's eye visualization similar to commercial autonomous vehicle systems.

Methods/Approach

The design of the autonomous car was created in SolidWorks, consisting of 25 individual parts assembled together. A total of thirteen chassis designs were made, tested, and improved. After creating the designs, the parts were fabricated using 3D printing. Many initial prototypes failed mechanical tests, exhibiting issues such as excessive flexing, instability, and lack of sufficient strength. The complete design and fabrication process to reach a final sturdy model took approximately seven weeks.

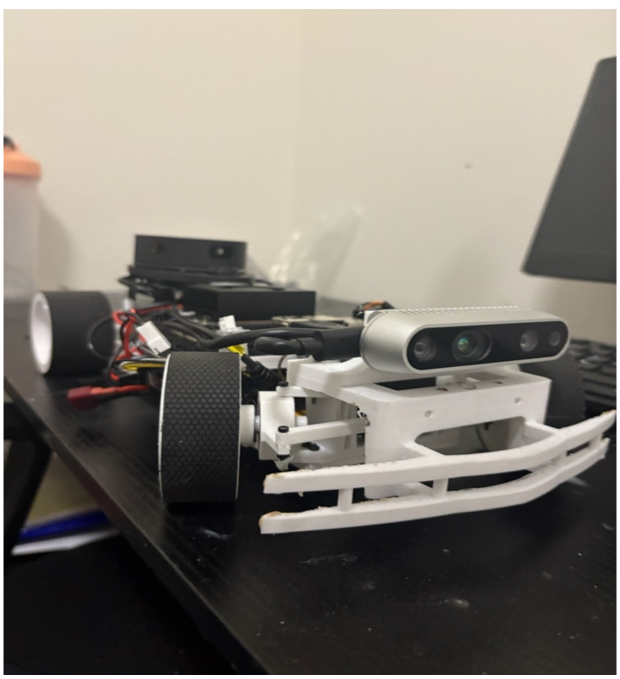


Image: Final Design of Car

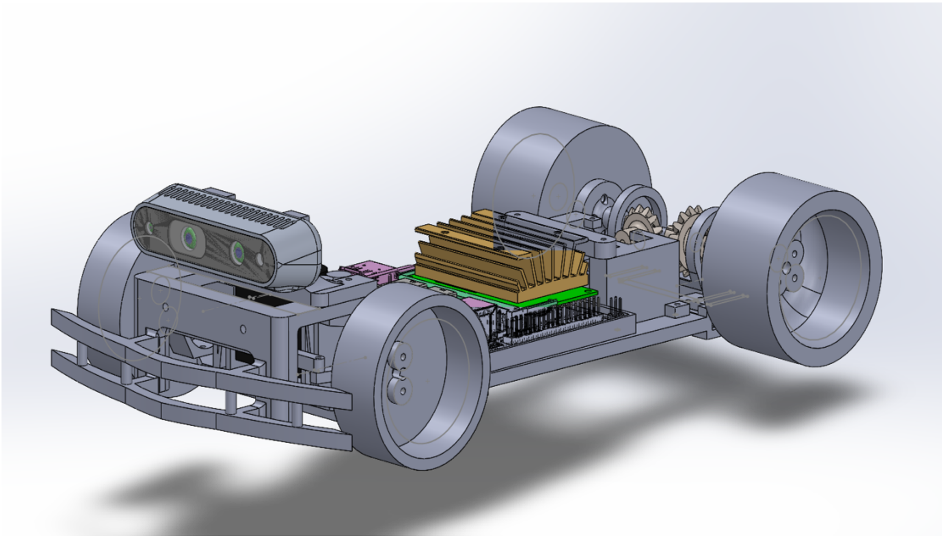


Figure 1: Assembled car

The parts were printed using Bambu Lab’s A1 Mini 3D Printer with PETG filament. PETG was selected due to its balance between strength, durability, and ease of printing. In addition to 3D printed components, several off-the-shelf mechanical parts were incorporated, including 3 mm and 8 mm bearings bought from Amazon. The assembly was joined together using M2 screws for structural connections. To make car more balanced and to prevent the structure from flexing, two metel rods run along front and back bottom plate of car.

Initially, a high-RPM RC race car motor was considered. A custom reduction gearbox was designed in SolidWorks and printed using PETG to reduce the motor speed from 8000 RPM to approximately 350 RPM. Although successful initially, the gearbox proved unreliable over time, suffering from issues such as noise, the need for frequent lubrication, and mechanical failure under continuous operation. Consequently, the high-RPM motor was replaced with an Injora DC motor featuring an integrated metal-gearbox, offering significantly higher reliability, lower noise, and greater robustness.

For steering, a **single-point push-pull mechanism** with a rectangular linkage was developed. The front wheels were mechanically coupled via a rectangular solid bar, ensuring synchronized turning. One wheel was connected to the servo motor through a bent rod (L Shaped), allowing steering movement through the servo's rotation.

