NTNU
Kunnskap for en bedre verden

Department of Mechanical and Industrial Engineering

TMM4150 - Machine Design and Mechatronics

Design and Construction of Battlebot for Competition

Group 6

28th November 2025

Deliverable for **TMM4150 - Machine Design and Mechatronics**		
Report Title: Design and Construction of Battle for Competition		
Report No.: 1	**Date of first issue:** 28th November 2025	
Summary: This report has the purpose of documenting the complete design, development, and construction process of a battlebot for the course *TMM4150 - Machine Design and Mechatronics*. It covers an explanation behind choices made during the different design phases, and a reflection of the result and performance in competition.		
Report written by: Aleks Polanski Florian Narum Dani Malte vor dem Esche Mohamed Elwalid Fadul Nicolò Pera Victor Matheus Berglund William Endresen		
Date of this revision: 28.11.2025	**Revisions:** 0	**Pages:** 18

Table of Contents

List of Figures

List of Tables

1 Introduction

Throughout the course TMM4105 - Machine design and mechatronics, we were challenged to make a battle-bot while learning about mechatronics, micro controllers, single-card computers, actuators, machine design, mechanics, dimensioning and other technical subjects.

Our team consisted of 7 students from the mechanical engineering department. Even though we were from the same department, we had different strengths, so we divided into groups.

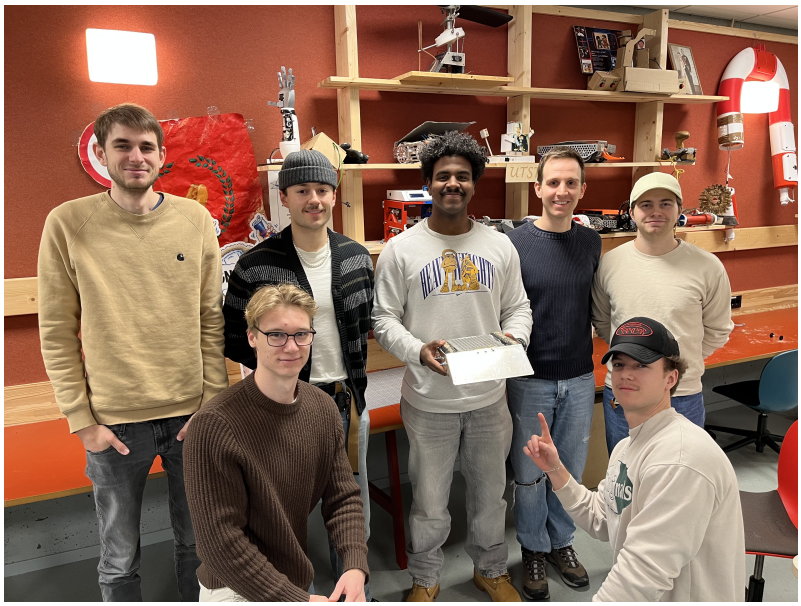


Figure 1: Team photo
Standing from the left: Malte, Florian, Mohamed, Nicolò and William
Crouching from the left: Aleks and Victor

To organise the work efficiently and make sure all tasks progressed at a steady pace, the group agreed early on to divide the workload according to the main strengths and preferences of each member. Even though everyone contributed across multiple areas when needed, each group member had one primary responsibility and one secondary task. This structure helped keep the project moving, especially during weeks when certain parts of the build required more attention than others.

The work was split into four main areas: CAD design, manufacturing, electronics, and programming. The distribution of tasks was as follows:

- **Aleks Polanski** – CAD, Manufacturing
- **Florian Narum Dani** – CAD, Manufacturing
- **Malte vor dem Esche** – Electronics, Programming
- **Mohamed Elwalid Fadul** – Electronics, Programming
- **Nicolò Pera** – CAD, Manufacturing
- **Victor Matheus Berglund** – Electronics, Programming
- **William Endresen** – CAD, Manufacturing

The mechanical design, electronics and software, and manufacturing remained tightly connected throughout the project. Changes in one discipline often required adjustments in the others; for

instance, updates in the CAD model influenced both the placement of electronic components and the manufacturability of the parts. For this reason, the sub-groups coordinated regularly and shared responsibility for decisions affecting the overall design.

This structure proved effective in keeping the workflow organised while still allowing flexibility. Whenever unexpected issues appeared, such as redesigning 3D-printed parts, adjusting tolerances, or solving integration problems, members from different sub-groups collaborated to resolve them quickly. This collaborative way of working was essential for completing the robot within the tight time constraints of the course. Figure 2 shows the timeline for how the groups organized the workflow.

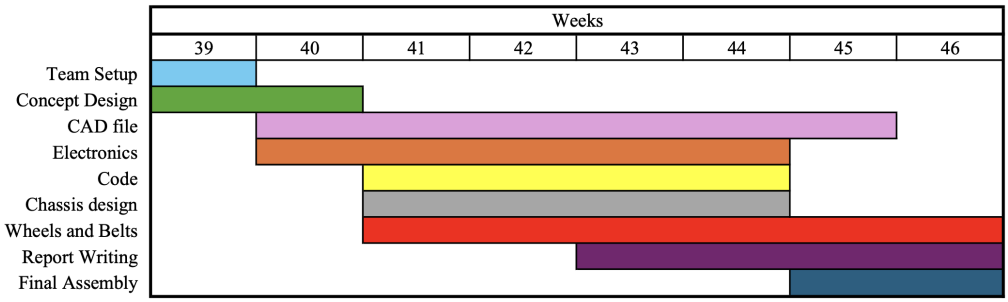


Figure 2: GANTT chart of tasks timeline

1.1 Motivation

The primary motivation behind our design was to construct a fully functional robot using only components available within the laboratory facilities, thereby avoiding reliance on external suppliers. This objective required focus on simplicity and robustness. Rather than pursuing a complex solution that might compromise reliability or fail to reach completion, the team prioritised a simpler and more robust design. We prioritised developing a straightforward design that could be manufactured, assembled, and tested with a high degree of efficacy.

The motivation behind an uncomplicated, but well-executed system, was to outperform a more complex design by eliminating unnecessary failure points. Consequently, we focused on ensuring that every subsystem could be realised using accessible materials and processes, reducing the risk of delays and integration challenges.

In forming our concept, we also drew inspiration from successful designs used in previous competitions. Observing characteristics shared by past winning robots provided valuable insight into which design elements tend to perform reliably under the specific conditions of the battlebot arena. These observations guided us toward a configuration that balances simplicity with proven effectiveness.

1.2 Concept and Strategy

To design and build an effective combat robot, it is essential to establish a clear set of design constraints. These constraints were derived from two primary sources: the official competition rules and the goals and requirements defined by the group. The internal goals were developed after a strategic analysis of previous competitions, where observed trends and outcomes provided valuable insight into successful design choices.

After reviewing recordings of previous matches, the group concluded that a defensive strategy would be most suitable for the given weight class and arena conditions. The decision was based on observations that offensive weapon systems had limited effectiveness against robust defensive designs while also consuming a significant portion of the available weight budget, constraining other subsystems. Previous projects showed that such systems did not provide sufficient offensive capability, and setting off development time to a dedicated weapon would have detracted from the propulsion system, which in this project functions as both the primary defensive and offensive

mechanism. By omitting a primary weapon, more of the weight allowance could be allocated to the propulsion and structural systems, enhancing both traction and durability. Furthermore, many matches were decided through pushing engagements that followed head-on collisions. In these scenarios, the robot with the superior traction and torque typically prevailed. Consequently, the team prioritized maximization of push power through a propulsion-focused design.

