Team ID: **FE24_324_02**

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

Description	Brand	Remarks	Quantity	Details
Servo Motor		Steering Actuator	1	MG996R
12v 380rpm 1.4kgcm Brushed DC Motor		Main Actuator	1	12v 380rpm 1.4kgcm
Full Metal Differential Gear Set		Differential Gear and Driver Gear for Robot rear shaft	1	
Raspberry Pi 4 Model B	Raspberry	Main Control Board	1	
High precision Ultrasonic Range Finder		Ultrasonic Sensors	3	US-015
Raspberry Pi Camera Module 3	Raspberry	Camera	1	
Motor Driver		Motor Drivers	1	TB6612FNG
Ball Bearings		Ball Bearings	1	6x13x5 mm
Plastic Wheels		Driving Wheels	1	80mm
Small Plastic Wheels		Steering Wheels	1	SPG330/SPG50 (80mm)
HMC5883L Module Triple Axis compass		Triple Axis compass	1	HMC5883L
Lipo Pack	Turnigy	Power Supply	1	5000mAh 3S 25C W/XT-90
LM2596 3A Buck Module with Display		Voltage regulator	1	LM2596 3A
Custom designed PCB		Integrate other components together	1	
NO Push button		Push button to start/stop robot	1	
1k ohm resistor		Resistor for push button	1	
220-ohm resistor		Resistor for LEDs	3	

RGB cathode LED		RGB LED for display	1	
Switch		Turn on/off controller and motor	2	
Screw terminal		Connect battery and motor	4	

OVERALL VEHICLE DESIGN

Size

Our robot is at a shorter than the depth of the parking lot. This allows us to drive directly into parking, without the need to do parallel parking. A problem we encountered while doing such a design was that the physical components such as servo motor, differential gears and DC motor combined were too long. Therefore, we had to custom print a differential gear axle that was shorter.

Component layout

Our robot has 3 layers available for mounting. The bottom layer is used mainly for actuation. The DC motor, differential and steering axle is at the bottom layer. Due to the large size of the batteries, the 2nd layer is dedicated to the batteries with a small part mounting the servo motor. The final top layer has the rest of our electronics such as the Raspberry Pi, the custom-made PCB, camera and voltage regulator. These components were placed here for easy access.

MOBILITY MANAGEMENT

Chassis

For our chassis, our team decided to use a tricycle approach instead of typical Ackerman steering configuration. This is reducing the steering mechanism's complexity while also reducing turning radius. his approach simplifies the steering mechanism while significantly reducing the turning radius, allowing for sharper angles and tighter manoeuvrers. This is particularly advantageous in competition scenarios where obstacles and pillars are closely spaced, requiring the vehicle to make quick and precise turns in either direction.

Our design deviates from traditional tricycle layouts by placing the steering mechanism at the rear and the differential gear system at the front. This configuration is intended to optimize the length of the robot while accommodating the ultrasonic sensors. One of these sensors is mounted at the front, where it benefits from existing space around the differential gear system. Placing the sensor here avoids the need for additional material spacing, which would have been necessary if it were positioned near the steering mechanism.

The rear-wheel steering allows the vehicle to pivot more sharply around the front wheels, making it extremely agile in tight spaces. This enables more precise adjustments when navigating through an obstacle course which is particularly useful when passing through obstacles.

Steering

Using a tricycle approach, our vehicle uses a single steering axle, controlled directly by a servo motor. This allows us to simplify the steering design while also allowing a wide range of steering angles for optimal mobility.
The servo motor will be the steering mechanism of our vehicle. For our steering mechanism we only required the turning axle to turn about 90 degrees
The MG996R servo motor can rotate 90 degrees and has been tested to be more accurate than its previous models.

Actuation

DC Motor & Full Metal Differential Gear Set:

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.

So, 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 rayed 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.

The main function of the differential is to allow the wheels to turn at different rpms. 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

The motor we chose provides sufficiently high RPM and torque whilst not being too expensive. Our needs as calculated above, are sufficiently met with a relatively controllable yet high max speed and sufficient power to weight ratio for a small robot.

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
Raspberry Pi 4:	5V, 3A
Battery:	5000mAh 3cell 25C, 11.1v

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.

