Design and Implementation of Mechatronic Systems

Tecnológico de Monterrey

Integrative Technical Report.

Semester 8
Team: Supertronics & Automation

Members:
Saúl Jesús Cuervo Méndez A01735937
Adán Francisco Cruz Ramírez A01736676
Raymundo Barbosa López A01735779

May 31, 2025

Index

1) Introduction

The increasing demand for autonomous systems capable of operating in uncertain environments has driven the development of intelligent mobile platforms that integrate perception, control, and decision-making. This project presents the design and implementation of an autonomous vehicle prototype that performs lane following, obstacle avoidance, and stop sign recognition using computer vision and LiDAR-based perception, implemented under the Robot Operating System (ROS) framework. The system combines mechanical, electronic, and software components to achieve a reliable and robust navigation strategy.

This work aims to validate an automation proposal through simulations and experimental testing, ensuring compliance with quality, safety, and productivity standards. The integration of cutting-edge technologies—such as deep learning models for object detection, PID-based control algorithms, and sensor fusion—enables the autonomous platform to adapt to dynamic scenarios. The development is supported by a systematic research process that includes a comprehensive review of the state of the art in mechatronic systems and autonomous navigation.

2) Project Definition

Sponsor / Stakeholder:
Dr. Hugo Gustavo González Hernández

Purpose and Objectives:
Develop an autonomous 1:10 scale Traxxas race car for indoor navigation, capable of lane following, traffic sign recognition, and static obstacle avoidance. The system integrates mechanical, electronic, and software components under ROS, applying computer vision and LiDAR data for decision-making and control.

Product:

- Autonomous vehicle with integrated perception and control systems
- Technical report in IEEE format

Timeframe:
9 weeks

Location:
Mechatronics Laboratory, Tec de Monterrey campus Puebla

Resources:

Team of 3 students; available hardware includes RPLIDAR A3M1, webcam, Jetson Xavier NX, omnidirectional Wi-Fi antennas, and Traxxas chassis with motors and steering servo.

Constraints & Risks:

- Limited time due to academic obligations
- ROS complexity and sensor integration challenges
- Hardware failures or delays
- Reduced testing time (3–4 effective weeks)

Scope:

- Lane detection and following
- Recognition of one traffic sign (STOP) on straight segments
- Avoidance of at least one static obstacle
- Basic control via PID
- Validation through controlled indoor experiments

Out of Scope:

- Dynamic obstacle handling
- Multi-sign detection
- Outdoor operation or GPS-based localization

3) Problem Identification and user needs

The team selected this project after identifying a significant gap: the institution currently lacks a ready-to-use platform capable of participating in advanced autonomous vehicle competitions with defined minimal requirements. This absence limits students' opportunities to engage in hands-on, competitive mechatronics challenges that develop both theoretical and practical skills.

The project aligns with our technical interests and leverages our existing knowledge in robotics, control systems, and computer vision, enabling efficient development within the established timeframe. Additionally, faculty members expressed the need to form a competitive team to represent the institution in national and international events, establishing a clear institutional demand.

From a user perspective, the project addresses multiple needs:

- Providing a functional, scalable autonomous vehicle platform that facilitates hands-on learning and experimentation with navigation, traffic sign recognition, and obstacle avoidance.
- Offering a tested foundation to future students and researchers, reducing development time and allowing them to focus on improvements rather than starting from zero.

- Enhancing practical skills relevant for professional careers and competitive examinations, thereby increasing student readiness for industry challenges.
- Meeting institutional goals by contributing a competitive project that elevates the school's presence in robotics contests.

Addressing these needs promotes continuity, knowledge transfer, and advances the institution's capacity to innovate in mechatronics education and research.

4) Solution Proposal

The project develops a 1:10 scale autonomous racecar using a Traxxas chassis, equipped with an USB webcam, Lidar A3M1, Jetson Xavier NX, motors, and a steering servo.

The system uses the webcam and OpenCV within ROS to detect a red stop sign via HSV filtering and contour analysis, stopping the vehicle upon detection.

For navigation, the vehicle follows a single yellow line on the right using HSV thresholding and Hough Line Transform. A PID controller adjusts steering based on the line's position offset to maintain smooth tracking.

Obstacle avoidance relies on the Lidar detecting objects within 1 meter. The vehicle stops and performs a programmed evasive maneuver by reversing and steering around the obstacle.

Control commands for steering and speed are published via ROS topics. The integrated system is tested on an indoor track at the Mechatronics Laboratory to validate its autonomous navigation and obstacle avoidance capabilities.

5) State of the art

Introduction and Context
The development of scaled autonomous vehicles, exemplified by platforms like F1TENTH, has driven significant progress in autonomous driving research and education. Since its emergence, the F1TENTH platform has become a widely adopted benchmark in academia and competitions, supporting studies ranging from classical trajectory planning methods to advanced deep learning algorithms.

Despite the wealth of research, the field remains broad and somewhat fragmented, posing challenges to direct comparison of methodologies and identifying a clear state of the art. To mitigate this, recent efforts focus on standardizing evaluation through open benchmarks, enabling objective and reproducible comparisons between diverse approaches.

Main Approaches in F1TENTH Autonomous Driving
Research in this domain generally divides into two categories:

- Classical Methods:
  These include particle filter-based localization to estimate vehicle position under uncertainty, model predictive contouring control (MPCC) for trajectory optimization and tracking, and reactive obstacle avoidance algorithms such as Follow-the-Gap, which navigate through free spaces detected in the environment.

- Learning-Based Methods:
  End-to-end reinforcement learning approaches train vehicles to make driving decisions directly from sensor data, bypassing explicit trajectory planning and enabling adaptive behavior learned from interaction with the environment [3].

Comparisons and Evaluations
Standardized benchmarks assess methods on criteria such as control frequency, localization accuracy, reward function design, and training environment realism. Studies indicate that trajectory optimization and tracking methods currently achieve the fastest lap times, followed by real-time online planning strategies [4].

Relevant Prior Work
To ground our project, we reviewed key research closely aligned with our objectives, highlighting five influential studies:

No.	Paper title	Journal / Source	Quartile	Why is it related to or important for your project?	Quotes
1	Automation of a tow-tractor for the autonomous delivery of materials in an industrial complex	International Journal on Interactive Design and Manufacturing (IJIDeM)	Q2	It presents a flexible navigation system for AGVs that integrates sensors like cameras, LiDAR, and encoders to enable autonomous route tracking and obstacle avoidance—key capabilities for recognizing traffic signs and evading static obstacles. The two-layer control architecture enhances safety and	0

				adaptability without requiring environmental modifications, making it suitable for integration into small-scale autonomous vehicles like JETRACER.	
2	Line Follower Robot: Design and Hardware Application	2012 International Conference on Informatics, Electronics and Vision, ICIEV 2012	N/A	Explains the basic principles of line-following robots, including the use of sensors and advanced programming for navigation. Mentions the use of sensors to detect black lines on white backgrounds (or vice versa), which is essential for proper navigation of the autonomous cart.	27
3	Control for balancing line follower robot using discrete cascaded PID algorithm on ADROIT V1 education robot	2015 International Electronics Symposium	N/A	It features an advanced control strategy that combines three PID controllers (for balance, speed, and line following). This could help us improve the cart's stability and precision on the track. Shows experiments demonstrating the effectiveness of cascade PID control.	35
4	Teaching Autonomous Systems at 1/10th-scale : Design of the F1/10 Racecar, Simulators and Curriculum	Proceedings of the 51st ACM Technical Symposium on Computer Science Education	N/A	The paper "Teaching Autonomous Systems at 1/10th-scale: Design of the F1/10 Racecar, Simulators and Curriculum" presents a ROS-based autonomous racing simulator for the F1/10 platform. It is designed to enhance education in autonomous systems by providing a safe and accessible testing	22

				environment. The simulator allows developers to test control, perception, and planning algorithms without requiring physical hardware.	
15	Autonomous navigation system of indoor mobile robots using 2D lidar	Mathematics	Q2	The paper "Autonomous Navigation System of Indoor Mobile Robots Using 2D LIDAR" describes the development of a prototype autonomous robot that uses a 2D LIDAR sensor to measure distances and navigate indoor environments. The LIDAR sensor emits a pulsed light beam to determine the distance between the robot and surrounding objects, contributing to the robot's autonomy without human intervention.	19

Table 1. State of the art table

A comprehensive list of consulted literature is available here.

6) Research methodology

This research project adopted a mixed-methods approach, combining both theoretical and practical methodologies to address the development and evaluation of an autonomous vehicle based on the Traxxas platform.

To establish the theoretical foundation, a documentary research methodology was used. Relevant academic literature, technical documentation, and prior studies related to autonomous navigation, computer vision, LiDAR-based perception, and control algorithms such as the Stanley controller were reviewed. This allowed for a comprehensive understanding of the technologies and algorithms implemented in the project.

From a technical perspective, the project followed an applied research methodology, in which existing theories and techniques were adapted to solve a real-world problem. The integration of hardware components (Jetson Xavier, camera, LiDAR, servo motor) and software modules (ROS, image processing pipelines) was guided by this practical orientation. The primary goal was to design, implement, and refine an autonomous navigation system capable of lane following, traffic sign recognition, and obstacle avoidance.

An experimental research methodology was also employed to validate the performance of the developed system. Several tests were conducted under controlled conditions, varying initial positions, vehicle speeds, and steering constraints. These trials aimed to assess how accurately the robot could follow a lane, stop at traffic signs, and maintain its trajectory using the PID controller.

Both quantitative and qualitative methods were used to analyze the results. Quantitative data, such as lateral error, processing time, and stop accuracy, provided measurable insights into system performance. Qualitative observations helped interpret the system's behavior in scenarios where sensor noise or incomplete data affected navigation.

The development of the autonomous vehicle followed an agile methodology, combining principles from SCRUM and Kanban. SCRUM enabled structured teamwork through sprints and reviews, while Kanban provided visual task tracking and workload management [1][2].

The technical process was divided into these key phases:

1. Requirements Analysis: Functional and technical requirements were defined, including lane following, traffic sign detection, and obstacle avoidance, based on project scope and competition needs.

2. Literature Review: Reference projects like F1TENTH and MIT Racecar were analyzed to extract best practices in sensing, control, and perception.

3. Prototyping and Validation: Early tests validated sensor integration (camera, LiDAR, encoders), motor control, and ROS communication in a controlled environment.

4. Incremental Integration: Modules were progressively added to the ROS ecosystem, with specific performance metrics evaluated at each stage (e.g., detection accuracy, latency).

5. Continuous Improvement: Based on test results, control parameters and perception algorithms were refined to improve overall system reliability.

6. Documentation and Version Control: All code and experimental data were managed using Git to ensure traceability and collaboration.