To further strengthen this capability, a ramp structure was integrated into the front of the robot. The ramp is intended to wedge underneath opposing robots, exploiting the uneven surface of the arena. Since all competitors must account for a minimum ground clearance to navigate the terrain, maintaining a ramp as close to the ground as possible increases the likelihood of wedging under an opponent upon contact. By wedging under the opponent, the opponent's drive system will be rendered ineffective, therefore providing and advantage in a pushing contests. This concept forms the core of the groups defensive strategy.

Figure 3: This years competitors

2 Process Description

Before starting to build a robot, a detailed elaboration of all the options, that might be suitable for a battle bot need to be examined. The whole process of finding and evaluating designs and choosing the optimal one is described in the following chapter.

2.1 Product Development

The product development phase translated the selected concept into a functional prototype through iterative design, simulation, and testing. This stage integrated mechanical, electronic, and software subsystems to ensure the complete operability and robustness of the robot under competition conditions. Starting from the CAD model generated in Fusion 360, each component was dimensioned to fit the 2 kg limit while maintaining mechanical stiffness. The chassis, made of laser-cut aluminium plates, was assembled using threaded joints to allow easy access to internal components. The track system was refined through several iterations to optimize traction and reduce friction losses, with 3D-printed PLA rollers and tensioning elements ensuring smooth motion. Special attention was paid to the mass distribution and center of gravity, which were validated through static and dynamic analyses.

2.2 Ideation

The ideation phase focused on generating multiple mechanical and functional concepts that could satisfy the competition requirements: a maximum weight of 2 kg, reliable mobility on mixed terrain, and structural robustness during collisions. Brainstorming sessions were carried out during the first week of the project, where each team member proposed different configurations, materials, and combat strategies. Ideas were discussed collectively and evaluated in terms of feasibility, effectiveness, and manufacturability.

2.2.1 Initial Concepts

Several potential layouts were sketched and modelled conceptually:

- Offensive robot with spinning weapon: designed to actively damage or destabilize opponents using a high-speed rotating disk. Although potentially powerful, this concept was quickly deemed impractical due to its high mass, energy consumption, and the mechanical complexity required to integrate a safe and reliable weapon system.
- Wheeled robot: a lightweight, agile solution that offered good speed on flat terrain. However, considering that the combat arena includes zones with sand and uneven surfaces, wheels would have provided insufficient traction and could easily lose grip.
- Tracked defensive robot: a compact, low-profile robot designed primarily to resist hits and overturn opponents using a wedge-shaped front plate. This configuration prioritizes torque, stability, and durability rather than speed or attack power.

2.2.2 Concept Selection

After comparing the advantages and limitations of each concept, the team voted on the design of the robot (Figure 19&20) and converged towards the defensive tracked robot configuration. The decision was based on its superior traction on sandy terrain, balanced mass distribution, and structural simplicity. The wedge design provided a strategic advantage, enabling the robot to lift or destabilize the opponent through contact rather than direct impact.

2.2.3 Concept Refinement

Once the main concept was selected, the team refined the design by defining:

- Approximate geometry and dimensions of the chassis, ensuring a low center of gravity and a compact structure
- Material selection
- Electronic layout, integrating motors, a microcontroller, and drivers
- Overall assembly strategy to facilitate quick maintenance and access to internal components

The ideation process provided a solid foundation for detailed CAD modeling and subsequent prototyping. It ensured that all critical design choices were made early, based on both functional reasoning and collective agreement within the group.

2.3 Design Constraints

The project was governed by both external and internal constraints. Externally, the competition organizers specify a number of mandatory requirements, including the maximum weight limit, restrictions on weapon types, and the integration of a battery management system (BMS). Internally, additional constraints were established to align with the chosen defensive strategy.

A low overall height was prioritized to enable the robot to engage effectively with opponents via the front ramp. A reduced height also contributes to a lower center of gravity, thereby improving stability during collisions. Another important consideration was the arena environment. Many past matches were lost due to robots becoming immobilized in the sandpits. While the optimal approach is to avoid these hazards altogether, the design must still accommodate recovery from such situations. To address this, belt-driven propulsion was selected, as it offers improved traction and manoeuvrability on loose or uneven surfaces.

Given the strategic emphasis on pushing power, the drivetrain was designed to deliver high torque output. As a trade-off, top speed was considered less critical. To provide a clear design target and guide component selection, a nominal speed of 1 m/s was defined.

2.4 Decision Making

At the start of the project, the team explored several possible layouts and propulsion strategies before narrowing the options down to something the group could realistically build within the 2 kg limit. Most decisions were made gradually, based on short discussions, quick sketches, and early tests done in the workshop rather than through a fixed plan. By comparing what worked in practice with what the group could manufacture reliably, the team moved step by step toward the final defensive tracked concept.

2.4.1 Locomotion and Combat Strategy

Early in the project, the team compared different propulsion and combat approaches to see what would actually work in the mixed sand–metal arena. Wheels were considered first, since they are lighter and mechanically simpler, but some quick tests and previous competition footage showed that they tended to lose traction or sink slightly in the sand. Tracks, on the other hand, offered much better grip and stability, even though they added some weight and mechanical complexity. This made the tracked solution the more reliable choice for the environment we were dealing with.

In parallel, the group discussed whether the robot should include any kind of active weapon. Although several ideas were proposed, the weight limit quickly became a decisive factor. A spinning

disk, flipper, or similar mechanism would have required a significant portion of the mass budget, leaving less room for a robust chassis and a strong drivetrain. Combined with the fact that many past matches were decided by pushing power rather than damage, the team gradually moved toward a defensive concept. This also aligned with the idea of integrating a low wedge at the front, allowing the robot to get underneath opponents and control engagements without relying on a dedicated weapon.

The combination of these two choices formed the foundation of the final design, balancing practicality, reliability, and the constraints of the competition.

2.4.2 Material and Manufacturing Decisions

The robot's structural frame was made of aluminium, as the laboratory facilities available to the group provided automatic laser cutting for this material. Aluminium was chosen for its high stiffness and impact resistance, ensuring that the robot can handle collisions without deformation. The track rollers were designed with notches to increase grip on the tracks. This geometry also made them easy to 3D-print in PLA. Additive manufacturing contributed to reduced weight and simplified the production process. This hybrid solution balanced robustness and mass while keeping fabrication simple and inexpensive.

2.4.3 Mass Estimation and Validation

The total mass of the final design was roughly estimated using the Fusion 360 model, which accurately represents all the real components. This allowed the team to confirm compliance with the 2 kg competition limit before fabrication and assembly. The CAD-based estimation ensured that design modifications would not exceed the allowable weight while preserving structural integrity.

	Weight budget		2000
Components	Weight	%	Weight left
Motor x 2	120	6	1880
Motor attachment system x 2	26	1.3	1854
Front cover	180	9	1674
Wheels x 4	197.2	9.86	1476.8
Belt (estimated) x 2	137	6.85	1339.8
Bottom plate	140	7	1199.8
Top plate	124	6.2	1075.8
Side part x 2	354	17.7	721.8
M3 x 12	13.08	0.654	708.72
Back wheel x 2	94.5	4.725	614.22
Axle for assy (M5) x 4	18.08	0.904	596.14
Door hinge x 2	12	0.6	584.14
Washer DIN 125-A x 12	5.16	0.258	578.98
Side part connector x 4	13.2	0.66	565.78
M3 x 10 ISO 4017 x 12	12	0.6	553.78
Side part x 2 NEW	236	11.8	317.78
Unused weight	317.78	15.889	

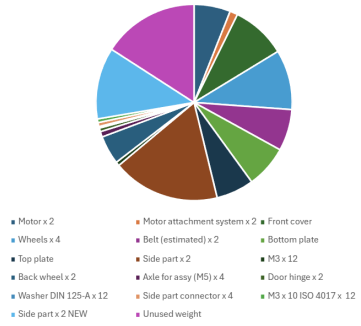


Figure 4: Weight budget distribution

Overall, the decision-making process was collaborative and pragmatic: each choice reflected a balance between performance, manufacturability, and the strict mass constraint imposed by the competition rules.