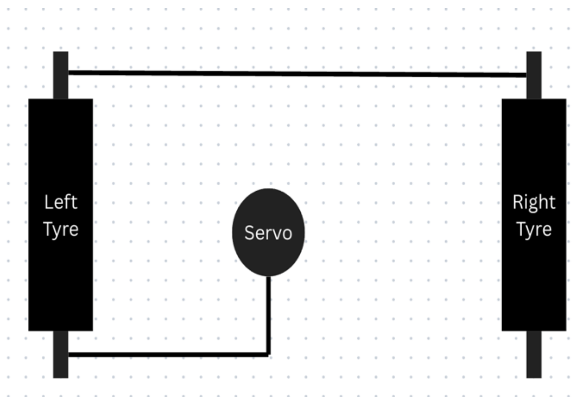


Figure 1.12: Steering Mechanism

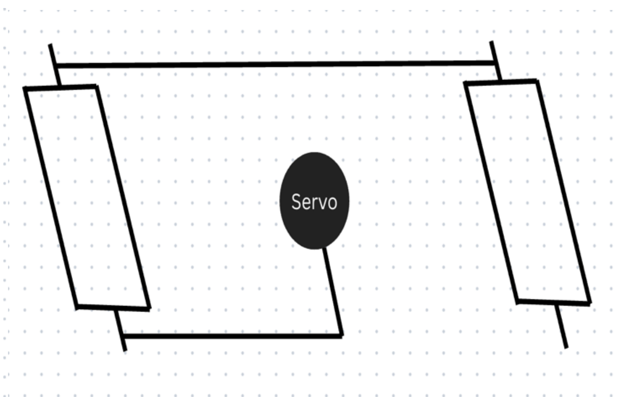


Figure 1.13: Steering mechanism at theta angle

The rear axle integrated a bevel gear system (Figure 1.3) to transfer torque from the motor to the wheels, with a 90-degree transmission between motor shaft and axle. Both bevel gears were designed to have 30 teeth each. A custom connector hub, also designed in SolidWorks, linked the axle to the wheels, each used 4 M2 screws to mount wheels to hub, facilitating easy replacement of drivetrain components if failures occurred.

Power was supplied by two independent battery systems. A 20,000 mAh external battery powered the Jetson Orin Nano, while a 1500 mAh 7.4V 35C LiPo battery exclusively powered the motor and ESC. The electronics were mounted systematically, with the PCA9685 motor driver placed above the battery compartment, and all wiring organized using jumper wires for modularity and easy maintenance. The servo was connecetd to BEC of ESC (5V 3A).

The motor and ESC used operates at 1500ms pulse which is neutral. If ESC is supplied with 1500-2000ms pulse, the car moves forward while below 1500 makes it go backward.

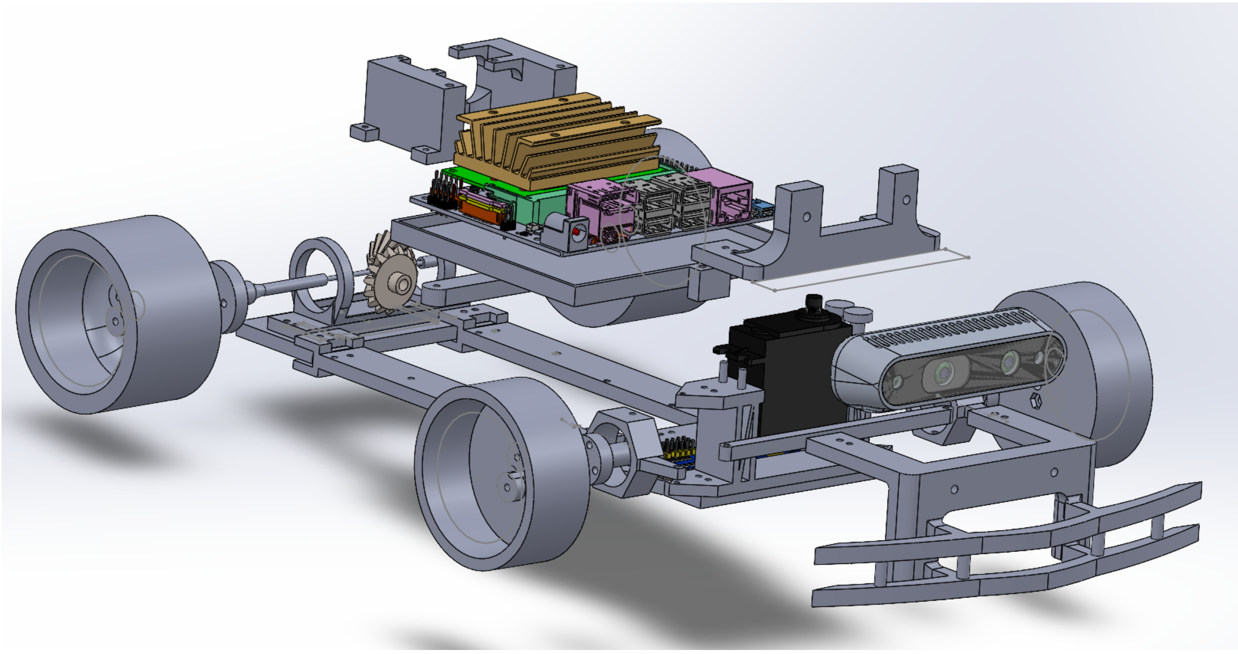
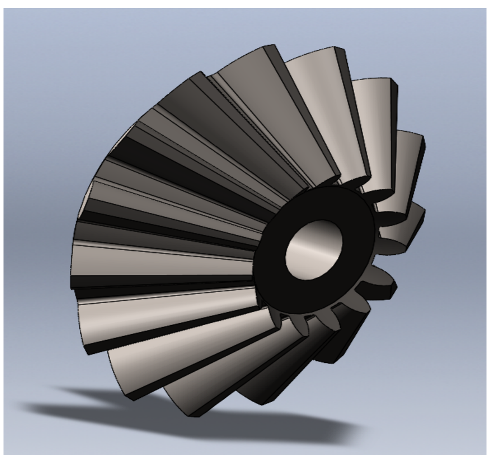