This structured and iterative approach enabled the team to develop a reliable and functional autonomous system ready for real-world challenges.

7) Proposal Development

The project aims to develop a scaled autonomous vehicle capable of navigating a predefined indoor track, detecting STOP signs, and avoiding static obstacles. The system must meet the following functional objectives:

- Complete at least one lap without human intervention.
- Detect and respond appropriately to a STOP sign placed on the track.
- Avoid static obstacles using real-time reactive control.
- Operate under a maximum speed of 0.2 m/s for safety and control.

System Architecture
The solution is organized into three functional stages: input, processing, and actuation.

Inputs
- Camera (RGB Video Stream): Used for traffic sign detection.
- RPLIDAR A3M1 (2D Point Map Cloud Data): Used for obstacle localization.
- Manual Start Signal: Triggered by the user to initiate autonomous behavior.

Processing
The Jetson Nano board handles all onboard processing, including:

- Vision model inference for STOP sign recognition (YOLOv5).
- Obstacle detection from LiDAR data.
- Motion planning and decision-making.
- ROS 1 node communication and integration.

Outputs

- PWM signals to control both the steering servo and the motor driver, enabling autonomous movement.

- Debug logs and detection flags for monitoring performance.

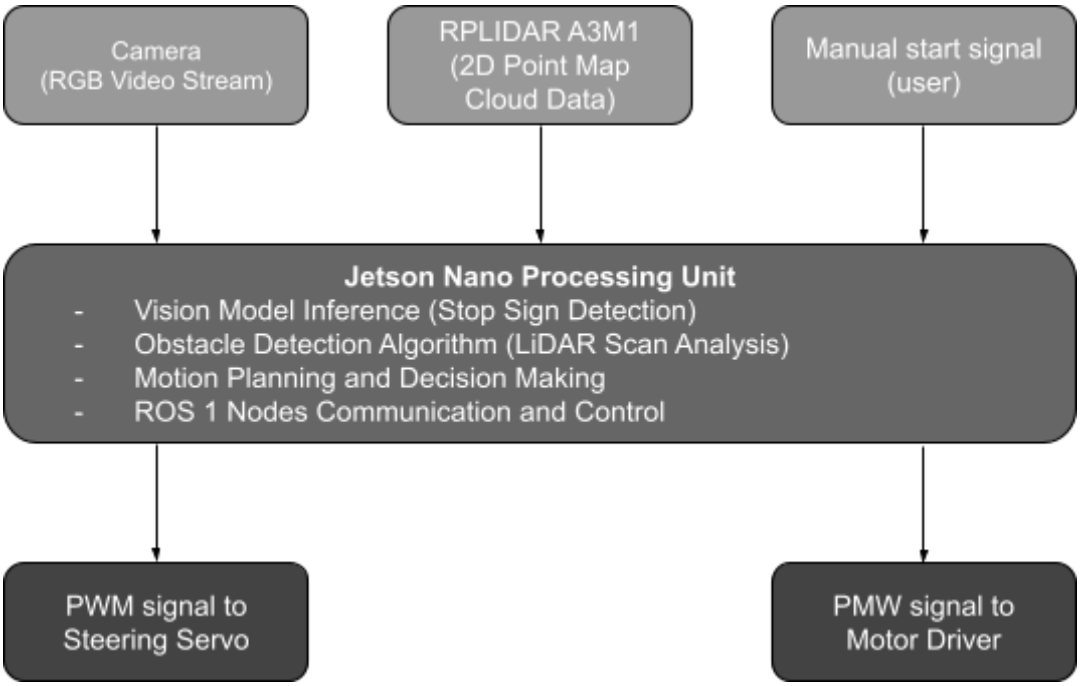


Figure 1. System architecture

This modular architecture promotes robust system integration, scalability, and ease of debugging throughout the development process.

8) Project Planning

The project was structured using a Gantt Chart that divides development into seven progressive phases, each assigned to specific team members with clear deliverables and deadlines. This planning ensured a systematic workflow, optimized resource use, and alignment with the project's technical and educational goals.

Project Phases

1. Project Definition and Planning
   Selection of the use case, justification of objectives, and creation of the roadmap.

2. Conceptual and Detailed Design
   Architecture design, subsystem specifications, and selection of components.

3. Hardware Integration
   Mechanical assembly, sensor mounting, and circuit design.

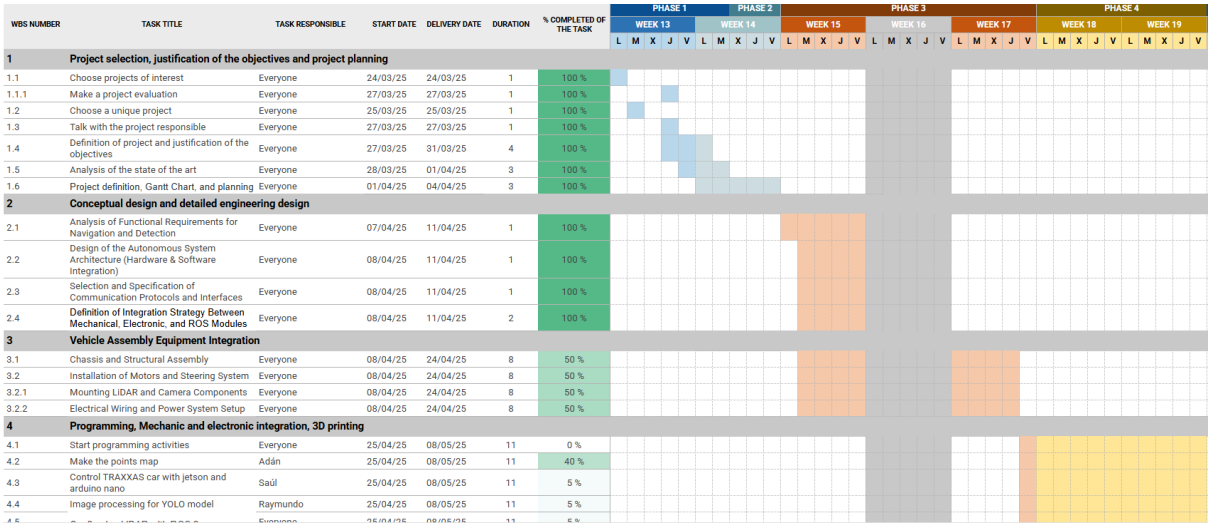
4. Software and Electronics Integration
   ROS node development, sensor communication, 3D printing of parts, and PID tuning.

5. Simulation and Algorithm Testing
   Testing detection and control algorithms in a virtual ROS environment with synthetic STOP signs and obstacles.

6. Physical Testing
   Iterative real-world trials for system validation, metric logging, and behavior tuning.

7. Final Prototype and Documentation
   Assembly of the final vehicle version, performance demonstration, and submission of the technical report.

This planning structure supported a modular, traceable, and validation-driven development process, which was key to meeting competition challenges and competency standards.



WBS NUMBER	TASK TITLE	TASK RESPONSIBLE	START DATE	DELIVERY DATE	DURATION	% COMPLETED OF THE TASK
1	**Project selection, justification of the objectives and project planning**					
1.1	Choose projects of interest	Everyone	24/03/25	24/03/25	1	100 %
1.1.1	Make a project evaluation	Everyone	27/03/25	27/03/25	1	100 %
1.2	Choose a unique project	Everyone	25/03/25	25/03/25	1	100 %
1.3	Talk with the project responsible	Everyone	27/03/25	27/03/25	1	100 %
1.4	Definition of project and justification of the objectives	Everyone	27/03/25	31/03/25	4	100 %
1.5	Analysis of the state of the art	Everyone	28/03/25	01/04/25	3	100 %
1.6	Project definition, Gantt Chart, and planning	Everyone	01/04/25	04/04/25	3	100 %
2	**Conceptual design and detailed engineering design**					
2.1	Analysis of Functional Requirements for Navigation and Detection	Everyone	07/04/25	11/04/25	1	100 %
2.2	Design of the Autonomous System Architecture (Hardware & Software Integration)	Everyone	08/04/25	11/04/25	1	100 %
2.3	Selection and Specification of Communication Protocols and Interfaces	Everyone	08/04/25	11/04/25	1	100 %
2.4	**Definition of Integration Strategy Between Mechanical, Electronic, and ROS Modules**	Everyone	08/04/25	11/04/25	2	100 %
3	**Vehicle Assembly Equipment Integration**					
3.1	Chassis and Structural Assembly	Everyone	08/04/25	24/04/25	8	50 %
3.2	Installation of Motors and Steering System	Everyone	08/04/25	24/04/25	8	50 %
3.2.1	Mounting LiDAR and Camera Components	Everyone	08/04/25	24/04/25	8	50 %
3.2.2	Electrical Wiring and Power System Setup	Everyone	08/04/25	24/04/25	8	50 %
4	**Programming, Mechanic and electronic integration, 3D printing**					
4.1	Start programming activities	Everyone	25/04/25	08/05/25	11	0 %
4.2	Make the points map	Adán	25/04/25	08/05/25	11	40 %
4.3	Control TRAXXAS car with jetson and arduino nano	Saúl	25/04/25	08/05/25	11	5 %
4.4	Image processing for YOLO model	Raymundo	25/04/25	08/05/25	11	5 %

Figure 2. Gantt Chart

Link to access the Gantt Chart: Diagrama de Gantt

9) Conceptual Design

The conceptual design of the autonomous vehicle focuses on establishing the overall structure and main components needed to perform lane following, stop sign

recognition, and static obstacle avoidance within a controlled indoor environment. The approach prioritizes modularity and integration through the ROS framework, balancing performance, scalability, and ease of implementation.

The system is organized into three core functional blocks: perception, decision-making, and actuation.

- Perception: An RGB camera and a 2D LiDAR sensor capture environmental data. The camera processes real-time images to detect stop signs using color filtering and contour analysis, enhanced by a deep learning detection model (YOLOv5) to improve accuracy. Meanwhile, the LiDAR identifies and locates static obstacles near the vehicle's path.

- Decision-Making: The gathered data is processed to determine the appropriate actions. The lane-following algorithm extracts the position of the yellow line on the right using HSV thresholding and Hough Line Transform. When a stop sign or obstacle is detected, the system halts or performs evasive maneuvers. A PID controller calculates steering adjustments to maintain stable and precise driving.

- Actuation: Control commands are converted into PWM signals that regulate the motor and steering servo. Speed limits and safe stop mechanisms are implemented to ensure reliable operation within the testing environment.