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

We use a separate Servo battery since our main battery provides to much power to the servo even with a regulator.

Sense

Controller: Raspberry Pi Module

We chose to use a Raspberry Pi instead of an Arduino because a Raspberry Pi is an operating system, allowing us to run more complex operations. Additionally, Raspberry Pi and Arduino are both popular choices in robotics projects, providing extensive online resources for support. We're also using VNC Viewer to remotely access and control the Raspberry Pi, which adds convenience and flexibility during development and testing. Moreover, since we plan to utilize the OpenCV library, which requires Python, Raspberry Pi is the ideal choice for our project.

- **Processing Power**: The Raspberry Pi 4 Model B is equipped with a quad-core ARM Cortex-A72 CPU, providing significantly more processing power compared to other microcontrollers like Arduino. This allows it to run complex algorithms, such as computer vision or machine learning, in real-time.
- **Versatility and Flexibility**: the Raspberry Pi 4 offers access to a vast ecosystem of software and programming languages. This flexibility enables easy integration with various sensors, cameras, and software libraries needed for advanced robotics tasks.
- **Image Recognition with OpenCV**: The Raspberry Pi 4 supports the Raspberry Pi Camera Module, which enables image recognition tasks. Coupled with OpenCV,

a widely used library for computer vision, the Raspberry Pi 4 can perform real-time image processing, object detection, and visual navigation. This capability allows your robotic vehicle to recognize objects, track movements, and make decisions based on visual inputs, providing a significant advantage in a dynamic environment.

- **Connectivity**: Built-in Wi-Fi and Bluetooth on the Raspberry Pi 4 are crucial for wireless communication, remote control, or IoT integration, which is particularly useful in competitions for connecting multiple devices. This is especially useful when coding and troubleshooting.
- **Multiple I/O Options**: The Raspberry Pi 4 offers multiple GPIO pins, USB ports, and HDMI outputs, allowing for interfacing with a wide range of peripherals, sensors, and modules without needing additional hardware.
- **I2C BUS connection**: The Raspberry Pi 4 provides SDA and SCL connection for I2C. This is because both the Gyro (mpu6050) and Magnetometer (TB6612FNG) use I2C. By using the I2C bus, the gyro and magnetometer will have their unique I2C bus address, preventing data overlap. I2C is used for these components since it allows for data exchange between microcontrollers with minimum wiring.

Ultrasonic Sensor: US-015

For our vehicle, we have strategically placed three ultrasonic range finders at the front, left, and right sides. These sensors play a crucial role in manoeuvring by helping the vehicle avoid obstacles like traffic pillars and maintaining a safe distance from the external and internal boundary walls of the field. The side sensors are particularly useful for detecting the side walls, ensuring the robot stays centred. At the front, we've positioned both a camera and an ultrasonic sensor to detect obstacles and measure distances effectively.

We're using the US-015 model, which offers a measuring range from 2 to 400 cm with an accuracy of up to 1 mm. Its stability and consistency make it ideal for precise robotics operations like this one.

Camera: PI Camera

The Raspberry Pi Camera is mounted on the front top of the vehicle, serving as the 'eyes' of the robot. Its primary function is to detect obstacles and identify their colour, which allows it to send signals to the robot on which direction to turn. By placing the camera in this position, it optimizes the vehicle's ability to efficiently detect obstacles.

We chose the Raspberry Pi Camera Module 3 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.

