Team ID:

Team Name: **Team Monke**

Team Members: **Lim Jun Hong David, Chong Qi Yuan, Sachin Ilangovan**

TEAM MONKE ENGINEERING JOURNAL

WRO Future Engineers (FE)

ABSTRACT

The purpose of this competition is to design, fabricate, assemble, program, integrate and operate a self-driving car using computer vision.

This vehicle will be made using Raspberry Pi as its main controller and other off the shelf electronic components. The vehicle will also be operated as a steering robot, like real life cars.

Upon completion of this project, the vehicle should be able to navigate effectively between obstacles using sensor fusion and computer vision.

A copy of this Engineering Journal can also be found on GitHub
Other notes that we took are also found in the doc/ subdirectory.

Contents

INTRODUCTION

Project Background

In an ever-increasing automated world, the use of Autonomous Guided Vehicles (AGV) has become more prevalent. Furthermore, with recent improvements in Artificial Intelligence, specifically Computer Vision (CV), the integration of CV with AGVs is in high demand. Some possible applications with this integration include:

- Commercial: Self-Driving cars, housekeeping robots
- Industry: Fully autonomous factory robots
- Defence: Autonomous drones

Objectives

This project aims to create a self-driving car using Computer Vision via OpenCV and a Raspberry Pi camera. This will be assisted using Ultrasonic sensors to accurately identify distances. Furthermore, magnetometers will be used to determine the heading of the AGV to further refine turns overall sensor integration.

Scope of project

This project will be done using a Raspberry Pi. Therefore, design and fabrication of all components will be required. Integration of electronic components such as sensors, motor drivers, motors etc will also be required.

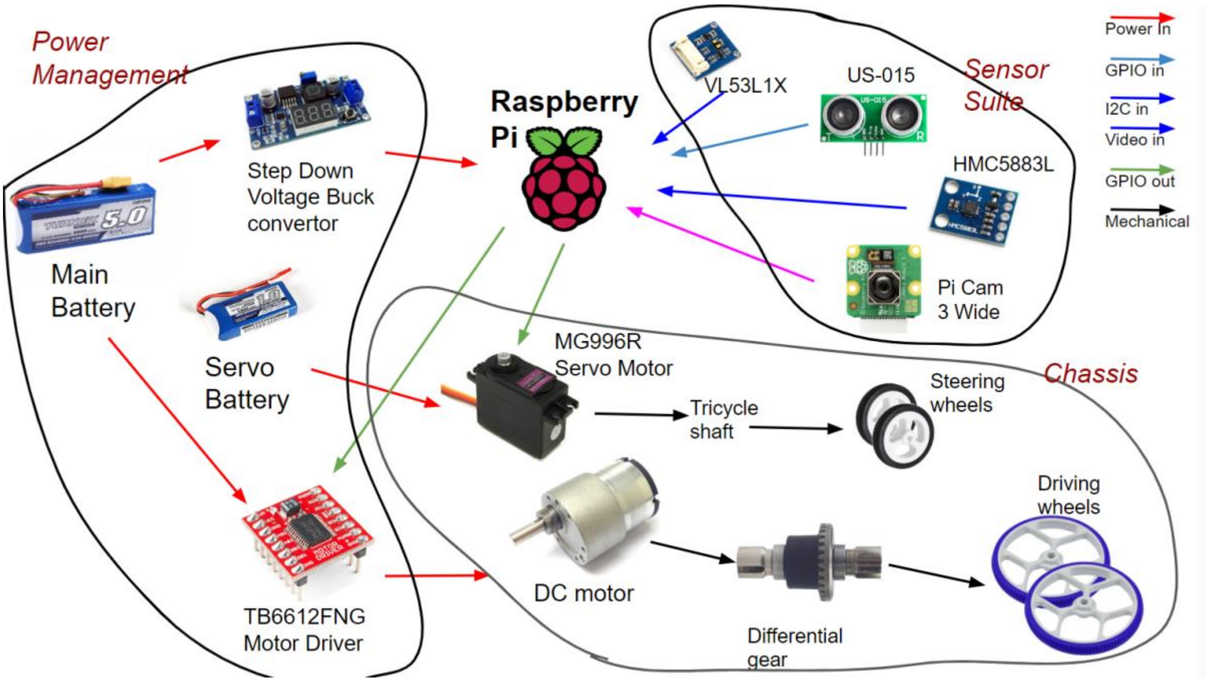
HARDWARE

HARDWARE OVERVIEW

Component	Quantity	Remarks
Raspberry Pi 4 Model B	1	Brain of Vehicle
Custom designed PCB	1	Provide connections, support, and a compact design for easy integration.
Raspberry Pi Camera Module 3	1	Colour detection
TB6612FNG Motor Driver	1	To regulate a motor's speed, direction, and efficiency.
DC 12V 10RPM -1000RPM Gear Motor Gear Box Reversible High Torque	1	provides propulsion to vehicle
Full Metal Differential Gear Set	1 set	much easier when turning a corner
MG996R Servo Motor	1	Provides steering
Turnigy Nano-Tech Plus 1300mAh 4S 14.8V 70C 140C LiPo Battery XT60 3D Multirotor	1	Power source for vehicle components
1000mAh battery	1	Servo Battery
LM2596 3A Buck Module with Display	1	Buck convertor
Bicycle Light Set Type-C Rechargeable 150Lumen	1	Headlight Infront of vehicle to help pi cam detect and differentiate the color better
Ball Bearings 6x13x5 mm	4	Provides smooth rotation for wheels
80mm Plastic Wheels	4	front and back wheels
HMC5883L Module Triple Axis compass	1	
High precision Ultrasonic Range Finder US-015	3	to help avoid obstacle, 1 on front, right and left of the vehicle
Rear steering	2	To steer

OVERALL VEHICLE DESIGN

System Diagram:



Size

In our initial design for WRO SG, our robot was intentionally larger than the parking lot's depth to facilitate direct parking without needing to parallel park. However, we encountered a challenge: the physical components, including the servo motor, differential gears, and DC motor, were too large and long. As a result, we had to custom print a shorter differential gear axle. Despite our efforts, the vehicle still seemed larger than the parking area, requiring us to redesign the layout to better accommodate all the components.

After observing the smaller, more compact designs of our competitors' vehicles, we decided to rethink our design and aim for a more compact form. The primary reason for our robot's original size was the components we used, so our first step was to research alternatives that could deliver the same or better performance but in a more compact package.

Component layout

Our robot retains the dual-level plate configuration from our original design, which provides mounting space for the components. The bottom plate is used for actuation and sensor mounting, containing ultrasonic sensors and the light source.

In the original design, the DC motor, differential gear, and steering axle were all mounted beneath the bottom plate. This configuration resulted in a taller, bulkier robot. Additionally, due to the large size of the batteries, the top of the bottom plate was dedicated to battery mounting, leaving only a small portion for the servo motor.

For the new design, we repositioned the differential gear and motor to the top of the bottom plate. Along with using smaller, more powerful motors and batteries, we managed to reduce the overall size of the robot. This allowed us to free up space for the ultrasonic sensor, which was previously constrained by the larger components.

We also redesigned our steering mechanism, which will be discussed in detail later in this journal.

The final top plate houses the “brains” of the vehicle, including the Raspberry Pi, custom-made PCB, camera, and voltage regulator. These components are placed on the top plate for easy access and maintenance.

MOBILITY MANAGEMENT

Chassis

For our chassis design, our team adopted a modern four-wheel layout, with two of the wheels dedicated to steering. In our initial design for WRO FE SG, we used with a tricycle approach rather than the conventional Ackermann steering configuration. This choice allowed us to reduce the complexity of the steering mechanism while also achieving a significantly smaller turning radius, ideal for navigating sharp angles and tight manoeuvrers. Such agility is crucial in our competition environment, where closely spaced obstacles and pillars demand quick, precise turns in either direction.

The tricycle design, while effective in our first competition, required an increase in vehicle height due to the larger axle size. Furthermore, this configuration resulted in instabilities of the robot during high-speed turns since both steering wheels are sharing the same axle. There was also an issue where the steering axle would break constantly, and the servo motor would be overstrained due to large forces acting through 1 single area. Although this configuration simplified steering and reduced turning radius, we identified opportunities to further refine the vehicle's profile for enhanced stability and manoeuvrability. We aim to lower the vehicle's centre of gravity without compromising its ability to execute rapid turns at high speed and large angles to meet the better performance standards.

Our original layout also diverged from traditional tricycle setups by situating the steering mechanism at the rear, paired with a differential gear system at the front. This unique configuration allowed us to optimize the vehicle's length while seamlessly incorporating front-mounted ultrasonic sensors. The sensor placement near the differential system eliminates the need for additional structural spacing, which would have been necessary, had the sensor been positioned closer to the steering assembly.

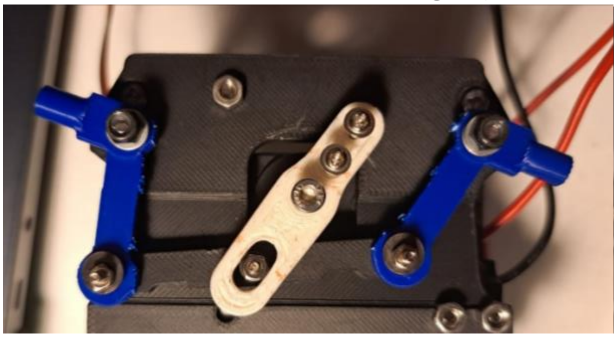
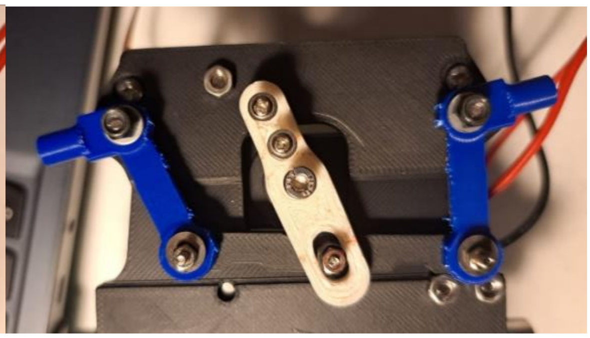
Following additional research and design evaluations, we have decided to transition to an Ackermann steering mechanism with a linkage system. This revised approach will be discussed in more detail in the next section. By integrating the Ackermann mechanism, we retain the advantages of rear-wheel steering, allowing the vehicle to pivot sharply around the front wheels. This modification gives the vehicle a more robust design while enhancing its manoeuvrability in confined spaces, enabling precise adjustments essential for navigating through complex obstacle courses.

Steering

Using an Ackermann steering configuration, our vehicle utilises an actuator link connected to wheel pivots on each end of the mechanism, enabling precise left and right steering control. The Ackermann system used is the positive Ackermann, where the inner wheel turns more than the outer wheel. This is because the inner wheel will have a higher steering angle. The angle was calibrated such that it aligned with the driving wheels to minimise unnecessary slip angle.

With a steering angle of about 37.7 degrees and a Wheelbase of 115mm, our turning radius comes out to be about 152.6mm. However, this is with a conventional steering where the steering wheel is situated at the front, not the back. Even though we reduced the steering angle, we have also successfully reduced our wheelbase. Therefore, our turning radius does not increase too much from our previous design.

To actuate this movement, we designed a custom linkage system where a uniquely crafted servo horn is affixed to the actuator link. This servo horn has a slot that allows the actuator link to slide back and forth as the servo rotates, translating rotational motion into smooth, controlled steering adjustments.

The core of this steering mechanism is the MG996R servo motor, chosen for its reliability and precision. For our requirements, the custom horn needs to pivot about 90 degrees or less. The MG996R servo motor's ability to perform accurate 90-degree rotations makes it ideal for this application, as demonstrated in previous iterations where it provided consistent and responsive steering performance. Given its proven effectiveness, we will once again integrate the MG996R servo into our updated design to maintain the vehicle's agility and ensure a wide range of steering angles for optimal mobility.

