WRO - Future Engineers Category

Team “Null”
SOUTH AFRICA

Engineering Journal

Contents

Introduction

Welcome to Team Null's engineering journal. Team Null is made up of Michael Shepstone (19 years), Raaid Fajandar (18 years), our coach Cody Williams and Chihuahua, our vehicle. We are from South Africa. As a team, we have been working on the Future Engineering challenge for the past 2 years. Chihuahua, the vehicle, has evolved many times to meet the challenging needs of the Future Engineers Category of the World Robotics Olympiad 2024. This document is relevant to the latest version in our “Chihuahua” series of robots (The 7th iteration).

Picture from left to right: Michael Shepstone, Cody Williams and Raaid Fajandar

Mobility Management

Motor Selection and Performance

A 6 Watt, 12 Volt brushed DC motor with an integrated gearbox(Figure 1.1) was used, providing 282 RPM (Revolutions Per Minute) at the output shaft to drive the rear wheels. This motor was chosen for its good mixture of torque, RPM, low EMI (Electromagnetic Interference) and a convenient form factor.

Additionally, this motor’s excess of torque allows it to be precisely controlled using PWM (Pulse Width Modulation), as there is no ramp-up/down when providing the motor with lower power, resulting in the vehicle responding immediately with the commanded speed.

At that stage, we pursued using a very similar motor, but with a speed of 615 RPM at the output shaft. Unfortunately, in addition to having insufficient torque at lower power levels to properly drive the robot, this faster motor caused issues with the steering control loop. There is an approximately 100 milliseconds delay between the time a steering offset is measured and the correction is applied.

This is a mixture of the delay in reading from the gyroscope, the processing of the data, and the response of the servo. This delay is acceptable with negligible effects at lower speeds, but causes the vehicle to oscillate when driving. Unfortunately, this resulted in unacceptable inaccuracies in the positioning of the vehicle, especially in the obstacle challenge rounds.

Therefore, the motor was replaced with the original, lower speed, higher torque motor.

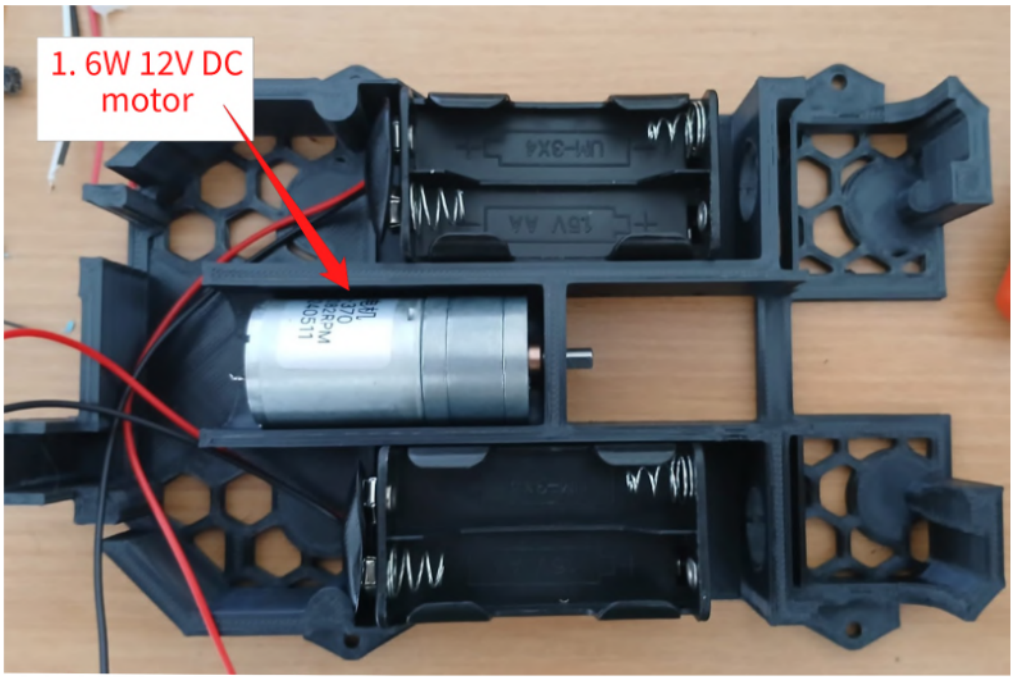


Figure 1.1 - Picture of the drive motor

Chassis Design Evolution

The chassis of the vehicle presented a unique challenge. Valuable lessons were learned from the vehicle's performance in the 2023 National Finals. Notably, the turning circle of the vehicle was far too large, and needed to be made smaller.

Additionally, in order to fit within the confines of the parking lot, without the need for actual parallel parking, the vehicle could be no longer than 200mm at any point.

Furthermore, during practice with a new vehicle (The 6th iteration of the design) it was determined that the vehicle must be as compact and small as possible to provide the greatest margin of error (Figure 1.2, Labels 1 and 3) while manoeuvring through the course. A redesign was therefore required, to create a far thinner vehicle body(Figure 1.2, Labels 2 and 4), opting to stack components vertically as opposed to laying them out horizontally. Finally, as a result of feedback, this design was to include cable channels for better cable management.

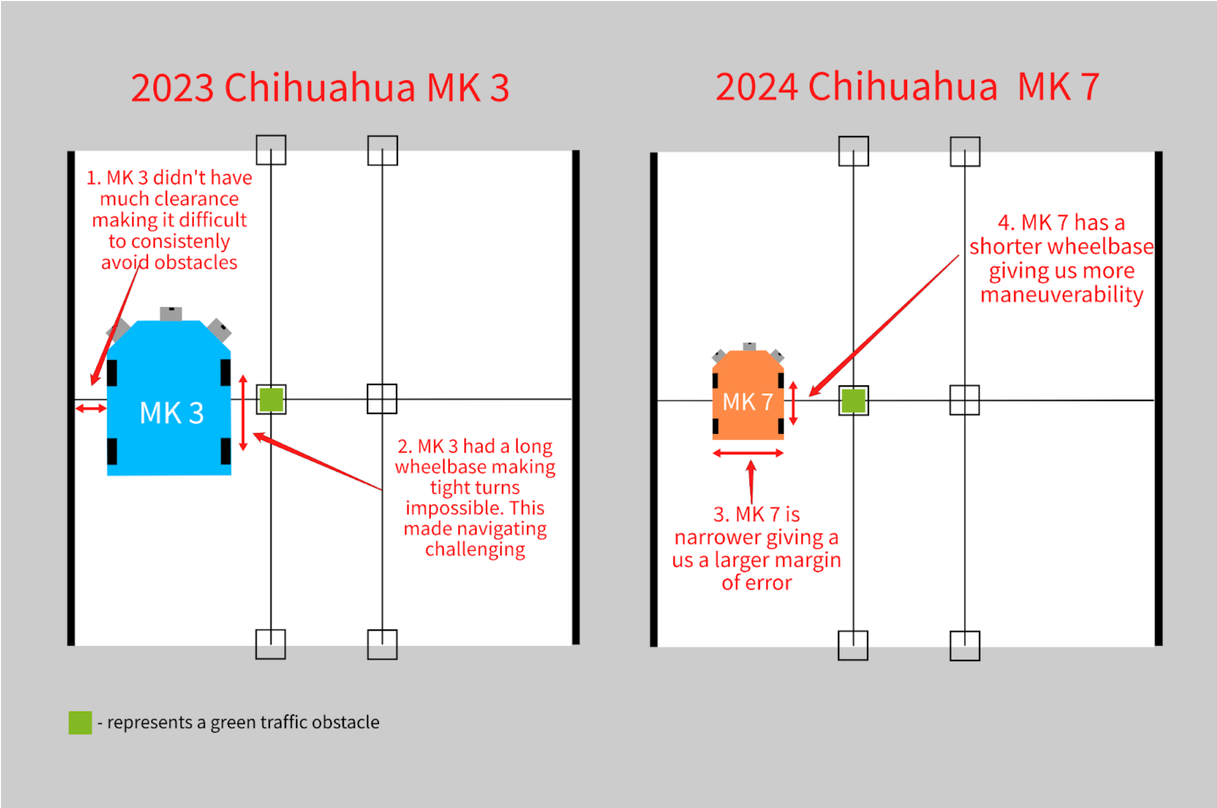


Figure 1.2 - Showing some differences between our robot, Chihuahua Mk 3, from 2023 compared to Chihuahua Mk 7 in 2024.

Powertrain and Gear System

There is a single drive axle which drives the rear wheels. This axle is connected to a small gearbox which is driven by the motor mentioned previously. The gearbox reduces the output of the motor in a ratio of 4 to 1. Due to the motor and gearbox having different shaft sizes, it was necessary to design and 3D print a drive shaft to couple the motor and gearbox. The motor is mounted perpendicular to the drive axle(Figure 1.3).

The drive axle is connected to the wheels by constant velocity joints. We chose this mechanism so that the rear wheels could drive the car and also aid in steering at the same time. This enabled us to build a 4 wheel steering system which gives greater manoeuvrability(More details are provided later).

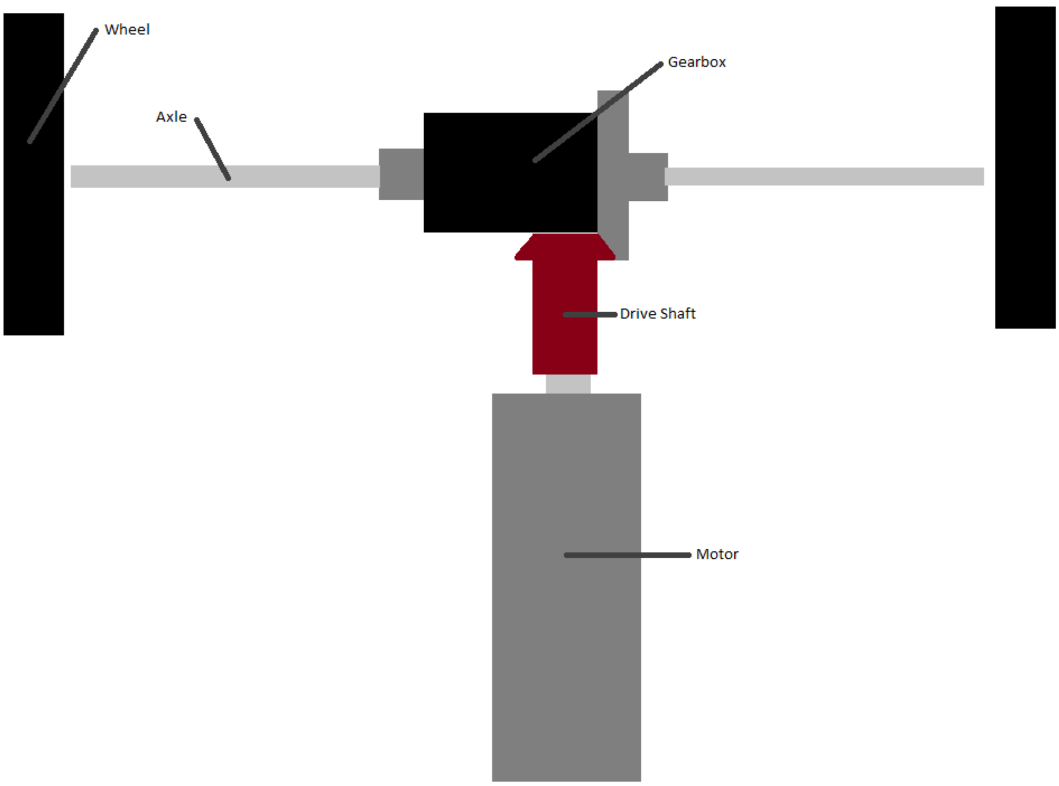


Figure 1.3 - Simplified diagram of the powertrain

Drive Shaft Development

The development of a reliable drive shaft required three design iterations and multiple prototypes with different materials. Initially, the shaft was printed with PLA at 20% and then 100% infill. Although PLA is strong, it has low shear stress resistance, causing frequent failures at the gearbox input (Figure 1.4, Label 1), requiring extensive disassembly for replacement.

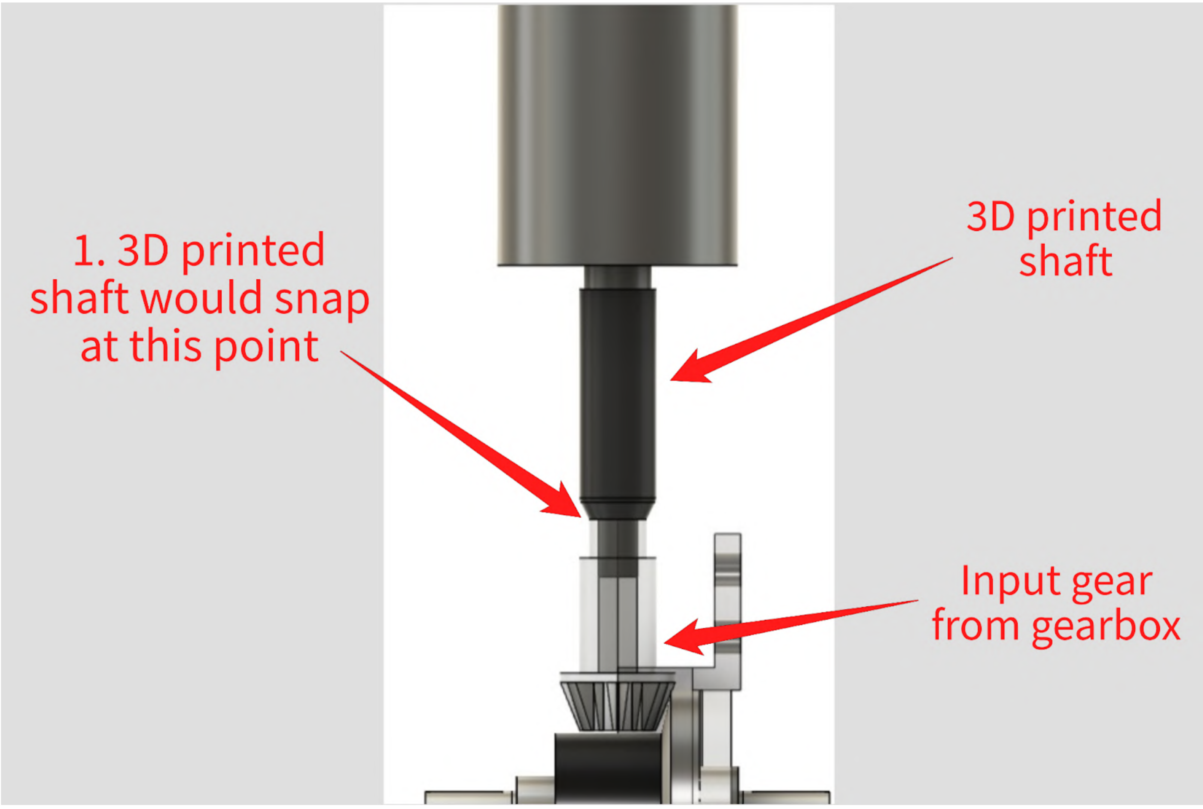


Figure 1.4 - Showcase of the typical failure point of the 3D printed drive shaft when made from PLA

The next iteration used flexible TPU filament to absorb shock and reduce shear stress. While TPU worked initially, it was too soft, causing the motor and gearbox slots to wear down quickly and causing the shaft to skip under torque.

The final version, made from PETG filament, provided the right balance of durability, hardness, and shear resistance, and is now the material in use.

Additionally, printing the shafts vertically led to weak layer lines, resulting in shearing. Horizontal printing was not feasible due to complex geometries, and machining from aluminum was ruled out due to tool limitations.

Steering System and Turning Mechanism

A key goal for this competition was achieving a significantly smaller turning circle. To accomplish this, we implemented an all-wheel steering system, inspired by the “Tesla Cybertruck.” The system uses gears and linkages to invert the steering motor's motion (Figure 1.5, Label 1) and transfer power to both the front and rear wheels (Figure 1.5, Label 2), turning them in opposite directions (Figure 1.5, Label 3). This drastically reduces the vehicle’s turning radius. The wheelbase was also minimised to further enhance steering performance.