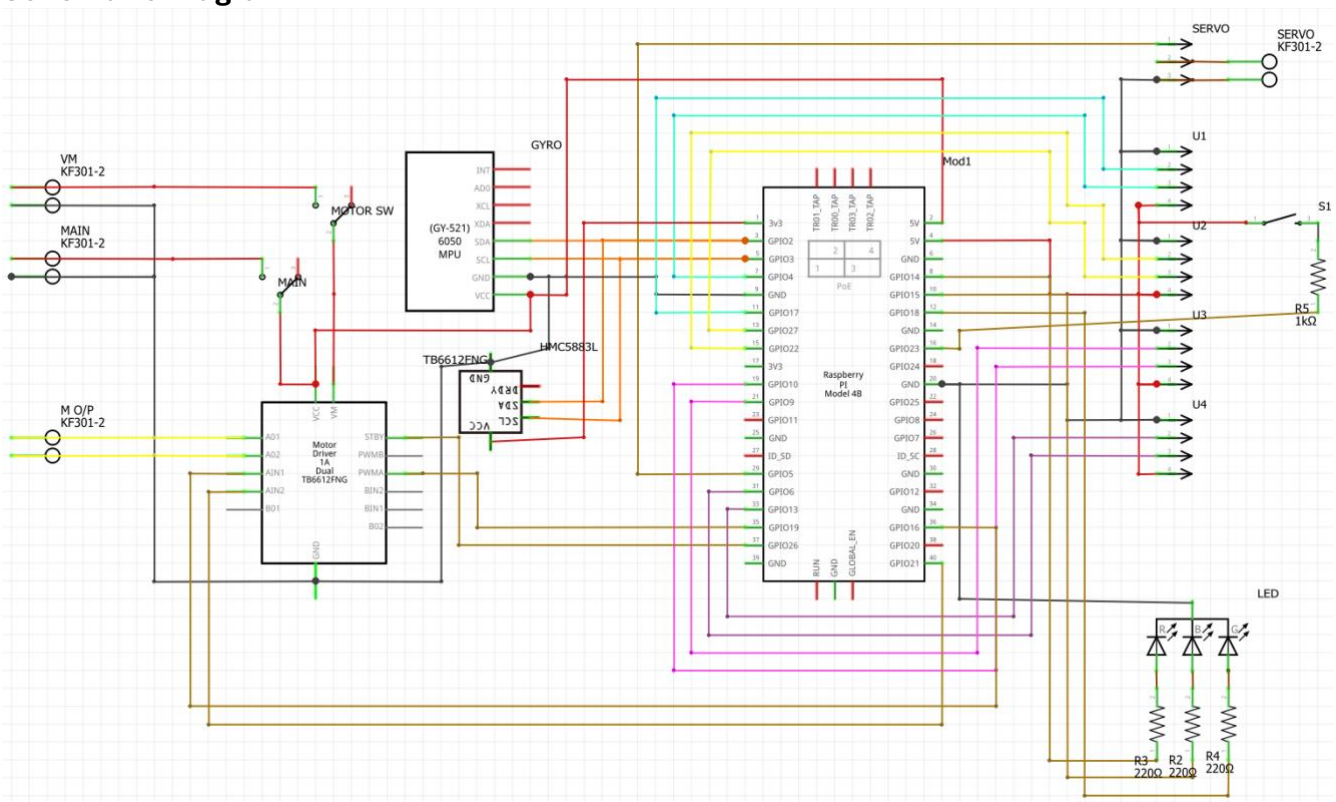
Magnetometer: HMC5883L Module Triple Axis compass

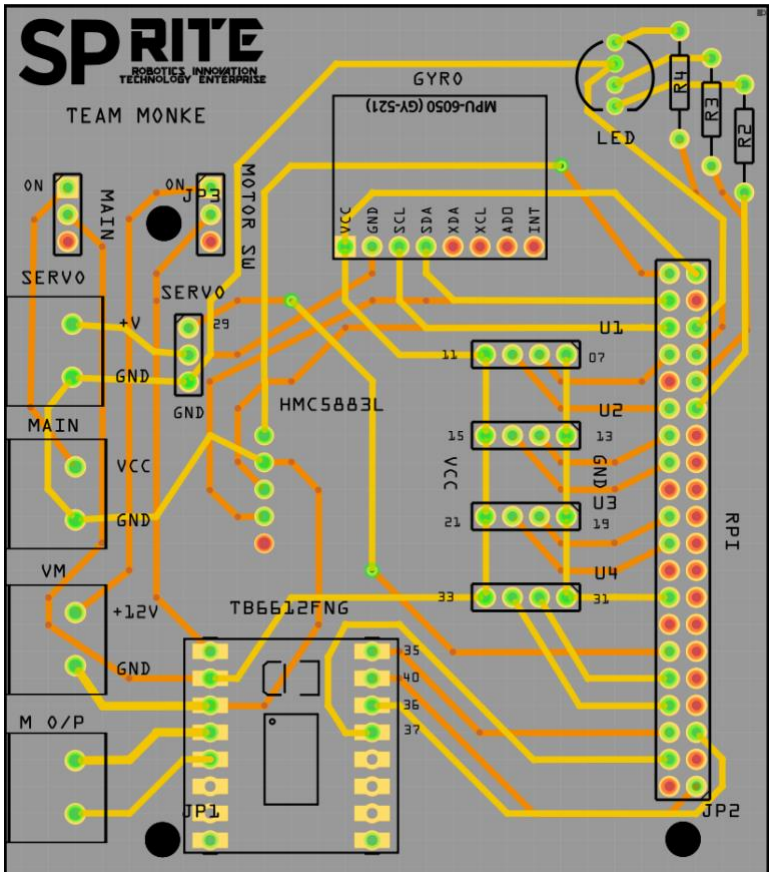
We are using the HMC5883L is a magnetometer sensor to derive the direction of our robot. This will enable the robot to be able to make more refined movements. For example, the robot will be able to determine how many rounds it has done.
The main reason we chose the HMC5883L was due to its accessibility and price and our usage purpose. A gyroscope was not suitable since we required a direction heading and a gyroscope would require algorithms and further modifications to achieve that. The HMC5883L allowed us to obtain the direction of the robot at a relatively low price.