Actuation

DC Motor & Full Metal Differential Gear Set: (Old Version)

At the start of planning and design we initially were planning on using a 3V - 6V Dual Axis TT Gear Motor but after some thought we realised that it would not be very efficient and logical as that motor would not have enough power to be able to move the vehicle at a required speed and would be very slow and would also be difficult to implement a differential gear set to it.

During our Singapore WRO FE, after some thought and research we landed on the 12v 380rpm 1.4kgcm Brushed DC Motor. From research, this motor provides higher speed and higher torque. The motor is mounted with a 30:1 gearhead that can produce 137mN.m torque with 380RPm rated 1A,12V. A full metal differential gear set is connected to the DC motor.

The use of differential gears allows the wheels to move at different rpms. They also serve a purpose of speed reduction while effectively transmitting torque. This allows smooth turning and cornering and optimal power distribution by dividing the force equally between them but still allowing them to follow paths of different length when turning. This further decreases the turning radius, reducing tyre wear, and improving stability.

For example, in a scenario where the vehicle must turn right, the left wheel must move a longer distance than the right wheel, which means that the left wheel needs to rotate at a higher speed. This is where a differential would be useful.

DC Motor:	137mNm, 380rpm
Mass of robot:	1.45kg

Torque/kg = 94.5mNm/kg
Power to weight ratio = 1255W/kg
Max speed of 380 rpm = 1591.7mm/s

New Motor Selection: DC 12V 1000RPM Gear Motor High Torque (Current Design):
For our international competition entry, we have revised the chassis design to be more compact, prompting us to select a smaller yet more powerful motor that aligns with the space-saving design goals. We are now utilizing a DC 12V 1000RPM high-torque electric micro speed reduction geared motor with an eccentric output shaft. This motor, with a 1:3 reduction ratio, allows us to achieve both high RPM and substantial torque in a more compact form factor. Key specifications include a rated torque of 0.15Kgf.cm and a D-shaped output shaft, enhancing stability in coupling with other components.

Torque/kg = 115mNm/kg
Power to weight ratio = 1208W/kg
Max speed of 1000 rpm = 4188.8mm/s

This new motor offers several advantages over the previous 12V 380RPM motor:

1. Compact Design: The 37mm-diameter gearbox and reduced motor size enable a more compact overall vehicle design, which is advantageous for tight manoeuvres and space-constrained paths in our obstacle course.
2. Higher Speed: The new motor's 1000RPM no-load speed is considerably higher than the previous motor, enhancing maximum speed potential without compromising control.
3. Enhanced Torque in a Smaller Package: Despite the smaller size, this motor provides sufficient torque to meet the demands of our compact design while maintaining stability and performance.
4. Eccentric Output Shaft: This design feature contributes to smoother operation and better power transfer, helping to optimize control during sudden directional changes.

In summary, switching to the DC 12V 1000RPM high-torque motor supports our design objectives by maximizing speed, agility, and space efficiency, making it a valuable improvement for our international competition entry. This also allowed us to use a smaller battery, reducing the overall size of our robot. Despite being smaller, it is visible that the Power to Weight ratio of the robot is approximately the same with a higher max speed.

Motor driver: TB6612FNG

To control our DC motor actuator, we require a motor driver. We chose the TB6612FNG since it met our power and actuation needs. It outputs 1.2A at up to 15V. This matches well with our motor which can take from 1A – 5A at 12V. This allows us to provide sufficient power to our motor while not overloading it or overspending on excessive equipment.
In addition to forward and reverse control, the TB6612FNG offers braking and standby modes. The brake mode quickly stops the motor, while the standby mode reduces power consumption when the motor is not in use.

POWER AND SENSE

Power

Consumption

DC motor:	1A, 12V at max load
Battery:	Turnigy Lipo Pack 5000mAh 3S 25C W/XT-140C LiPo Battery
New DC motor:	**0.15A, 12V**
New Battery:	**1300mAh 3S 11.1V 70C LiPo Battery, 12.6V maximum**
Raspberry Pi 4:	5V, 3A

At maximum load, the battery provides sufficient power with a comfortable excess. The main component consuming power is our controller, the Raspberry Pi at 3A. Power to the Raspberry Pi from the battery is first regulated by the Buck module to prevent overload. As for our main actuator, the DC motor, is controlled by the Motor driver which receives power directly from the battery. This is because the Motor Driver's VM is rated at 15V, higher than the battery's. This allows us to receive maximum power to the main actuator without any loss of power.

New components:

Upon downsizing our robot, we required a smaller battery and smaller motor. Therefore, we decided on our new set of motors and batteries. At a maximum voltage of 12.6V, the battery is more than capable of powering the 12V motor. The most notable difference is that the battery's capacity has dropped from 5000mAh to just 1300mAh. However, since our motor current draw has also dropped to just 0.15A, this decrease in capacity is offset. Overall, the battery would simply last shorter, requiring more frequent changes and charging.

Servo Battery: 1000mAh, 7V
Servo: 7.2V max

We use a separate Servo battery since our main battery provides too much power (Voltage * Current) to the servo even with a regulator.

Sense

Controller: Raspberry Pi 4B+

We chose Raspberry Pi SBCs over Arduino Microcontroller due to their higher processing power, enabling us to run complex operations like computer vision. It also offers greater flexibility for integrating software and hardware. We use VNC Viewer for remote control, which simplifies development and testing. Since we plan to use the OpenCV library for image processing, the Raspberry Pi's support for Python makes it the ideal choice.
The Raspberry Pi 4 handles real-time tasks like computer vision fairly well which may be harder on microcontrollers like the Arduino Uno.
Thanks to its versatility with software and programming languages, the Raspberry Pi makes integrating essential robotics components like sensors and cameras much easier.

- **Real-Time Image Processing with OpenCV:** The Raspberry Pi Camera Module, paired with OpenCV, enables real-time object detection and visual navigation, allowing our robot to recognize and react to its environment. OpenCV provides a range of functionality that makes computer vision far easier to execute, especially with python.
- **Wireless Connectivity:** Built-in Wi-Fi and Bluetooth allow for seamless communication, remote control, and troubleshooting, especially during competitions.
- **I/O:** The Raspberry Pi 4 offers plenty of GPIO pins, USB ports, and HDMI outputs, making it easy to connect various sensors and peripherals.
- **I2C:** The Raspberry Pi 4's I2C bus allows communication for connection with multiple components while reducing wiring complexity.

While the Raspberry Pi may not be as efficient at handling low-level tasks like communication with components than an Arduino, it more than makes up for it with the ease of programming using python and OpenCV.

Ultrasonic Sensor: US-015 & **Time of Flight Sensor**

For our vehicle, we continue to rely on ultrasonic sensors to aid in obstacle detection and safe navigation. However, for our international competition entry, we have enhanced our design by introducing Time-of-Flight (TOF) sensors on the sides of the vehicle, replacing the ultrasonic sensors previously positioned there. We now use the US-015 ultrasonic sensor exclusively at the front, paired with a front-mounted camera, while the TOF sensors cover side detection.

The US-015 ultrasonic sensor at the front provides reliable obstacle detection and distance measurement, essential for navigating around pillars and maintaining safe distances from boundary walls. It operates within a range of 2 to 400 cm with 1 mm accuracy, making it ideal for precision in close-range navigation.

Datasheet: US-015 Datasheet

TOF Sensors on the Sides of the Vehicle:

Our new VL53L1X TOF sensors have a range of up to 4 meters and offer enhanced accuracy and extended range capabilities over ultrasonic sensors. The TOF technology enables rapid and accurate distance measurements with minimal interference, making it well-suited for detecting obstacles along the vehicle's sides. This improvement ensures that the vehicle can maintain better lateral positioning, improving manoeuvrability in confined spaces by precisely measuring the distance to side walls and other obstacles.

Switching to TOF sensors also contributes to a more responsive and robust system, as TOF sensors are less susceptible to environmental noise and provide faster refresh rates. This improvement will enable our vehicle to respond to nearby obstacles more accurately and quickly, allowing for sharper adjustments and a more stable alignment within the field.

The VL513X TOF Sensor uses a i2c bus for interfacing which makes circuitry for the PCB much easier than using ultrasonic sensors.

The sensor consumes a maximum of 20μA during measurement which is far lower than the US-015 Ultrasonic Sensor’s 2.2ma operating current, making it more power efficient for our battery.

By combining ultrasonic and TOF sensors, we achieve a robust and efficient sensing system, enhancing the robot’s overall performance and positioning accuracy as it navigates the obstacle in the field.

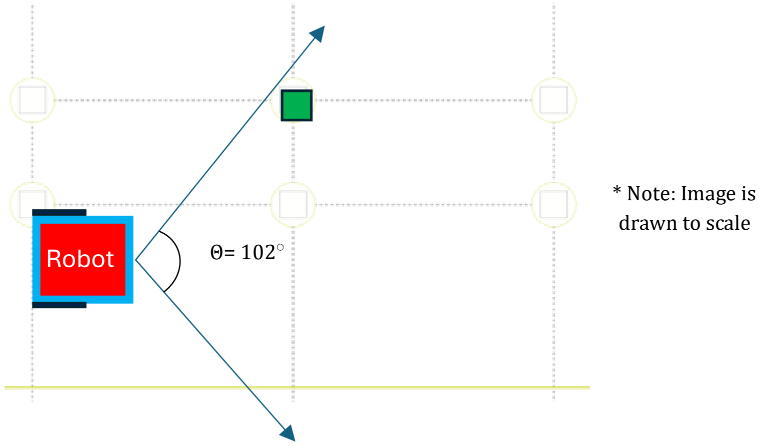
VL51L1X Datasheet: Datasheet

Camera: Raspberry Pi Camera 3, Wide Angle Variant

A Raspberry Pi Camera is mounted at the front top of our robot, serving as the 'eyes' of our robot. Its primary function is to detect obstacles and identify their colour, which allows the robot to determine the corresponding action to take.

We chose the Raspberry Pi Camera Module 3 wide angle as it is the latest model, offering advanced features such as phase detection, which facilitates image recognition. This capability allows the system to differentiate between green and red pillars, guiding the vehicle's directional turns accordingly. Additionally, the camera supports various default modes, including 480p at 120 frames per second. With such a high frame rate, our robot can receive near-instantaneous information, ensuring responsive and accurate movement. Furthermore, we chose to upgrade to the wide-angle version for internationals as it has a wider Field of View (FOV).

The variant has a horizontal FOV of 102 degrees. This makes it easy to detect traffic signs when they are close by. Take for instance the starting position of the robot in the obstacle challenge.



In this scenario, when the robot is placed on the right most side and the nearest obstacle is on the left most possible position, it is still able to detect it with the camera without moving back.

The camera also has a resolution of **4608 x 2592** which is much larger than the Pi Camera 2 at **3280 × 2464** and Pi Camera 1 at **2592 × 1944**. This allows for higher precision distance calculation if required.

Thus, with the raspberry pi wide angle camera 3, we are able to perform fast image processing along with capturing close by objects, hence making it a good choice for our robot camera.

More on info on the Camera 3 can be found here: Raspberry Pi Camera 3 Datasheet

Compass: HMC5883L Module Triple Axis Magnetometer

The HMC5883L magnetometer compass module is employed to provide precise heading information for our robot. This module interfaces with the Raspberry Pi via the I2C bus, making integration straightforward and reliable.

With an average current draw of just **0.1mA** in measurement mode, the HMC5883L has a minimal impact on the **5000mAh** LiPo battery, preserving power for other components. Additionally, the module offers a heading accuracy of **±1°**, which is sufficient for maintaining alignment over extended distances. For example, a 1° heading error over a 1-meter distance results in a lateral drift of only **1.75** cm, an error margin that is acceptable within our design constraints.