The main components and their interfaces are summarized below:

Component	Function	Interface
Traxxas Chassis	Mechanical base and movement	PWM control for motor and servo
Jetson Xavier NX	Processing and control	ROS nodes for sensors and actuators
USB RGB Camera	Visual data capture	USB interface, OpenCV processing
RPLIDAR A3M1	Obstacle detection	Serial communication, ROS driver
PID Controller	Steering regulation	ROS node, PWM output

Table 2. Components

This design considers key factors such as modularity to facilitate development and incremental integration; real-time operation to respond appropriately to

environmental stimuli; scalability for future feature additions; and safety by enforcing speed limits and automatic stops in emergency situations.

10) Analysis of the project's ethical dilemmas

Even though this is a prototype, there are important ethical points to keep in mind when developing autonomous systems.

Safety and Responsibility

Since the vehicle runs in a controlled environment, we made sure it detects obstacles, stops at signs, and can safely stop if something goes wrong. Tests were always supervised to avoid accidents.

Transparency

The software was built in a modular way with clear documentation so others can understand how it works and verify the results.

Data Privacy

The sensors collect only the information needed for navigation and analysis. No personal data is recorded or shared.

Bias and Reliability

We tested the system in different conditions to make sure it works well in various situations, avoiding any bias from the training data.

Academic Integrity

All tools and references were properly cited, and we followed academic honesty rules throughout the project.

11) Detailed engineering

a. Norms and standards

To ensure safety, quality, and reliability in mechatronic systems, it is essential to consider technical standards that apply both generally and specifically, depending on the type of application. In this project, we use hardware and software components typical of prototype-level autonomous vehicles, which implies compliance with certain standards.

General standards

These standards are relevant to systems that integrate mechanics, electronics, and software, such as this autonomous vehicle:

- ISO 12100:2010 – Safety of machinery – General principles for design. Establishes design principles to reduce risks in mechanical systems.

- IEC 60204-1 – Safety of machinery – Electrical equipment of machines – Part 1: General requirements. Applies to the installation of electrical components such as motors and controllers.

- IEEE 830-1998 – Recommended Practice for Software Requirements Specifications. Recommends how to clearly and effectively document software requirements [5].

- ISO/IEC 25010:2011 – System and software quality models. Helps evaluate the quality of embedded software systems.

Specific Standards

For prototype-scale autonomous vehicles like the Traxxas and the use of frameworks such as ROS, the following standards are important:

- ISO 26262:2018 – Road vehicles – Functional safety. This standard focuses on the functional safety of electrical and electronic systems in road vehicles. Although our project is a prototype, its principles of safe design are applicable in academic and research contexts.

- IEEE 1872-2015 – Standard Ontologies for Robotics and Automation. Defines key concepts and ontologies so that robotic systems, including those based on ROS, can be interoperable and standardized [6].

- REP-2000 (ROS Enhancement Proposal). Specification aimed at standardizing the structure of packages and nodes in ROS, promoted by the ROS developer community.
- ISO 8373:2012 – Robots and robotic devices – Vocabulary. Defines clear terminology for describing robotic devices, their components, and capabilities.

Standards Related to Sensors and Electronic Components

Since the system includes sensors like LiDAR, cameras, and microcontrollers, it is advisable to consider the following standards:

- IEC 60825-1 – Safety of laser products. Directly applies to the use of LiDAR sensors, ensuring the device operates within safe limits.
- JEDEC JESD22-A104 – Thermal shock testing for electronic components. Useful for validating the resilience of devices such as the Jetson Nano under real-world operating conditions.
- IEEE 1451 – Smart Transducer Interface Standards. Applies to the communication between sensors and microcontrollers, recommending standardized formats for data acquisition.

b. Plans and diagrams

Technical drawings

Schematic design:
Represents how the connection must be between the Arduino Nano and a pair of three pin connectors in the PCB.

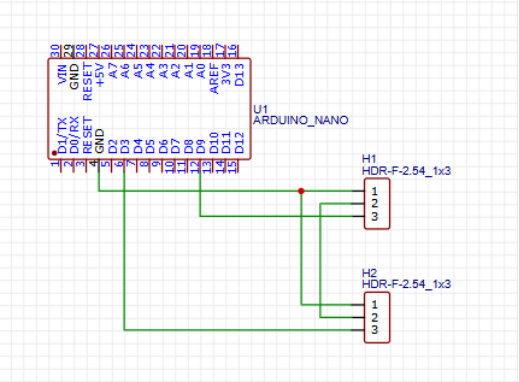


Figure 3. PCB schematic design

PCB design:

A centralized distribution PCB is proposed that integrates connectors for servo motors.

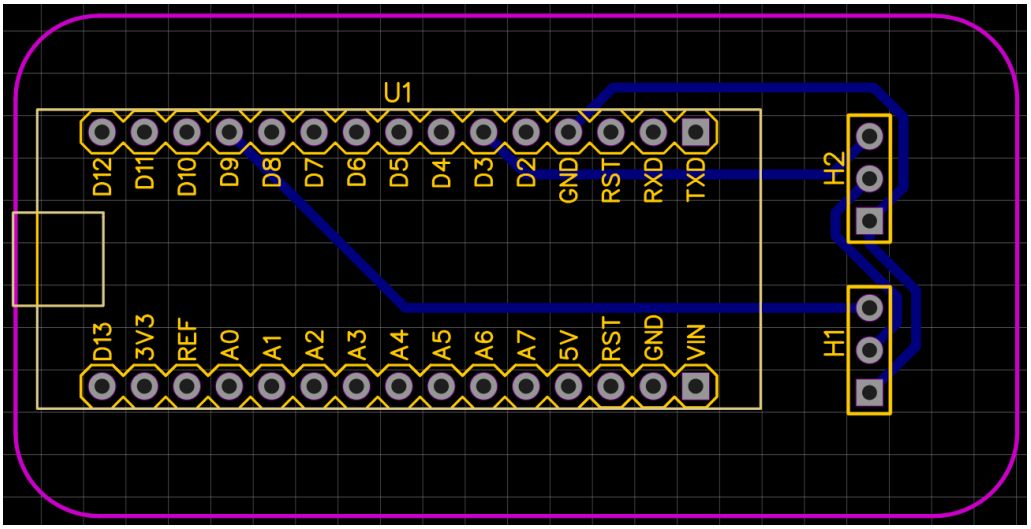


Figure 4. PCB design

Technical drawings of the chassis components:

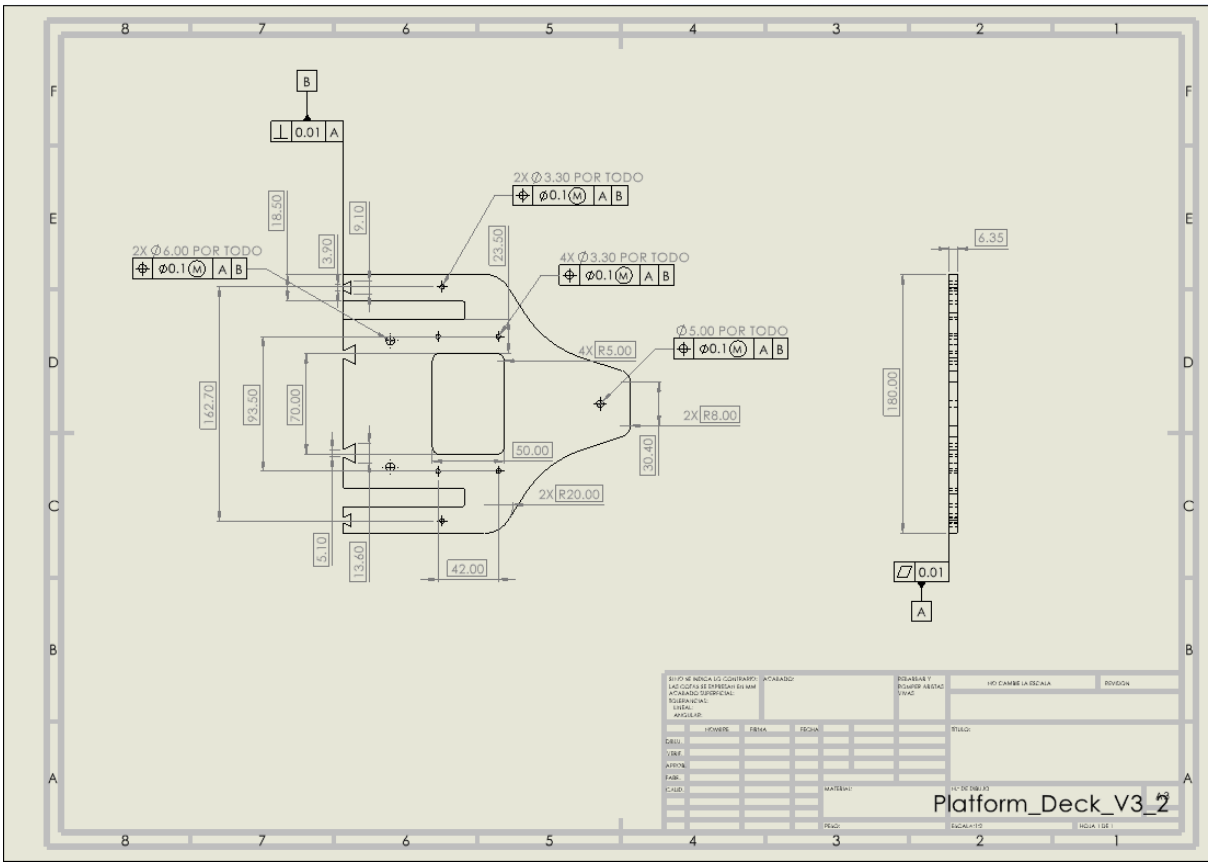


Figure 5. Platform deck female technical drawing

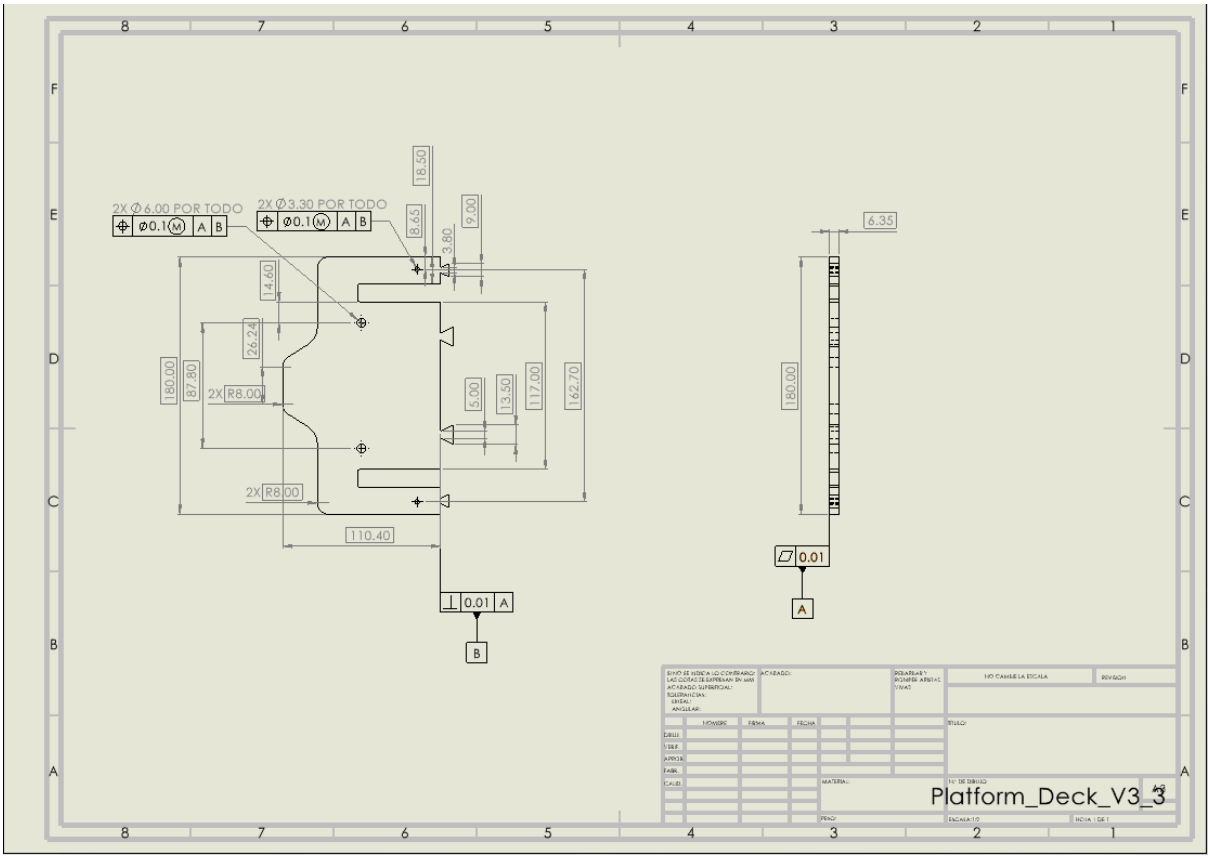


Figure 6. Platform deck male technical drawing

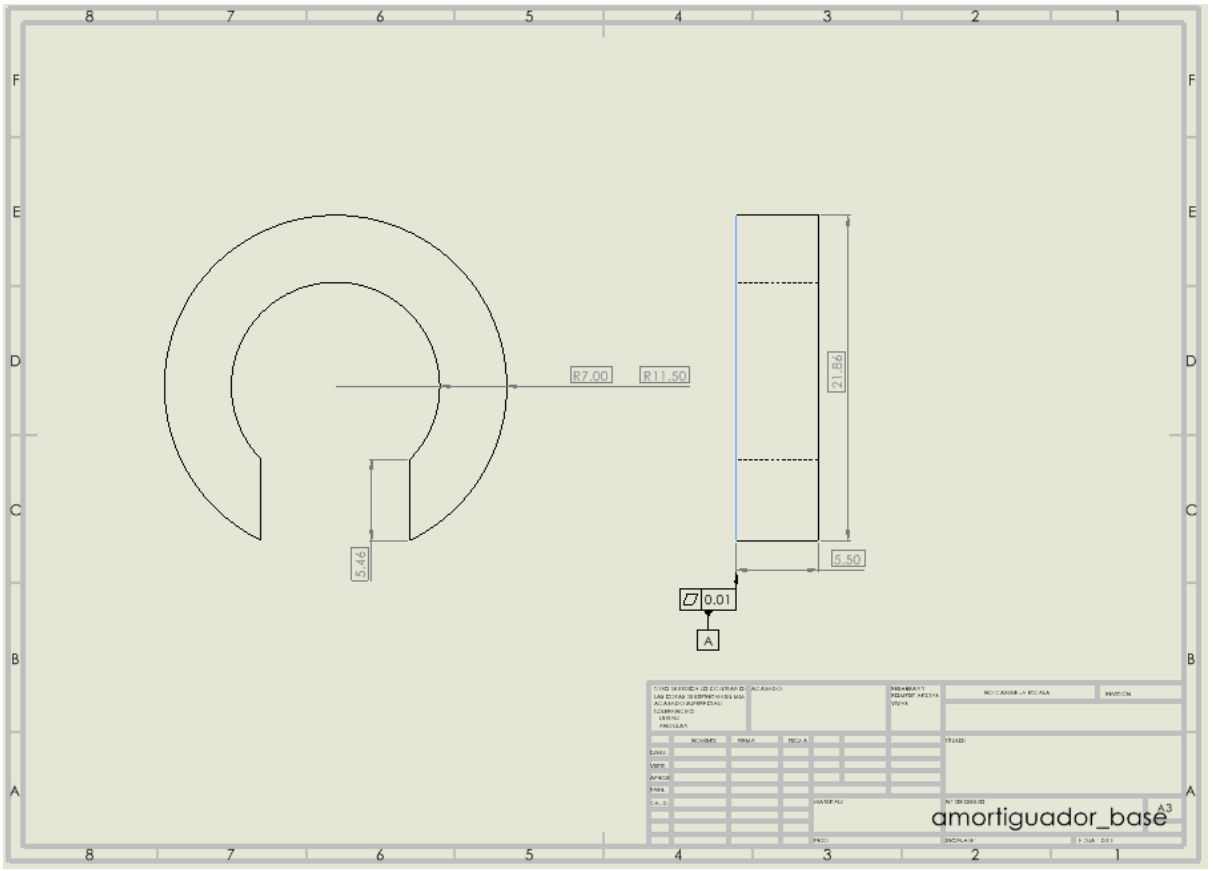


Figure 7. Spring base technical drawing

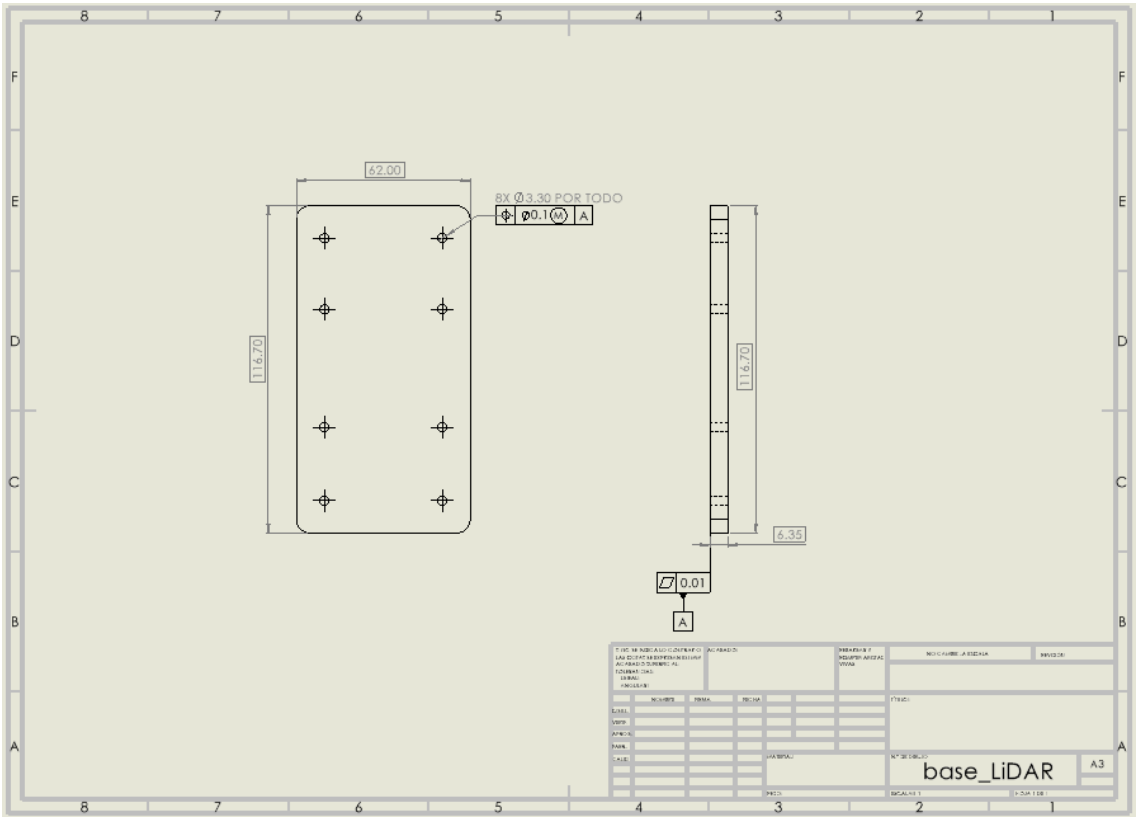


Figure 8. LIDAR base technical drawing

c. Tolerances, GDT, and quality

To ensure precision and functionality in the mechanical assembly of the prototype, the following tolerance and GD&T principles are applied:

- Dimensional tolerance: ±0.1 mm on 3D-printed components to ensure proper fit of sensors and servos.
- Clearance fit between the steering axles and mechanical parts to avoid interference.
- Straightness and parallelism are guaranteed on the chassis platform to maintain the stability of the LiDAR sensor.
- Quality control: Visual inspections, electrical continuity tests, and functional tests of the mounted sensors will be performed

d. Materials and components

The components and materials used for building the Traxxas are the following ones.

Materials:

- Traxxas RC car chassis
- PLA
- M6 and M3 bolts and nuts

Components:

- Jetson Xavier
- RPLIDAR A3M1
- Logitech Webcam C922
- Arduino Nano
- Li-Po Battery 11.1V/5200mAh
- Traxxas ESC XL-5 Waterproof
- Traxxas NiMH battery
- Traxxas Motor Titan 550 12T
- Traxxas High-Torque Waterproof Servo

This list of components includes the ones strictly needed to build and assemble the Traxxas vehicle. However the team only acquired the materials and components that had not been already purchased by the mechatronics department. This will be covered in more detail in the next section.

e. Procurement

For the procurement process, a Bill of Materials was necessary in order to have the total cost of the project if the department were to acquire some components again. The only component that required a purchase process was the LiPo battery and it was carried out by the professors of the mechatronics lab. The team only made the quotation of the battery with the store and it was sent via email to the mechatronics lab professors who later completed the purchase.