Integration: Printed Circuit Boards

To integrate all our sensors and microcontrollers, we required something to combine them together and provide the necessary power with I/O connection. Therefore, we chose to use a custom printed PCB. Two other options were available to us as well. A breadboard or a perforated board. Breadboards was not a very long-term solution, and breadboard connections are not very secure, particularly on a moving robot. Not to mention mounting of the breadboard would also cause issues. Perforated boards offer more secure connections, but the soldering of all connections will be time consuming. Furthermore, we wanted a solution that could last and could be reused for other projects. Therefore, we chose to use a PCB since it allowed us to have secure connections, enable stacking of header pins on the Raspberry Pi and less soldering required. This also provided a valuable learning experience in how to PCB design.

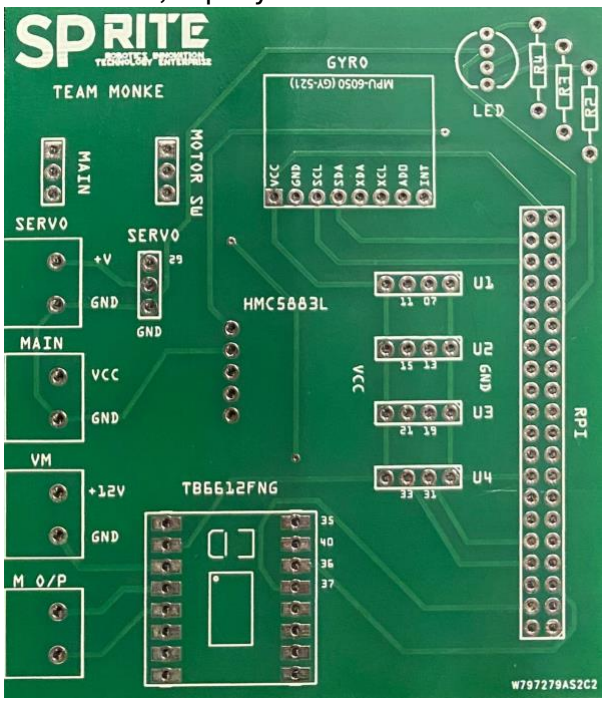
Schematic Diagram:

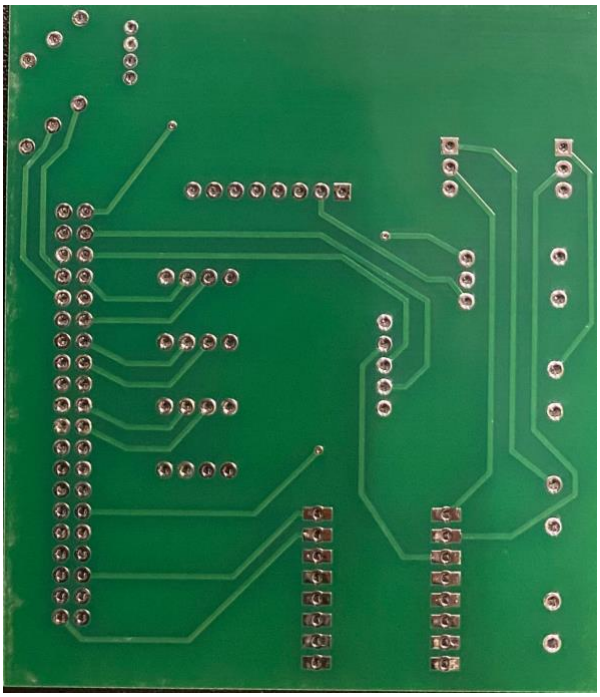




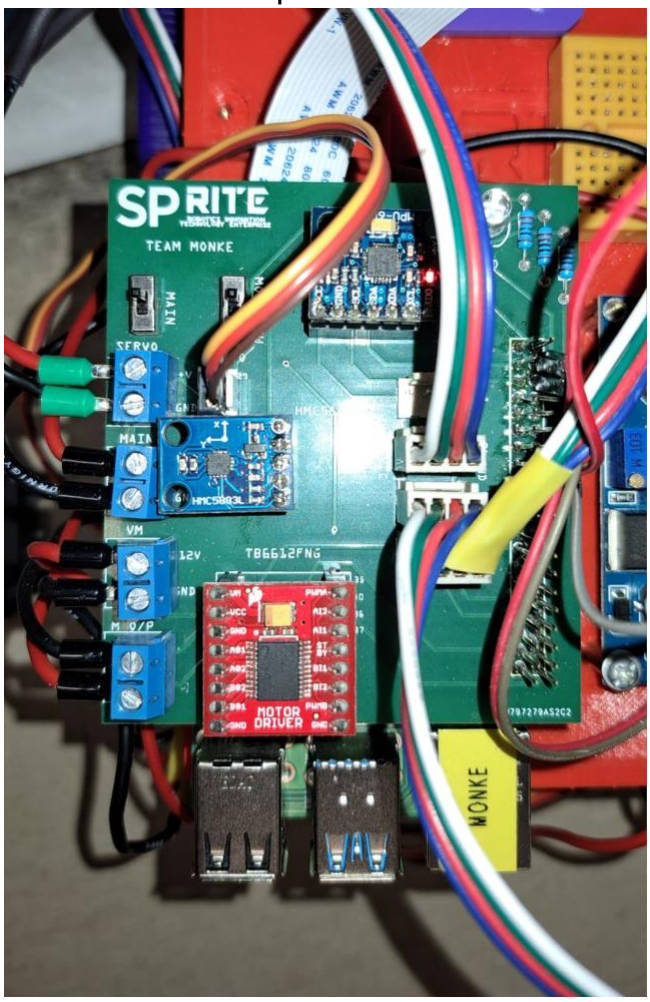
Actual PCB, Top layer

Bottom layer





PCB with all components mounted on our robot

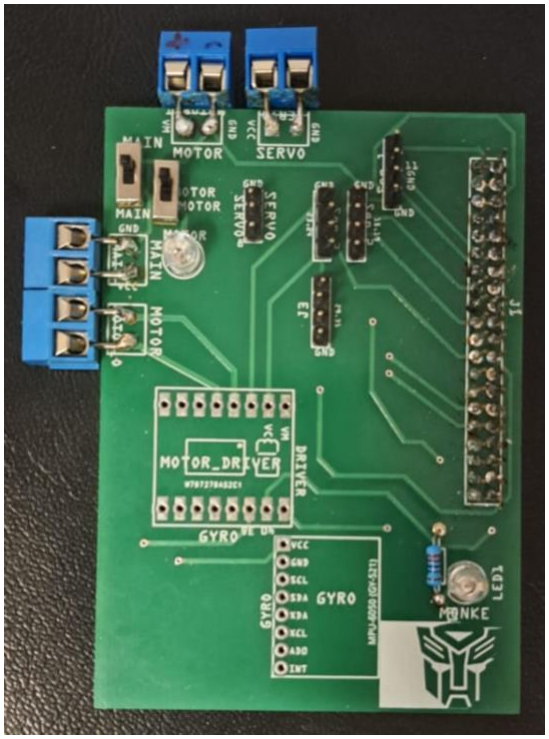


DEVELOPMENT MILESTONES

Fabrication

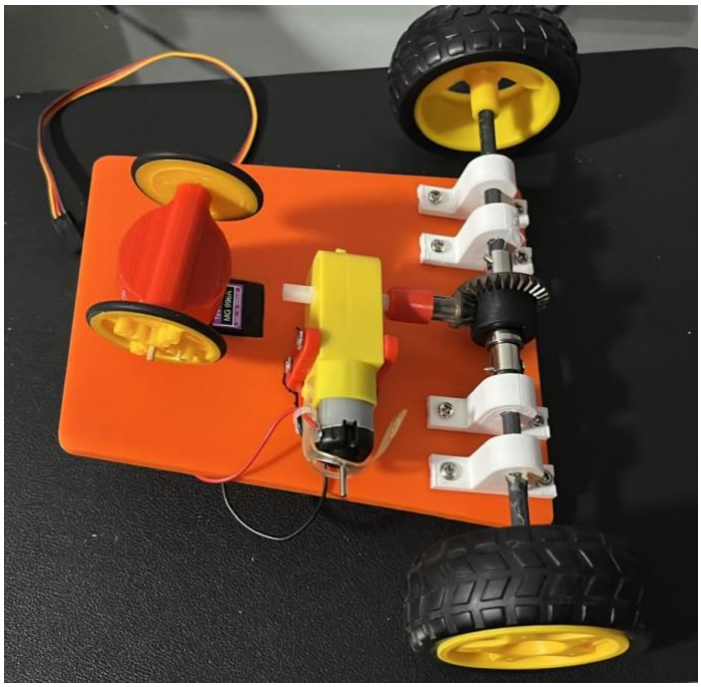