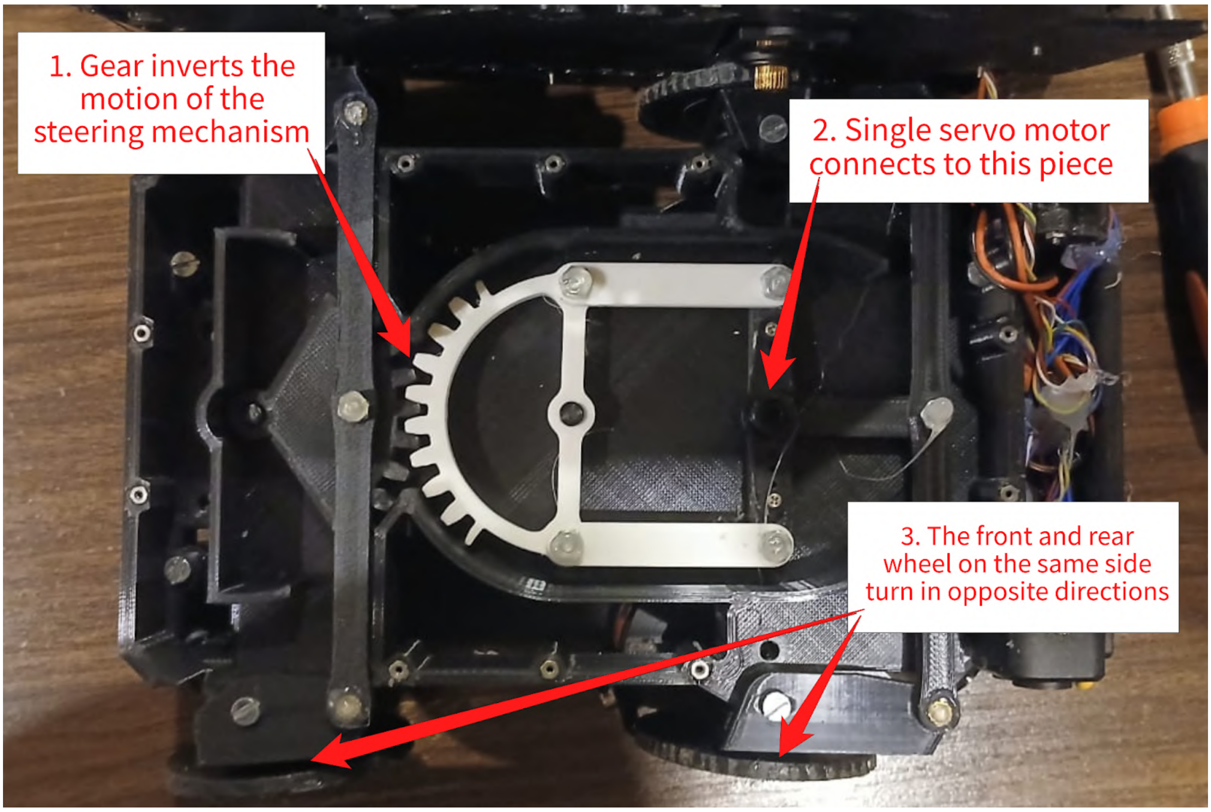


Figure 1.5 - Overview of the 4 wheel steering mechanism

A single, powerful servo motor (14KG MG996R) provides the mechanical power to move the steering mechanism. Initially, plastic 9-gram servos were used, but they burned out quickly under the steering system’s load, as the steering system has a large amount of friction from all the gears and linkages involved. Switching to the more powerful metal gear servo ensured better reliability. While even more powerful servos could work, they are unnecessary and could potentially damage the system if an error occurs.

Power and Sense Management

Power Management

The vehicle has two main power systems connected by a common ground for data transfer.

1. **Main Power System (figure 2.1.1, green block)**: Powered by eight 1000mAh 14500 lithium ion battery cells, these packs feed the drive motor, the steering servo and the microcontrollers and separate 12V, 7V and 5V circuits respectively. Each circuit has its voltage achieved through the use of a 3A buck converter, bringing the approximately 14.4V of the battery pack down to the required levels.

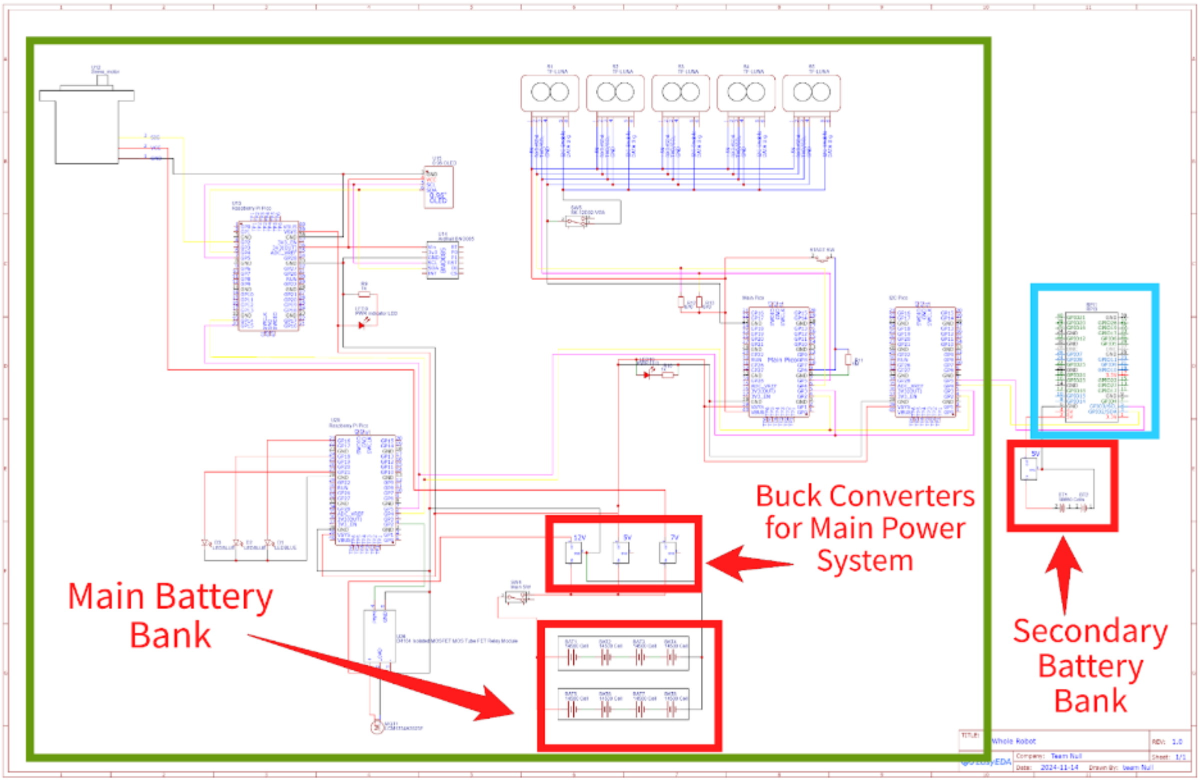


Figure 2.1.1 - Wiring diagram of the entire robot showing the main (Green block) and secondary power (Light blue block) systems

Initially 18650 cells were used for this circuit, as they had an excellent capacity, were relatively cheap and were widely available. Unfortunately, when the vehicle was downsized, the original set of 4 18650 cells did not fit in an easily accessible location (The batteries had to be somewhere where they could easily be swapped out, allowing us to continuously drive with minimal delay). Therefore, 14500 cells were used as a direct replacement. Using the same chemistry as the 18650s, they produced the same voltage as them, and were able to act as a “drop-in” replacement. However, due to their limited capacity (1000mah as opposed to 3000mah), we opted to utilise two packs of four, wired in parallel.

To ensure the longevity of both systems, the batteries are swapped out when the voltmeter monitoring the main battery voltage reads below 14V. This provides us with a battery life of approximately 1.5 hours. When the batteries are swapped, they still have 40-50% of their charge remaining.

2. **Secondary Power System (figure 2.1.1, blue block)**: Powered by two 3000mAh Samsung 18650 battery cells, which feed into a single 5A adjustable buck converter. This system exclusively powers the Raspberry Pi 4 and webcam.

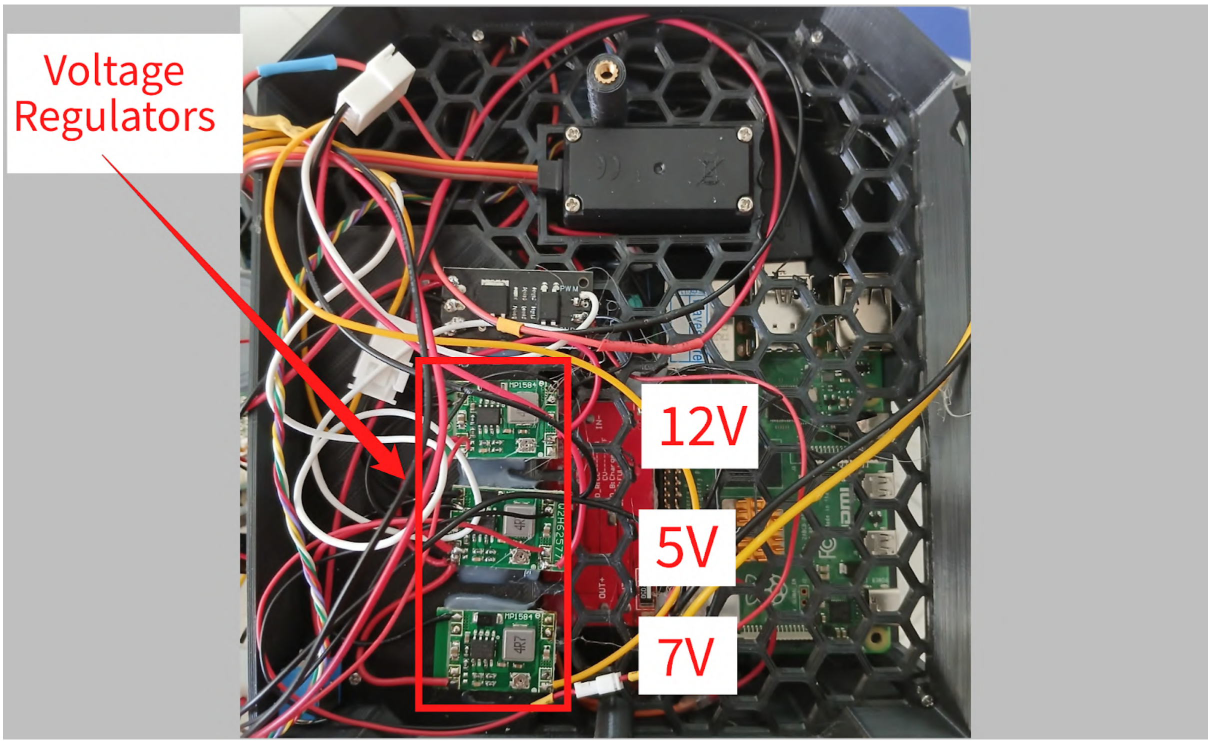


Figure 2.1 - Voltage regulators for the main power system

The separation of power systems was necessary due to current draw spikes from the drive and servo motors, this would cause a very small voltage drop in the power system. The Pi 4, which controlled the camera system, was especially sensitive to these voltage drops and would reboot as a result of them occurring. This is catastrophic for obstacle challenge runs. Therefore the decision was made to utilise two separate power systems (sharing a common ground) to eliminate this issue.

The secondary battery pack is not monitored, and is thus swapped out alongside the primary pack. When the primary pack is swapped, the secondary pack has approximately 60% of its charge remaining.

Sense Management

Sensors on the vehicle:

1. 5 x TF-Luna Time Of Flight (TOF)
2. 1 x Generic webcam (sourced from a team member's desktop computer)
3. 1 x Adafruit BM085 gyroscope

Mounting Positions.

Time of Flight(TOF) Sensors

All the TOF sensors are positioned in the front of the vehicle on specific mounts. These mounts ensure that not only are these sensors positioned and angled correctly increasing overall accuracy, but also ensure that they are mounted symmetrically. (Slight asymmetric mounting of sensors on previous iterations caused differing vehicle behaviour between directions). Each mounting point contains a slight outline for the sensor, as well as a hole in which a single M1.25 bolt can be attached through, securing the sensor in place. It was initially thought that the asymmetric mounting pressure would cause an issue for individual sensors, but this turned out not to be the case.

The 5 mounts are positioned around the front of the vehicle, facing:

- 90 degrees left
- 45 degrees left
- Directly forward
- 45 degrees right
- 90 degrees right

All the sensors are mounted with their midpoint being 30mm off the ground. This was found to be the ideal height to mount the sensors, providing a good balance between ground clearance and potential impact on the wall. This is with the exception of the sensors facing 90 degrees left and right, which, due to clearance issues, had to be mounted at a height of 45mm.

The mounting positions of the TOF sensors are visually aided by Figure 2.3 and Figure 2.2 below.

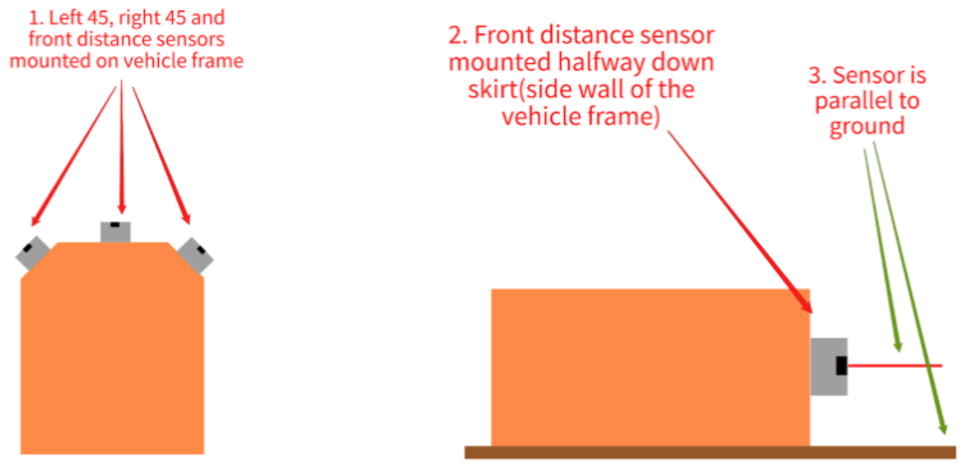


Figure 2.2 - Diagram showing front, left 45 and right 45 distance sensor mounting positions

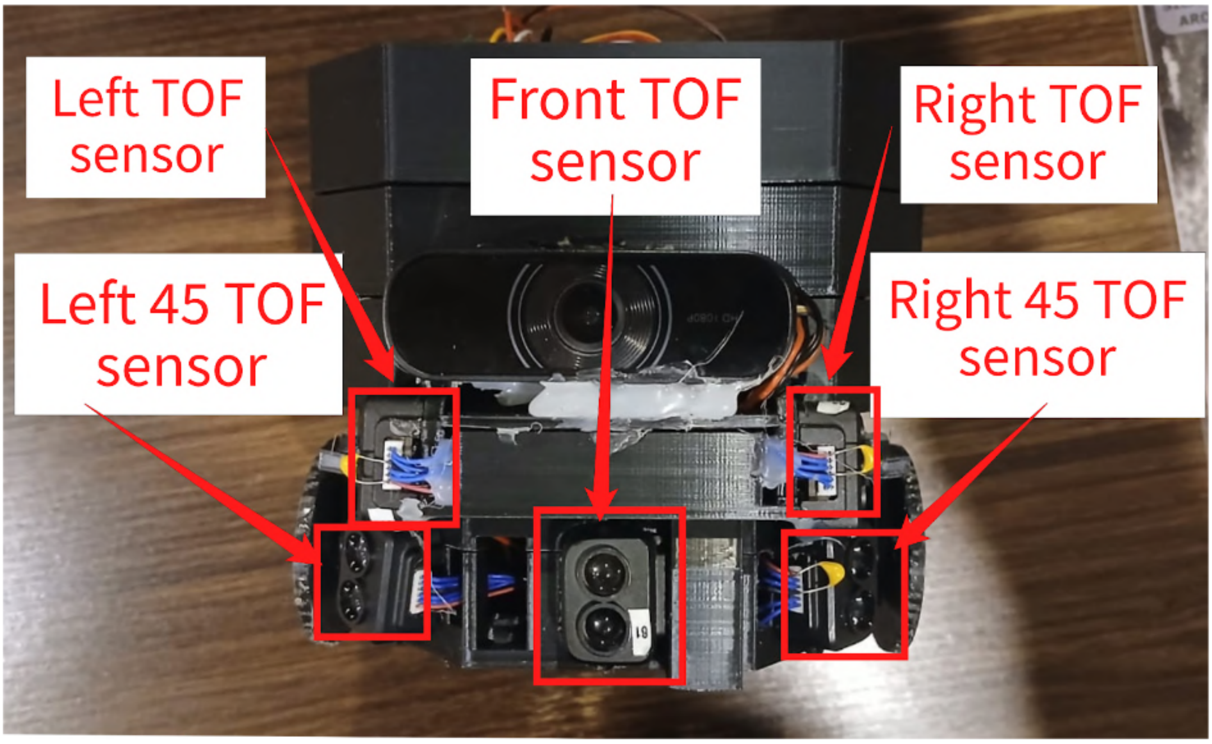


Figure 2.3 - Annotated photo of TOF mounting

It was highly important that the sensors are mounted parallel to the game mat, as failure to do so can result in inaccurate readings (Figure 2.4)