Figure 1.2: Exploded View of car



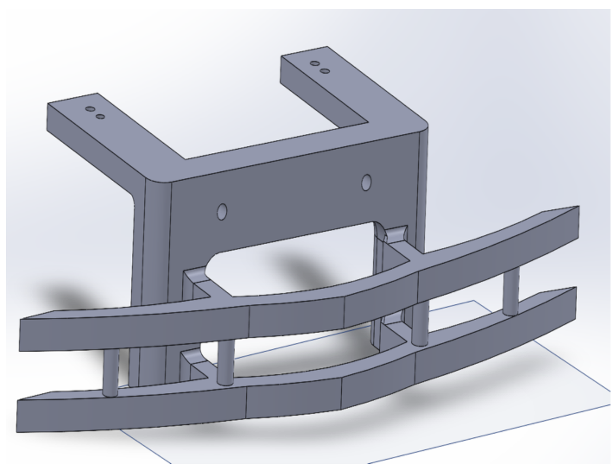


Figure 1.3: Design of gear

Figure 1.4: Design of front shield

Sensor Integration

Two main sensors were employed:

1. Youyeetoo A1M8 LIDAR: Mounted above the motor casing, this 2D, 360-degree LIDAR with a 12-meter scanning radius was tasked with generating a virtual safety boundary (bounding box) around the vehicle. If any object was detected

approaching or entering this bounding box, corrective control signals were sent to the motor and steering to avoid collision.

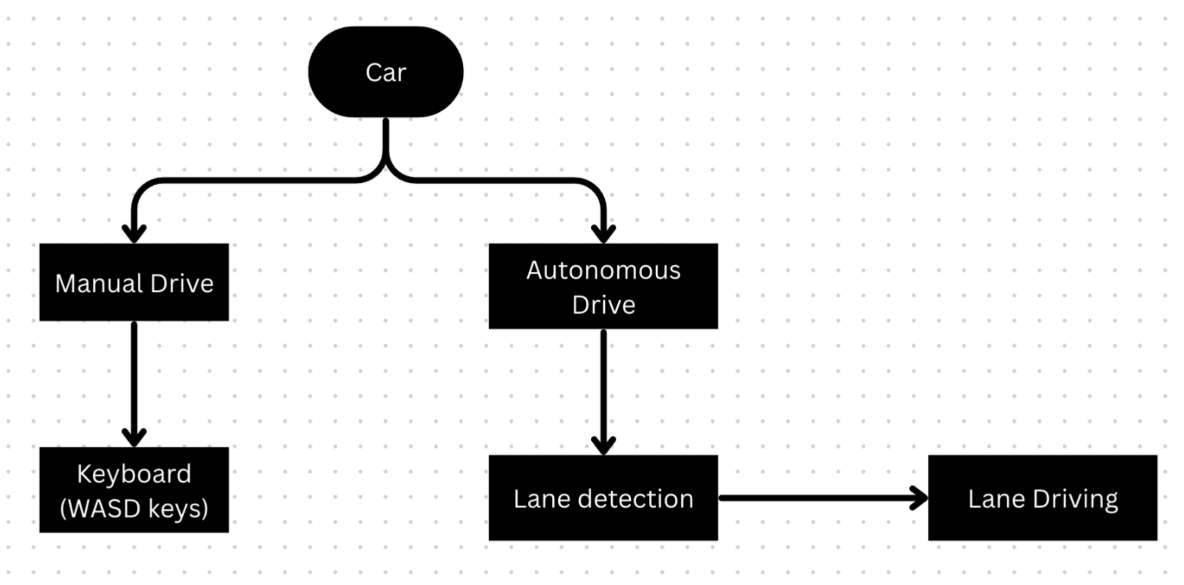
2. Intel RealSense D435i Camera: Mounted at the front of the car, the camera served as the primary perception sensor. It provided both depth data and an Inertial Measurement Unit (IMU) feed. For lane following, only the lower portion of the camera's depth feed was processed, focusing on identifying and tracking white tape lane markings on the ground.

Software Architecture

There are two methods of operating the car.

Manual Method

The manual method uses terminal to manually send commands to car. The usual WASD keys are used to operate the car.



The Autonomous Method:

During the initial stages, a Raspberry Pi 4B+ was used for computation; however, it was quickly replaced by the Nvidia Jetson Orin Nano due to increased sensor data demands and frequent crashes due high load of data from multiple sensors. The Jetson Orin Nano provided superior performance at a relatively low cost, making it well-suited for real-time robotics applications.

The system was built using ROS 2 Humble. Custom ROS 2 nodes were developed in Python for the following functionalities:

1. Sensor Data Acquisition: Nodes subscribed to LIDAR and camera topics and gathered real-time data streams.
2. Lane Detection: Depth images from the RealSense camera were processed using OpenCV techniques to detect lane boundaries formed by white tape on the ground.
3. Obstacle Detection: LIDAR data was used to monitor the safety perimeter around the vehicle, identifying potential obstacles within a critical distance.
4. Control Command Generation: Based on lane position and obstacle information, control commands (PWM signals) were generated and sent to the PCA9685 driver to adjust steering and motor speed.
5. Data Fusion: LIDAR and camera data were combined at the decision-making layer to ensure robust navigation even when one sensor’s data was compromised.

The motor controller interpreted PWM signals, where a 1500 ms pulse indicated a neutral state, pulses above 1500 ms corresponded to forward movement, and pulses below 1500 ms indicated reverse movement. The effective range was between 1000 ms and 2000 ms.