2.5 Electronic components & testing

In the beginning, there were a couple of discussions about the components that should be used. The main three components in question were the motor, the driver and the microcontroller, as most of the other components followed these choices. For the controller and micro controller, a choice was made relatively fast, and a PS4 controller and an "ESP32 Dev Board" were chosen. The ESP was chosen for its higher clock speed in comparison to an Arduino, as well as its wireless connectivity capabilities that was needed to connect the PS4 controller over Bluetooth.

For the motor, there was the choice between brushed and brushless DC motors. Both come with certain advantages and disadvantages. Brushless DC motors have the advantage that they can provide a lot of power in a small lightweight package, but the problem is that power is only produced at higher RPMs, so a transmission would be needed. Brushed DC motors on the other hand are heavier and do not produce nearly as much power as their brushless equivalent. However, they don't spin as fast, and have the advantage that they often come with gearboxes already attached. In the end, a 24V brushed DC motor was chosen with a 20.8 reduction ratio.

After choosing the components, everything was wired on a breadboard to make sure that the concept we thought about earlier was indeed functioning. Figure 5 shows the prototype circuit built on a breadboard. This setup includes the two brushed DC geared motors, the motor driver module, the voltage regulator, and the microcontroller. All components were loosely interconnected using jumper wires.

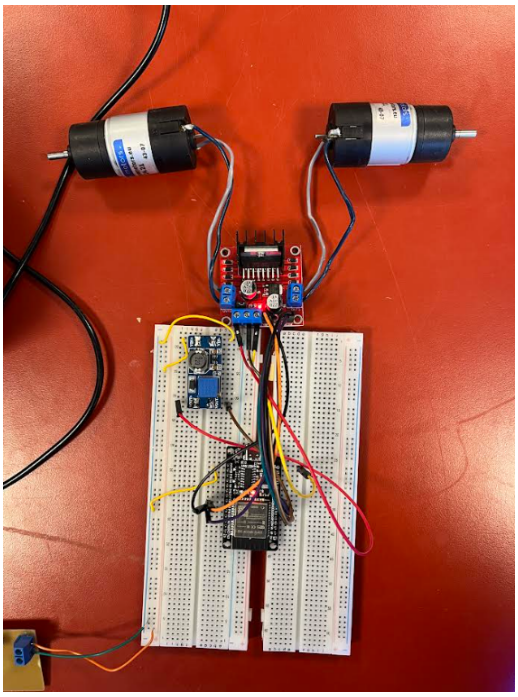


Figure 5: Prototype of the electronic control circuit used for testing.

2.6 Mechanical testing and validation

Multiple field tests were conducted on surfaces replicating the competition arena and the arena itself. These tests focused on traction performance, turning capability, and structural resistance under impact. Data from these sessions guided incremental adjustments to both mechanical- and control systems. The final prototype achieved consistent motion, high grip on loose surfaces, and reliable operation throughout all testing sessions, confirming the robustness of the chosen defensive tracked configuration.

3 Product Documentation

The following chapter presents the final design of the robot, including detailed documentation of the mechanical structure, electronics, and manufacturing processes.

3.1 Final Design

The final design was chosen due to weight, manufacturability and strength. We made close to 40 iterations, all of which had minor changes from the previous version.

Figure 6: Final design in fusion360

For simulation we used Fusion 360 and applied a total force of 40N distributed in both the bottom-plate and axles throughout. This force was chosen based on the weight of two bots at 2 kg.

$$F = mg \approx 2 * \left(2kg * 10m/s^2\right) = 40N$$

Fusion estimated a safety factor of roughly 8. This is more than enough, and made us lock in on the final design.

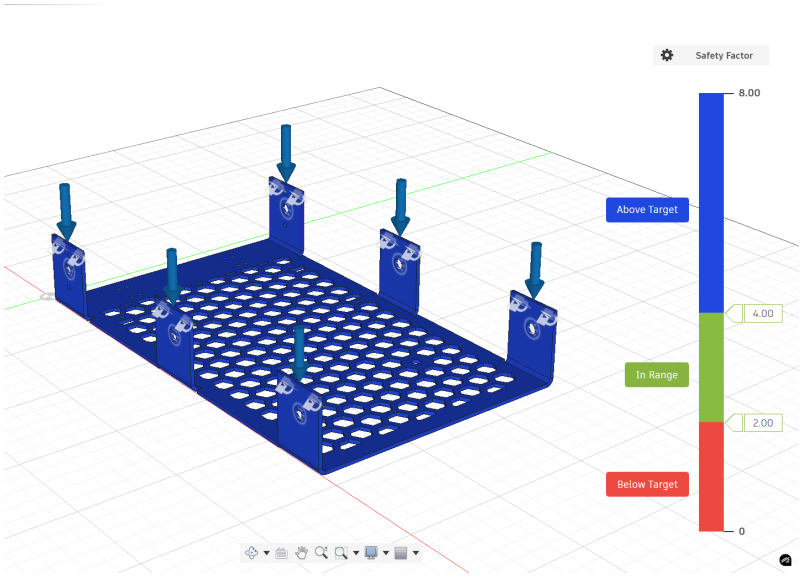


Figure 7: Simulation of bottom plate in Fusion 360

3.2 Production Methods

The manufacturing approach was defined mainly by what we could reliably produce in the workshop. Most structural components, such as the aluminum plates for the chassis, were cut with a water jet. This method gave us good accuracy and repeatability, but the long queue times meant that these parts had to be finalised early in the project.

For the remaining components, we relied heavily on 3D printing. This allowed quick iterations when dimensions or clearances needed adjustments. Several parts went through multiple printed versions before reaching their final shape, something that would not have been practical with metal machining.

Manual machining was used only for minor operations, such as cleaning edges or widening holes, since the available lathe and mill did not provide the precision needed for tight fits.

Overall, the final production strategy emerged naturally: aluminum for the load-bearing frame and 3D-printed plastic for parts requiring frequent iteration or low stiffness. This combination kept the manufacturing process simple while allowing for quick redesigns during the build.

3.3 Mechanical

From the very first CAD draft, we consistently prioritized designing according to the specific production methods intended for each component. One of the challenges we encountered was that not all team members were familiar with the manufacturing techniques that would be used. This occasionally led to suboptimal designs, particularly for parts intended to be 3D-printed, which caused minor delays and the need for redesign. To address this, the group reworked the affected components and reprinted them. Although this step was necessary to achieve the final design quality we aimed for, the redesign phase could have been avoided through a more thorough initial understanding of the production constraints.

3.3.1 Wheels

With the maximum allowed weight for the battle-bot being two kilograms (plus an additional 600 grams for the battery), it was crucial to maintain a lightweight design while staying within the specified limits. This also provided some flexibility to redistribute "dead weight" strategically around the robot to achieve the desired weight balance and stability. During the evaluation of the first wheel iteration, it became apparent that the design was excessively heavy and severely over-dimensioned. Since the battle-bot is designed with six wheels, any unnecessary mass in a single wheel would be multiplied by six. Consequently, reducing wheel weight became a key focus area for the CAD team.