In addition to the challenge of mounting these sensors, they are also highly sensitive to EMI and input voltage fluctuations, both of which can cause them to “fall” off the

I2C bus, resulting in erroneous distance resorts from the sensor. This was remedied in both hardware and software, through the use of a 0.1uF ceramic capacitor between the 5V and ground lines of each sensor, smoothing out the power to an acceptable standard. Additionally, the bus speed was reduced from 400 khz to 100khz. Finally, the function used to obtain the distance measurement from the sensor has the ability to reboot the sensor mid-run should an issue be detected. These three remedies, working together, was enough to completely solve this issue

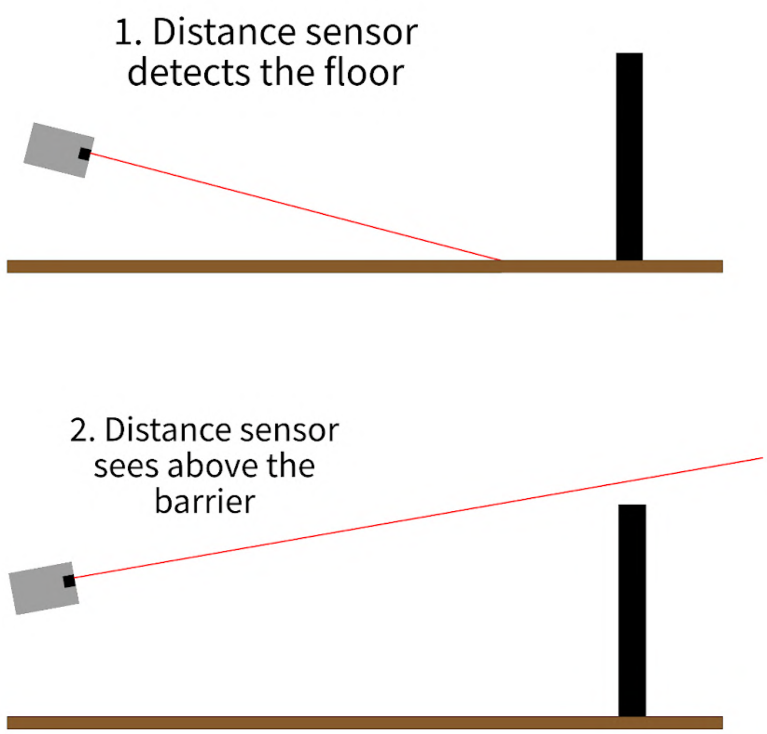


Figure 2.4 - Diagram showing that the distance sensors give unwanted values when it is not mounted parallel to the ground.

Webcam

The webcam has a greater amount of flexibility regarding mounting height than the TOF sensors, this is due to the ability to correct for mounting height offsets in software. However, the camera still needed to be looking forwards. It is therefore positioned on a “shelf” above the forward looking TOF sensor, as shown by figure 2.3, although it is not labeled.

Gyroscope

The gyroscope is mounted to a protoboard as a part of the steering system. This sits towards the back of the vehicle(Figure 2.6).

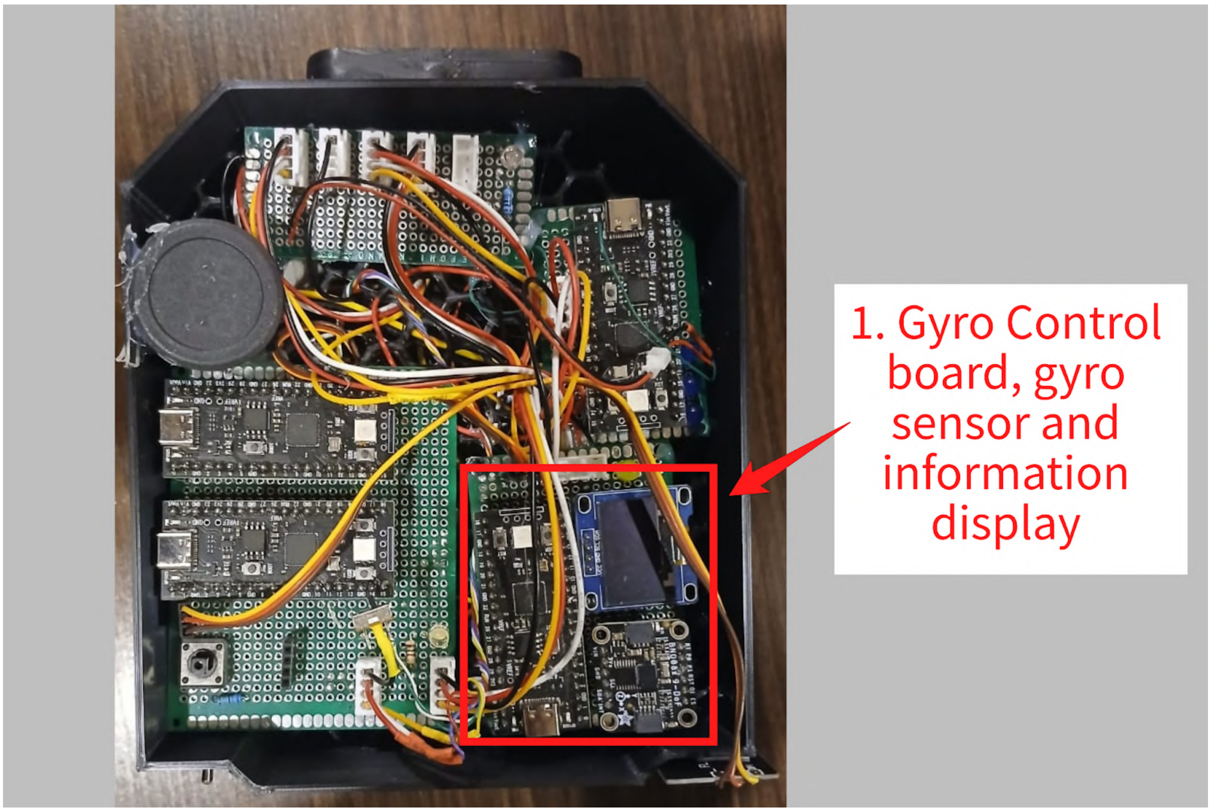


Figure 2.6 - Diagram showing the mounting position of the gyroscope and the pi pico that it is connected to

Vehicle Control

There are 4 microcontrollers and 1 Single Board Computer (SBC) to control the car(See figure 2):

1. 1 x Pi Pico for controlling the drive motor
2. 1 x Pi Pico for reading data from the gyro and managing the steering servo
3. 1 x Pi Pico that acts as the "main" controller
4. 1 x Pi Pico for I2C communication between the SBC and the main controller
5. 1 x Pi 4 that runs a python script leveraging OpenCV to detect obstacles

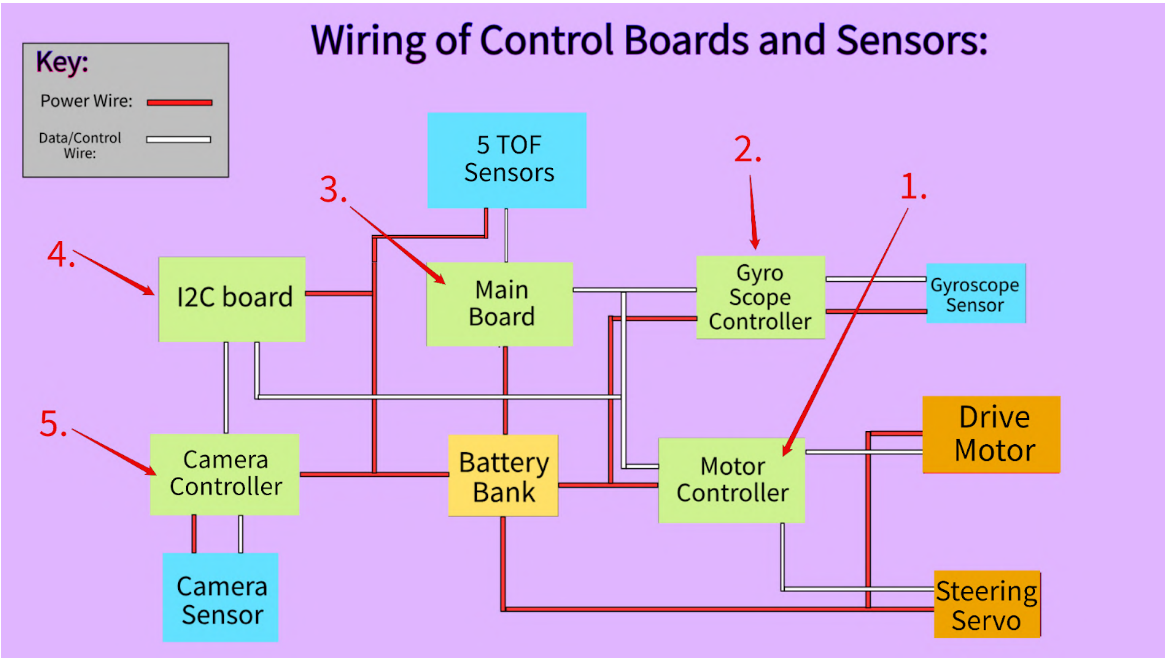


Figure 2.7 - High level overview of power and data connections

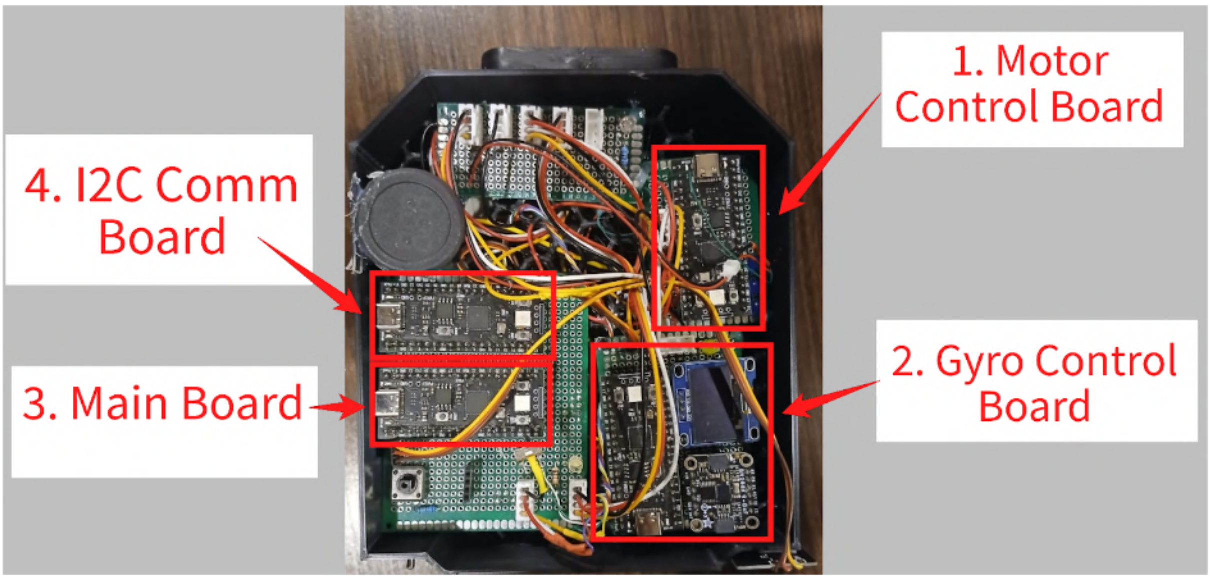


Figure 2.8 - Positions on Control Boards on the vehicle

The system involves multiple Pi Pico boards, each with two I2C buses: one for communication with I2C devices on the board and another for inter-board communication. However, the "I2C board" uses both buses for communication between boards.

Board Functions:

1. **Main Board** (Figure 2.7, Label 3): Handles decision-making, manages TF-Luna sensors, and processes data from other boards connected to the secondary I2C bus. It acts as the master on both buses.

2. **Gyro Board** (see label 2, Figure 1): Manages the vehicle's steering feedback using a BMO 085 gyroscope sensor and an OLED display. It receives target angles and reports the true angle to the main board, acting as the master on the primary bus and slave on the secondary bus.

3. **Motor Board** (see label 3, Figure 1): Controls the drive motor's speed and direction. It receives speed commands from the main board via the secondary bus and is a slave on this bus.

4. **Camera Board** (Pi 4, see label 4, Figure 1): Tracks objects by reporting their position, size, and colour on its primary I2C bus, acting as the master on this bus.

5. **I2C Board** (see label 5, Figure 1): Acts as an intermediary between the main and camera boards. It resolves the issue of the Raspberry Pi 4's inability to act as an I2C slave. It caches the camera board's reports, making them available for the main board when requested.

This modular setup, despite initially raising concerns about latency, works effectively for the vehicle's operations.

Why these boards?

Pi Picos were chosen for the control boards as they offered a nice balance of ease of use, cost and compute power. We used the arduino IDE to write C++ code for these boards. This code can be quickly uploaded to the boards with relative ease.

In conjunction with their physical benefits, these boards are also highly versatile in what they can do. They have hardware support for up to 2 I2C buses, as well as multiple serial and other communication interfaces.

It must be noted, however, that these boards have a very weak internal voltage regulator, and therefore should never power something above 200mA of current draw directly.

The pi 4 was chosen as the image processing board as it was the only board on hand with the power for image processing. Although other versions of the pi could work, such as the pi 3, pi 2, and pi 5, as well as all the variants thereof, provided they have similar specifications, we decided to use what was on hand as opposed to investing in another board for minimal gain.

The pi 4 is running [Raspbian](), a fork of the Debian operating system specifically for the Raspberry pi. It is running python code. Should this code need to be modified, it needs to be modified on the pi itself. A wireless keyboard dongle is semi-permanently inserted into the pi’s USB port within the vehicle for this reason.

Obstacle Management

The TOF (Time of Flight) sensors are used to detect the distance from the boundary walls to the left, right and front. This data is used in combination with the gyroscope to navigate around the course.

Open Challenge

During the open challenge round, the vehicle uses only distance sensors and a gyroscope for navigation, while the camera reports data to the I2C cache board, it is never read by the main board.

The vehicle begins by driving forward, using data from both the left and right TF-Luna sensors to determine the correct direction to turn. When a wall on either side “falls away,” it identifies this as the inner wall of the course, determining the direction to traverse (clockwise or counter-clockwise). Once the turning direction is established, only the sensor pointing toward the inside wall is polled, waiting for the wall to fall away before initiating a turn (See label 1 below, Also see flowchart 1).

After a turn, the vehicle waits until it has reached its target angle, which is 90 degrees from the angle it was targeting before the turn (the vehicle starts at 0 degrees). This step is necessary because the TF-Luna sensors cannot measure the distance from a wall if they are more than 5 degrees off perpendicular to the wall. Once the vehicle reaches its target angle, it checks that the wall is present before waiting for it to fall away (See label 2 below, Also see flowchart 2). This ensures the vehicle does not make two rapid turns and double back on its path, which could cause it to get stuck against a wall.

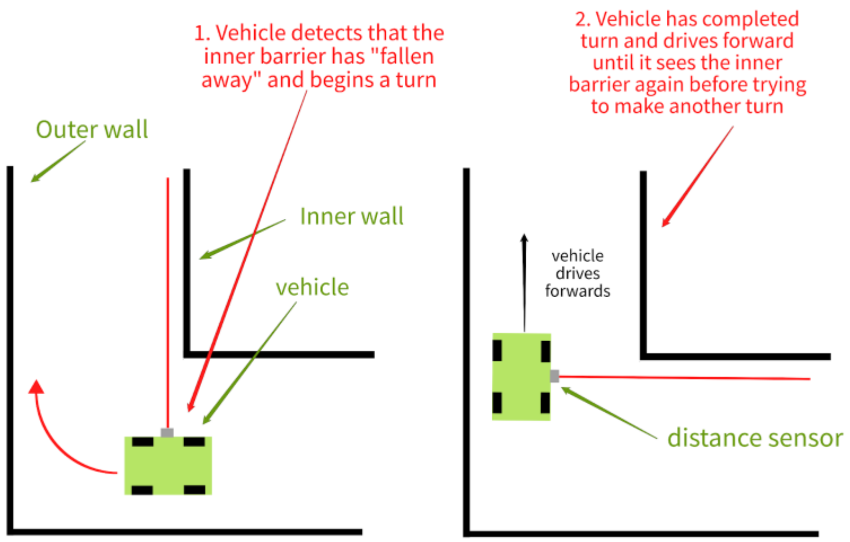


Figure 3.1 - Diagram showing the vehicle making a turn in the course.