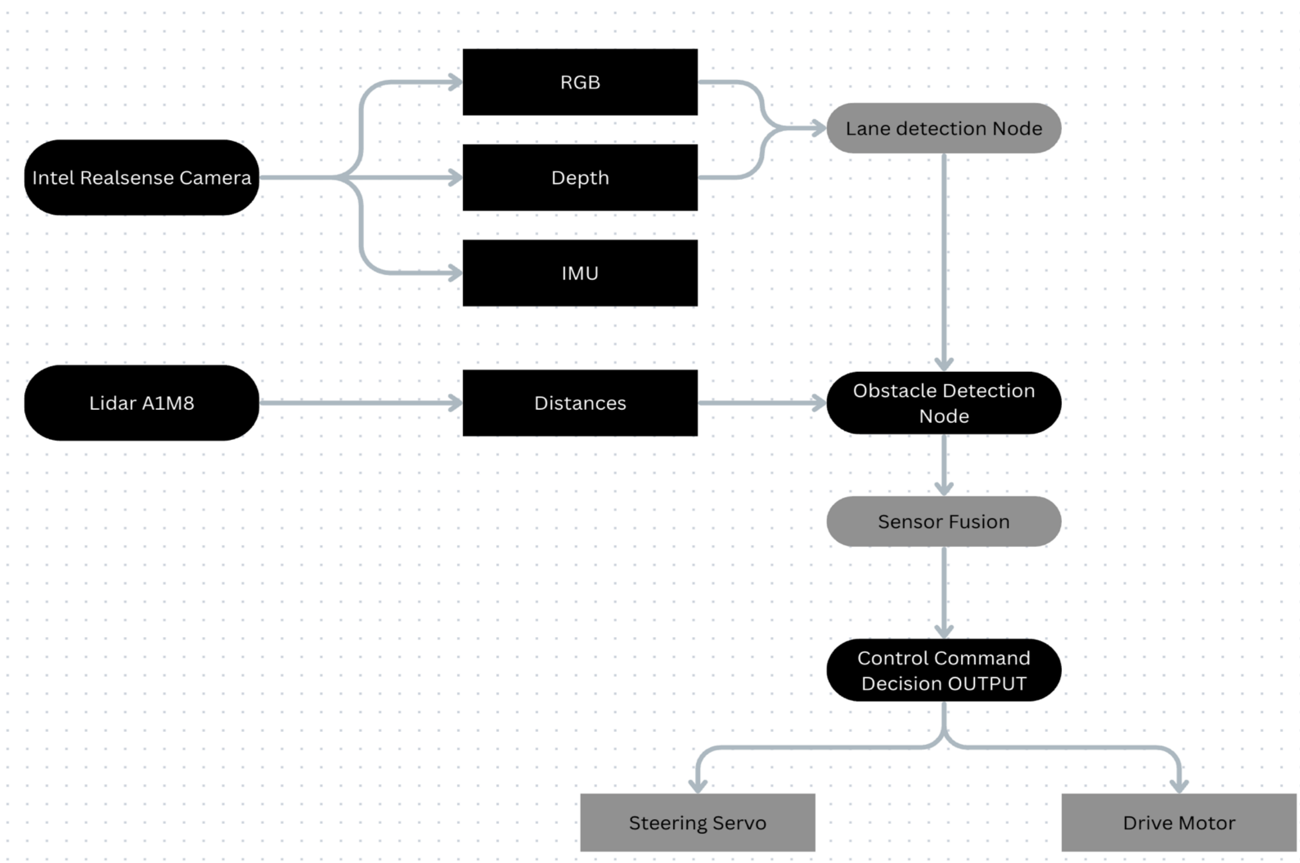
PID Controller Implementation

To achieve smoother and more accurate lane-following behavior, a Proportional-Integral-Derivative (PID) controller was implemented for the steering control system.

The PID controller took the error between the center of the detected lane and the center of the camera's field of view as input. It then generated correction signals to the steering servo to minimize this error over time.

1. The Proportional (P) term provided immediate corrective action proportional to the magnitude of the lane offset.
2. The Integral (I) term accounted for accumulated past errors, helping to eliminate steady-state drift.
3. The Derivative (D) term predicted future error based on its rate of change, allowing the controller to dampen oscillations.

Careful tuning of the PID was done to achieve a balance between fast lane-centering response and stable, oscillation-free driving. Without PID control, the car exhibited sharp, jerky steering corrections; with PID implemented, lane-following behavior became smooth and significantly more reliable.



Design Challenges and Solutions

Several challenges were encountered throughout the project:

1. Mechanical Durability: Early prototypes were too fragile. The infill was increased to 20% and wall loops were increased to 6. Material selection was improved by switching to PETG and reinforcing critical structural components.
2. Motor Reliability: The initial gearbox solution was abandoned in favor of a commercially available motor with integrated metal gears, increasing motor mechanism integration.
3. Processing Power: The Raspberry Pi lacked sufficient computational ability for real-time perception and control. Upgrading to the Jetson Orin Nano resolved these issues and enabled multi-sensor integration.
4. Sensor Noise: LIDAR and camera data had occasional noise. Smoothing filters and outlier rejection methods were applied in software to improve accuracy.
5. During initial testing the car failed to respond as it was detecting the floor as nearest object, and it was getting detected inside bounding box. The bouding box was then shortened and the issue was fixed.
6. The car was confusing other white color things with lanes and then getting stuck. It was tried to fix by cutting out the top part of camera feed but that failed multiple times.
7. Jerky steering correction were corrected after implementing PID.