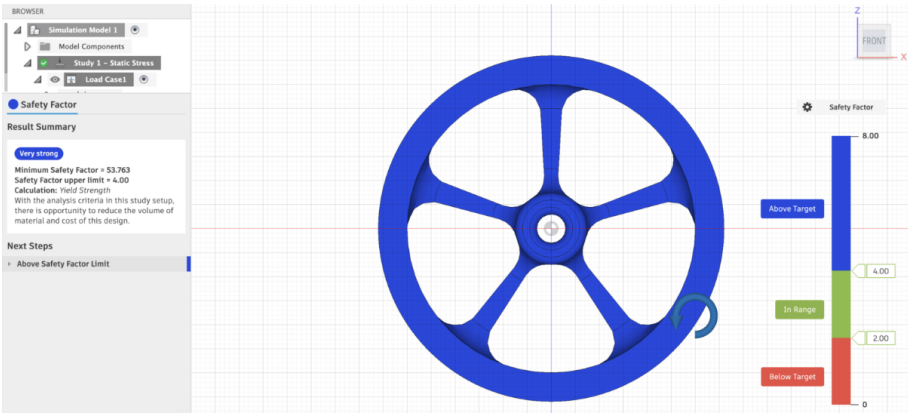


Figure 8: Structural analysis of the wheel in Fusion 360

The applied tangential force on the wheel is $F = 40\,\mathrm{N}$ with a lever arm of $r = 0.03\,\mathrm{m}$. The resulting total torque is therefore calculated as

$$T = F \cdot r = 40\,\mathrm{N} \times 0.03\,\mathrm{m} = 1.2\,\mathrm{Nm}$$

Since the torque is distributed equally between two driving wheels, the torsional load per wheel becomes

$$T_{\mathrm{wheel}} = \frac{1.2\,\mathrm{Nm}}{2} = 0.6\,\mathrm{Nm}$$

This calculation was used to validate that the geometry of the wheel and hub design could withstand the applied torque without exceeding the material's yield limit. The analysis also served as a reference for comparing wheel designs during weight optimization, ensuring that the structural integrity was not compromised by material reduction.

To obtain empirical data regarding the wheel's structural performance, a manual torsion test was performed using pliers to apply a twisting force to the wheelbase, as shown in Figure 9. The deformation and subsequent failure observed during this test represent a load far greater than what we expect the wheel to experience under peak operating conditions that we calculated. Based on this observation, it can be concluded that the wheels will not be subjected to comparable torsional stresses during actual use, and the design is therefore considered structurally sufficient.

Figure 9: Empirical destructive torsion test performed on wheel prototype using pliers.

3.3.2 Battery mounting

The first design iteration of the battery mounting structure, shown in Figure 10, was made of ABS plastic and had a total mass of 26 g. While this version provided high rigidity, the design was significantly over-dimensioned considering the loads and boundary conditions it would experience in operation. The battery does not generate substantial mechanical loads, vibrations, or torsional forces that would justify such a rigid and heavy structure.

In the second design iteration, shown in Figure 11, the component was optimized with reduced material usage and simplified geometry. The mass was reduced to only 5 g, representing a weight reduction of roughly 80% compared to the initial version. Despite the large decrease in mass, the

structural integrity remained sufficient for the intended application, as the part primarily serves as a positional fixture rather than a load-bearing element. This was tested in a field study.

This redesign demonstrates the importance of evaluating the functional requirements of each component before over-engineering its geometry. By minimizing unnecessary rigidity and material thickness, both production time and material usage were reduced without compromising performance.

The battery was mounted near the middle of the robot for an even weight distribution, as the lead-acid battery was by far the heaviest component of the robot.

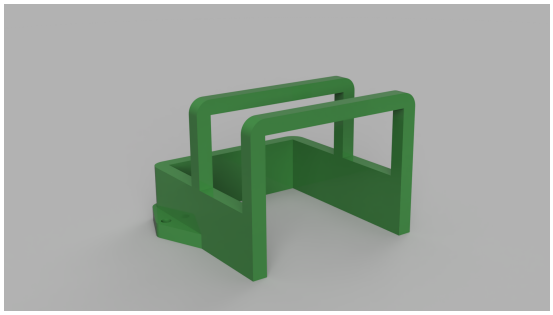


Figure 10: Battery mounting point – first design iteration (26.383 g)

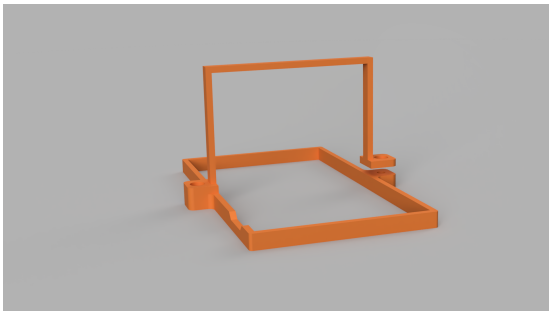


Figure 11: Battery mounting bracket – second design iteration (5.428 g)

3.3.3 Outer casing & tracks

The robot's outer shell Figure 12, consists of a top and bottom plate with bent sides, designed to provide structural rigidity. A hexagonal pattern was integrated to reduce their weight by approximately 50%. These plates formed the main chassis for the robot. On the right and left sides of the robot two track sub-assemblies, Figure 13, were mounted to house the tracks, wheels and axels while acting as protective side plates. The track sub-assemblies were optimised for easy detachability as every fault and mandatory change to the electrical system found during testing and optimization required detaching the tracks.

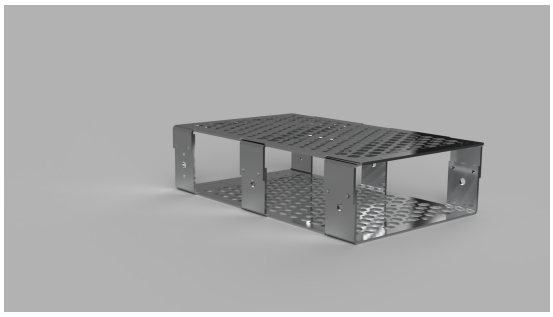


Figure 12: Outer shell

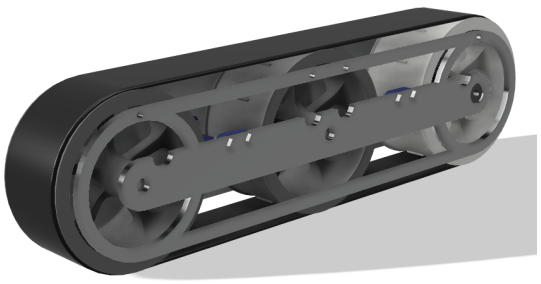


Figure 13: Side parts

3.3.4 Motor mounting

The motors were primarily mounted to the main frame of the robot using two screws parallel to the axis of the motor. These screws had the function of transferring the torque of the motors to the chassis. To further secure the motors, each had a 3D-printed holder that grabbed around the motor near the back. These two measures ensured that the motors would not move under applied torque or bending forces.

3.3.5 Electrical component rig

To secure all the necessary electrical components of the robot, a plate was printed where the electrical components would be mounted. The L298N and the BMS-board were secured to the plate with nuts and bolts, while the step-up converter and the ESP32 were glued down, due to a lack of good mounting holes in either of the components, as well as the limited space on the rig. The rig itself was also bolted to the chassis with four bolts.