This sequence of steps is repeated until all 3 laps are completed.

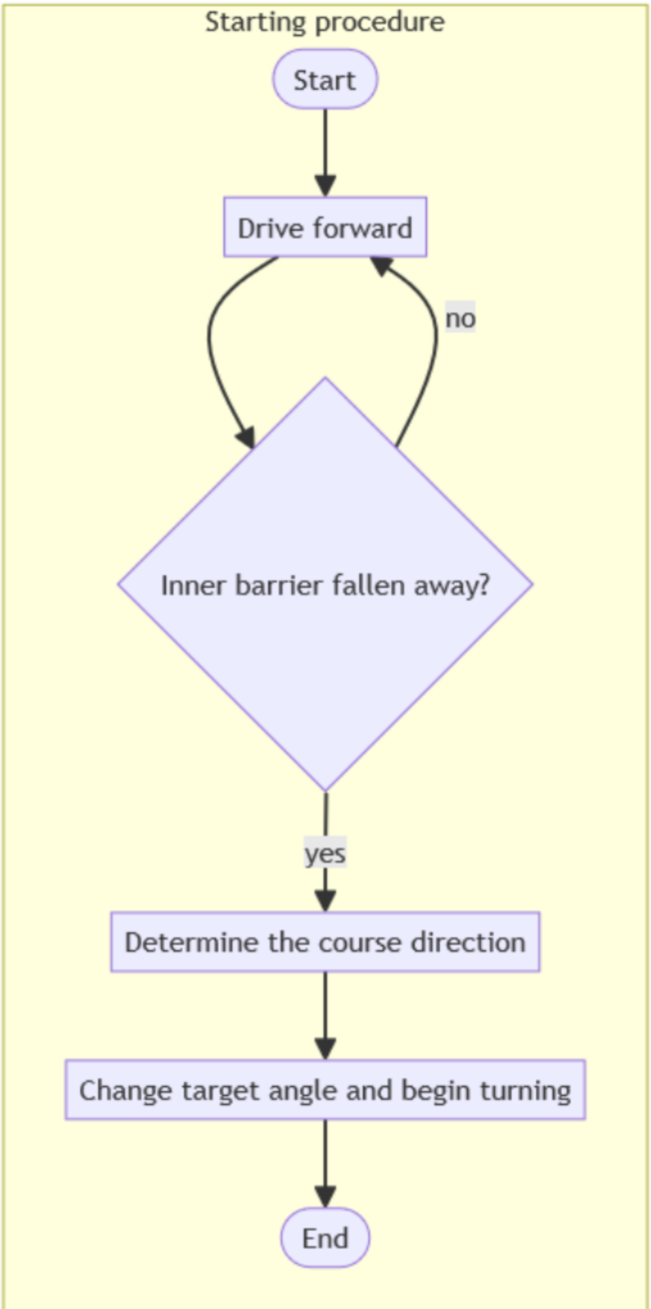


Figure 3.2 - Flow chart 1 showing the starting procedure to determine the course direction

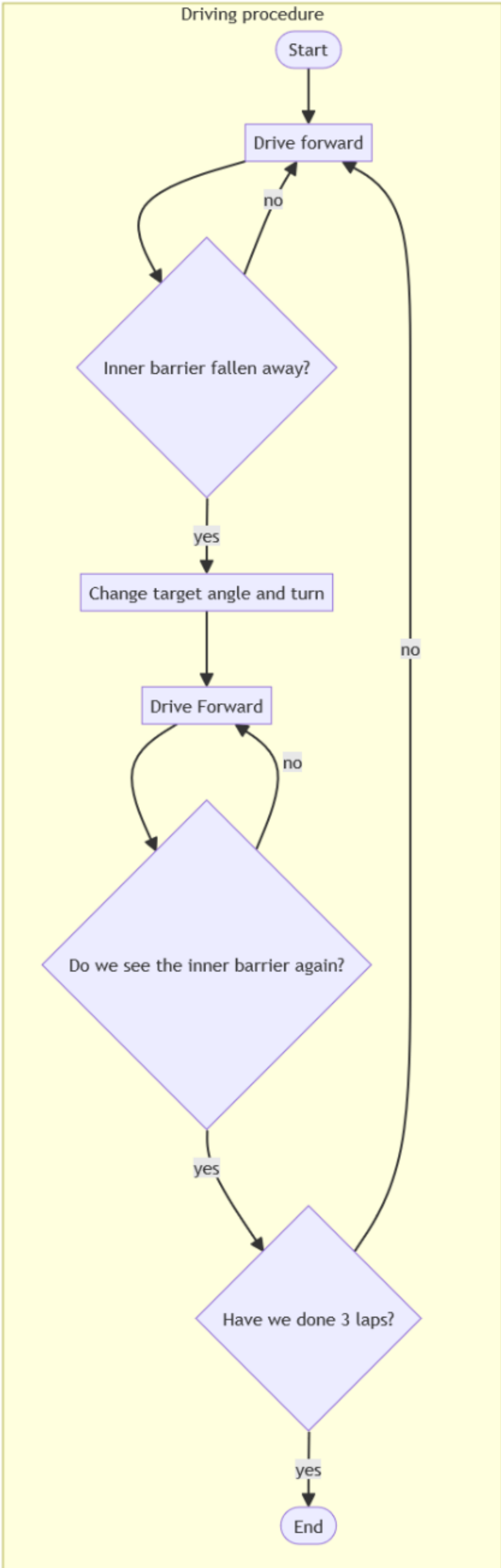


Figure 3.3 - Flow chart 2 showing the logic while navigating the course in the open challenge

Obstacle Challenge

During the obstacle challenge, the vehicle follows the same steps as the Open Challenge but it also takes into account obstacles detected by the camera board.

The avoidance of obstacles is divided into two main sections, avoidance of obstacles while the vehicle is travelling in a straight section, and avoidance of obstacles while the vehicle is turning.

The course is divided into three logical lanes, the left, right and centre lanes. The left lane is as close to the left wall as possible, and vice versa to the right. The centre lane is slightly biassed towards the outside lane, be it left or right, depending on the direction around the course. This is to facilitate the reuse of code between the lack of object and travelling in the outside lane in the presence of the parking block.

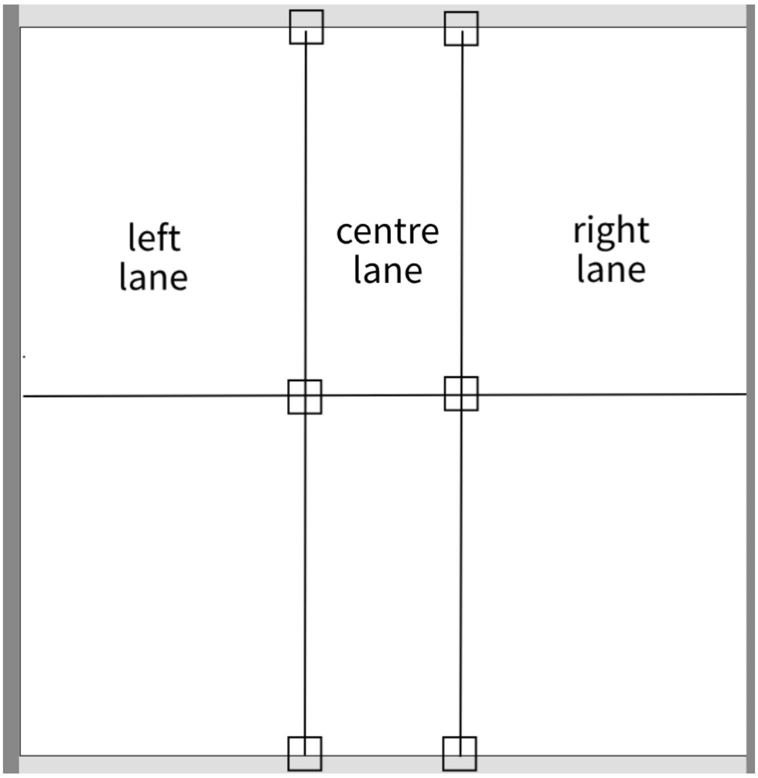


Figure 3.4 - Diagram showing how the straight sections of the course are logically divided

While the robot is navigating the course, it is building a map of it. This increases the reliability of obstacle handling after the first round and also allows us to reliably locate the parking bay on the course(We need to be able to avoid the parking bay until the last round where the vehicle needs to park).

Obstacle Avoidance in Straight sections

Just after the first obstacle has passed, a frame is captured to the cache. Based on this frame, the vehicle will make the decision to navigate to the left or right lane in the same manner.

- Should the lane the vehicle is currently in be the lane it needs to navigate to, no navigation action will be made. Should a lane change be necessary(See diagram 1 below), the vehicle will angle itself at 45 degrees, and use the appropriate TF-Luna sensor to measure its distance from the wall it is approaching(See diagram 2 below). This will allow it to successfully navigate to and hug the wall on the side of the course as dictated by the colour of the block observed.

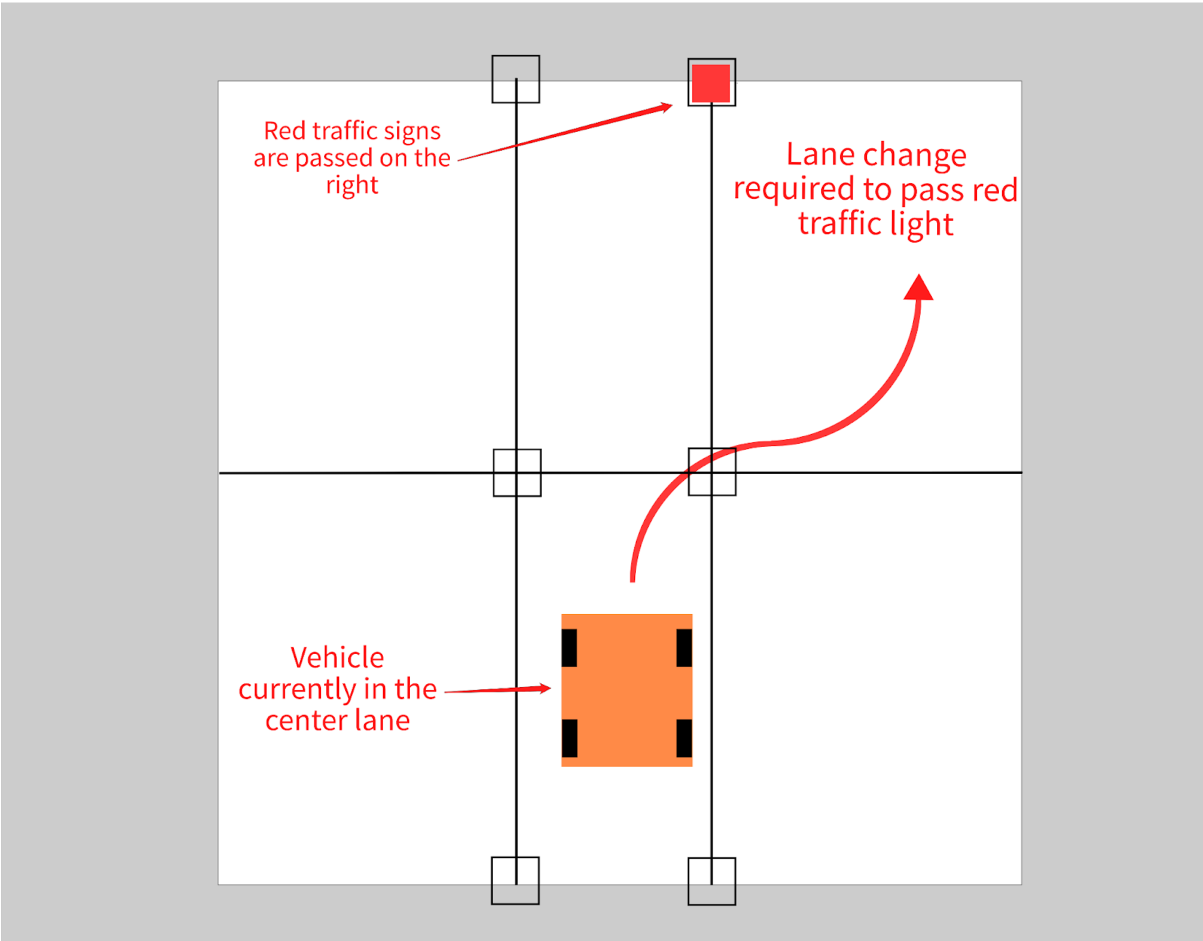


Figure 3.5 - Diagram 1 showing an example case of when a lane change is required

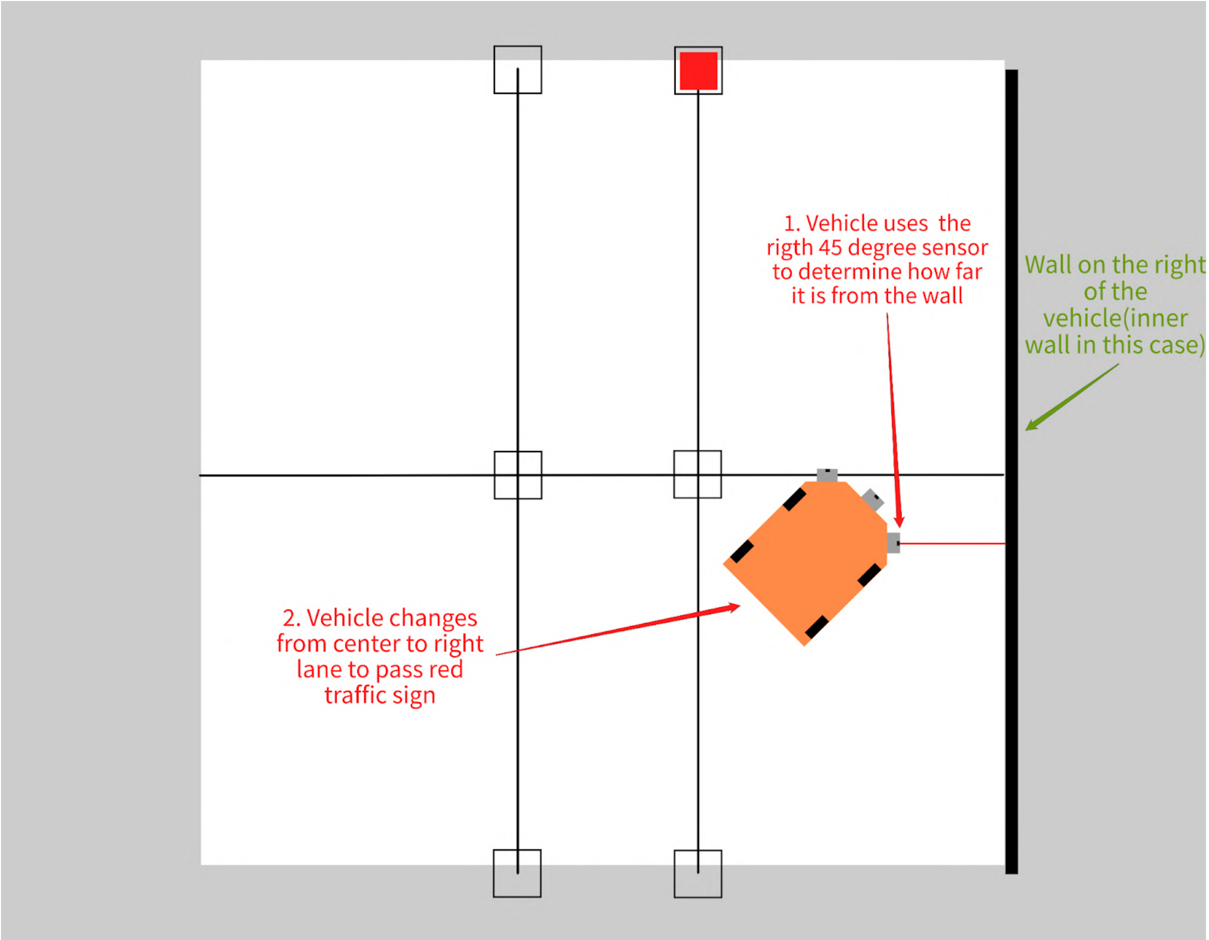


Figure 3.6 - Diagram 2 showing the execution of a lane change

- Should no block be observed, the vehicle will remain in the lane it is currently in.
- Should the parking area be observed, the vehicle will make no adjustment if the target lane is the inside lane (whether this is the left or right lane depends on the direction the vehicle is travelling). If the vehicle is targeting the right lane, or is currently inside the right lane, the vehicle will target the centre lane instead(See diagram below).