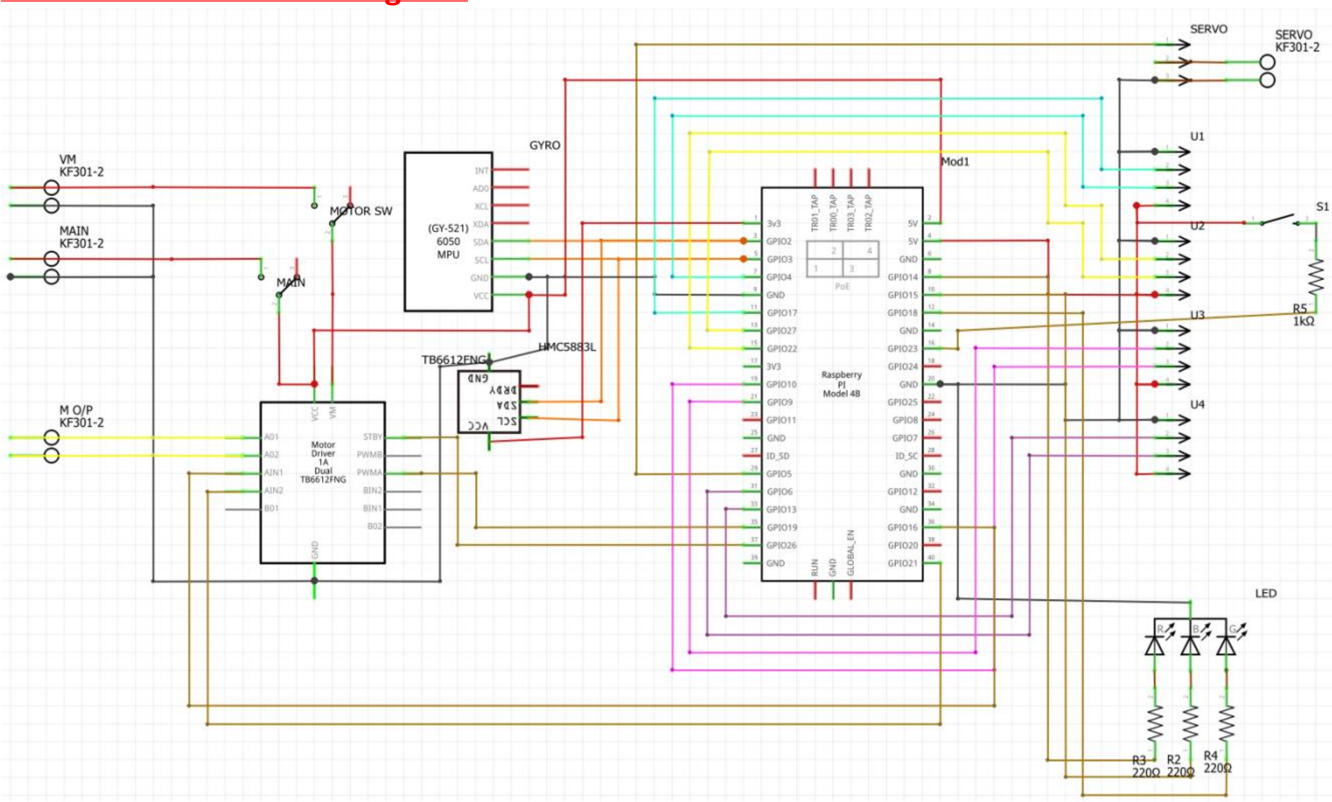
The HMC5883L is a cost-effective solution that balances accuracy with low power consumption, making it an ideal choice for ensuring our robot's directional stability throughout its tasks.

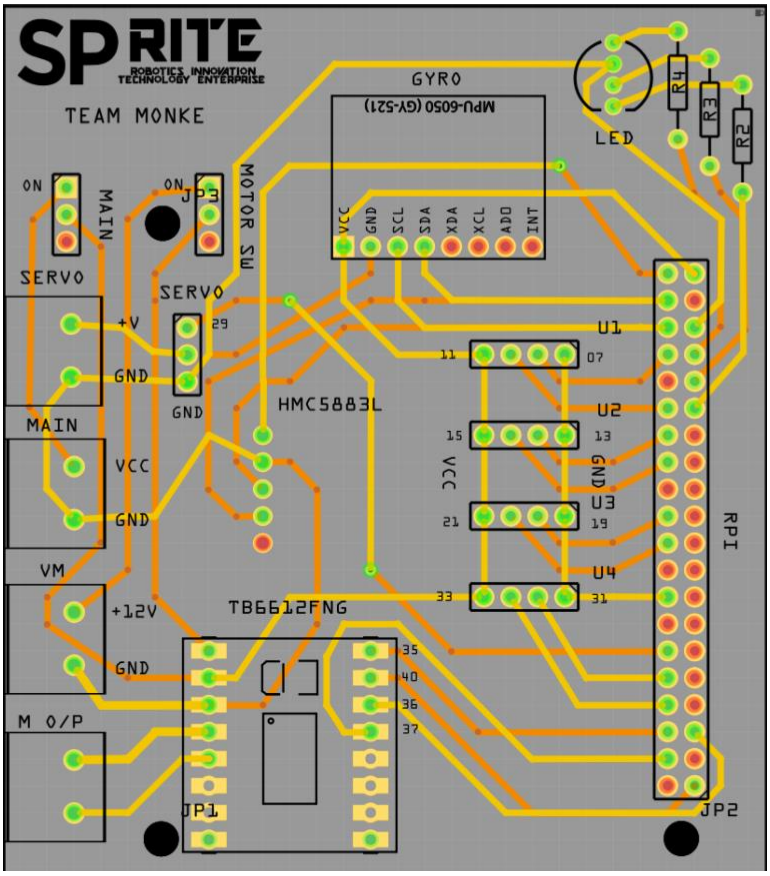
The specifications of the module can be found here: [HMC5883L datasheet]

Integration: Printed Circuit Board

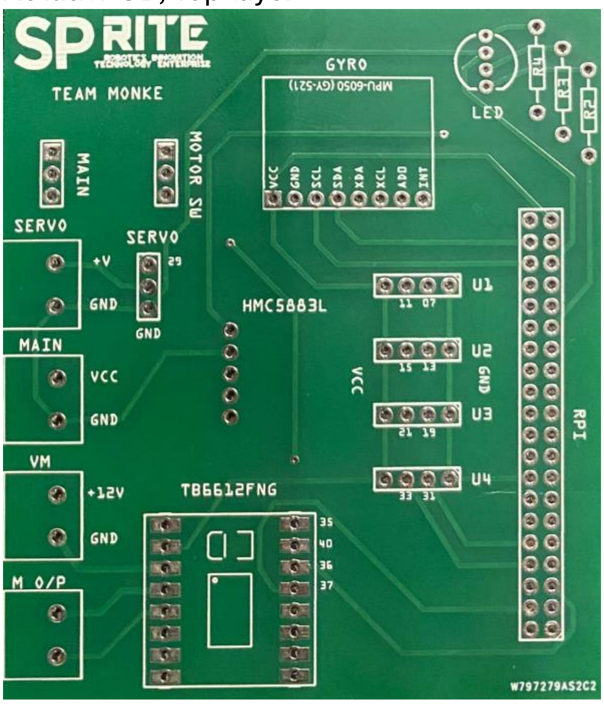
To connect all our sensors and components together, we chose to use a custom designed PCB. We could build our circuits on breadboards or perforated boards, but breadboards are not meant as a long-term solution since their connections are not very secure, especially on a moving robot. On the other hand, perforated boards offer more secure connections, but soldering of connections would be time consuming, difficult and more likely to be wrong. Thus, we chose to use a PCB since it allowed us to have secure connections, with less soldering required, and lasts long. This also provided a valuable learning experience in how to PCB design.

WRO FE SG Schematic Diagram:

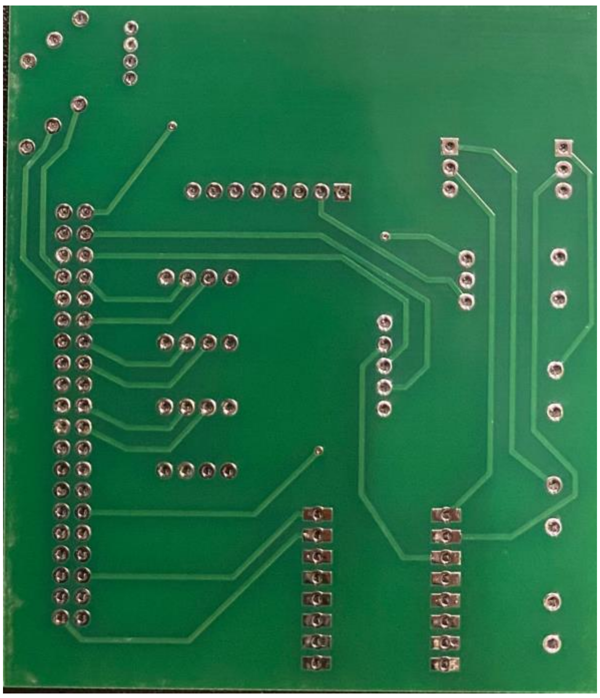




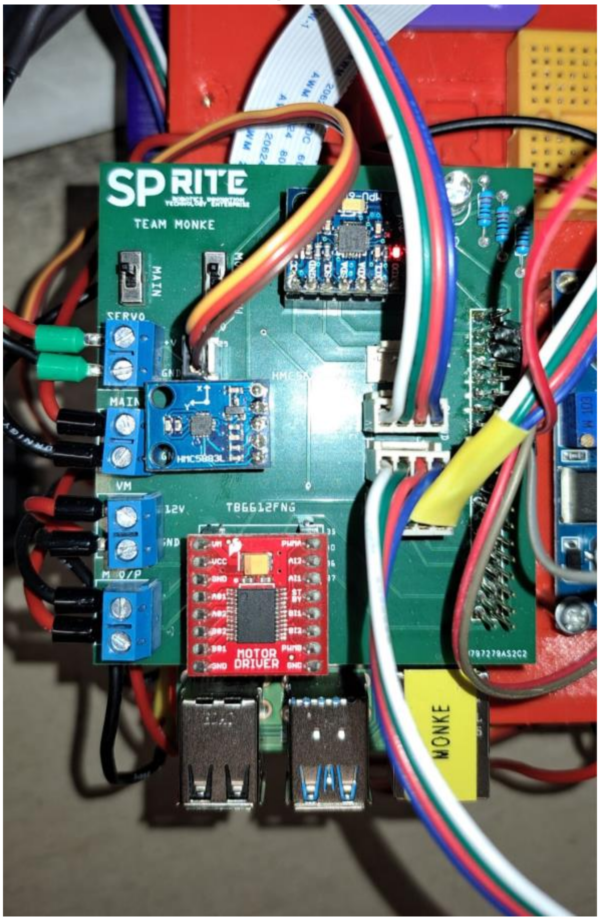
Actual PCB, Top layer



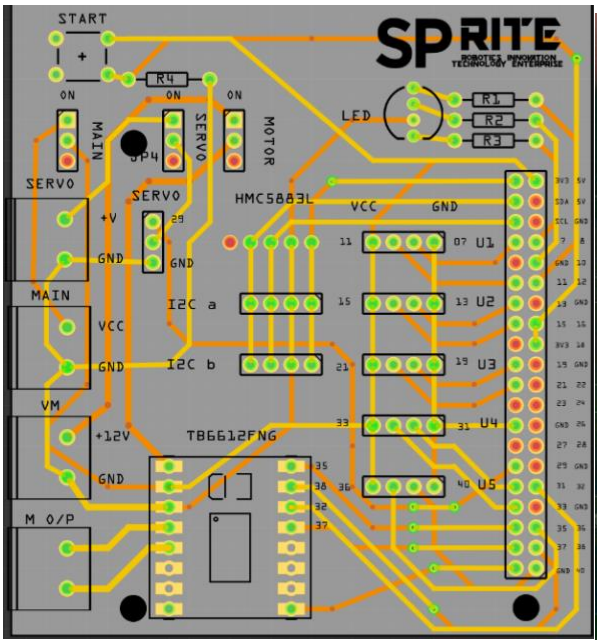
Bottom layer



PCB with all components mounted on our robot



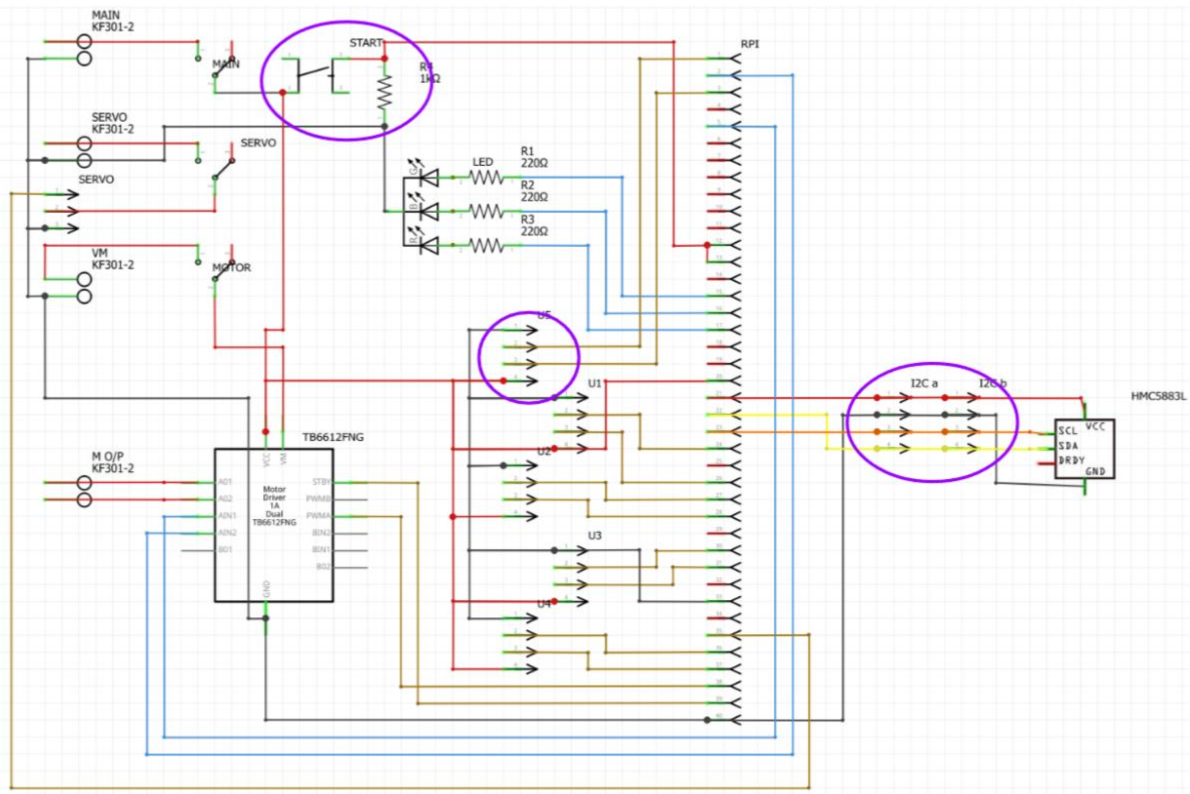
WRO FE International Türkiye:



As seen in our updated PCB design and schematic diagram, we made a few improvements while retaining much of the previous layout. In the earlier version, we missed adding a push-button component, which we quickly addressed by connecting it to a small, perforated board. However, to make our design more compact and streamlined for this competition, we decided to create a new PCB based on the original layout.

The updated PCB includes a few key changes:

- **Push Button:** The push button is now integrated directly into the PCB, eliminating the need for additional components.
- **Gyro:** We removed the gyro, as it wasn't used in our final design due to inaccuracies and only occupied extra space.
- **Additional Connectors**: We added extra 4-pin I2C and GPIO, GPIO 3-pin connectors to accommodate any future components without requiring a redesign.
- **Motor Power Switch**: An additional switch now allows us to control power to the main motor separately.



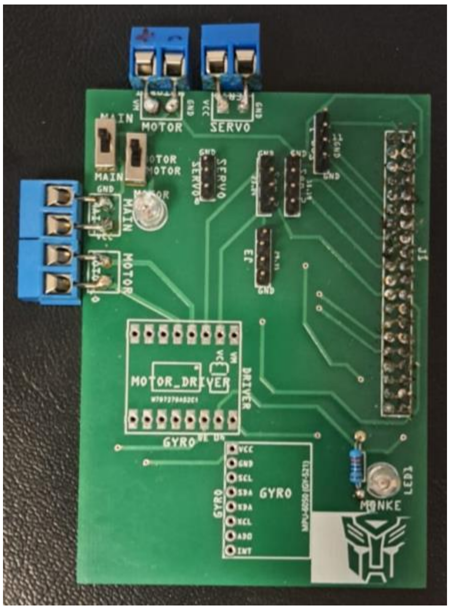
DEVELOPMENT MILESTONES

Fabrication

WRO FE SG:

PCB

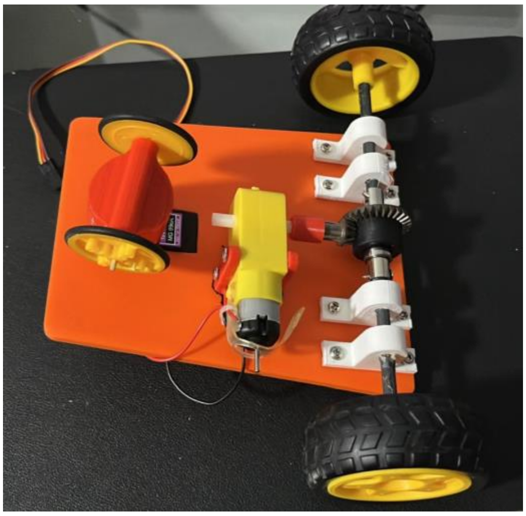
Old PCB



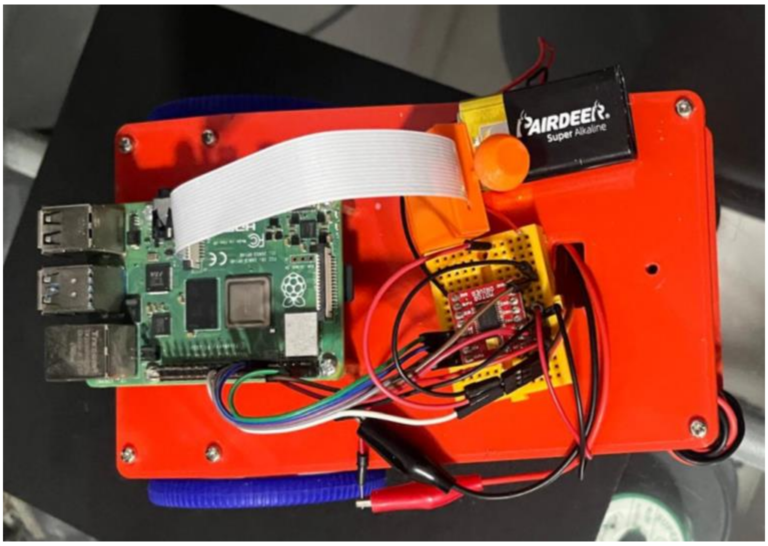
Visible from the image above, there were multiple improvements to be made to the PCB. For example, the screw terminals are the wrong size.

AGV

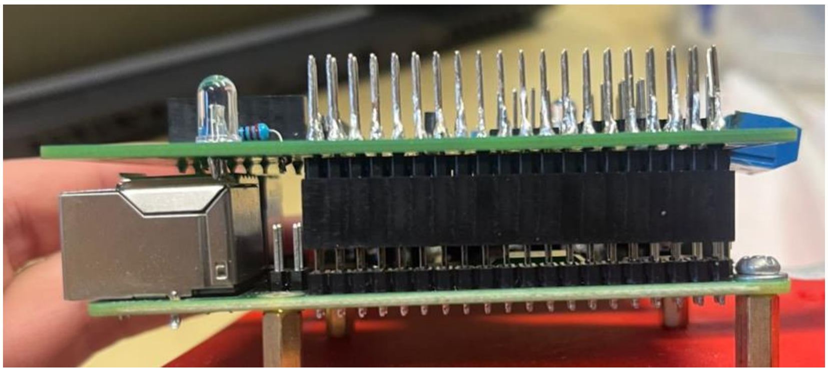
This is our very first AGV that we tried. This was meant as a proof of concept, testing out the differential gear drive and tricycle steering.



Assembly

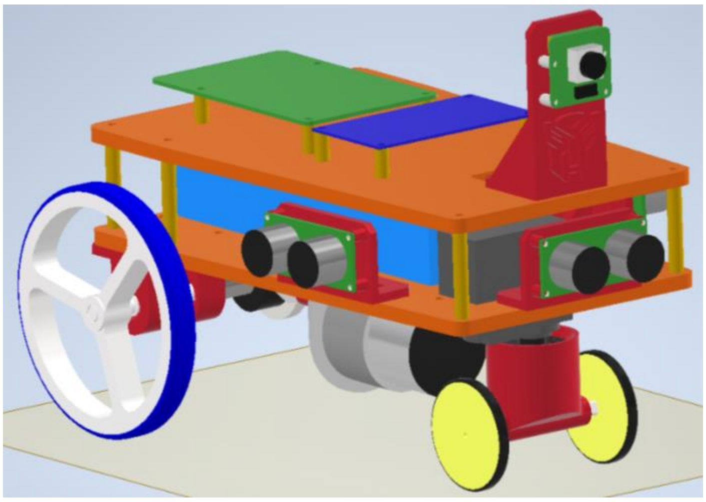


As our 2nd prototype, we used a breadboard to test out the motor driver. From insecure connections to messy wires, it was clear that we needed a PCB.



Upon assembling our PCB, we noticed that the Raspberry Pi’s USB and ethernet ports were blocking it.

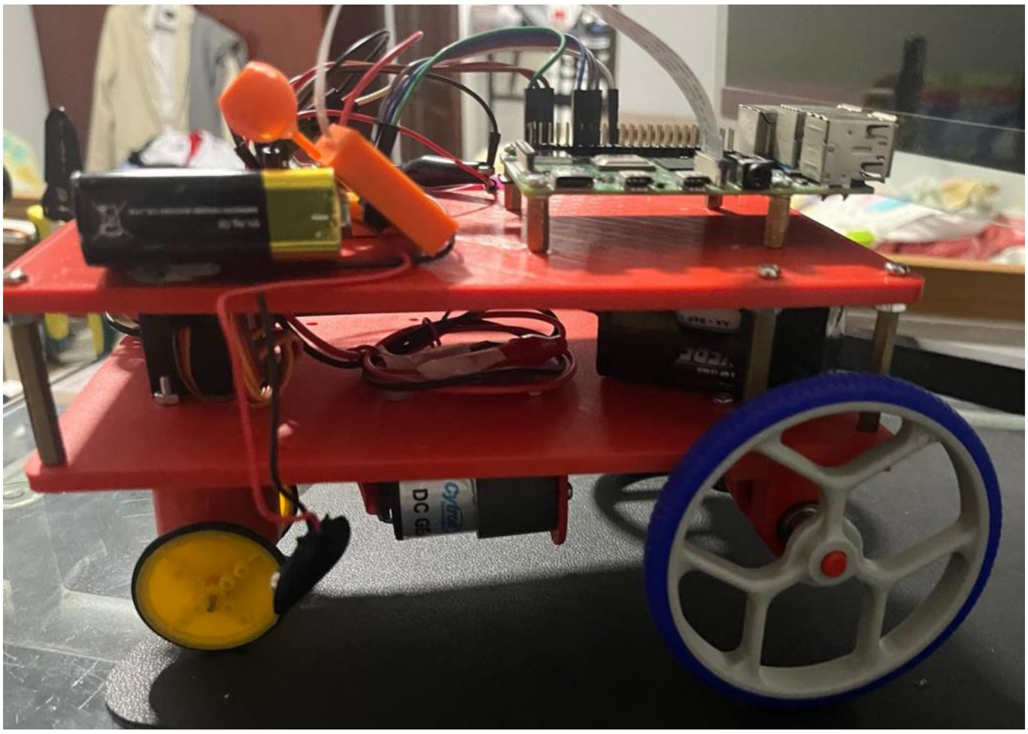
AGV



This is the 3D CAD file of our old AGV.