3.4 Electronics

The electronic subsystem was designed to ensure stable control and power distribution. Two DC motors were driven by an L298N dual H-bridge module, controlled by an ESP32 microcontroller. For the controller, a "PS4 controller" was connected via Bluetooth to the ESP32 providing easy controllability. Power was supplied by a mandatory 600mAh Lead-Acid battery. The control software was developed and uploaded using the Arduino IDE, which provided a simple and flexible environment for programming the ESP32 microcontroller. The program interpreted input signals from the PS4 controller and converted them into differential speed commands for the two DC motors. Custom mapping of the values was implemented to ensure optimal acceleration considering dead zones of either the controller and the motors and proportional steering response. Several iterations of code tuning were performed to improve responsiveness and stability, especially on uneven or sandy surfaces.

3.4.1 Component Description

- Geared DC motors: Two identical 24V brushed DC motors with integrated gearboxes were used to drive the robots tracks. These motors were selected for their compact size and high torque-to-weight ratio, suitable for a robot with a total mass under 2 kg.
- L298N Motor driver: A dual H-bridge motor driver was employed to control both motors independently. The driver allows bidirectional motion and speed control through Pulse Width Modulation signals from the microcontroller. The motor driver also supplies the ESP32 with 5V.
- ESP32 microcontroller: The ESP32 board handles the high-level control logic, including direction commands and PWM generation. It also provides Bluetooth connectivity for the PS4 controller interface, allowing full wireless control of the robot.
- Voltage regulator: A boost converter, MT3608, supplies a stable 24V to the motor driver from the 12V lead-acid battery, to drive the motors at their rated voltage.
- PS4 wireless controller: A DualShock 4 controller connects to the ESP32 via Bluetooth, sending throttle and steering inputs for fully wireless control.
- Battery Management System (BMS) and fuse: The BMS protects the 12 V battery from overcharge, deep discharge, and short circuits. A 4 A fuse adds additional protection and allows safe system isolation. A voltage divider was added to read out the voltage of the battery continuously on the ESP. This was implemented through two resistors connected in series to divide the input voltage. For calculating the voltage output, this formula was used:

$$V_{\text{out}} = V_{\text{in}} \cdot \frac{R_2}{R_1 + R_2}$$

The resistors were chosen to step down the maximum battery voltage of 12,7V to an ESP-safe 3,3V.

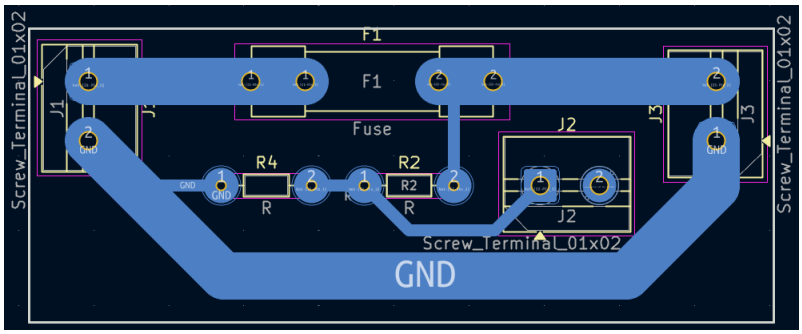


Figure 14: final design of the BMS-PCB

3.4.2 Control Logic and Code Implementation

To control the robot, several functionalities were programmed into the microcontroller. The ESP32 reads the user inputs from a PS4 controller, connected via Bluetooth, and interprets analog throttle and steering commands, as well as two buttons to generate PWM signals for the motor driver. As the robot uses two tank tracks, an algorithm was needed that implemented steering by slowing down the correct track, relative to the steering input provided by an analog stick and the throttle value read from an analog shoulder button on the PS4 controller. To connect the PS4 controller to the ESP, a library called "PS4controller" from GitHub was used.

To further control the robot, two more modes were implemented, each activated by holding down a button. The first mode implements a different steering mode. While holding down the "cross button" on the controller and giving a steering input, both tracks start moving, but in opposite directions, relative to the steering input value. Other than normal steering while driving, where one track only stops moving while steering, this allows turning on the spot.
The other mode is a crawl mode. When holding down the "circle button" while driving normally, the PWM output of the ESP is limited to lower duty cycle values in order to limit the speed of the battlebot

A wiring diagram responsible for motor control is shown below:

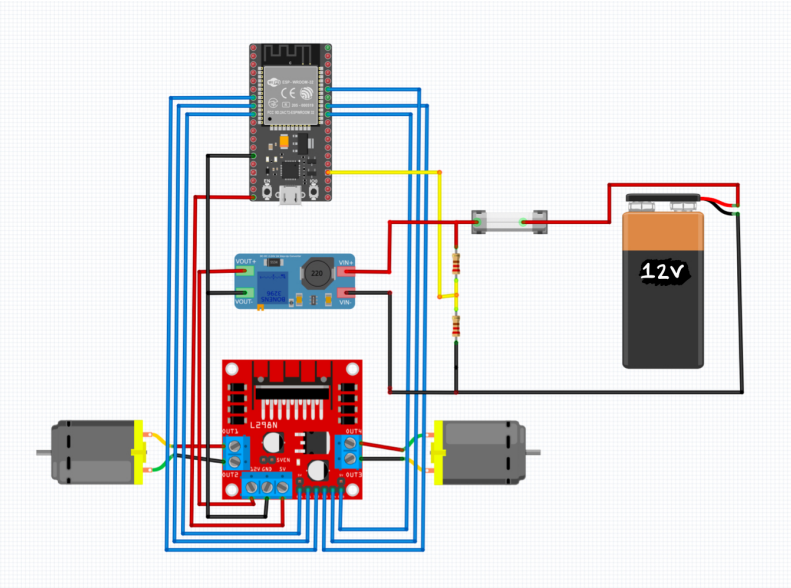


Figure 15: Wiring diagram for circuit.

A simplified block diagram of the motor control system is shown below:

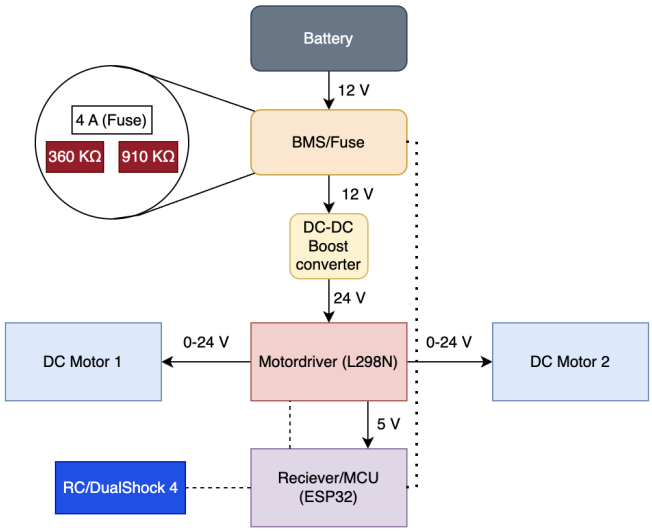


Figure 16: Block diagram for circuit.

3.5 Manufacturing and Assembly

We started out aiming to use a bit of everything in the shop. Turning, milling, waterjet cutting, sheet bending and 3D printing. Initially that was seen as a benefit as it kept our options open, making the process more flexible. Once we actually tried to run parts, the cracks showed. The lathes and mills in the flex area could not reliably hit the finish or accuracy we needed. The team also realised that any parts requiring waterjet cutting had to be prepared early, since the long queue for the machine could easily cause delays. Missing that time window would put it behind schedule with little chance to recover.