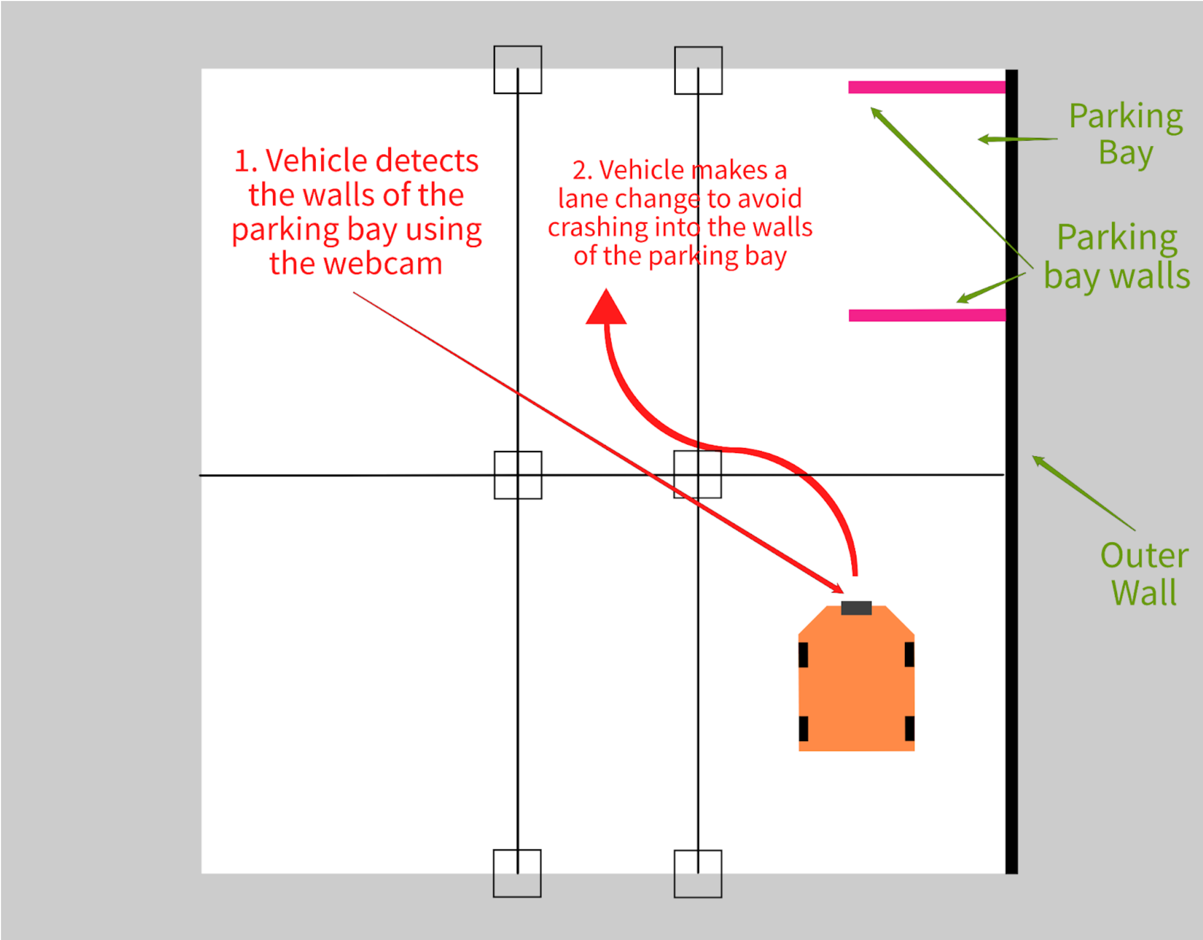


Figure 3.7 - Diagram showing how the vehicle avoids crashing into the parking bay area during the run

Obstacle Avoidance in Turns

The position in which the frame is captured depends on the vehicle position. Should the vehicle be in the inside lane(whether we are in the inside or outside lane is determined by the direction we are navigating the course. See label 2 on next page), the frame will be captured before the turn is initiated(See label 1 on next page).

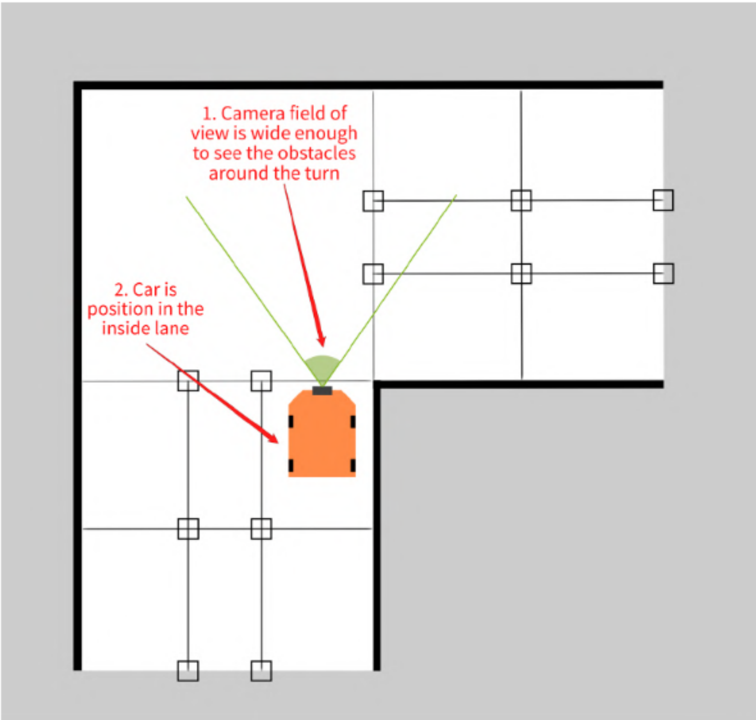


Figure 3.8 - Diagram showing the effective field of view of the web camera for obstacle detection around a turn(Note that this diagram does not represent the actual field of view)

Should the vehicle be in the outside lane, the vehicle will turn 60 degrees(See diagram below) before capturing a frame. Based on the contents of this frame, the vehicle will make one of six unique movements depending on the current and target lanes of the vehicle.

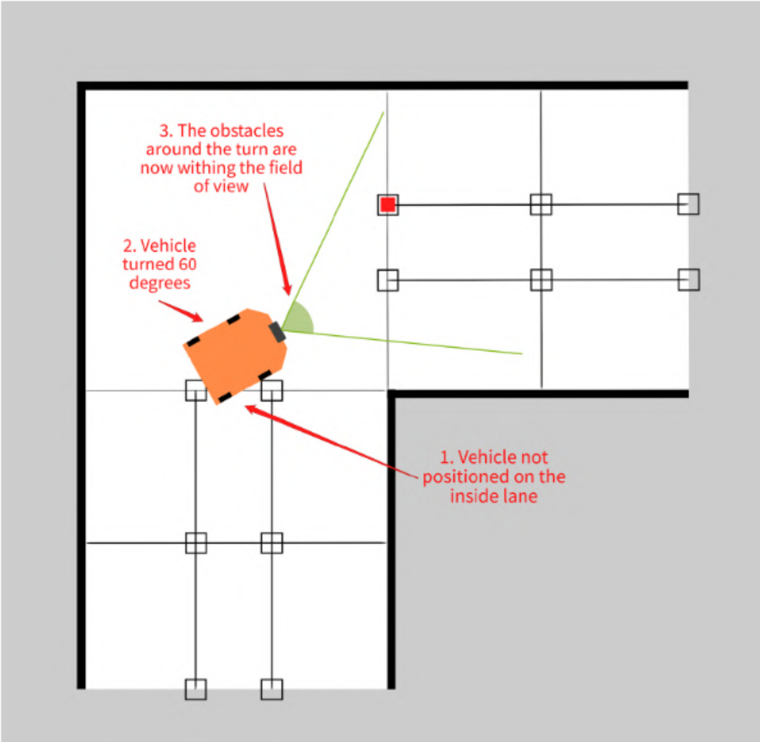


Figure 3.9 - Diagram showing how the vehicle needs to turn 60 degrees if it was not in the inside lane.

Due to the method of turning, while originating from the inside lane, the vehicle will never hit the parking area, due to it basing its turn off of the position of the block should the outside lane be targeted. This naturally puts it into the centre lane, should the parking block be present.

Should there be no block present(See label 2 below), and the robot originates in the inside lane(See label 1 below), the robot will use its front TF-Luna to determine the correct point at which to turn(See label 3 below).

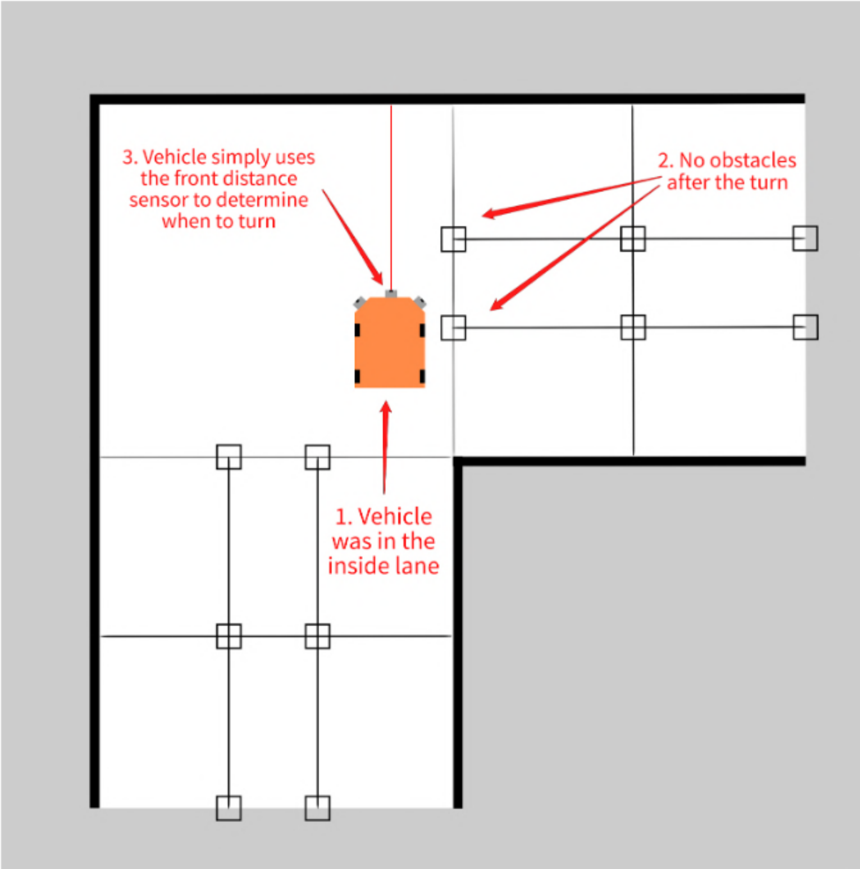


Figure 3.10 - Diagram showing how the vehicle simplifies the turning procedure if no obstacles are detected around a turn. The vehicle will turn into the outer lane

Should the parking bay be present, and the robot originates from the outside lane, the centre lane will be targeted instead of the outside lane. The vehicle then follows the same steps as previously mentioned.

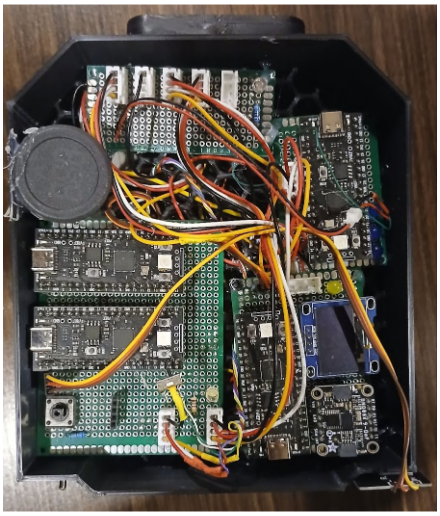
Parking Challenge

When navigating to the parking area. The robot will hug the inside wall until it reaches the segment where the parking is located. The robot will then navigate to the centre lane and use the outside TF-Luna sensor to determine the location of the parking area. It will then swing into it directly and stop the motor after 3 seconds. Due to the previously built map of the course, the camera is not needed for this operation.

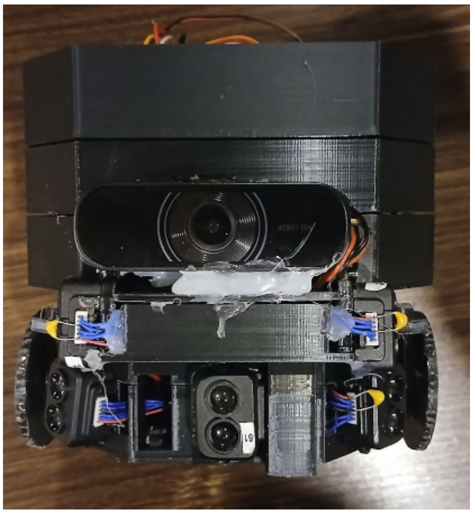
Photos of the Completed Vehicle

Additional (and higher resolution) images can be found here.

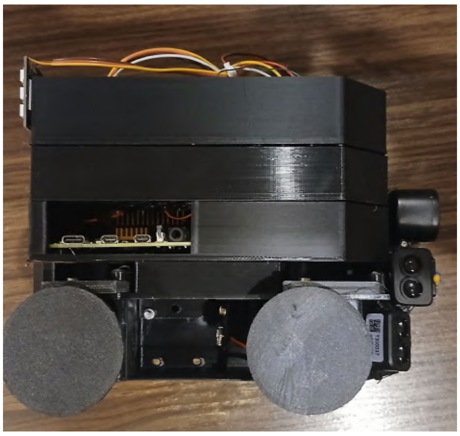
Top View

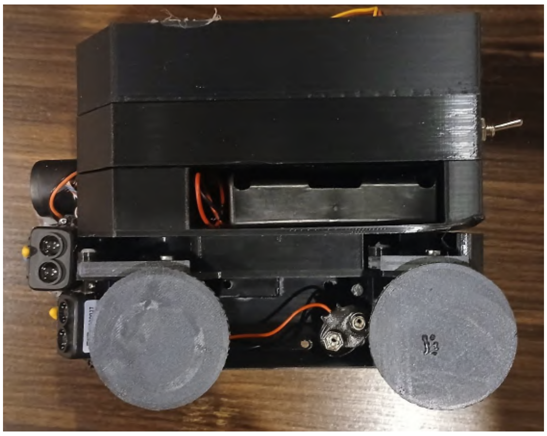


Front View

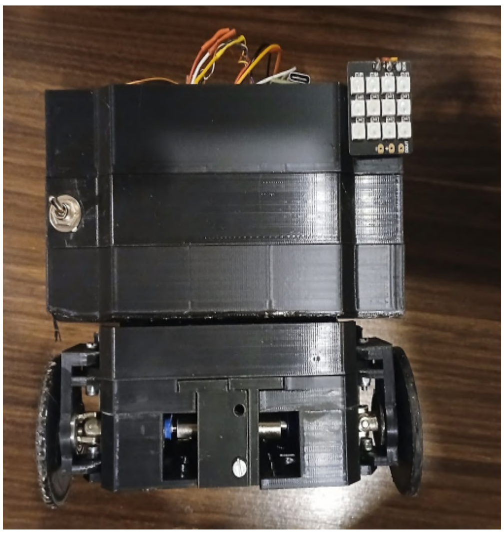


Side Views

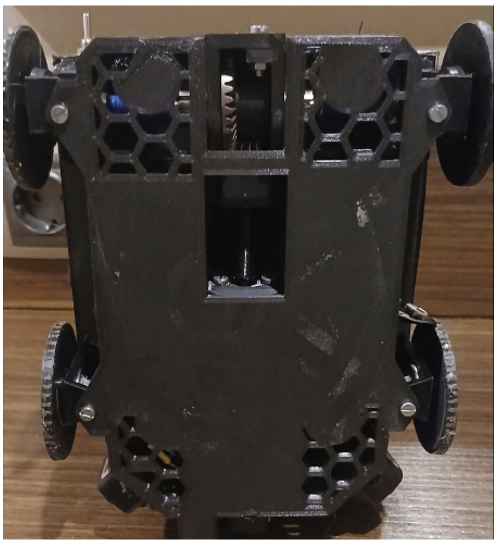




Back View



Bottom View



Photos of the Team

Formal photo of team (back) and club head (front)

Informal Photo

Performance Videos

Please see attached link for Team Null vehicle performance videos.