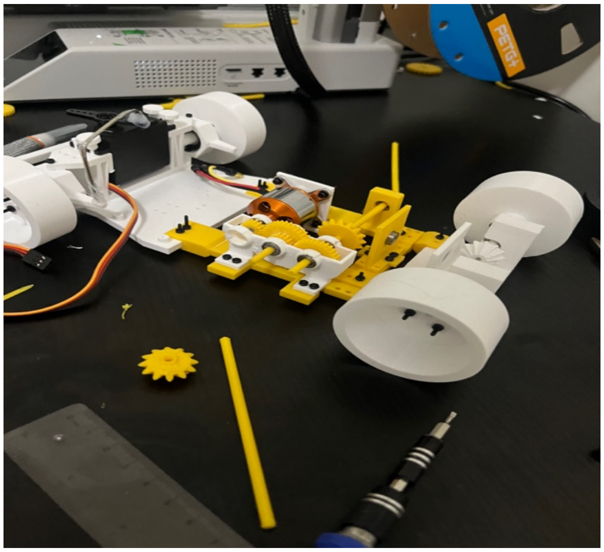


Figure: Initial Design of car

Bill of Materials

S.no	Part Name	Quantity	Price (USD)
1	Servo Motor	1	11
2	Intel RealSense D435i Camera	1	334
3	Nvidia Jetson Orin Nano	1	249
4	Injora Dc Motor with ESC	1	44
5	Bearing 8mm	2	0.9
6	Steel Rod	3	1.5
7	Lidar A1M8	1	99
8	PCA 9685	1	5
9	Jumper Wire	15	4.8
10	M2 Nut Bolt	30	0.36
	Total		**749.56**

Total Cost excluding printing material = $749.56

Design Parts:

ITEM NO.	PART Name	QTY.
1	CarFront_base	1
2	CarServo_mount	1
3	Carfront_tyre_bearing_ mount	2
4	Carmacro-servo2	1
5	CarCap_servomount	2
6	Carfront_tyre_mount	2
7	Part7^Assem11	1
8	Part8^Assem11	1
9	Part11^Assem11	1
10	Carsteering_link	1
11	CarTyre_front	2
12	CarCopy of Part5^Assem1	1
13	Carback_tyre_mount	2
14	CarTyre_Big	2
15	Part17^Assem11	1
16	Carback_mount	1
17	CarMid_link_1	1
18	Carmid_link_2	1
19	CarJetson_mount	1
20	Carmotor_mount	1
21	Metric - Straight bevel pinion 1.5M15PT 15GT 14.5PA 9FW --- 15O7H17MD4N	1
22	Metric - Straight bevel pinion 1.5M16PT 15GT 14.5PA 9FW --- 16O7H17MD2N	1
23	Carmotor_mount_link	1
24	Part21^Assem11	1
25	Part24^Assem11	1
26	Carbackwheen_bearin g_mount	1
27	Copy of Copy of Part2^Assem11	1

28	CarFront_Shield	1
29	CarCamera_mount	1
30	Copy of Part19^Assem11	1
31	CarRealSense_D435	1
32	142-13449-1000-A02.stp	1
33	142-63448-1GEN_A00.stp	1
34	PORG_LEAF_SPRING_DRAWN.stp	1
35	PORG_LEAF_SPRING_SCREW.stp	4
36	PORG_HEX_STANDOFF.stp	2
37	PORG_SYSTEM_SCREW.stp	4
38	155-0576-000.stp	1
39	Car12Channel PWM	1
40	Intel RealSensecamera	1
41	Lidar_A1m8	1

Parts = 41

Total Parts = 54

Results and Conclusion

The final design successfully achieved the core goal of autonomously following lanes marked by white tape on the ground. Lane detection was implemented by detecting the tape edges and then averaging the detected lines to generate a straight centerline path for navigation. For straight-line segments, the system performed reliably, maintaining smooth and centered motion between lanes. When encountering curved paths or corners, the car was programmed to use one of the lanes as a reference and maintain a fixed distance from it. This strategy improved navigation around moderate curves, although performance degraded for sharper turns exceeding 70 degrees.

Obstacle detection using the LIDAR was successful in identifying walls and nearby objects, providing rapid feedback to the obstacle detection node. The car was able to stop or adjust its motion appropriately when encountering objects within its safety bounding box.

Overall, the car demonstrated smooth and accurate lane following on straight sections, with reliable obstacle avoidance behavior in controlled environments.

Several findings emerged from the implementation and testing process:

- Filtering based solely on detecting the white color of the tape proved **unreliable** under different lighting conditions or when multiple white objects were present. More advanced image processing methods (such as perspective transformations or deep learning-based segmentation) would be necessary for improved performance.
- The car struggled with sharp turns due to the lane-following algorithm's limitations and the mechanical steering constraints of the rigid linkage system. More complex path-planning algorithms could enhance performance in the future.
- The addition of a PID controller for steering significantly improved stability. Before PID, the car exhibited jerky and delayed corrections, but after PID tuning, the motion became smoother and more centered.
- A major lesson learned was the **importance of testing edge cases** (e.g., overlapping white marks, broken lane markings) to evaluate robustness.

In conclusion, a fully functional small-scale autonomous lane-following car was successfully designed, fabricated, and tested. It was able to follow white tape lanes and perform basic obstacle avoidance in a controlled indoor environment. The project integrated mechanical design, electronics, sensor fusion, and real-time software development using ROS 2 and the Nvidia Jetson Orin Nano platform.