In order to represent the cost of the projects a purchase list was created:

Quotations and purchase list

Bill of Materials (BOM) is as follows:

Component	Supplier/Reference	Unit Price	Quantity	Total
Jetson Xavier	NVIDIA	$149	1	$149
Logitech Webcam C922	Logitech	~$138	1	$138

RPLIDAR A3M1	SLAMTEC	~$400	1	$400
Li-Po Battery (11.1V, 5200mAh)	Electronic Store “...”	$70	1	$70
PLA	Amazon	$20	1	$20
Miscellaneous (Cables, Mounts, Fasteners)	Local Supplier	$50	-	$50
Theoretical cost of the project				$827

Table 3. Bill of materials

f. Feasibility and ROI

Feasibility Analysis.

Technical Feasibility: The project leverages mature and widely used components in robotics research, such as the Jetson Nano, RPLiDAR, and computer vision algorithms like YOLO. These technologies are well-documented, with abundant open-source support and ROS (Robot Operating System) integration. The platform is capable of executing real-time perception and control tasks, making it technically viable for applications such as autonomous navigation and traffic sign recognition.

Operational Feasibility: The proposed system has been partially implemented with functional subsystems: traffic sign detection using YOLO, object avoidance using LiDAR data in RViz, and communication between Jetson and Arduino via serial for PWM control. The team has demonstrated the ability to integrate and test these components, indicating a strong operational foundation. With continued testing and tuning, full system integration is achievable.

Schedule Feasibility: Given the project timeline of ten weeks, the tasks have been adequately divided and assigned to team members. With focused sprints and time management, final system testing and refinement are feasible within the remaining time frame.

Risk Assessment: Risks such as hardware failures, sensor noise, or integration bugs exist, but they are mitigated by the modular approach taken, which allows testing and validation of each subsystem independently. The use of simulation environments (e.g., RViz and Gazebo) can also serve as backups for testing in case of physical limitations.

Return On Investment (Educational Perspective).

Instead of traditional financial ROI, this project delivers value through **educational and developmental returns**, which are critical for student growth in STEM fields.

Knowledge Acquisition: Students gain hands-on experience with embedded systems, sensor integration, robotics middleware (ROS), real-time control, and AI-based perception—skills directly relevant to industries such as autonomous vehicles, automation, and robotics.

Skill Development: The project fosters critical thinking, problem-solving, teamwork, and technical communication. Students apply theoretical concepts in control systems, computer vision, and mechatronics to real-world challenges.

Portfolio and Career Impact: Successfully executing and documenting this project can significantly enhance students' professional profiles. It can serve as a capstone or research showcase in resumes, grad school applications, or job interviews.

Academic Contributions: The outcomes of the project can be turned into technical reports, conference presentations, or even journal submissions, contributing to the body of knowledge in robotics education and autonomous systems.

Motivation and Engagement: By working on a tangible, high-impact project, students remain engaged and motivated. The project's challenge level and potential for real-world applications increase satisfaction and learning retention.

12) Implementation

The implementation of the project has been divided into three areas which describe key aspects of the project. Here is an explanation of each category.

a. Mechanics

The mechanical part of the project includes the design and manufacturing of the platform deck used to hold the Jetson Xavier in the chassis; the platform deck is based on the roborace car page which contains the specifications for designing an F1TENTH deck suitable for the competition.
Because of the size of the Bambu printers in the lab, our deck had to be divided into two parts so the 3D printing of the part could happen. For the design of this joint, a Milano tail shape was chosen leaving both parts as follows.

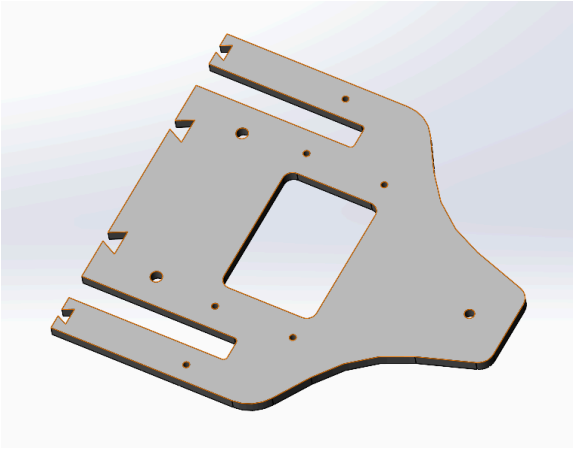


Figure 9. Platform design female

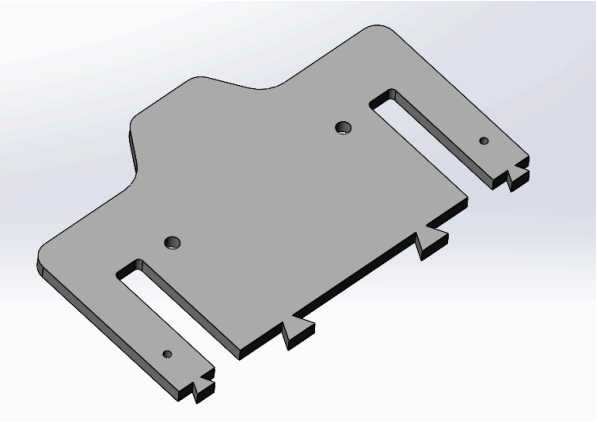


Figure 10. Platform design male

The complete assembly of both parts should look like the following:

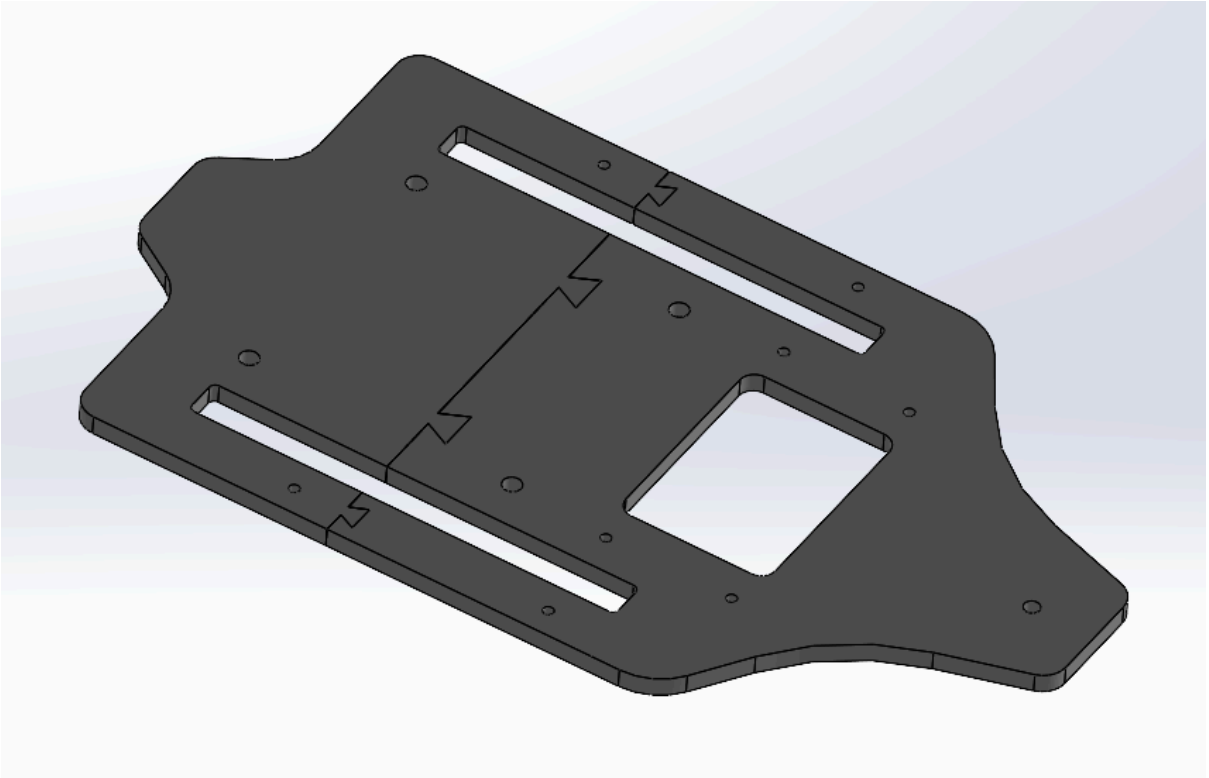


Figure 11. Platform assembly

This assembly considered all the GD&T tolerances and quality terms and was then fixed to the Traxxas chassis so the Jetson Xavier could be attached to it.

Then, a base was needed in order to keep the LIDAR in place and so that it won't interfere with its detection range.
In this case, the part could be printed as a whole. At first holes were made to attach it to standoffs and to the LIDAR body, but then the equipment showed malfunction when screwed to the base so the use of a double sided tape was needed. Nevertheless, the base fulfills its function.

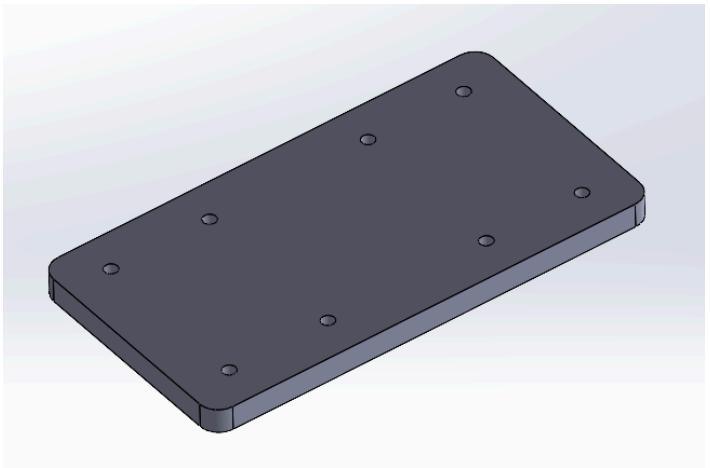


Figure 12. LIDAR base

The final mechanical component designed was a base for the springs of the car. The bases attached to the suspension system of the LIDAR were the ones designed by previous users of the car, so with time the bases started to rip apart.
That is why the team designed a new base that looks as the following:

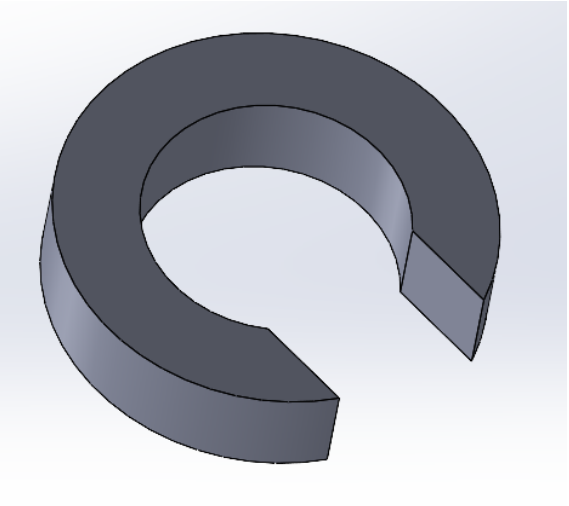


Figure 13. Spring base

With this base the car no longer has the problem of a soft suspension but instead has a stiff suspension now, as a racecar should have.
The following image shows how all the mechanical components were attached to the chassis of the car accomplishing its GD&T and quality parameters.