Open Category Clockwise:
https://drive.google.com/file/d/1u9WIaTdv1vVr_2TocBf5nXeMcmsZVSX0/view?usp=sharing

Open Category Anti-clockwise:
https://drive.google.com/file/d/1rUEM5G98ogTVsLS80-4uWdPW7KKapgZX/view?usp=sharin
g

Obstacle Course: Clockwise:
https://drive.google.com/file/d/1A3ZOPgexcJJU8Gloadm9QQgIXQ2pb4we/view?usp=sharing

Obstacle Course: Anti-clockwise:
https://drive.google.com/file/d/18CCVhjMQ0LsvGY3-RUxEsRtBEZ069Jtl/view?usp=sharing

Parking:
https://drive.google.com/file/d/1vMxz0yG7lwOoRhPndcRS1Icij8LYPWNm/view?usp=sharing

GitHub Utilisation

The Team Null Github for this project is available here.

Appendix A

Bill Of Materials:

Material	**Quantity**
Major Mechanical Components	
PLA Filament	Approx 500g
56mm tyres	4
JGA25-370 Geared motor DC motor 12V electric gear motor high torque 282 rpm	1
Metal Differential And Drive Shaft Set For SCY 16101 16102 16103 16201 Pro 1/16 Brushless RC Car Upgrades Parts Accessories	1
MG996R (180 degrees) Servo Motor	1
6.125mm-12mm single channel ball bearing	4
Minor Mechanical Components	
20mm M2.5 standoff	4
6mm M2.5 bolt	4
5mm M2.5 standoff	4
5mm M3 standoff	4
10mm M3 standoff	2
M3 Nylon nut	2
M3 standard steel nut	9
M3 lock-nut	8
6mm M3 threaded Inserts	8
6mm M3 bolt	10
8mm M3 bolt	4
10mm M3 bolt	2
12mm M3 bolt	5
16mm M3 bolt	4

25mm countersunk M3 bolt	1
8mm M1.25 bolt	6
M1.25 standard nut	6
Major Electrical Components	
Raspberry Pi Pico	4
Raspberry Pi 4	1
Wide Angle 1080p Webcam	1
TF-Luna Time of Flight Sensor	5
Adafruit 9-DOF Orientation IMU Fusion Breakout	1
64x32 I2C OLED Display	1
18650 Battery Cell	6
2x 18650 battery holder case	3
3A Adjustable buck convertor	4
5A Adjustable buck convertor	1
5V relay module	4
LR7843 Isolated FET Module	1
4x3 LED matrix	1
Minor Electrical Components	
JST-XT 6 pin male-female connector set	2
JST-XT 4 pin male connector	7
JST-XT 4 pin female connector	9
JST-SM 3 in male-female connector set	1
JST-Micro 2 pin male-female connector set	1
5mm through-hole LED (Various colours)	19
2x20 2.54mm female header	1
1x40 2.54mm female header	10
1x40 2.54mm male header	10
10kΩ Resistor	5
1MΩ Resistor	1

Through-hole momentary switch	1
Solder	
Wrapping wire	

Printing Instructions

All parts for the vehicle were printed on an Ender 3 V2 Neo 3D printer with a 0.4mm nozzle in standard PLA (Polylactic Acid) filament.

All the part files that need to be printed are available as .step files at https://github.com/Michael-she/Chiuaua-MK6/tree/main/models/Components

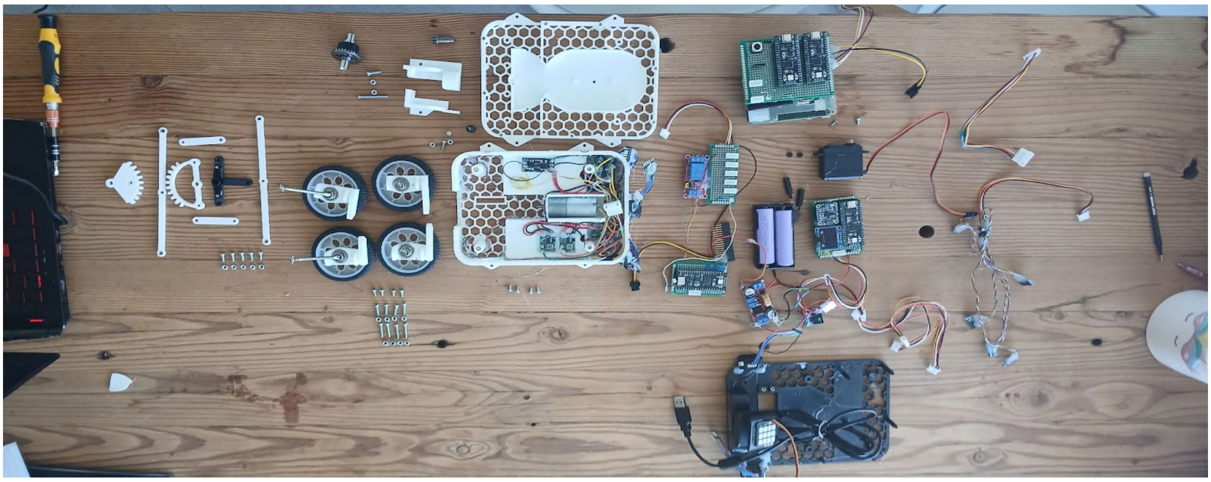
The parts were printed with tree supports, at a layer height of 0.16mm, 0.8mm wall width and a standard infill of 25%. This is with the exception of the printed motor shaft, which was printed with 0.4mm walls and 100% infill. This decision was made after the motor shaft printed with standard settings sheared off, leaving a piece of it embedded in the differential drive gear, which was very tricky to remove.

The files were sliced using Cura Slicer. Should you want to copy the configuration exactly, the Cura configuration is available here.

Some of the files are printed with TPU and PETG. They use slightly different Cura settings. The links to those configurations are available here and here respectively.

Some files may have to have their orientation changed away from the default for optimum printing performance. This is usually achieved through rotating the part to align any large flat faces against the bed, thus minimising overhangs.

Assembly Instructions

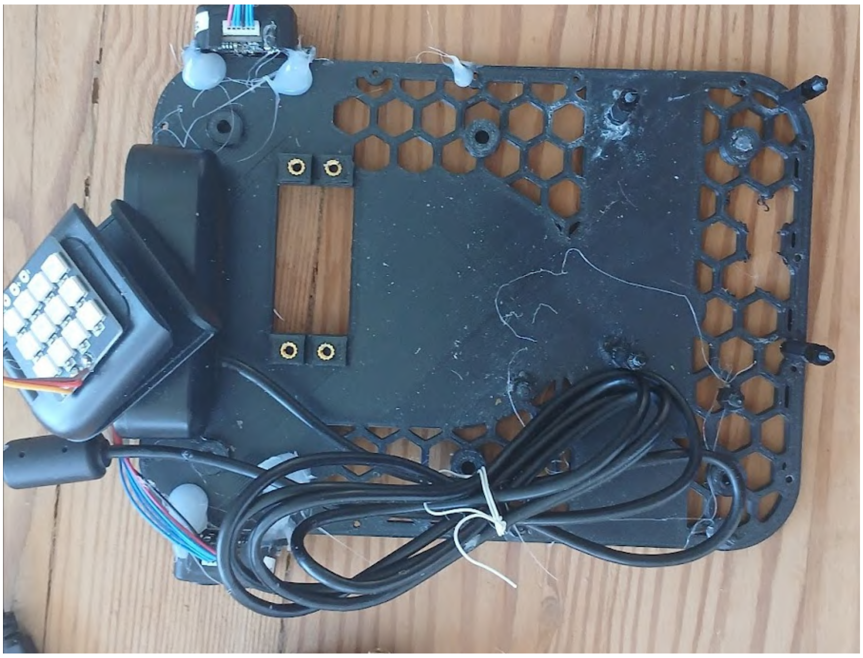


Overview of all the parts of our vehicle

Once electrical components have been assembled as specified by the circuit diagrams (covered later in the document). The assembly of the mechanical components can begin .

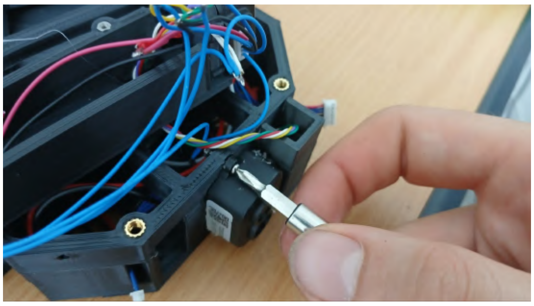
Please note, some of the photos show components already mounted, these components have been glued down and could not easily be removed for the purpose of this document. All series of images explaining a process should be read left to right, top to bottom.

Top Frame



Picture of the top frame where our microcontrollers and some sensors are mounted

1) Attach the TF-Lunas to the 3d printed frame. When these actions are complete, the result should resemble the image above. Below are close-ups of each component in their position:

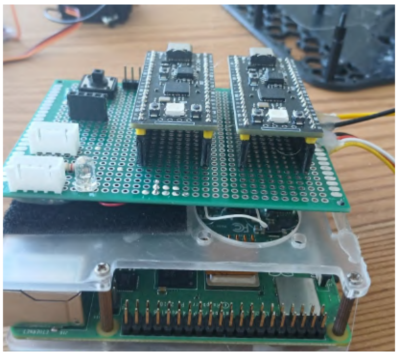


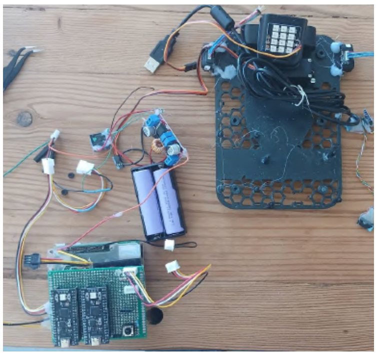
2) Glue the 4x3 LED Matrix to the back of the webcam mount as shown below.

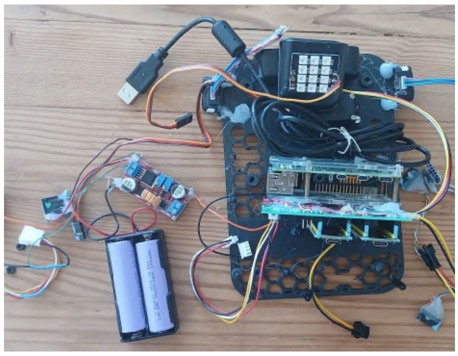
3) Push 4, 6mm M3 threaded inserts into the holes surrounding the servo mount. The result should be as pictured in the image on the right.

4) Attach the steering servo, the servo should be placed in the hole, and secured with two 6mm M3 bolts. The image below shows this process in progress.

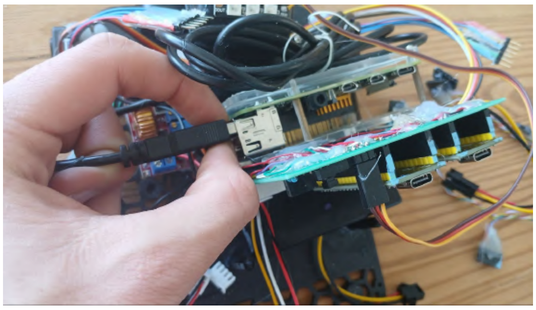
5) Use a small amount of hot glue to attach the Raspberry Pi and main board to the top frame as shown in the images below.

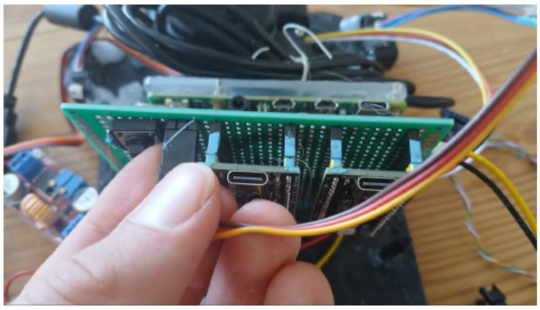


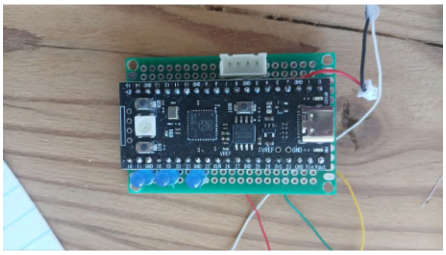
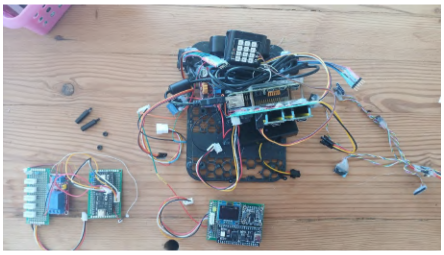
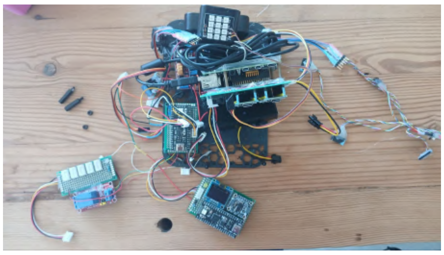




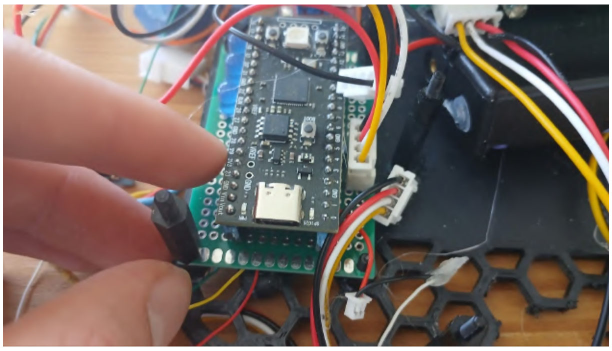
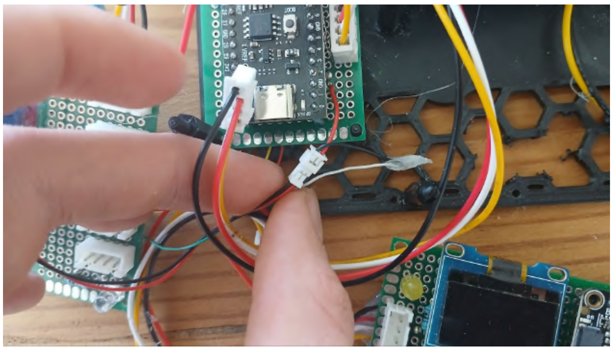
6) Connect the webcam and LED matrix to their relevant ports as shown below.

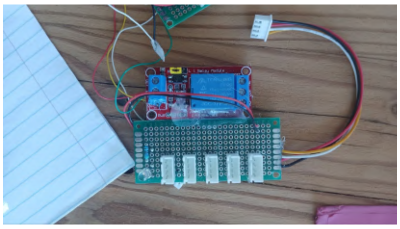
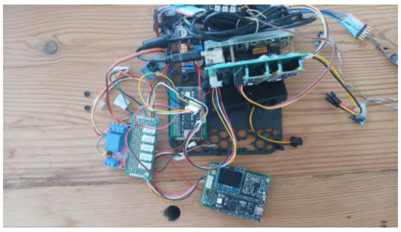
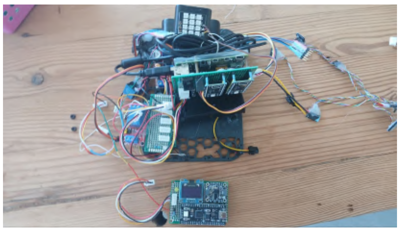




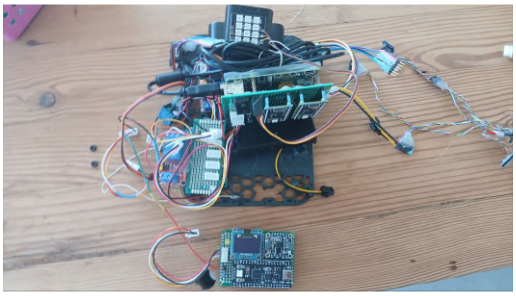
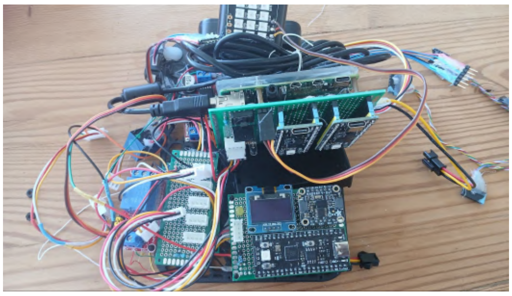
  

7) Attach the 10mm M3 standoff to the 5mm M4 standoffs holding the motor control board. This should secure the board in place and provide a mounting point for the wiring hub and motor direction relay. Attach the communication wire (male-male 4 pin JST-XH) to the motor board now, as it will be difficult to reach it later. Attach the wiring hub and motor direction relay to the standoffs above the motor control board. The process is outlined in the series of images below.