Apart from this several enhancements could be pursued:

- Implementing more advanced lane detection methods, such as neural network-based approaches (e.g., Nvidia's PilotNet) to handle complex environments and lighting variability.
- Improving path planning around curves and intersections using trajectory prediction algorithms.
- Adding a real-time birds-eye-view visualization system to better monitor the car's environment and decision-making.
- Exploring dynamic obstacle avoidance and overtaking capabilities for multi-object environments.
- Conducting extensive outdoor testing with different surface materials and lighting conditions.

This project provided valuable hands-on experience in integrating hardware and software for autonomous navigation and highlighted the complexities involved in real-world robotics applications.

Appendix 1

Some of the important part's assembly drawing:

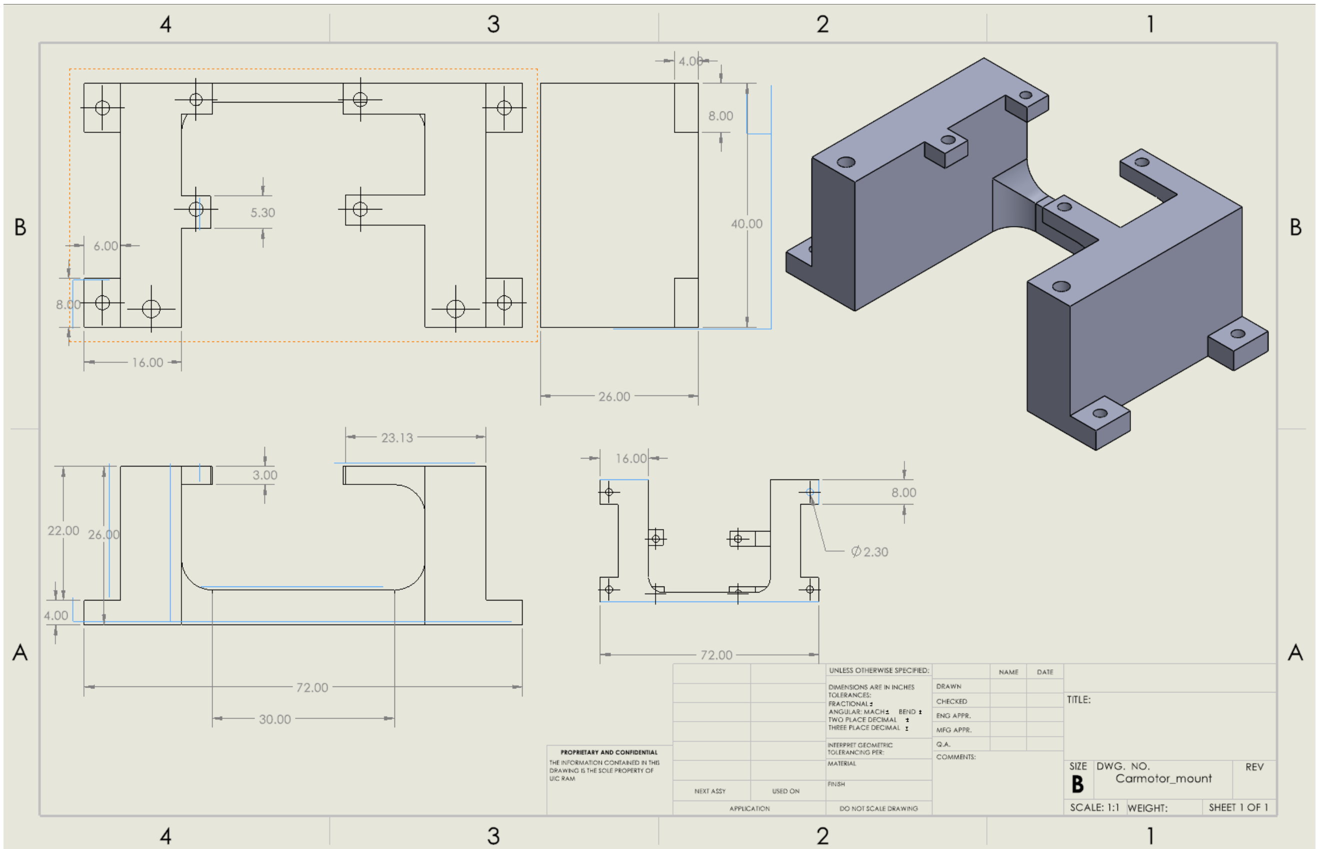


Figure 2: Motor mount on which Injora Motor is mounted

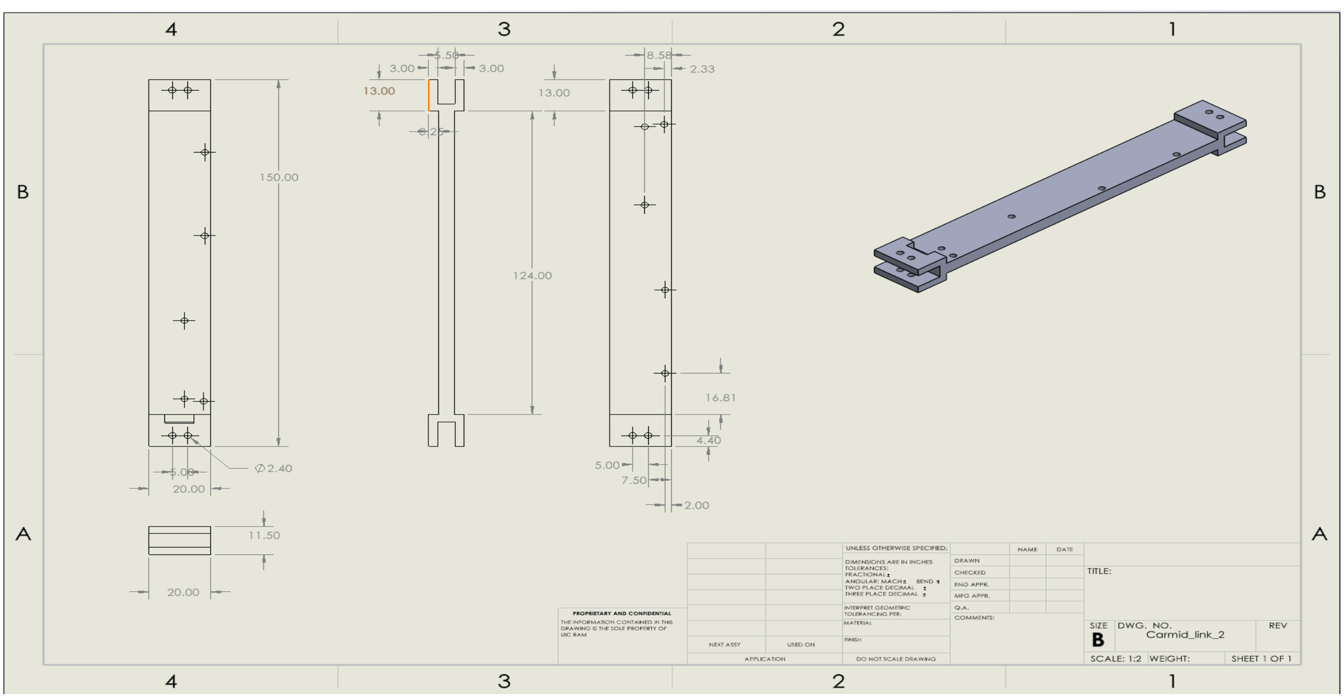


Figure 3: The base link which connects front plate and back plate of car.