The robot is facing a different direction and is much longer than our current version.

As our final prototype, we started integrating the camera and OpenCV control.



Troubleshooting

Previously, we connected the servo motor directly to the battery. This caused the servo to overload and fry. Therefore, we are now using a smaller dedicated battery for the servo.

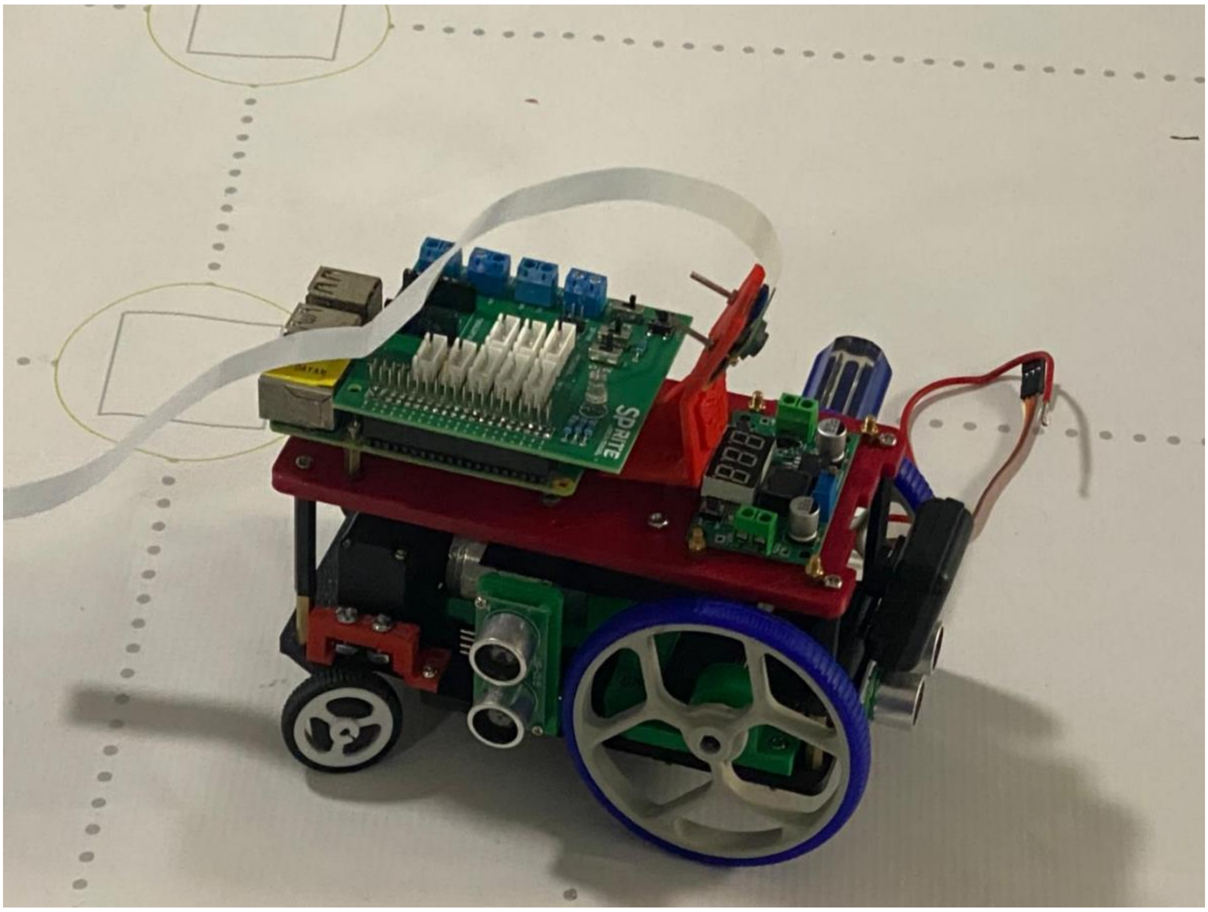
With our current steering configuration, our servo overload due to direct steering forces. When steering at sharp angles, the servo must exert a large amount of power and experiences forces of the robot pulling it forward.

WRO FE International Türkiye:

The main objective of redesigning the robot for the Internationals was to downsize our robot and use the Ackermann steering mechanism. Furthermore, we also reevaluated some of the sensors we used and opted for higher performance ones that costed more.

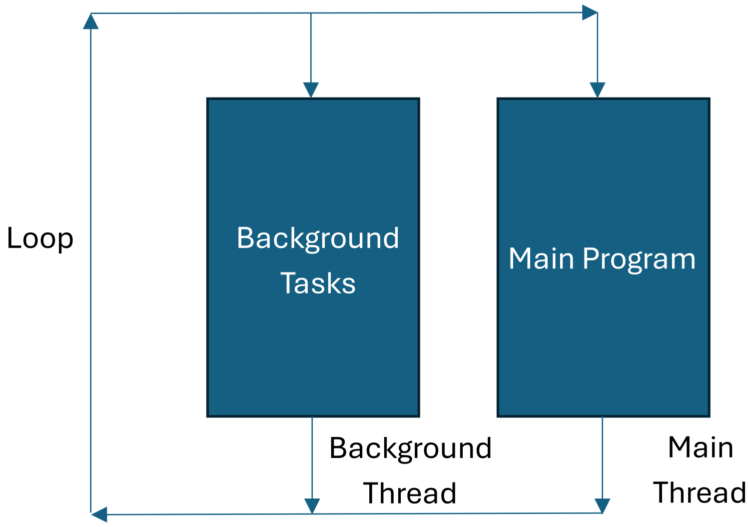
The most noticeable differences are:

1. Drastically reduce size of robot, decreasing wheelbase and therefore turning radius
2. Use Ackermann steering for better reliability, robustness and stability
3. Explore the use of better sensors such as the TOF sensor, Pi cam 3 wide angle
4. Improve PCB by adding more features and connections



PROGRAM AND ALGORITHMS

Program Structure:



To allow for easier programming, the processes of our program are split into the background tasks and the main program. This eases programming as we do not need to constantly call the sensors. The background tasks are run continuously in the back with no obstruction. Tasks include reading sensors and detecting obstacles. This allows for fast reaction of our robot to the changes in its environments as the sensors are constantly updating.

The main program is run in the main thread which is where the logic and algorithms for reacting to different scenarios is carried out. This allows for use of loops which makes programming complex logic easier.

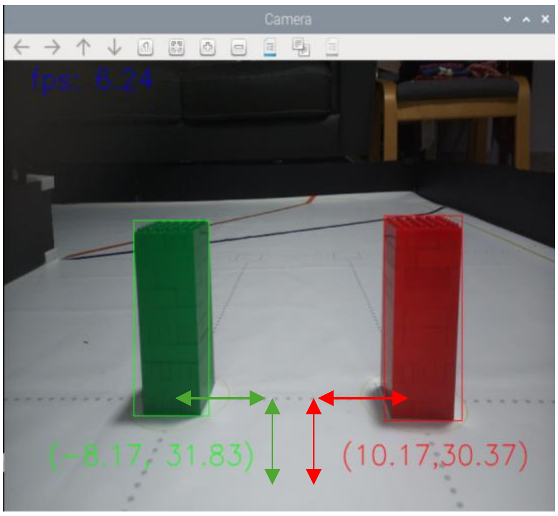
Obstacle Challenge:

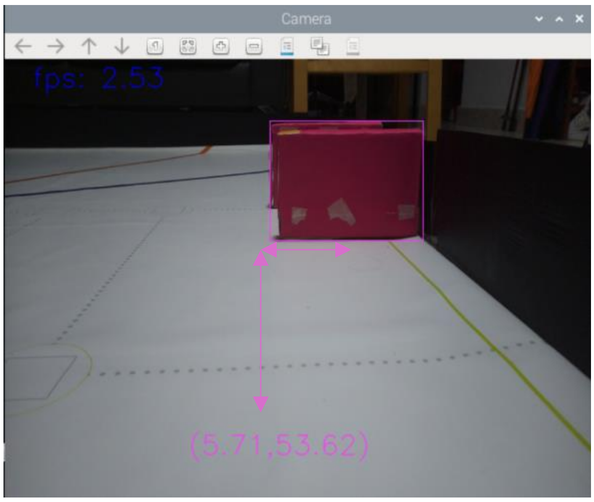
This section will discuss the methods and strategy implemented for obstacle management.

The strategy for obstacle management can be broken down into **4** steps:

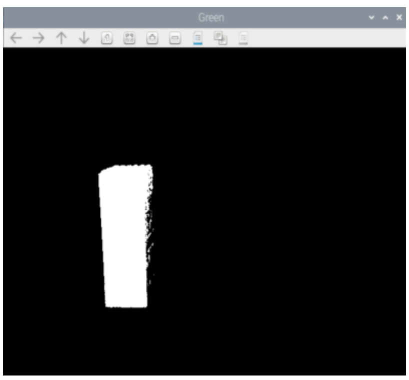
1. Look for obstacles:
   - The obstacles to avoid:
     - Green traffic sign, red traffic sign, Pink Parking Lot Borders
   - To detect these obstacles, we use their **HSV** (Hue, Saturation, Value) values and apply a binary mask onto the camera image.
   - We then use OpenCV's "findContours()" function to extract the largest bounding box from the mask

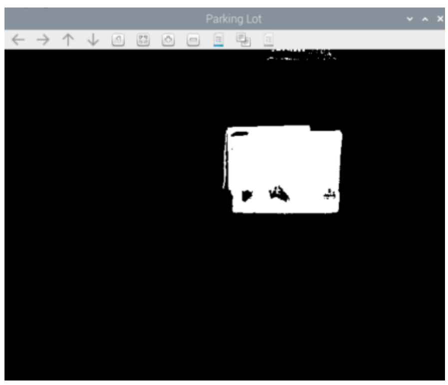
Camera Image





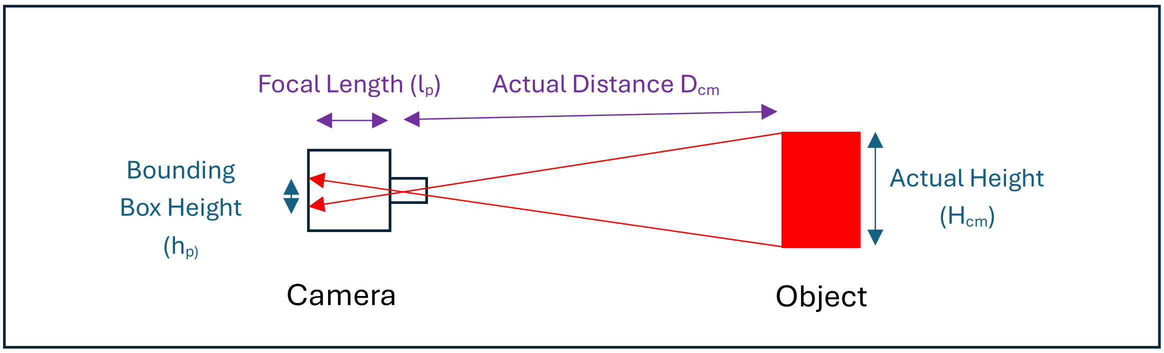
Binary Masks





2. Get object's location relative to robot:

- With the object’s bounding box, we can estimate its relative location by using the dimensions of the bounding box and the know dimensions of the object.
- This method involves the use of **similar triangles.**
- **Requirements:** <u>known object size</u>, <u>focal length of camera</u>