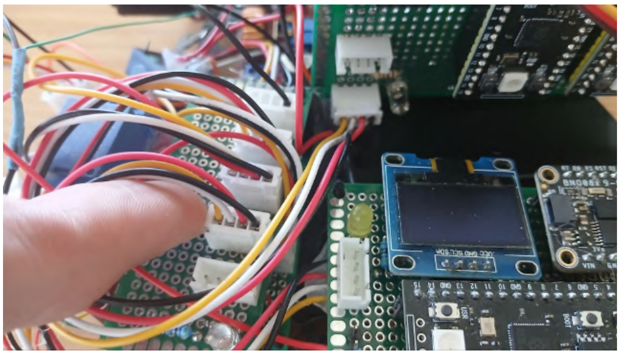
8) Attach the 4 20mm M2.5 standoffs to the 5mm M2.5 standoffs. Place these through the mounting holes on the gyroscope board and secure with the M2.5 standard nuts. Place this entire assembly onto the upper board as shown in the below images and glue the standoffs in place

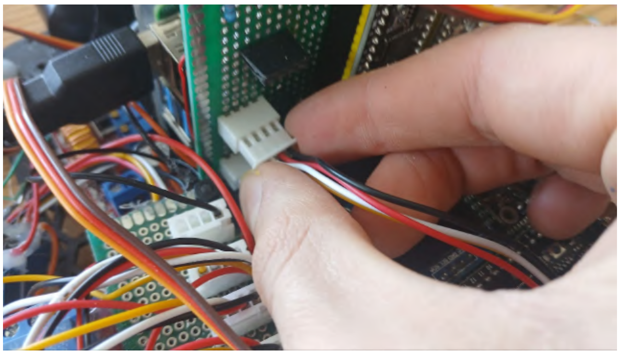
 

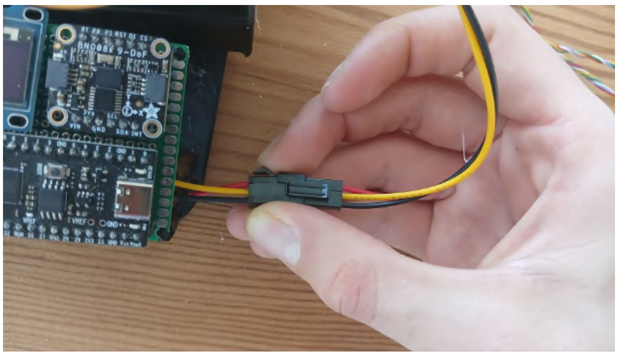
9) Connect all boards to the wiring hub. Connect the Pi to the main board. Connect the servo to the gyroscope board. Attach the TF-Luna wiring harness to the main board

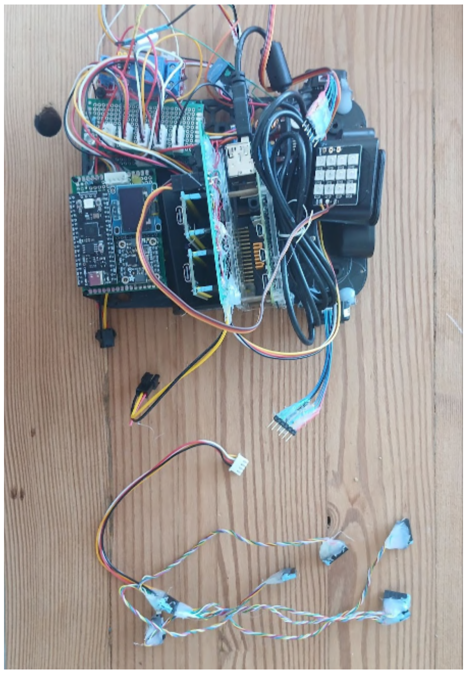
Lower Frame

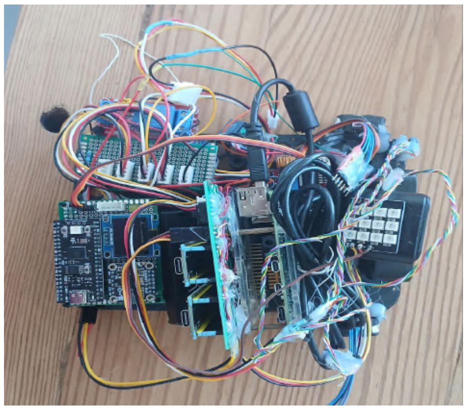
1) Mount all the 3A voltage regulators in the frame, as well as two of the 5V relay modules. These should be connected as specified in the wiring diagram. Although it is not necessary, 12 downward facing LEDs were added that are used to indicate the switched on state of the vehicle. Holes were also drilled in the bottom of the frame to











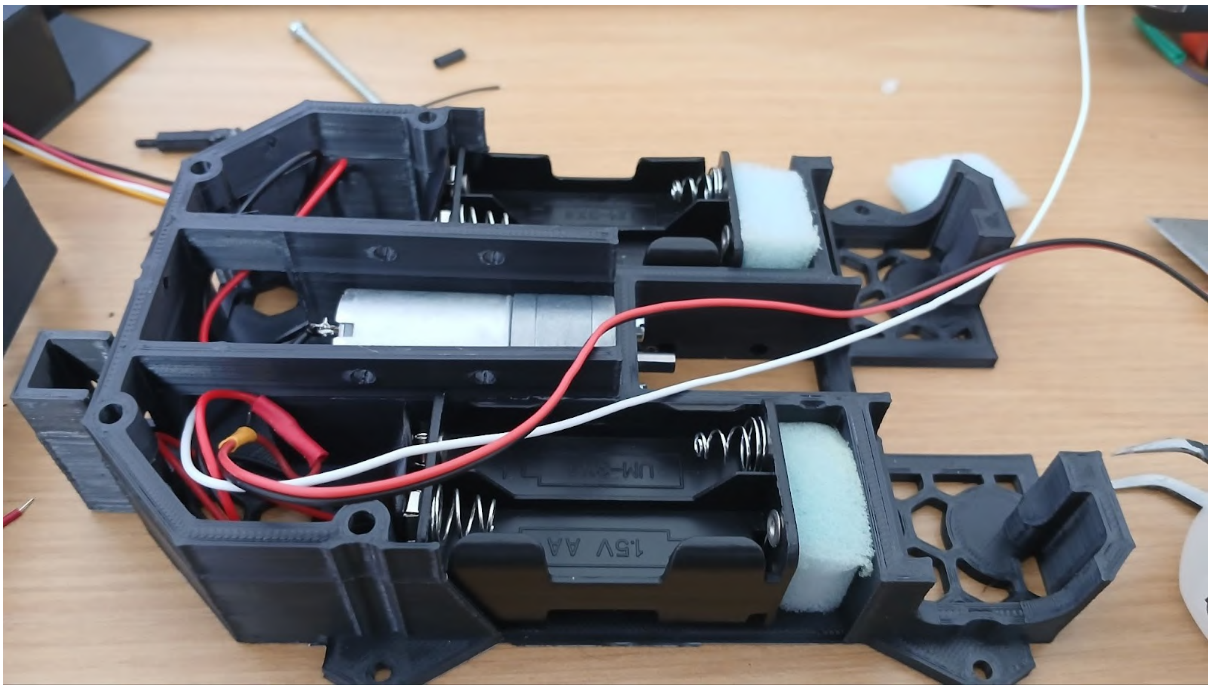
provide access to the power switch and a voltage monitor.

2) The motor must also be attached to the lower frame. This is done by aligning the motor with the boltholes, lowering it into the frame against the back wall before

pushing it forward; it is then secured using two M3 6mm bolts. The motor may have to be rotated slightly in its mount to correct the alignment with the mounting holes.

3) The front facing TF-Lunas need to be mounted. Use a small amount of hot glue to attach a sensor facing directly forwards, left 45 degrees and right 45 degrees. Although the up down direction does not need to be closely monitored (as long as they are aligned with the frame itself, it will be fine) the left-right angle of the two angled sensors needs to be closely monitored as difference in angles between these sensors will cause differences in behaviour between the two directions of the vehicle.

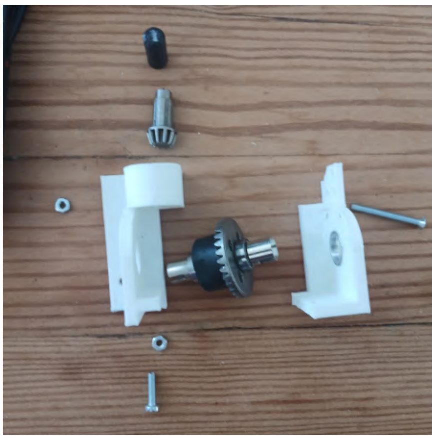
The **final product** is pictured below:

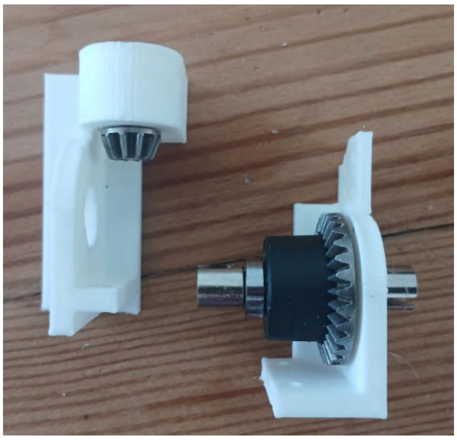


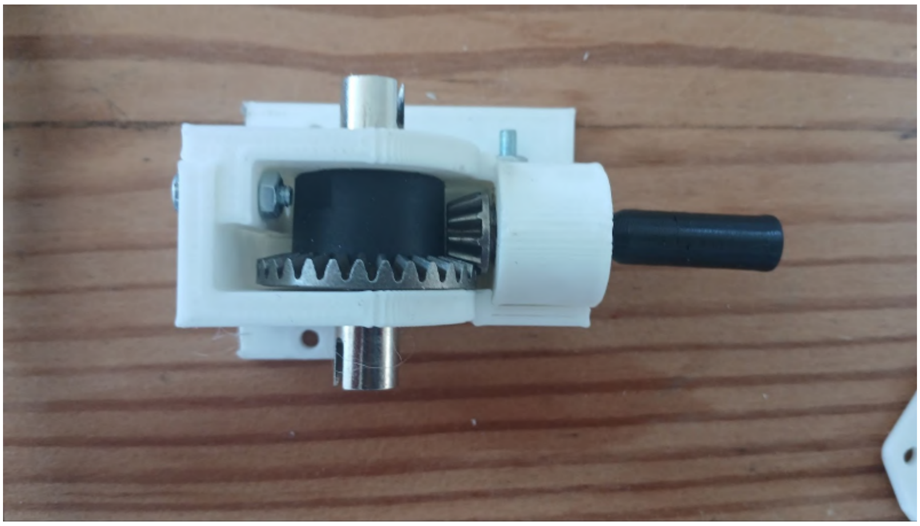
Middle Frame

1) Assemble the wheel subassemblies. The outside surface of the bearings should be coated with two-part epoxy before being inserted into the 3d printed wheel mounts. Care needs to be taken not to get any of this epoxy into the bearing mechanism, which will stop it from rotating. Once the epoxy is dry, the wheel shafts can be inserted into the bearings, the wheels can then be mounted onto the shafts, and secured with the included bolt. This should be repeated for each wheel. Ensure that the back wheel shafts are mounted to the back wheel steering mounts. The image below shows the final product of this process

2) Assemble the gearbox frame. Place the gears into the frame as shown in the first two images. Carefully combine the two halves of the gearbox, ensuring that the gears mesh properly. Once the gearbox is combined, thread the 30mm countersunk bolt through the hold below beneath the small gear, before securing it with a nut. The same can be done with the 12mm bolt and the top hold on the left side of the final image. Secure this with a nut, taking care not to drop the nut into the frame, as it can prove difficult to remove. Once the fame has been fully assembled, the motor shaft can be inserted into the hole in the small gear. If the assembly is correct, rotating the motor shaft should cause all the gears to rotate

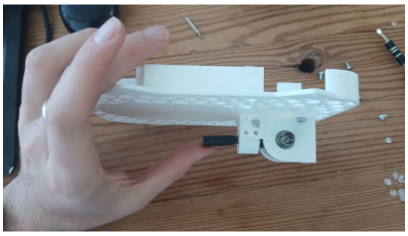






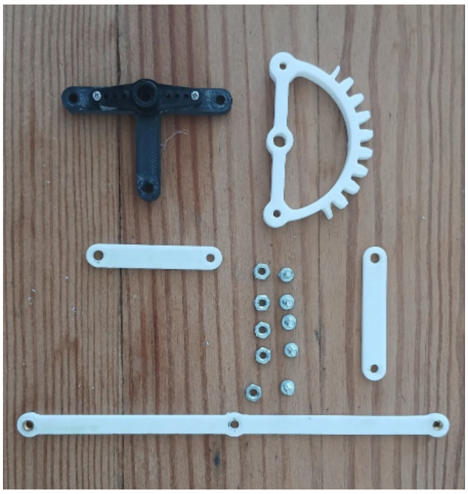
3) Mount the gearbox to the middle frame. Insert two 10mm bolts into the mounting holes as shown in the 2nd image. The gearbox frame should be mounted to the middle frame, with the plate against the middle frame. Once secure with bolts, the final assembly should resemble the 3rd image.

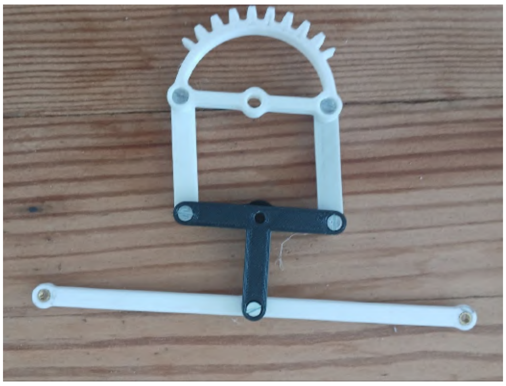




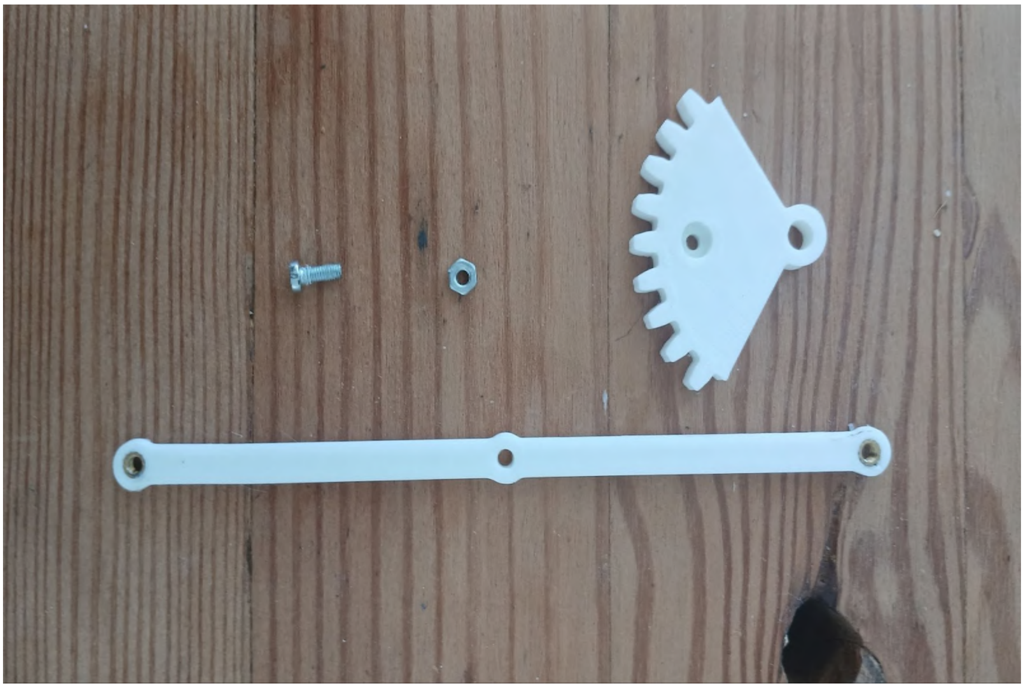
4) Insert M3 threaded inserts into the 4 outer holes of the steering rods. This is pictured on the right.

5) Assemble the forward steering mechanism. Glue the servo horn onto the steering actuator (black component in the left image). Place five 8mm M3 bolts into the three countersunk holes on the steering actuator and the two 4 countersunk holes on the gear. The image below left shows the result of this process. Place the spacers over the bolts on the in-line holes on both the gear and steering actuator. Place the steering rod onto the forward bolt on the steering actuator. Secure all the bolts in place using standard M3 nuts. Tighten the bolts until the mechanism can barely hold its shape when held at 90 degrees in the air, do not tighten it beyond this point. Place a small amount of hot glue on the tightened bolts to prevent them coming loose as a result of movement and vibration.