After taking a step back to review its approach, the team decided to change direction. Instead of trying to use every available manufacturing process, it focused on the ones that could be carried out reliably and consistently. That meant keeping as many parts as possible in plastic, either 3D printed or cut flat and only using metal where we truly needed stiffness, wear resistance or strong threads. The side effect was helpful. The lower mass gave the team some margin to add weight later where it was most useful, helping achieve the desired weight distribution. With that room we can move parts around after early tests and place the center of gravity where it actually helps the bot.

This was closely linked to the planned fighting strategy. Since the team relied on getting underneath other robots, the front of the bot had to sit as low as possible without catching on the floor. At the same time, the design had to prevent the opponent's weight from tipping the robot when climbing onto the ramp. A lighter structure with strategically placed weight lets us keep the front down, keep traction where it matters and avoid being levered over when someone rides up our ramp. Cutting the number of processes also made life simpler. It also meant fewer setups, fewer fixtures, fewer handoffs, reduced tolerance stack-up, and faster iterations whenever a part broke or needed adjustment.

In short, we traded variety for control. The design now prefers print friendly geometry and simple manufacturing routes, with the option to add weight only where it improves stability and handling. That change reduced risk and lined the build up with how we actually test and compete.

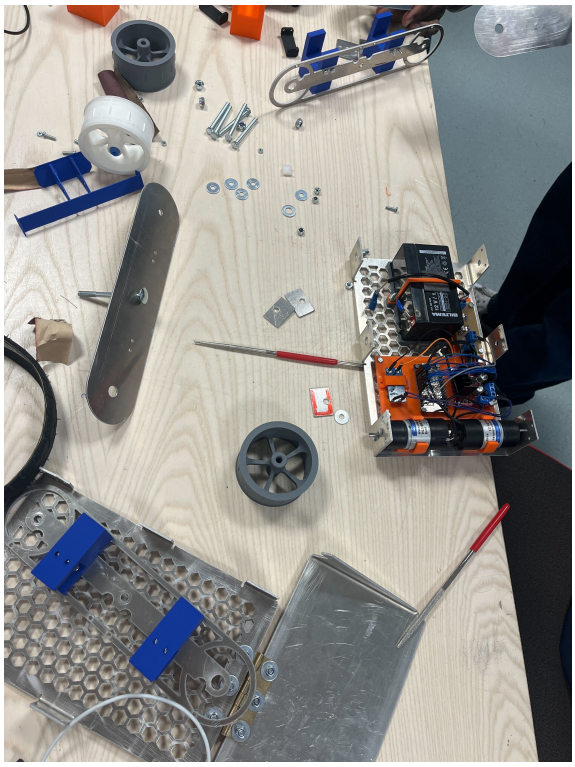


Figure 17: Assembly process of the robot

Our previous testing looked promising, which led to the final assembly. We thought this would be a LEGO-like assembly, but realized that the frame was bending in on itself due to tensions from the weight and weight placement in our bot-build. This leads to a less than desired spacing between the chassis elements causing too much friction in the interface between wheel and side-parts. We also discovered that the tightness of the belts caused friction on the axles leading to an adjustment of the belt length.

While not having a specific top hinge in mind while making the design, we had to manually cut the hinge holes ourselves.

3.6 Bill of materials

This section summarizes the main components of the robot as currently defined by the concept and design choices. A complete BOM can be found in the appendix.

Table 1: Bill of Materials

Component	Material	Process
Top plate	Aluminum	Laser cutting
Bottom plate	Aluminum	Laser cutting
Side panels	Aluminum	Laser cutting
Structural frame/chassis	Aluminum	Laser cutting
Wedge front	Aluminum	Laser cutting
Track rollers	PLA	3D printing
Tracks belts	Rubber	Sewing
Track hubs	PLA	3D printing
Fasteners set	Steel	Off-the-shelf
Electronics enclosure	PLA	3D printing
Battery pack	—	Off-the-shelf
Motor drivers	—	Off-the-shelf
Drive motors (2x)	—	Off-the-shelf
Main controller	—	Off-the-shelf
Wiring & connectors	—	Off-the-shelf

3.7 Summary

The robot used tracks to ensure stability and avoid becoming trapped in the sand pits of the arena. The outer shell integrates top and bottom aluminium plates with a hexagonal pattern to reduce mass while preserving stiffness, plus side panels to shield critical components. Structural elements in aluminium are included where higher impact resistance is required. Track rollers are 3D-printed in PLA to minimize weight and simplify fabrication. All choices are constrained by the competition mass limit of 2 kg and were preliminarily validated via CAD-based mass estimation in Fusion 360.

Figure 18: Robot after competition

4 Discussion and Conclusion

With the completion of the robot and the results in the competition, the following sections evaluate its performance, compare the results with the goals established earlier in the project, and provide solutions to problems.

4.1 Results

The battle bot lost the first battle in the competition, by getting wedged in-between the outer wall of the arena and the opponent. Therefore we lost the competition by not being able to move any more.

4.2 Strength and Weaknesses

When designing a fighting robot there is no ideal design without knowing who your competitors will be. We concluded that having a flush metal cage, would be the strength required to win. This led to the weakness of not having any weapons attached and strictly relying on pushing the opponents.

4.2.1 Concept and Strategy

One strength of the chosen strategy is that it is strong against most weapon systems. By not including a weapon, the weight budget was better used on armour and propulsion. Another strong feature was the pushing ramp, which functioned both as a defensive and offensive element. By keeping both strategy and concept simple, possible failure points that could cause the bot to lose on its own were reduced. This also made it easier to ensure a good quality of the solutions implemented, as there weren't that many to focus on.

A weakness in the strategy is that the bot lost its offensive advantage if it didn't manage to wedge itself under the opponent. If the opposing bot was slimmer than the pushing ramp, it could manage to drive over our bot because it was so low to the ground. This was very bad for our bot because it effectively reduced the pushing power to zero. Because of limited testing, optimal ramp angles were not considered. With an optimal angle, the bot should have been able to wedge under the opponent, while steep enough so that it cannot be driven up.

4.2.2 Design

One weakness of the design was the size. The bot had a suboptimal length-to-width ratio, which was detrimental to its maneuverability. In the end, the length did make it significantly more difficult to turn. Ending up in the sandpit was also one of the reasons a fight could have ended. Due to the length there was more sand pushing up against the sides of the bot, again limiting its maneuverability. Another flaw became apparent when mounting the side-panel construction. After assembly, we observed that the panels tilted slightly inwards due to over-tightening. Because we had limited spacer availability during testing, the panels were fastened directly to the chassis without any spacing. This resulted in unwanted friction between the panels and the wheels during testing, which further reduced maneuverability. A related weakness was the absence of ball bearings. Early in the design process, we decided not to include bearings in order to meet the weight target. This decision ultimately proved counterproductive. Running the wheels directly on the drive shaft generated significantly more friction than anticipated, reducing responsiveness and making precise control difficult. After the final weigh-in, it became clear that we had more than enough weight margin, and in hindsight, ball bearings should have been included to ensure smoother rotation and improved performance. The left over weight budget could have also been used to incorporate an offensive weapon or been allocated towards more powerful components.

4.2.3 Components

In terms of component choice, the main weakness of the bot was the motors. In the end, the motors that were chosen had too little power to drive the bot properly. Combined with the length-related design weakness discussed in the previous chapter, the motors could not provide enough torque to maneuver the bot reliably.