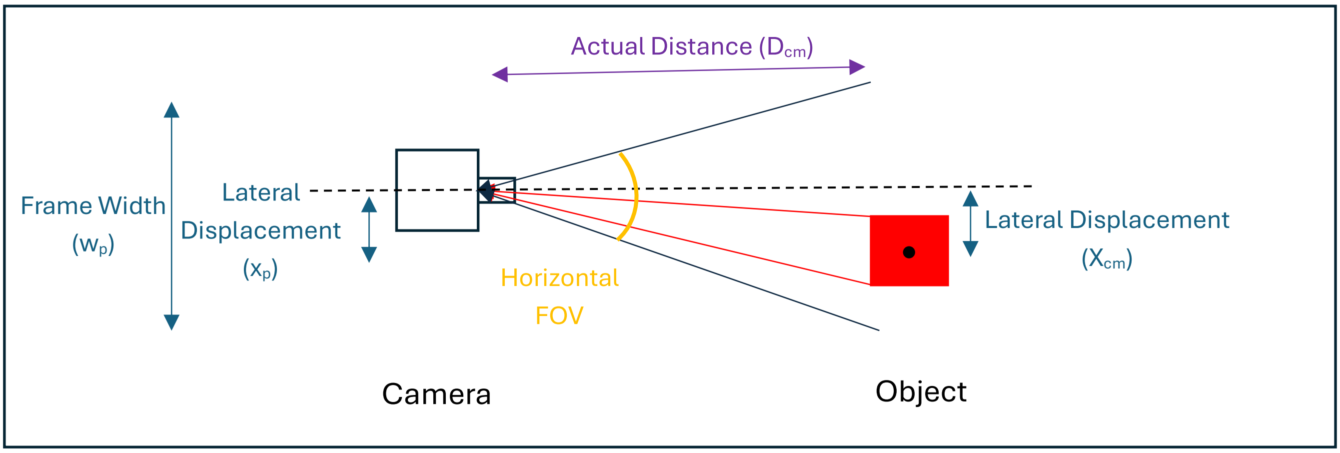


Side View

Object Distance can be calculated by :

$$D_{cm} = \frac{H_{cm}}{h_p} \times l_p$$

- To get the lateral displacement of the object to the robot, we can use simple trigonometry.

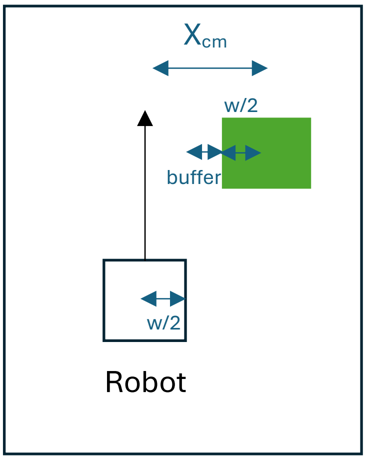


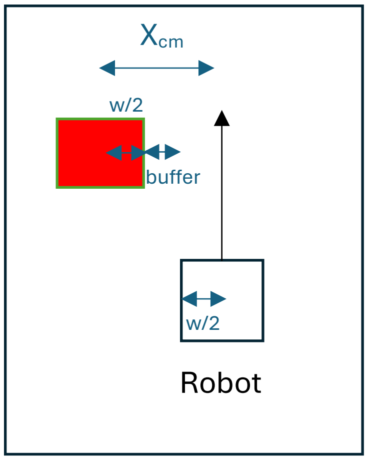
Top View

Lateral Distance can be calculated by :

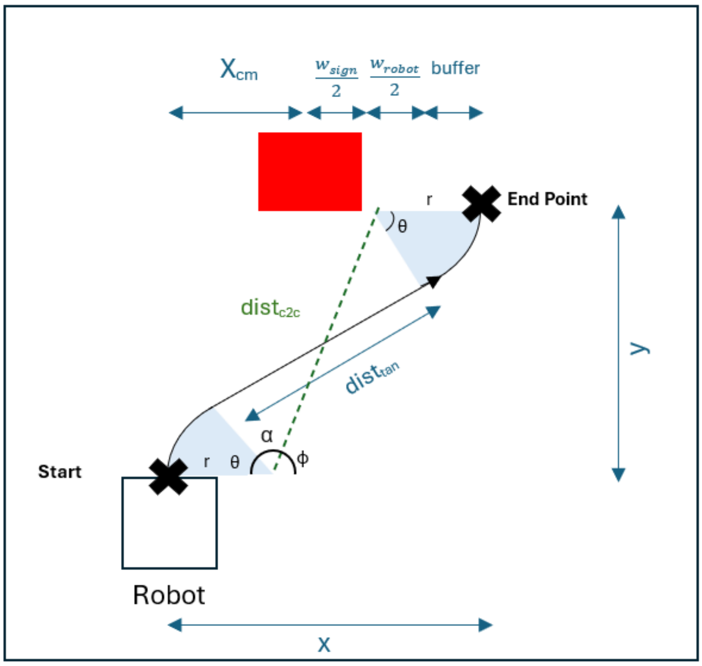
$$X_{cm} = \frac{x_p}{w_p/2} \times D_{cm} \tan\left(\frac{FOV}{2}\right)$$

3. Check if we are on the right side of the traffic sign:
   - For green, $\boldsymbol{X_{cm}} < -(\boldsymbol{Robot\_Width} + \boldsymbol{traffic\_sign\_width}/2 + \boldsymbol{buffer})$
   - For red, $\boldsymbol{X_{cm}} > (\boldsymbol{Robot\_Width} + \boldsymbol{traffic\_sign\_width}/2 + \boldsymbol{buffer})$
   - If we are not on the correct side of the traffic sign, we have to turn accordingly.





4. Turn to the correct side of the traffic sign:
   - Since we are not on the correct side of the traffic sign, we must turn to the correct side and avoid it.
   - We calculate the path required to get to the correct side, based on the object's estimated position.
   - The path taken is constructed from 2 arcs and a cross tangent that they both share. The radius of the arc is taken to be the same for simplicity.



Let,

Radius ***r*** = ***WHEELBASE***

X = $\boldsymbol{X_{cm}}$ + ***sign_width*** /2 + ***robot_width*** /2 + ***buffer***

Y = $\boldsymbol{D_{cm}}$

The angle $\boldsymbol{\theta}$, and distance $\boldsymbol{dist_{tan}}$, can be calculated using:

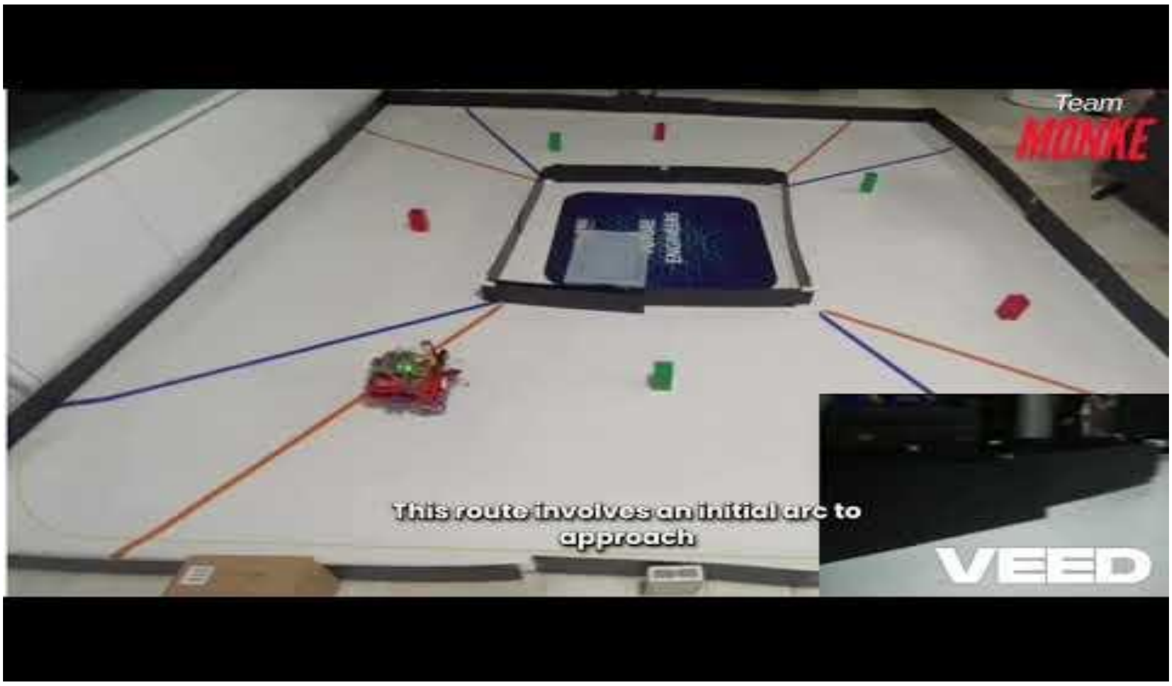
$$dist_{c2c} = \sqrt{(x-2r)^2 + (y)^2}$$

$$\theta = 180 - \alpha - \emptyset$$
$$= 180 - cos^{-1}\left(\frac{r}{\frac{dist_{c2c}}{2}}\right) - tan^{-1}\left(\frac{y}{x-2*r}\right)$$

$$dist_{\tan} = \frac{y - 2r * \sin(theta)}{\cos(theta)}$$

- Using the calculate values of ***theta*** and $\boldsymbol{dist_{tan}}$, we can drive the robot accordingly on the path we calculated.

https://www.youtube.com/watch?v=x_PWh__5Zlo



CONCLUSION/IMPROVEMENTS

Sensors

1) Instead of Ultrasonic sensors and TOF sensor, we could use a 360 Lidar sensor instead. Lidar is much more consistent and accurate compared to ultrasonic sensors and can provide more data on the robot's environment. Our team encountered some difficulties in the consistency of the Ultrasonic sensor.
2) We could also use another IMU other than the HMC5883L. This is because the HMC5883L is easily affected by high current wires inducing a magnetic field due to the Faraday's law of induction. Furthermore, it requires constant calibration to adjust to a new venue.

PCB

1) More push buttons can be added on the PCB so that we are able to start/stop the robot or give inputs during testing.
2) An option for a LCD/OLED display could be added to allow for easier debugging and troubleshooting in the testing phase.

Controller

1) A microcontroller could be used in place of the raspberry pi 4 which is an SBC (Single Board Computer). Although microcontrollers may be inferior in terms of the processing power they offer, they are often much faster and more suited for dealing with low level components such as ultrasonic sensors and motors. This would make the overall interfacing of the robot brain with its component much easier as the raspberry pi does add some complexity in doing so.

Algorithms

1) Instead of programming the step-by-step actions for the robot to take when it encounters a condition, we could employ machine learning to control the robot instead. We had plans to test out the suitability of supervised learning for the open challenge, however we came to realise that the lack of environmental information due to using only 3 ultrasonic sensors and 1 compass hinders the model's capability to learn the correct action to take. We could use the camera as input for the model but we have yet to test it.