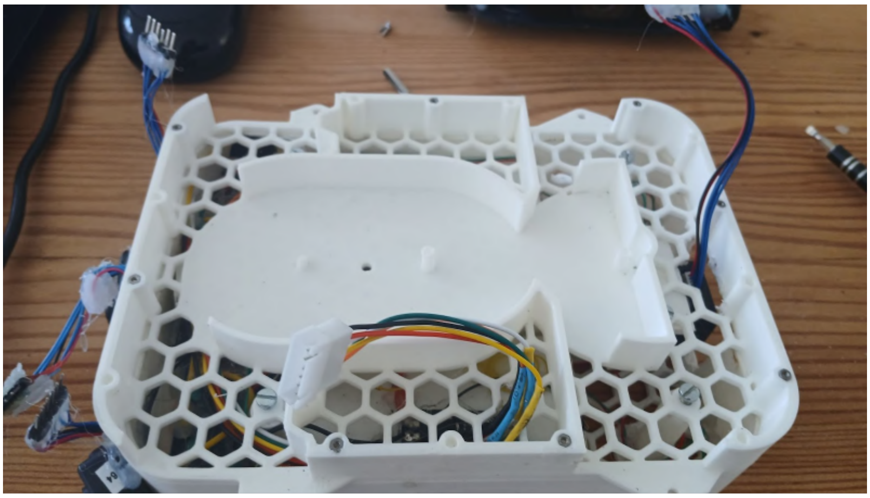



6) Assemble the rear steering mechanism. Place a 8mm M3 bolt into the countersunk hole on the gear, place this bolt through the steering rod. Tighten an M3 nut onto this bolt until you can easily push the steering rod with your finger, but it stops immediately after the pushing stops. Place a small amount of hot glue onto the nut. The images below outline the above process.

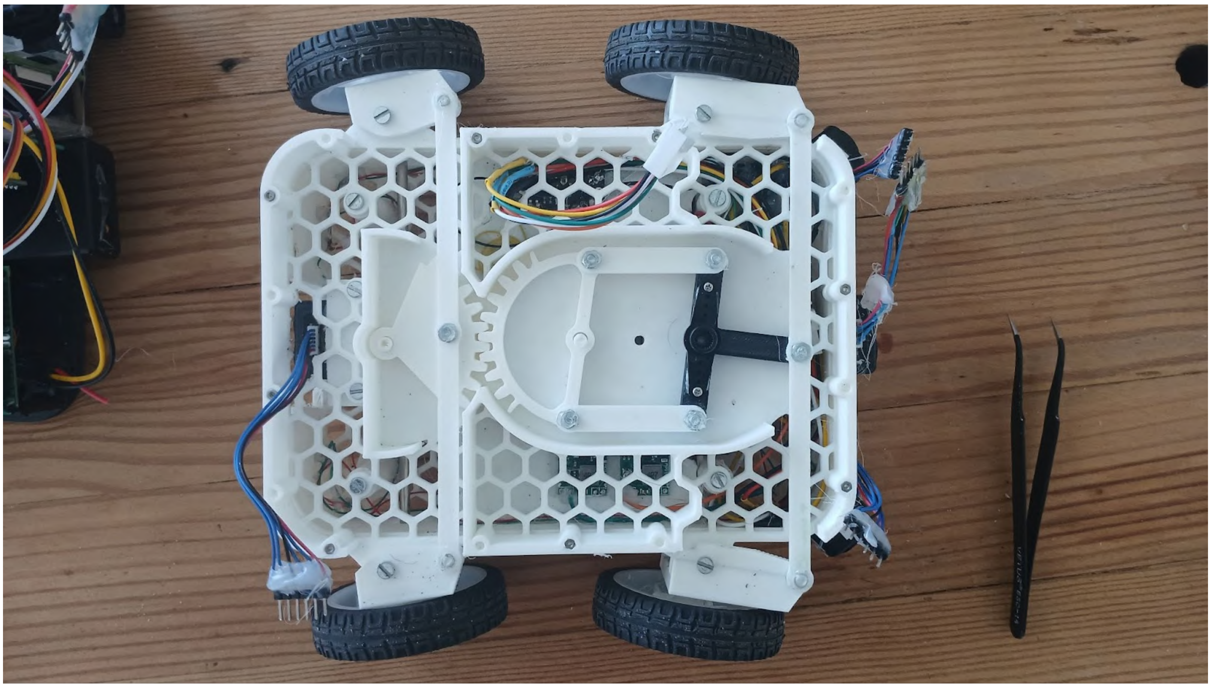


Final Assembly

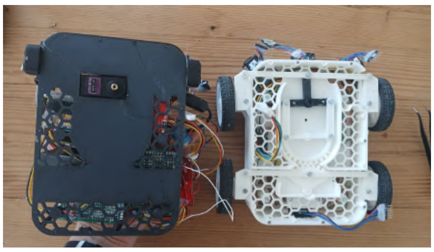
1) Attach the middle (main) frame and the lower frame. First, the pass-through connector needs to be connected between the middle and lower frames. Then, the frames need to be attached. This is achieved by aligning the rotational position of the motor and motor shaft. The main frame is then lowered onto the lower frame, with the gearbox mounted in-line with or touching the back wall of the lower frame. Once the main frame is resting on the lower frame, the main frame is pushed forwards, engaging the motor and motor shaft. Once this is complete the frame can be secured in place using 4, 6mm M3 bolts. The result of these actions should be represented by the image below.

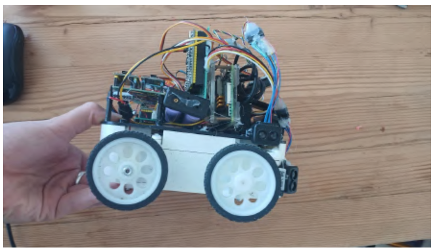


2) The wheels then need to be mounted to the combined frame. The main frame is placed between the two upper sections of the wheel mount and the lower section of the wheel mount rests on the lower frame. A 16mm M3 bolt is then inserted into the upper hole and secured with a lock nut. The same is repeated with the lower section, using a 12mm M3 bolt. These bolts should be tensioned similarly to those on the steering mechanism. When mounting the back wheels, ensure that the shafts engage the gearbox when mounting.
3) Place the steering mechanisms on the frame. The mounting points are sized so that there is only one position in which it will fit.
4) Insert 6mm M3 into the countersunk hole on the wheel mount. This should be screwed into its corresponding threaded insert on the steering mechanism. This should be repeated for all 4 wheel mounts. Place a small amount of hot glue on the bolt/threaded insert to prevent it from vibrating loose. Care needs to be taken in this step not to invert the vehicle while the steering mechanism is unsupported as it may fall out. The final product should look like the below image:



5) Mount the top frame to the rest of the vehicle. The wheels should be straightened and the servo centred prior to this operation. The top frame can then be carefully lowered onto the rest of the vehicle. Gently press down on the servo to engage it with the servo horn. This assembly can then be bolted down using the 8mm m1.25mm bolts. The process for this is outlined in the images below.





Control System Assembly

The control system for the vehicle was built using perforated boards, 2.54mm headers and wrapping wire.

This system consists of 4 individual boards that need to be manufactured.

1. Motor Control Board
2. Gyro Control Board
3. Main Control Board
4. I2C Parse/Control Board

These boards are connected to the overarching system through 2,3 and 4 pin JST (Japanese solderless Terminal) connectors. These connectors, although useful for serviceability, are quite bulky and not particularly rugged. They were therefore mostly omitted from the particular vehicle that competed.

In our case, we decided to combine the main control board and the I2C parse/control board. The boards were designed using EasyEDA. The schematics in PNG form as well as the original project can be found here

1. Motor control board

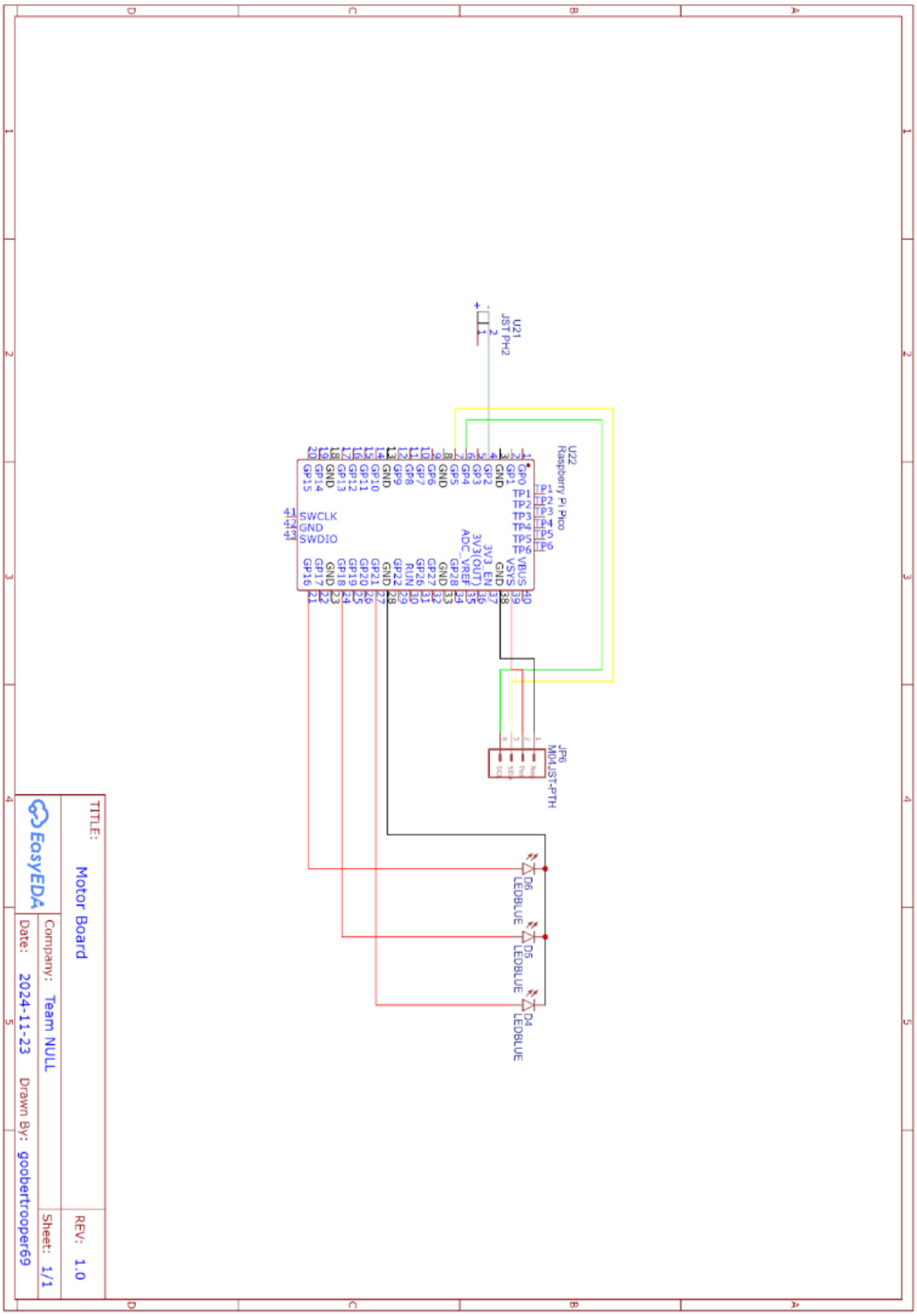


Figure 7.1 Motor board circuit schematic

2. Gyro Control Board

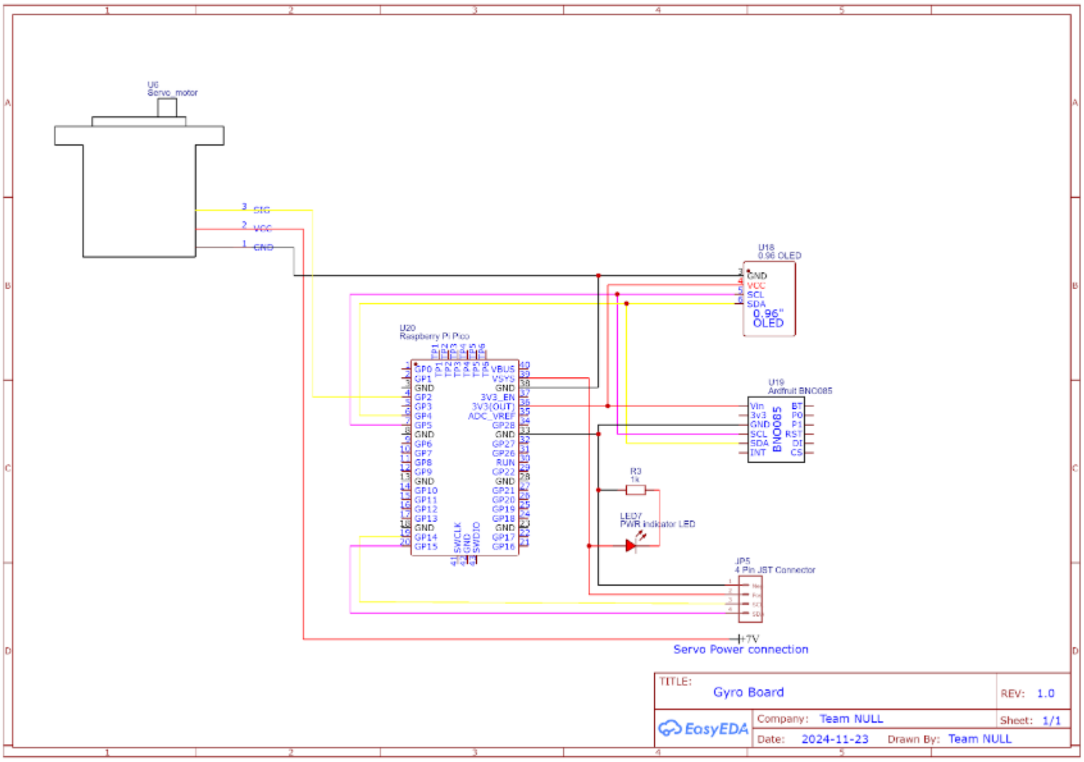


Figure 7.2 - Schematic of gyro control board

5. Main board

The main board refers to the main pi pico and the I2C pi pico. Both of these pi pico's have been placed on one prototyping board according to the following schematic.

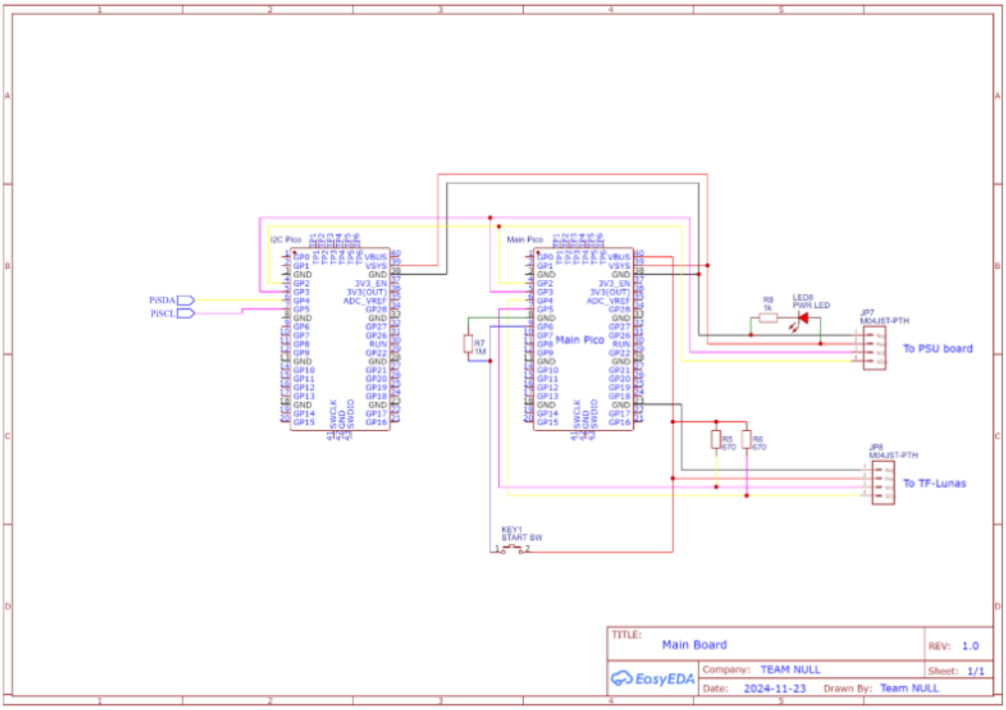


Figure 7.3 - Schematic of main control board