PCB

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.

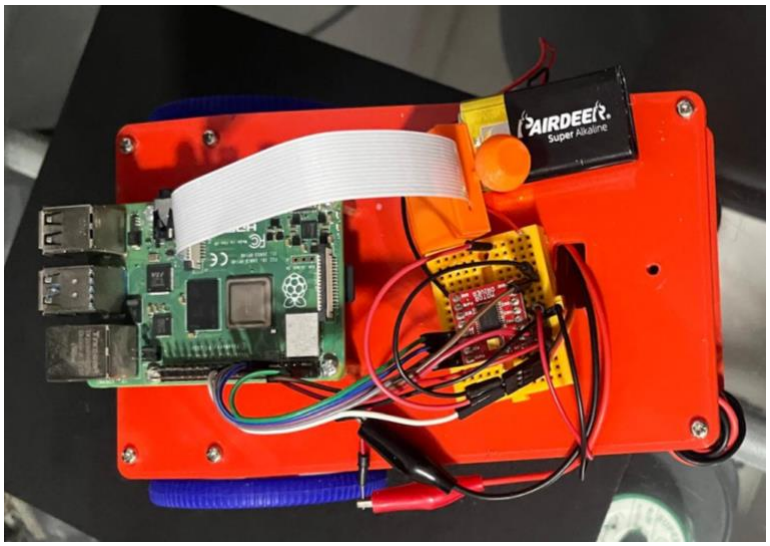
AGV



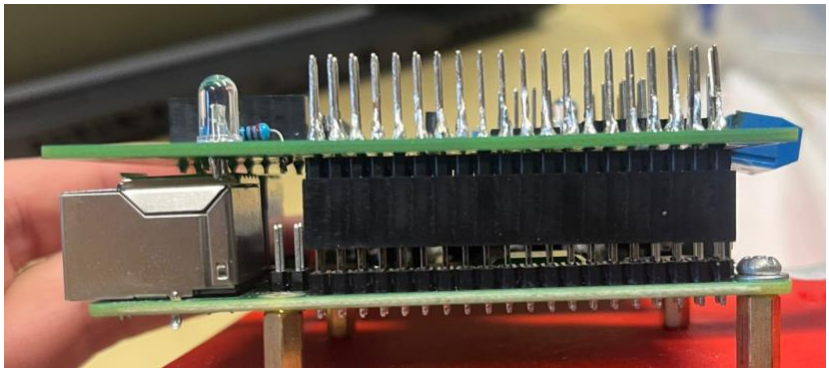
This is our very first AGV that we tried. This was meant as a proof of concept, testing out the differential gear drive and tricycle steering.