REFERENCES

WRO FE Rule Book:

- https://wro-association.org/wp-content/uploads/WRO-2024-Future-Engineers-Self-Driving-Cars-General-Rules.pdf

VNC Viewer References:

- https://www.google.com/url?sa=t&source=web&rct=j&opi=89978449&url=https://www.realvnc.com/en/connect/download/viewer/&ved=2ahUKEwiW_4zD2JeIAxUXS2wGHda5O6kQFnoECAkQAQ&usg=AOvVaw0UTdY8kQhw7EWigV5-iC08
- https://youtu.be/ikUH_wOesIY?si=3h6Y2FHPOmsgjrJ-
- https://youtu.be/ocWZ9QFgvXk?si=nr7zo171_FX9tkB3

GitHub References:

- https://github.com/World-Robot-Olympiad-Association/wro2022-fe-template

Learning Materials

HMC5883L Compass Module with Raspberry Pi:

- https://how2electronics.com/interfacing-hmc5883l-magnetometer-with-raspberry-pi/

Ultrasonic Sensor with Raspberry Pi:

- https://projects.raspberrypi.org/en/projects/physical-computing/12

Pi Camera with Raspberry Pi:

- https://projects.raspberrypi.org/en/projects/getting-started-with-picamera/0
- https://www.youtube.com/watch?v=0hrF8Wq8SSQ

Threading in Python:

- https://realpython.com/intro-to-python-threading/

OpenCV Documentation:

- https://docs.opencv.org/4.x/index.html

Software Control with Git:

- https://indico.cern.ch/event/724719/contributions/2981043/attachments/1638054/2754736/Git_tutorial.pdf

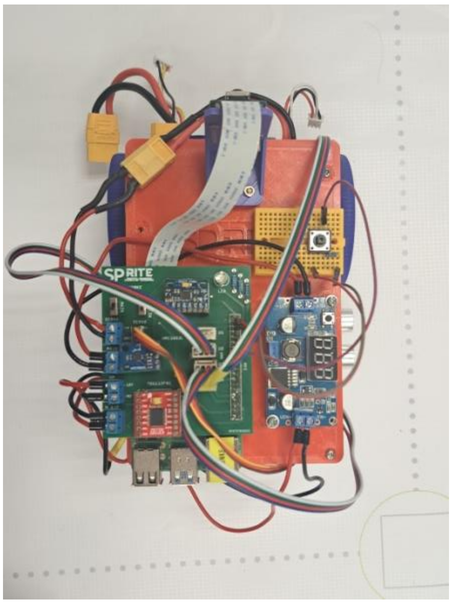
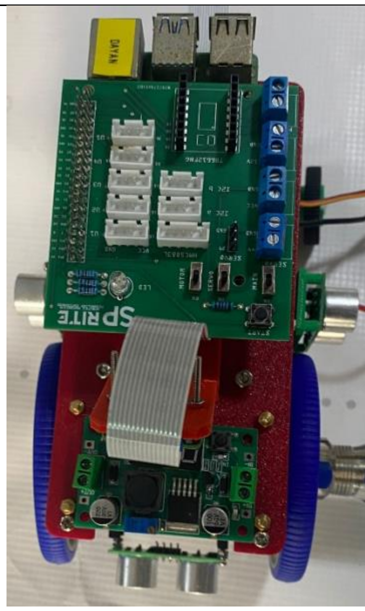
Python Code Documentation:

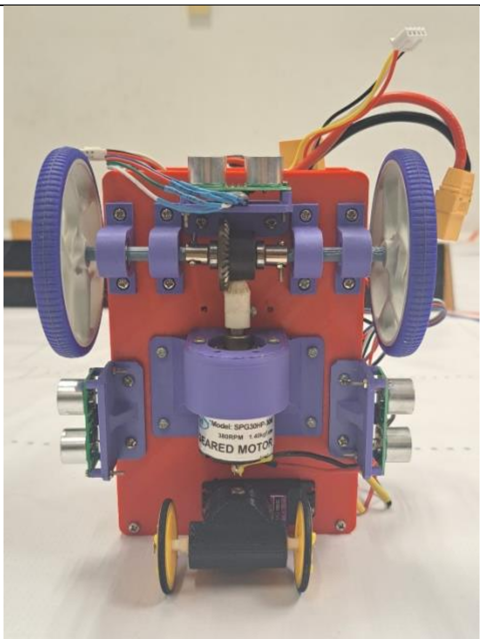
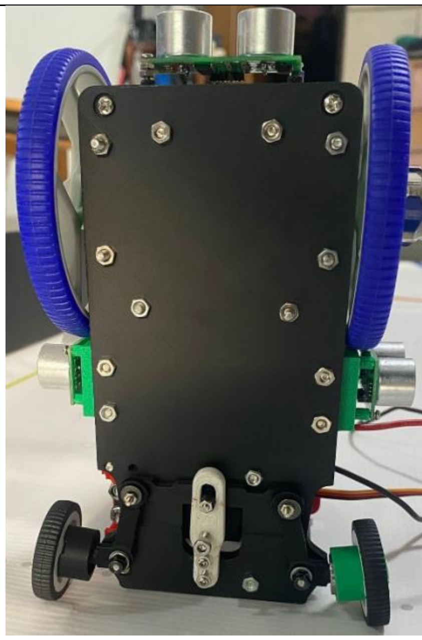
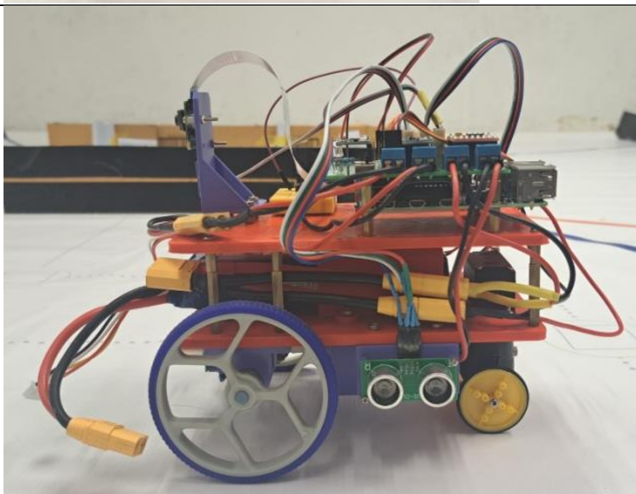
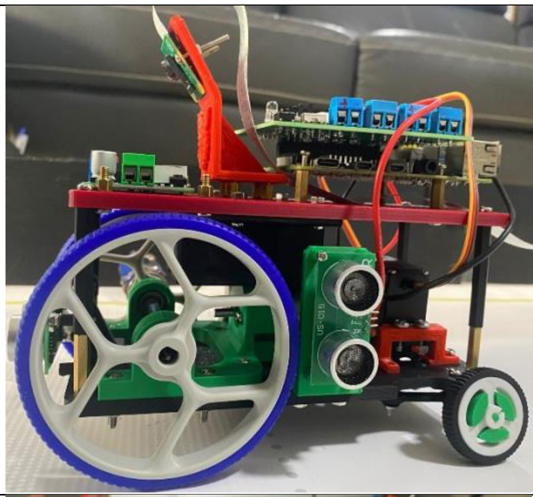
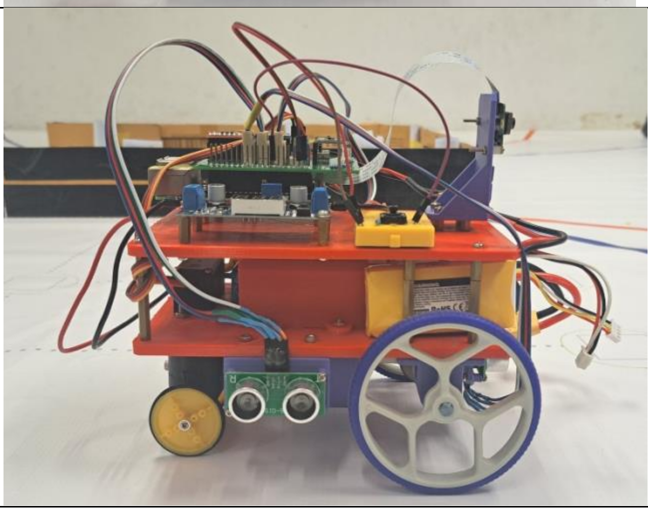
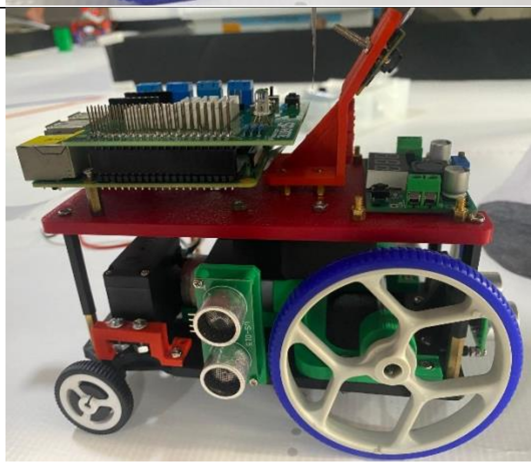
- https://realpython.com/documenting-python-code/

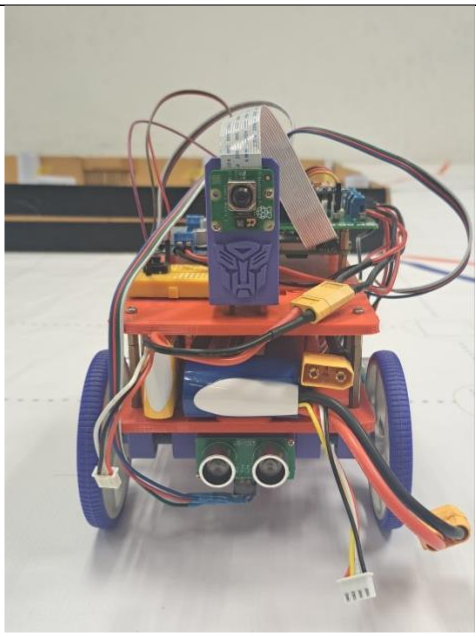
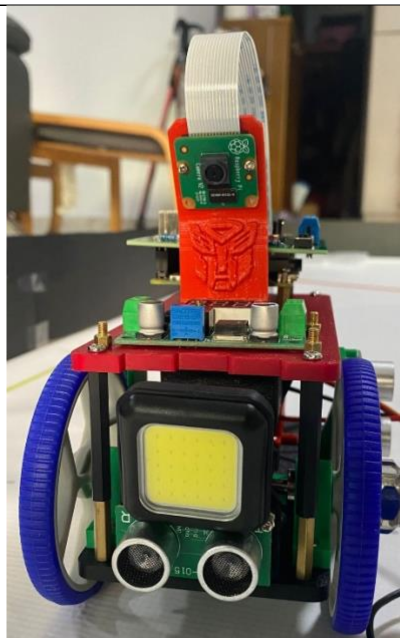
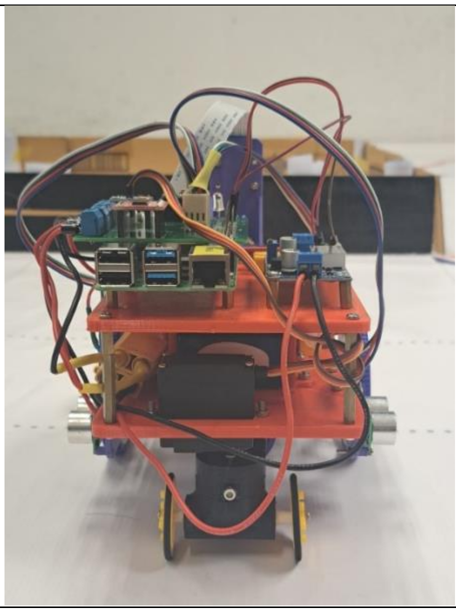
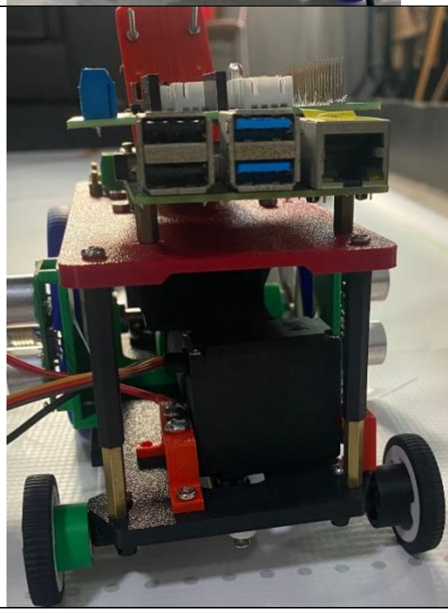
APPENDIX A