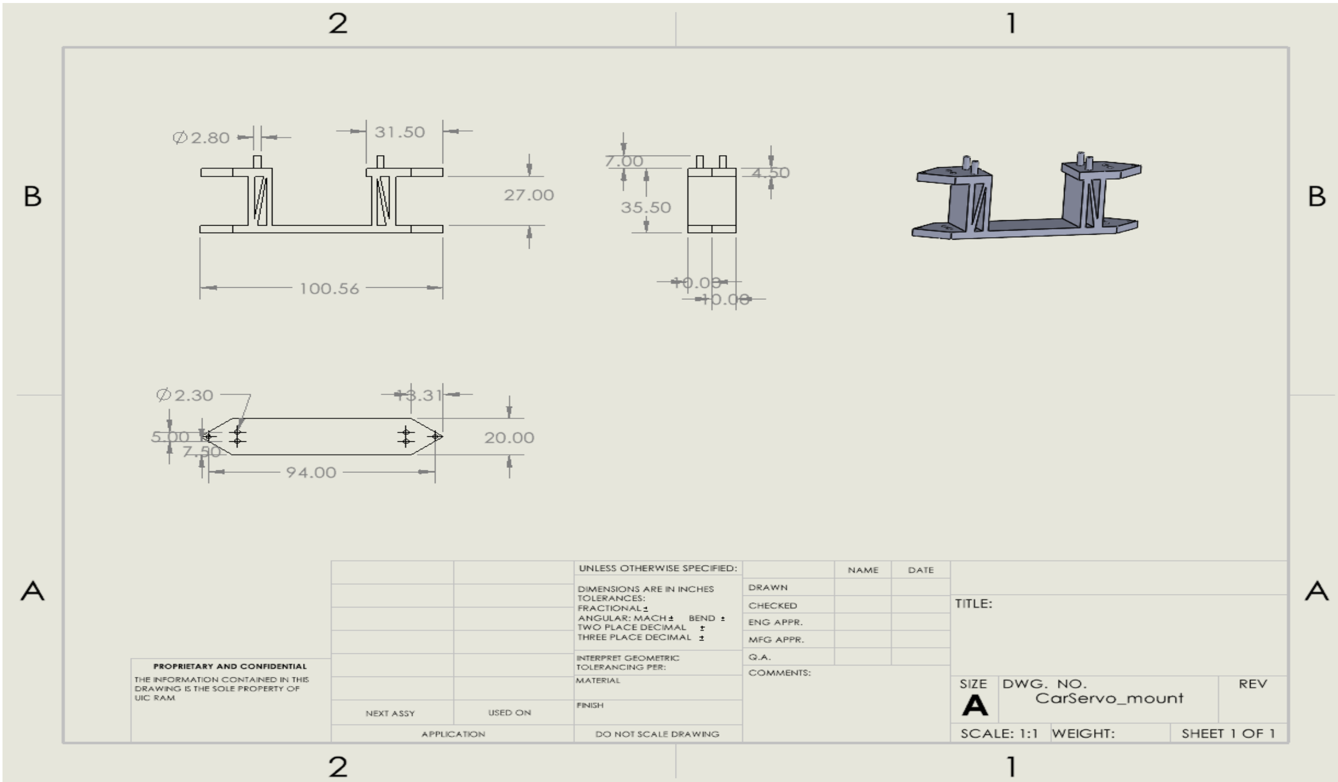


Figure 4: Mount on which servo is installed

Appendix 2