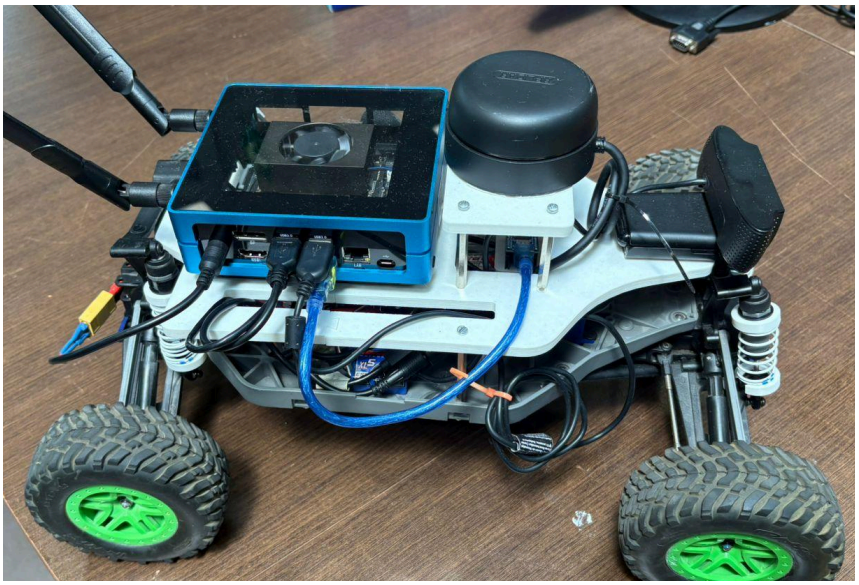


Figure 14. Traxxas chassis with all the components

b. Electrical/Electronical

The electronical part of the project comprehends the design and manufacture of a PCB to connect the Arduino NANO to the Jetson Xavier and to the motor ESC and servos.
The PCB was created using the schematic and PCB design described in the Plans and Diagrams section of this document, resulting in the following component:

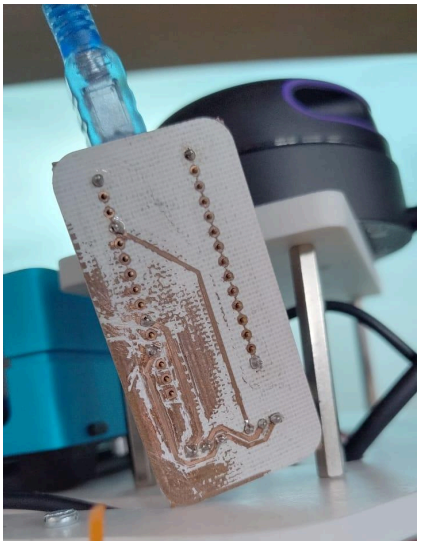


Figure 15. PCB in Traxxas

In this PCB the Arduino NANO was placed looking as follows:

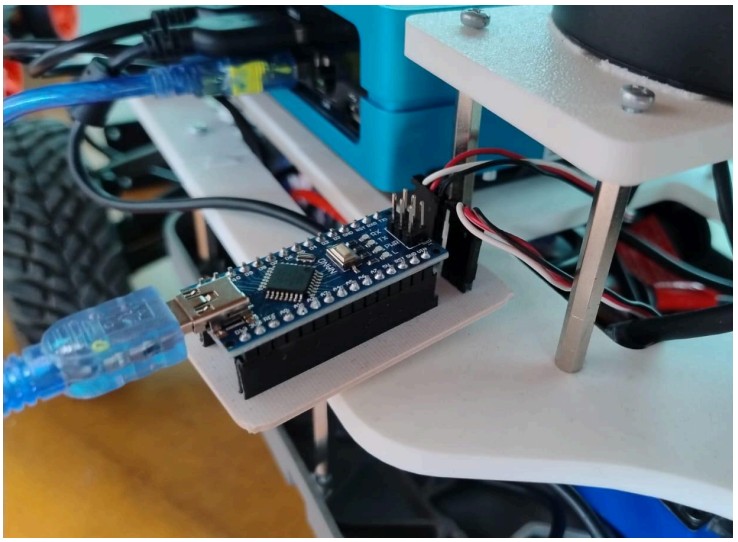


Figure 16. PCB in Traxxas 2

With this implementation, the pins of the Arduino are connected to the pin connectors directly via the tracks of the PCB enabling a clean integration of the

components without the use of several cables to connect the servo-like connectors to the Arduino.

c. Software

We used Ubuntu 18.0.4, an open-source Linux distribution. For programming, we used Python 2.7, in which we created a line-follower code that only detects the yellow color, creates a centroid at a certain distance from the yellow line, and uses a basic PID controller with a 0.6 proportional gain, a 0.0001 integral gain and a 0.1 derivative gain, to move the servos so they follow the correct path.

Similarly, we also support the ROS (Robot Operating System), an open-source framework that provides tools and libraries for developing robotic applications. It facilitates the construction of robotic systems by offering hardware abstraction, device control, package management, and message communication.

The ROS topics used connected the Webcam data, the LIDAR data and the serial communication with the Arduino so it could be possible that when the Python code executes, all the publishers and subscribers manage the data giving the implementation a real-time response when the camera detects the lane or the LIDAR detects there is an obstacle on its way.

13) Experimental protocol and test

The autonomous vehicle was tested in a controlled indoor track designed with two parallel yellow lines marking the boundaries of the lane. The vehicle was programmed to follow only the right lane line, maintaining a fixed offset to stay centered. A single red stop sign and one static obstacle were placed strategically along the path to evaluate object detection and avoidance capabilities.

Figure 17 . Mechatronics Lab Track

Test Setup

The testing environment was set up in the Mechatronics Laboratory. The vehicle used a USB camera for lane and stop sign detection, and a LiDAR sensor for obstacle detection. ROS was used for system integration and control.

Procedure

1. The car was positioned at the starting line and activated via ROS.

2. It followed the right-hand yellow lane using a PID controller.

3. Upon detecting the red stop sign, it was expected to stop completely.

4. If an obstacle was detected within a one-meter range, the car would initiate an avoidance maneuver.

5. The run ended when the vehicle completed a full loop or intervention was required.

Obstacle Avoidance Routine

The obstacle avoidance logic was implemented through a sequence of reversing and turning maneuvers. Various steering angles and time durations were tested iteratively to find the combination that allowed smooth navigation around obstacles without losing track alignment. The final routine reversed, paused, and turned at

angles of -45° and +40° with tuned delays, allowing the vehicle to bypass the obstacle and return to its lane.

Performance Metrics

- Average speed during tests was approximately 0.2 m/s.
- Stop sign detection had a high success rate, thanks to HSV-based filtering and contour detection.
- The obstacle avoidance routine triggered reliably when the LiDAR detected close-range objects.
- Lane tracking was visually verified to remain consistent, with minor deviations corrected by the PID controller.

Final Code

The final implementation combined all behaviors into a single ROS node, handling image processing, obstacle detection, and control commands. The code used real-time image cropping, color filtering, and Hough line detection to locate the right boundary line, while continuously publishing steering and velocity values.

This testing protocol allowed the system to be validated under realistic academic conditions, demonstrating reliable lane following, stop sign response, and reactive obstacle avoidance.

14) Results

The system was tested in three main scenarios: line following, stop sign detection, and obstacle avoidance. The car successfully maintained its trajectory using the right yellow line with an offset, recognized the stop sign reliably, and avoided obstacles using a predefined maneuver. The average speed reached was approximately 0.5 m/s, exceeding the initial goal of 0.2 m/s.

Figure 18. Traxxas vehicle as final product

Figure 19. Lane following action

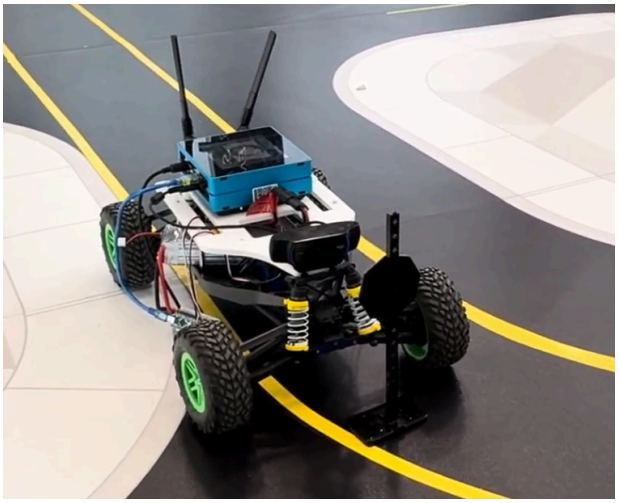


Figure 20. Stop sign detection

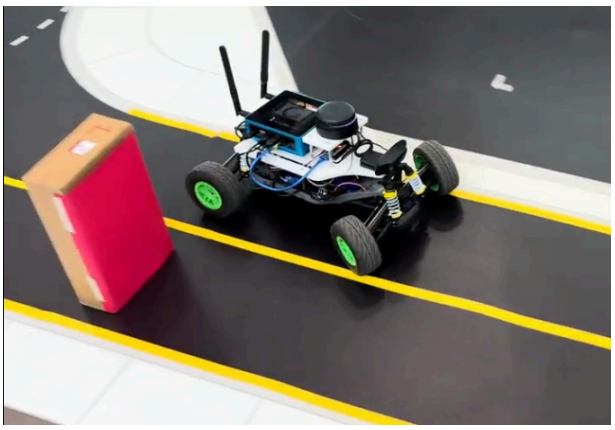


Figure 21. Traxxas performing an evasion maneuver.