Compared to the group's target of having the bot reach a top speed of 1 m/s, the bot was not up to par. However, given the limited motor options available, the group did the best with what was realistically achievable. After further review, it was concluded that the bot's speed was not the limiting factor for Group 6 not winning the battle bot competition; lack of torque, maneuverability and geometry played a more critical role.

Regarding other redesigns such as the battery mount and the wheels, the group's calculations and simulations were sufficiently thorough for the task at hand. None of these design solutions failed during the competition. This was beneficial, as the resulting weight savings allowed the group to allocate more mass to other, more critical parts of the bot, which in turn supported the chosen concept of a specific center of gravity.

For the center of gravity, we did not explicitly calculate an exact position along the x-axis. However, through iteration and testing, the group managed to achieve a center of gravity far enough back on the robot to avoid tipping over when another robot was on Group 6's ramp. The ramp itself was also considered a success, as it was close enough to the ground to engage opponents while not clipping any ledges or other imperfections on the arena surface.

4.3 Solutions

The solution to the turning problem would be to shorten the wheelbase of the robot, but this is also a trade-off as the bot would be easier to flip from the front. With more time, a more compact design with a more optimal length to width-ratio might have led to a better performance.

To improve the maneuverability regardless of the length of the bot, better motors could have been used. Going for stronger brushed DC Motors or faster spinning motors with a higher gear ratio, that could provide more power would be the obvious choice. Another choice would have been to use brushless DC motors. Brushless DC motors pack significantly more power into a smaller and lighter package, but would have increased the overall complexity of the bot, by introducing new challenges, such as gearing and controlling.

4.4 Conclusion

THe group gained valuable insights to several engineering disciplines throughout the development and manufacturing of the robot. Overall, it's reasonable to say that we had a solid robot, despite having some design flaws, that had a good fight in its first and only round. Developing and building the bot was quite the process with several ups and downs. The overall weekly progress was steady and we all acquired new skills, that are either mechanical-, electrical- or programming-related.

5 Appendix

5.1 Images

Propulsion vote:

Options: Drone, Wheels, Tracks, Self-balancing

Name	First vote	Second vote
Victor Matheus Berglund	Tracks	Wheels (Adjustable)
Florian Narum Dani	Tracks	Wheels
William Endresen	Tracks	Wheels
Malte vor dem Esche	Tracks	Wheels
Mohamed Elwalid Fadul	Tracks	Wheels
Nicolò Pera	Tracks	Wheels
Aleks Polanski	Tracks	Wheels

TRACKS WIN

Figure 19: Propulsion vote

Weapon vote:

Options: Piston, Pushing, Flipping, Rotating, Spring loaded, squirter, drill, hammer, taser, sand weapons, projectile, pressure washer, claws, grappling hook

Name	First vote	Second vote
Victor Matheus Berglund	Spring loaded flipping	Pushing
Florian Narum Dani	Pushing	Rotational flipping
William Endresen	Pushing	Piston
Malte vor dem Esche	Taser	flipping
Mohamed Elwalid Fadul	Pushing	Flipping
Nicolò Pera	Pushing	Flipping
Aleks Polanski	Pushing	Flipping

PUSHING WINS

Figure 20: Weapon Vote

5.2 Code

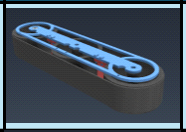
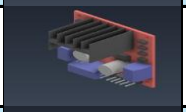
```
#include <PS4Controller.h>
#include <ps4.h>
#include <ps4_int.h>

#define RIGHT 1
#define LEFT 0

//Define Pins for controlling the L298N Board
const int ENA_RIGHT = 12;
const int ENA_LEFT = 33;

const int motor1pin1 = 14;  // IN1&2 for Motor RIGHT
const int motor1pin2 = 27;

const int motor2pin3 = 25;  // IN3&4 for Motor LEFT
const int motor2pin4 = 26;
```

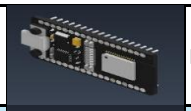
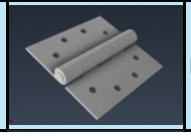
```
17
18 //Set PWM config
19 const int PWM_CHANNEL_RIGHT = 0;
20 const int PWM_CHANNEL_LEFT = 1;
21 const int PWM_FREQ = 5000;
22 const int PWM_RESOLUTION = 8;
23
24 //Define Pins for voltage read out
25 const int SENSVN = 34;
26 int voltVN;
27
28 //Controller input variables
29 int throttle = 0;
30 int stickValue = 0;
31 int steeringValueLEFT = 0;
32 int steeringValueRIGHT = 0;
33
34 int i = 0;
35
36 void setup() {
37
38   //pinMode Motor RIGHT
39   pinMode(motor1pin1, OUTPUT);
40   pinMode(motor1pin2, OUTPUT);
41   pinMode(ENA_RIGHT, OUTPUT);
42
43   //pinMode Motor LEFT
44   pinMode(motor2pin3, OUTPUT);
45   pinMode(motor2pin4, OUTPUT);
46   pinMode(ENA_LEFT, OUTPUT);
47
48   //pinMode ADC
49   pinMode(SENSVN, INPUT);
50
51   //Setup PWM channels
52   ledcAttachChannel(ENA_RIGHT, PWM_FREQ, PWM_RESOLUTION, PWM_CHANNEL_RIGHT);
53   ledcAttachChannel(ENA_LEFT, PWM_FREQ, PWM_RESOLUTION, PWM_CHANNEL_LEFT);
54
55   //set Motor speed to zero on start up
56   setMotor(0, RIGHT);
57   setMotor(0, LEFT);
58
59   //initialise Serial connection
60   Serial.begin(9600);
61
62   //initialise PS4 controller connection
63   PS4.begin( e8:9e:b4:d9:df:48 );
64 }
65
66 void loop()
67 {
68   //get battery voltage
69   getMeasurment();
70
71   //check if cross is pressed for turning on the spot
72   if(PS4.Cross())
73   {
74     stickValue = PS4.LStickX();
75     if(stickValue > 10)
76     {
77       setMotor(map(stickValue, 10, 128, 0, 255), LEFT);
78       setMotor(map(stickValue, 10, 128, 0, -255), RIGHT);
79     }
80     else if(stickValue < -10)
81     {
82       setMotor(map(stickValue, -128, -10, -255, 0), LEFT);
83       setMotor(map(stickValue, -128, -10, 255, 0), RIGHT);
84     }
85     else
86     {
87       setMotor(0, LEFT),
88       setMotor(0, RIGHT);
89     }
```

```
  }

  //if cross is not pressed, enter normal driving mode
  else
  {
    //call getSteering function to calculate needed motor speeds
    getSteering();

    //slow mode, if circle is pressed
    if(PS4.Circle())
    {
      steeringValueLEFT = map(steeringValueLEFT, -255, 255, -200, 200);
      steeringValueRIGHT = map(steeringValueRIGHT, -255, 255, -200, 200);
    }

    //call setMotor and output values to the L298N
    setMotor(steeringValueLEFT, LEFT);
    setMotor(steeringValueRIGHT, RIGHT);
  }
}

//trackSelect 0 = LEFT; 1 = RIGHT
//function to controll the L298N
void setMotor(int speedVal, bool trackSelect)
{
  //constain speed value
  speedVal = constrain(speedVal, -255, 255);

  //driving forward if speedVal > 0
  if (speedVal > 0)
  {
    if(trackSelect) //if right track
    {
      digitalWrite(motor1pin1, HIGH);
      digitalWrite(motor1pin2, LOW);
      ledcWriteChannel(PWM_CHANNEL_RIGHT, speedVal);
    }
    else //if left track
    {
      digitalWrite(motor2pin3, LOW);
      digitalWrite(motor2pin4, HIGH);
      ledcWriteChannel(PWM_CHANNEL_LEFT, speedVal);
    }
  }

  //driving backwards if speedVal < 0
  else if (speedVal < 0)
  {
    if(trackSelect) //if right track
    {
      digitalWrite(motor1pin1, LOW);
      digitalWrite(motor1pin2, HIGH);
      ledcWriteChannel(PWM_CHANNEL_RIGHT, abs(speedVal));
    }
    else //if left track
    {
      digitalWrite(motor2pin3, HIGH);
      digitalWrite(motor2pin4, LOW);
      ledcWriteChannel(PWM_CHANNEL_LEFT, abs(speedVal));
    }
  }

  //braking
  else
  {
    if(trackSelect) //if right track
    {
      digitalWrite(motor1pin1, LOW);
      digitalWrite(motor1pin2, LOW);
      ledcWriteChannel(PWM_CHANNEL_RIGHT, speedVal);
    }
    else //if left track
    {
```

```
      digitalWrite(motor2pin3, LOW);
      digitalWrite(motor2pin4, LOW);
      ledcWriteChannel(PWM_CHANNEL_LEFT, speedVal);
    }
  }
}

//get Steering function to calculate the needed outputs for the L298N
void getSteering()
{
  //read throttle Value and Map accordingly for backwards and forwards
  if(PS4.R2())
  {
    throttle = PS4.R2Value();
  }
  else if(PS4.L2())
  {
    throttle = PS4.L2Value() * (-1);
  }
  else
  {
    throttle = 0;
  }

  //get steering input and slow down the right track according to the input values
  stickValue = PS4.LStickX();
  if(stickValue > 10)
  {
    steeringValueLEFT = throttle;
    steeringValueRIGHT = steeringValueLEFT - map(stickValue, 10, 128, 0,
    steeringValueLEFT);
  }
  else if(stickValue < -10)
  {
    steeringValueRIGHT = throttle;
    steeringValueLEFT = steeringValueRIGHT - map(stickValue, -128, -10,
    steeringValueRIGHT, 0);
  }
  else
  {
    steeringValueRIGHT = throttle;
    steeringValueLEFT = throttle;
  }
}

//get ADC measurments and average them over 1000 samples for better battery
    observation
void getMeasurment()
{
  if(i == 1000)
  {
    int voltVN = analogRead(SENSVN);
    Serial.print( \tVN Value:  );
    Serial.println(voltVN);
    i = 0;
  }
  else
  {
    i++;
  }
}
```