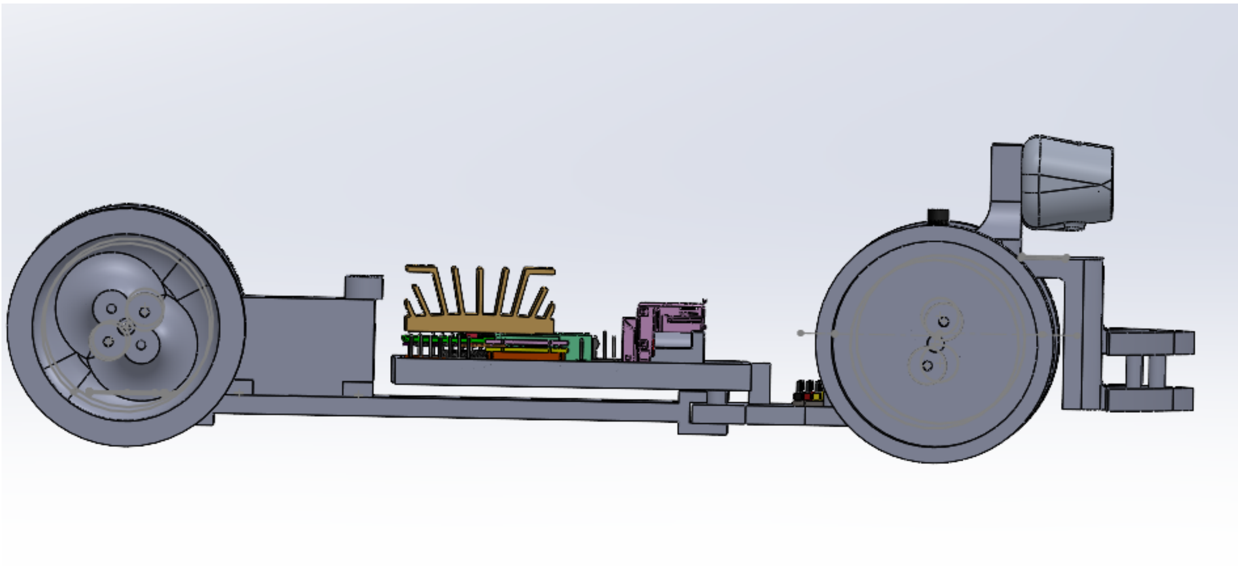


Figure 5: Side View of Car

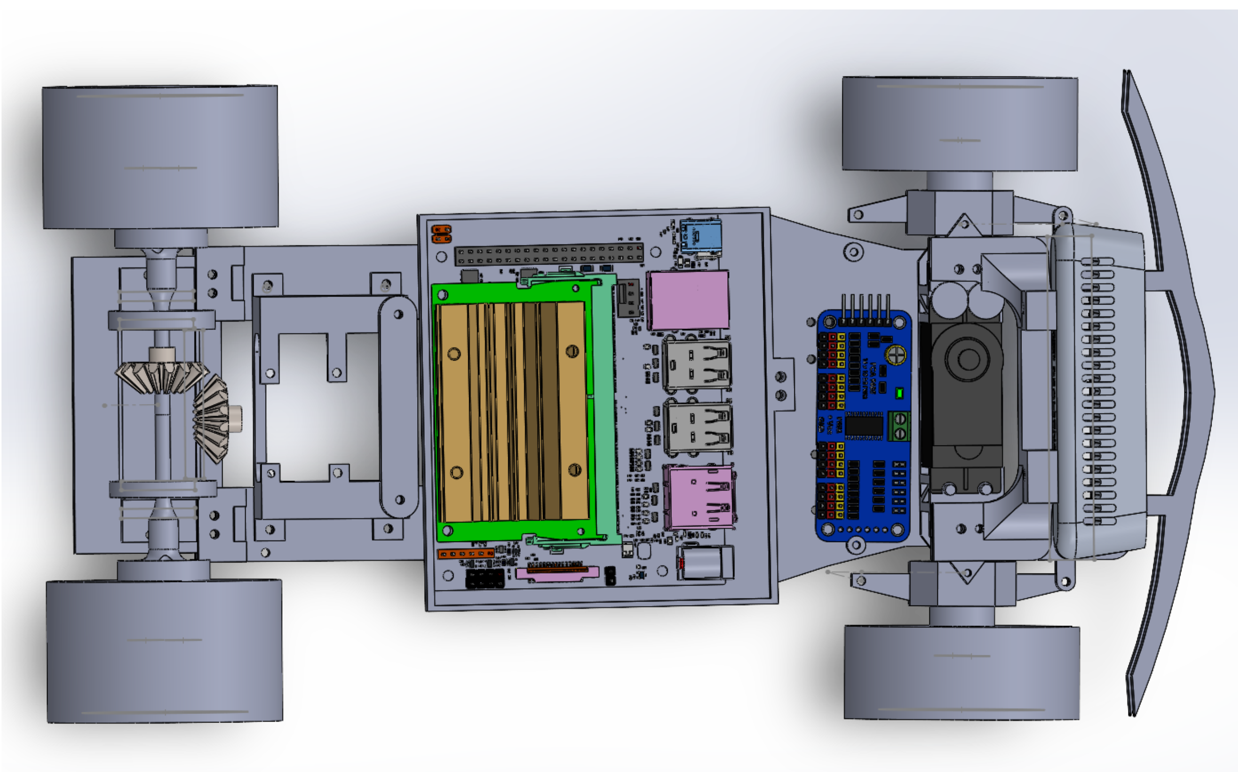


Figure 6: Top view of car