The following link contains all the videos regarding the actions described in Figures 19-21 as well as some footage of early stages of the project and more tests videos.
Link:
https://drive.google.com/drive/folders/1sZb3JNdUWP4B6KkjTr9l9Md0qb00Fcjb?usp=sharing

As shown below, all functions performed reliably under test conditions:

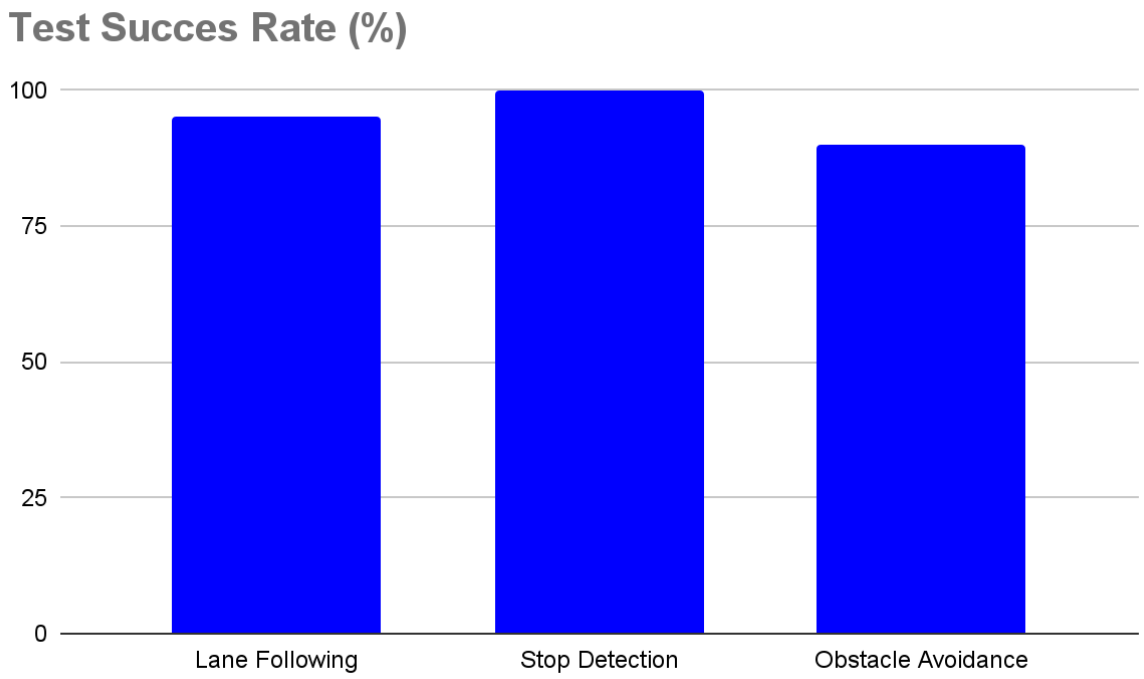


Figure 22. Test Success Rate

15) Conclusions

Adan:
This project represented a comprehensive learning experience, where we applied knowledge of mechanics, electronics, programming, and control to a functional system. Despite the technical challenges and limited time, we were able to develop an autonomous prototype capable of following lanes, recognizing stop signs, and avoiding obstacles. The use of ROS and modern technologies such as LiDAR strengthened our skills in industry-relevant tools. This work helped us with teamwork, problem-solving, and ethical responsibility. We left with a solid foundation for future autonomous robotics projects and valuable preparation for the professional environment.

Saul:
The development of the autonomous vehicle based on the Traxxas platform represented a comprehensive learning experience in the fields of robotics, control systems, computer vision, and embedded systems. Overall, the project not only met its technical objectives but also laid the groundwork for future enhancements and applications in autonomous systems. The knowledge gained can be extended to larger-scale platforms or adapted for other use cases such as search and rescue, delivery robotics, or smart transportation systems. This final result is a testament to interdisciplinary collaboration, continuous improvement, and the practical application of engineering principles in the context of autonomous vehicle development.

Raymundo:
The development of this project led to a functional autonomous vehicle capable of following a lane, detecting stop signs, and avoiding obstacles. By integrating perception, control, and actuation modules, and validating the system through real-world tests, we demonstrated a solid understanding of mechatronic design and automation. The approach was supported by ethical considerations and a review of current technologies, ensuring the solution was both responsible and technically sound.

16) References

[1] Atlassian. (s/f). *Qué es scrum y cómo empezar*. Atlassian.com. Recuperado el 4 de abril de 2025, de https://www.atlassian.com/es/agile/scrum

[2] Martins, J. (2025, enero 19). *¿Qué es la metodología Kanban y cómo funciona?* Asana. https://asana.com/es/resources/what-is-kanban

[3] Evans, B. D., Trumpp, R., Caccamo, M., Jahncke, F., Betz, J., Jordaan, H. W., & Engelbrecht, H. A. (2024). Unifying f1tenth autonomous racing: Survey, methods and benchmarks. arXiv preprint arXiv:2402.18558.

[4] O'Kelly, M., Zheng, H., Karthik, D., & Mangharam, R. (2020). F1tenth: An open-source evaluation environment for continuous control and reinforcement learning. Proceedings of Machine Learning Research, 123.

[5] IEEE Standards Association. (1998). IEEE 830-1998: IEEE Recommended Practice for Software Requirements Specifications. New York, NY: IEEE. https://doi.org/10.1109/IEEESTD.1998.88286

[6] IEEE Standards Association. (2015). IEEE 1872-2015: IEEE Standard Ontologies for Robotics and Automation. New York, NY: IEEE. https://doi.org/10.1109/IEEESTD.2015.7084073

Appendix 1: Further work

1. Multiple traffic sign detection: Expand the vision system to recognize other signs such as “Curve,” “Speed Limit,” or “No Entry,” using models trained on larger datasets.
2. Improved evasion system: Implement a more advanced evasion algorithm to avoid obstacles more smoothly and efficiently.
3. Integrating a Localization and Mapping (SLAM) system: Using LIDAR and cameras to build maps of the environment and enable more autonomous and robust navigation in new spaces.
4. Migration to ROS 2 and Python 3: Upgrade the system to more modern versions of the ROS ecosystem, which will improve compatibility, security, and long-term maintainability.
5. Uncontrolled Environment Testing: Evaluate vehicle performance in semi-structured or variable lighting environments to validate its robustness under real-world conditions.

Appendix 2: Technical memory for further work and developers

1. Code Structure

- catkin_ws/src/line_follower/scripts/pruebaadan.py

Main script that integrates line following, obstacle avoidance, and stop sign detection.

- catkin_ws/src/lidar_tools/scripts/lidar.py

Script that publishes lidar information and triggers avoidance routines upon detecting nearby objects.

- arduinotraxxas/traxxas.ino

Code loaded into the Arduino Nano that receives speed and direction commands via serial.

2. System Dependencies

- Ubuntu 18.0.4
- ROS Melodic
- Python 2.7
- OpenCV 3.4
- RPLIDAR ROS Driver

3. Recommendations for New Developers

Test all scripts individually before integrating them.

Document any changes to the evasion or vision logic.

Verify correct USB port mapping if changing Jetson or environments.

Continue using start.sh to launch all necessary nodes from a single command.

4. Hardware Considerations

Ensure secure mounting of the LIDAR and camera to avoid erratic readings.

Check that the ESC is properly calibrated with the servo motor.

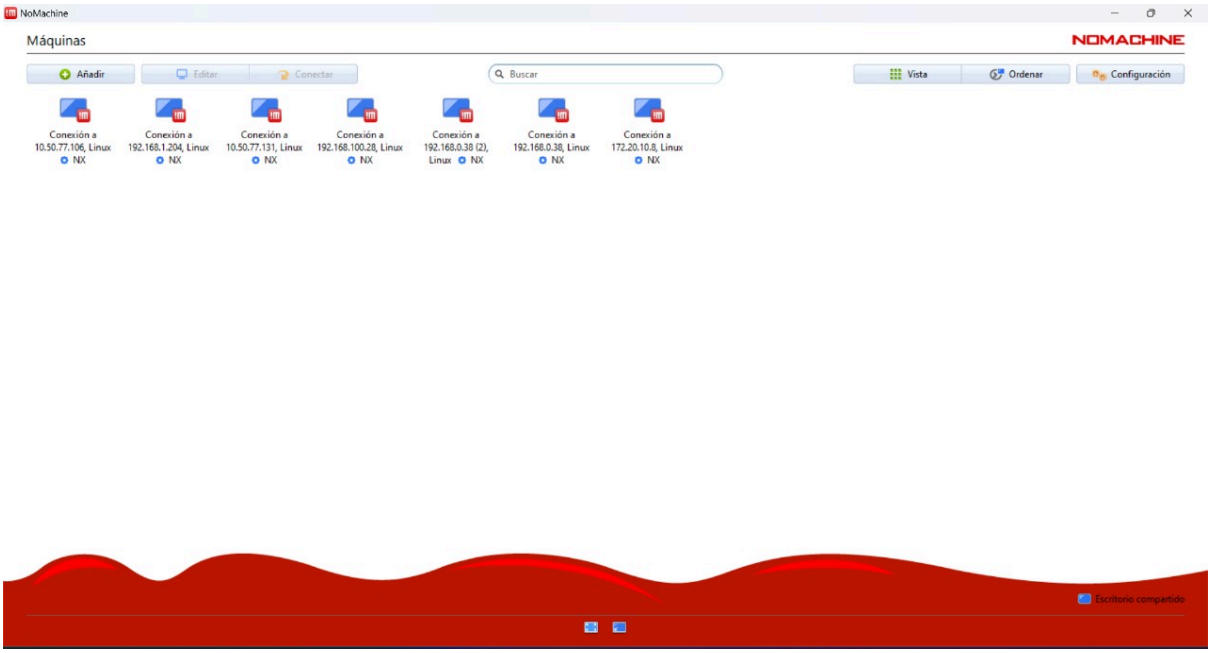
Confirm that the LiPo battery is fully charged and balanced before each use.

Appendix 3: User manual

The first thing you need is to start a virtual machine to be able to see the Jetson screen, the team used the "NoMachine" program since it is free and is already installed on the Jetson, so they only need to connect to the following IP: 10.50.77.106

It could also be connected to a separate screen since the Jetson has HDMI, the problem is that the fully assembled car already occupies the 4 USB ports, 2 ports used to connect and power the Lidar, one port for the camera and another for the Arduino, so you would need to disconnect 2 ports to connect a keyboard and mouse.