<u>Below are the images of our WRO FE SG & WRO FE International Türkiye AGV:</u>

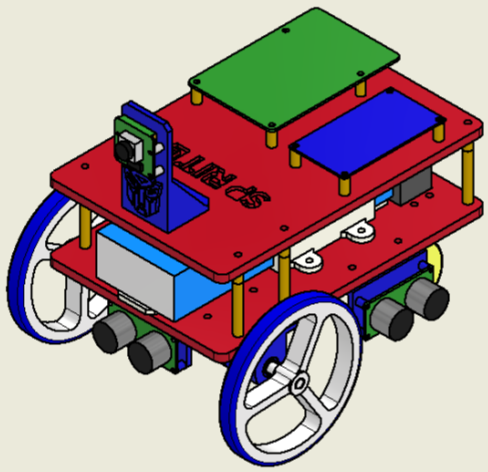
As previously stated in our chassis section of this engineering journal, we had redesigned our vehicle and from the photos below for reference and comparison you can clearly see that our new vehicle is much smaller than its previous predecessor almost by half the size being decreased and much compact and cleaner.

View	**WRO FE SG**	**WRO FE International Türkiye**
Top		

Bottom		
Left		
Right		

Front		
Back		

To View Engineering drawing in DWG format access respective files on GitHub



ISO
SCALE 1 : 1

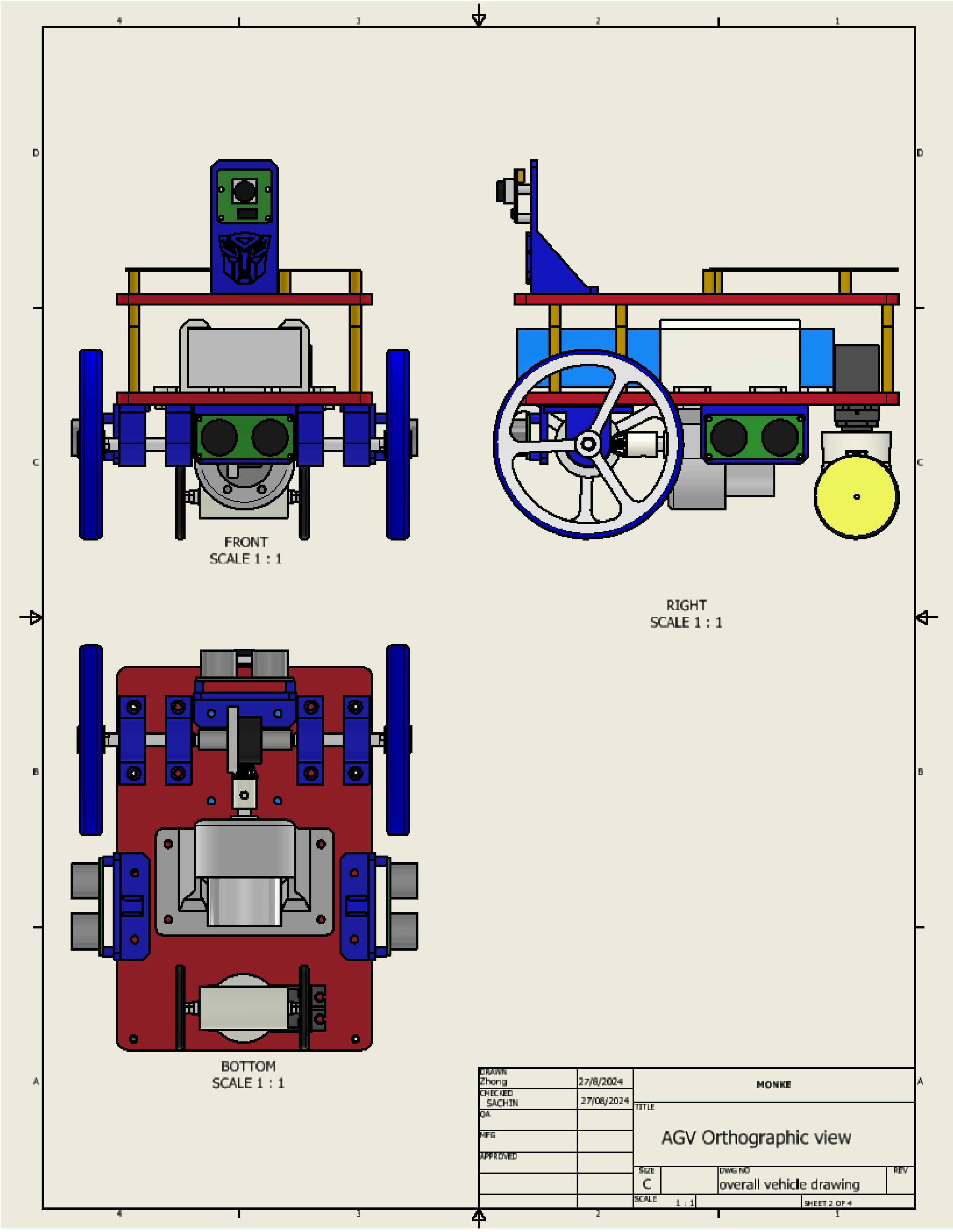
MATERIAL LIST			
TOTAL QTY	PART NUMBER	MATERIAL	DESCRIPTION
4	bearing holder	PLA	Axle holder for Driving wheel
1	servo motor HS-7955TG		Steering actuator
1	steering	PLA	Steering mount to servo horn
1	dfifferential	Steel	Differential Gear
2	front wheel	Plastic	Steering wheel
2	rear wheel 86mm	Plastic	Driving wheel
2	rear shaft v2	PLA	Driving axle
1	12v DC motor	-	Main actuator
1	Top Plate	PLA	Top plate for mounting components
1	Raspberry Pi	-	Controller
1	Base Plate	PLA	Bottom plate for mounting servo, battery etc
1	3s lipo battery	-	Power Source
3	ultrasonic mount	PLA	Mount for Ultrasonic sensor
3	US-015	-	Ultrasonic sensor
6	stand M3-L10	Brass	M3, 10mm stand
14	stand M3-L20	Brass	M3, 20mm stand
1	LM2596 Buck Converter	-	Voltage regulator for Raspberry Pi
1	camera mount	PLA	Raspberry Pi camera mount
1	raspberry pi camera v2	-	Camera for obstacle detection
4	stub	Brass	Connect Camera to camera mount
1	front shaft v3	PLA	Steering axle
1	steering bottom half	PLA	Connect steering mount to steering axle
1	driver gear adapter	PLA	Custom printed Differential Gear
1	3S lipo holder v2	PLA	Battery holder
1	12v DC motor mount v2	PLA	Motor mount
6	Ball Bearing 6mm x 13mm x 15mm	Steel	Ball bearings for steering axle and driving axle

DRAWN Zhong 27/8/2024
CHECKED
QA
MFG
APPROVED

MÖNKE

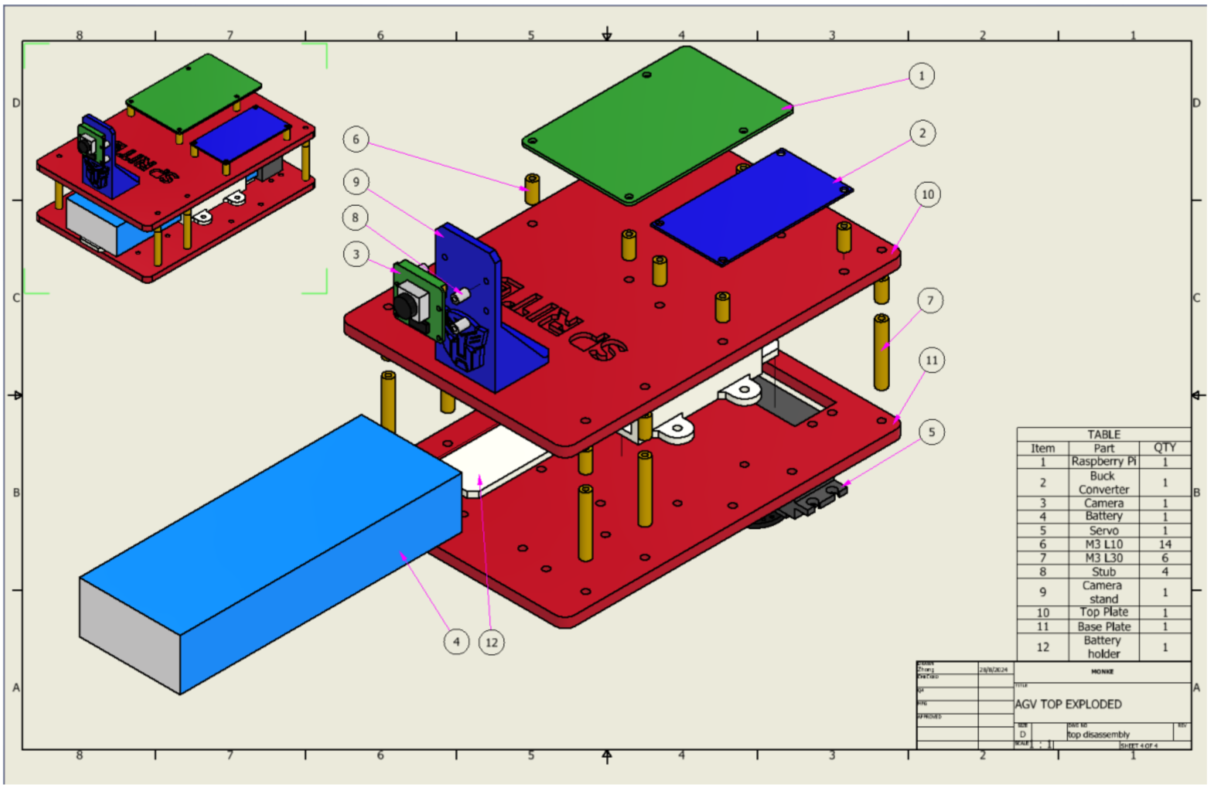
TITLE
AGV ISOMETRIC VIEW

SIZE C
DWG NO vehicle isometric view
REV
SCALE 1 : 1
SHEET 1 OF 4



FRONT
SCALE 1 : 1
RIGHT
SCALE 1 : 1
BOTTOM
SCALE 1 : 1
DRAWN
Zhong
27/8/2024
CHECKED
SACHIN
27/08/2024
QA
MFG
APPROVED
MONKE
TITLE
AGV Orthographic view
SIZE
C
DWG NO
overall vehicle drawing
REV
SCALE
1 : 1
SHEET 2 OF 4

Top part expanded



Bottom part expanded