5.3 Bill of materials

Item	Part Number	Thumbnail	BOM Structure	Unit QTY	QTY	Description
1	SidePartsAssy		Normal	Each	2	Assembly for wheels, easy attachment to rest of body
1.1	Axle for 3dprint		Normal	Each	1	Slit inside and outside
1.2	Inner SidePart - For production		Normal	Each	1	
1.3	Outer Side Part - For Production		Normal	Each	1	
1.4	Belt		Normal	Each	1	
1.5	Rear Wheel v1		Normal	Each	1	With slit and larger axle.
1.6	Wheel v1		Normal	Each	2	
1.7	M3 small thread, ISO 4017		Normal	Each	8	
1.8	SidePart Connector_Componen t1		Normal	Each	2	
2	Body Assy		Normal	Each	1	
2.1	Bottom Plate - For production		Normal	Each	1	
2.2	Top plate - For production		Normal	Each	1	
2.3	El-Components Rlg		Normal	Each	1	
2.3.1	L298n H-Bridge		Normal	Each	1	
2.3.2	MT3608		Normal	Each	1	

2.3.3	ESP32 wroom 32		Normal	Each	1	
2.3.3.1	Component1		Normal	Each	1	
2.3.3.2	Component1(Mirror)		Normal	Each	1	
2.3.3.3	Part30		Normal	Each	1	
2.3.4	El-Components RIg		Normal	Each	1	To hold in place el components
2.3.5	El-comp Spacer		Normal	Each	5	Spacers for the el-rig
2.3.6	CableSpacer		Normal	Each	1	
2.4	BatteryMotionBracket Assembly		Normal	Each	1	
2.4.1	BatteryMountingBracket-P1		Normal	Each	1	
2.4.2	BatteryMountingBracket-P2		Normal	Each	1	
2.4.3	Sealed Lead Acid Battery 12V 1.3AH v4		Normal	Each	1	
2.5	NewMotorHolders		Normal	Each	2	
2.6	Front Cover FOR PRODUCTION		Normal	Each	1	
2.7	DOOR HINGE v2		Normal	Each	2	Hinge (off the shelf part)

2.7.1	Component1		Normal	Each	1	
2.7.2	Component2		Normal	Each	1	
2.7.3	Component3		Normal	Each	1	
2.8	LuigB040x1ab		Normal	Each	2	
2.9	M3 small thread, ISO 4017		Normal	Each	30	
3	M3 small thread, ISO 4017		Normal	Each	8	

While preparing this work, the authors used AI-assisted tools for language editing and refinement. After using these tools/services, the authors reviewed and edited the content as needed and take full responsibility for the content of the published report. All analyses, interpretations, and conclusions presented in this work are the authors' own.

5.4 Declaration of AI aids and tools

NTNU

Declaration of AI aids and tools

Have any AI-based aids or tools been used in the creation of this report?

No ✓ Yes

If yes: please specify the aid/tool and area of use below.

Text

✓ **Spell checking.** Are parts of the text checked by: Grammarly, Ginger, Grammarbot, LanguageTool, ProWritingAid, Sapling, Trinka AI or similar tools?

Text-generation. Are parts of the text generated by: ChatGPT, GrammarlyGO, Copy.AI, WordAi, WriteSonic, Jasper, Simplified, Rytr or similar tools?

Writing assistance. Are one or more of the report's ideas or approach suggested by: ChatGPT, Google Bard, Bing chat, YouChat or similar tools?

If no, sign document. If yes, use of text aids/tools apply to this report -please specify usage here:

ChatGPT was used to correct grammar and sentence flow, as well as provide feedback on content.

Codes and algorithms

✓ **Programming assistance.** Are parts of the codes/algorithms that i) appear directly in the report or ii)have been used to produce results such as figures, tables or numerical values been generated by: GitHub Copilot, CodeGPT, Google Codey/Studio Bot, Replit Ghostwriter, Amazon CodeWhisperer, GPTEngineer, ChatGPT, Google Bard or other tools.

If yes, use of programming assistance aid/tools apply to this report - please specify usage here:

ChatGPT was used for last part of the code where a voltage monitor was implemented. The only difference was the change of amount of samples and variable name for a more comprehensive code

---

Images and figures

Image generation. Are one or more of the reports images/figures generated by: Midjourney, Jasper, WriteSonic, Stability AI, Dall-E or similar tools?

If yes, use of image generator aids/tools apply to this report – please specify usage here:

Other AI aids or tools. Have you used other types of AI aids or tools in the creation of this report?
If yes, please specify usage here:

✓ I am familiar with NTNU's regulations on artificial intelligence. I declare that any use of AI aids or tools are explicitly stated i) directly in the report or ii) in this declaration form.

Group 6, 27. nov. 2025,
Signature/Date/Place