If you use NoMachine to connect, it should look something like this:

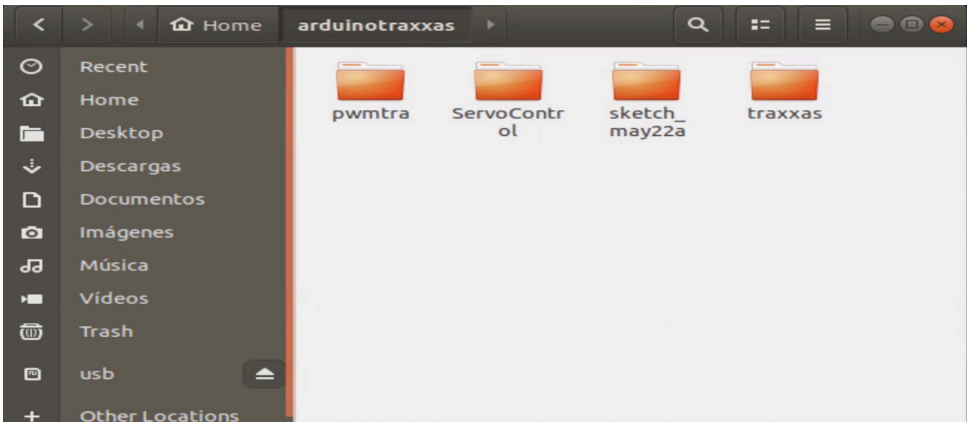


In this case, we have more than one connection, but you should only care about the first one. If you have any other questions, the following link is direct to an official Jetson website: https://www.waveshare.com/wiki/JetRacer_ROS_AI_Kit_Tutorial_II:_Install_Jetson_Nano_Image

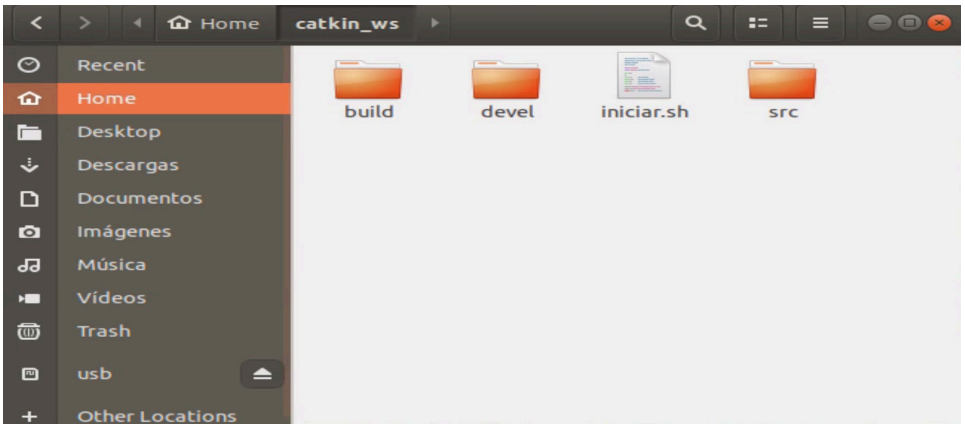
The Jetson runs on Linux, with the Ubuntu operating system, the version is 18.0.4 and all codes were programmed with Python 2.7.

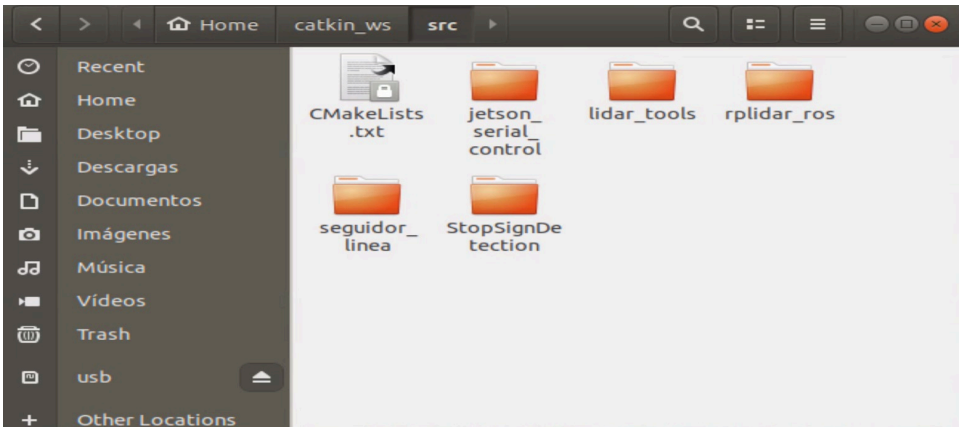
The first thing you need to do once you're connected to the Jetson is go to the files. There you'll find several folders, but the only two that were used are the folders called "arduinotraxxas" and "catkin_ws." The “arduinotraxxas” folder contains the Arduino codes that we need to control the motor and servos, as well as create the

publisher for controlling them. In this case, the Arduino file we ultimately used is located inside the "traxxas" folder.

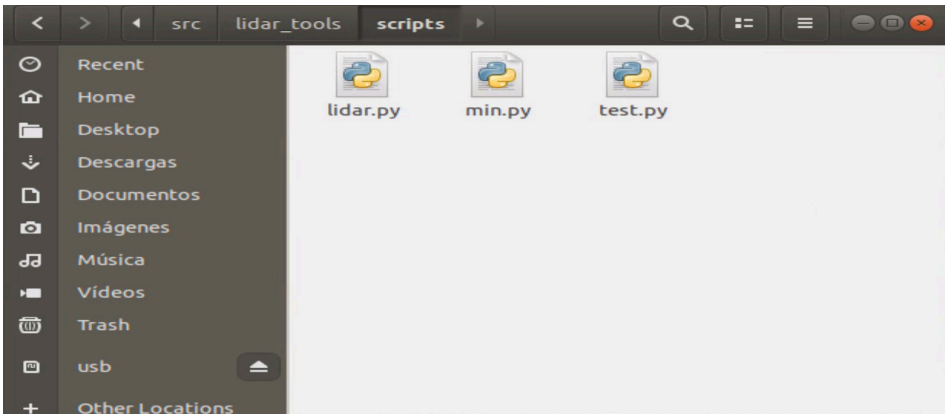


In the "catkin_ws" folder is all the important information, inside there are folders where libraries were installed for the use of the Lidar, the camera and the codes that were used for the Traxxas to work.

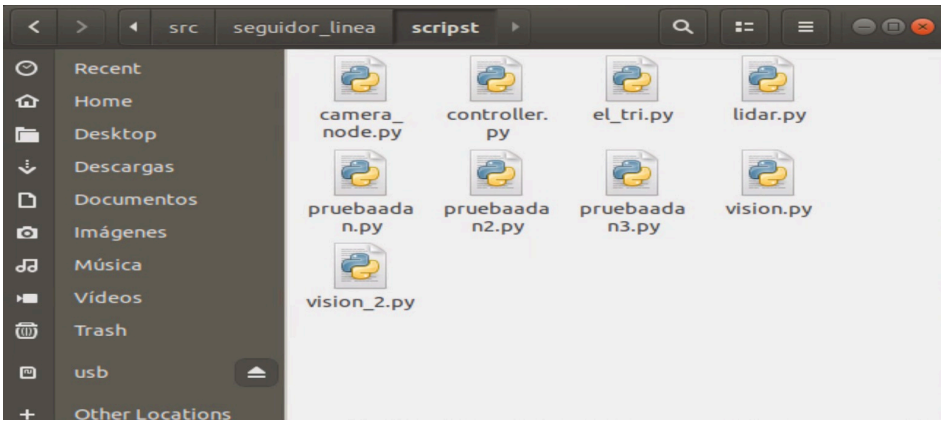




Our main codes are located inside two folders, the “lidar_tools” folder has the detection code towards the front of the Lidar, it also creates the lidar publisher so that the main code subscribes and depending on the information received performs the corresponding action, the necessary code is called "lidar.py”.



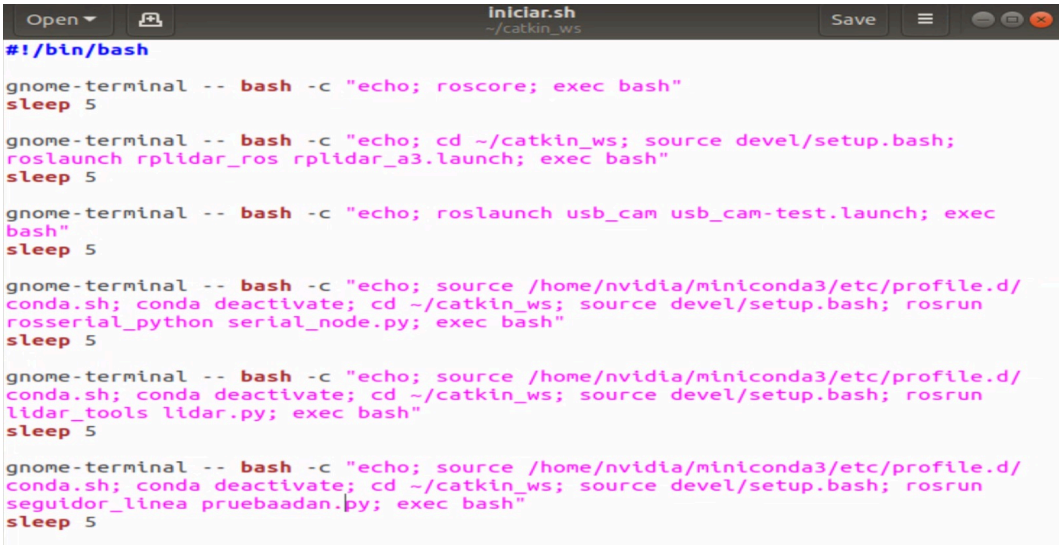
And finally, our main code is located inside the "line_follower" folder. Inside there, open another folder called "scripts" and there you can see some of the codes that we used during the project, but the important code is called "pruebaadan.py" with this code and the one mentioned above you can now use the Traxxas.



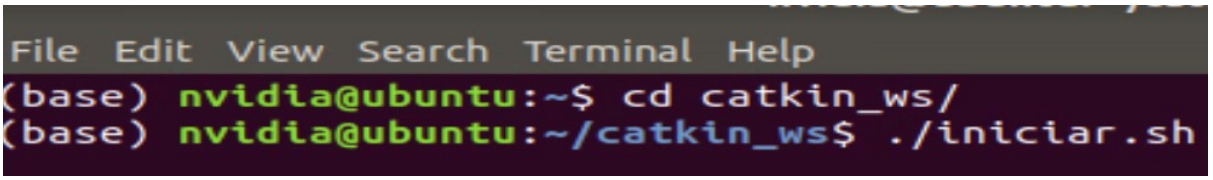
First, you must have the Traxxas properly assembled. The lithium battery powers the Jetson. The batteries are connected to the ESC controller, which you must turn on by pressing it until the light flashes. Make sure to upload the Arduino code in case the code was previously changed or if the board is different, in this case ours is an Arduino Nano. You must also have the four USB ports connected for the lidar, the camera, and the Arduino. Now, open a terminal and, inside it, the "catkin_ws" folder. You can do this with the command:

- cd catkin_ws/

Once inside that folder, you will see that there is a file called "iniciar.sh" this file is to open all the terminals necessary for the Traxxas to work, if you want to see which terminals we open and the commands that are executed you can open the file and you will be able to see the commands used.



Now all you need to do is run that file, and it will automatically open the 6 terminals needed for the cart to work properly.



Basically, those are all the steps you need to know and follow to get the Traxxas up and running. If you have any questions, you can also look in the document where we explain how it works in more detail.