Assembly

PCB

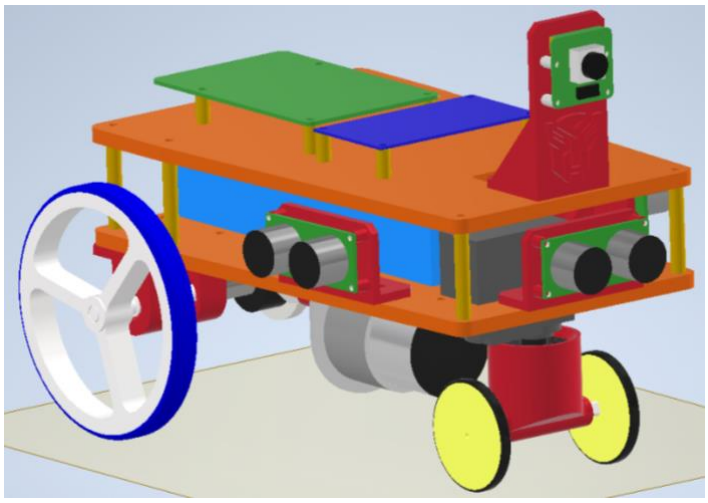


As our 2nd prototype, we used a breadboard to test out the motor driver. From insecure connections to messy wires, it was clear that we needed a PCB.



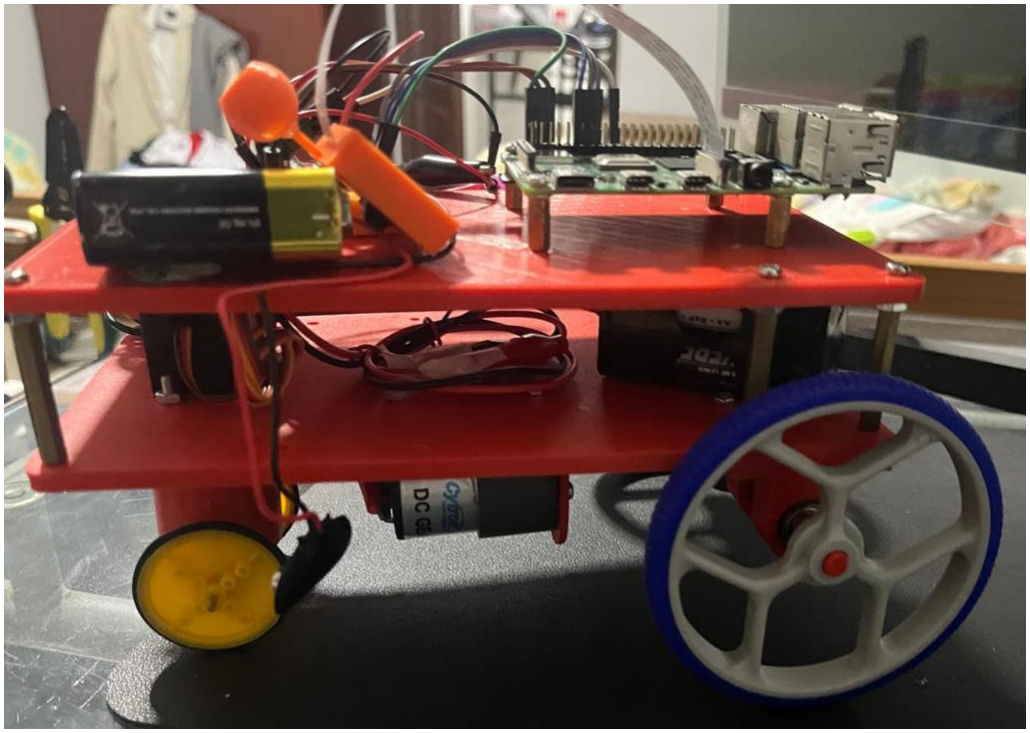
Upon assembling our PCB, we noticed that the Raspberry Pi's USB and ethernet ports were blocking it.

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.

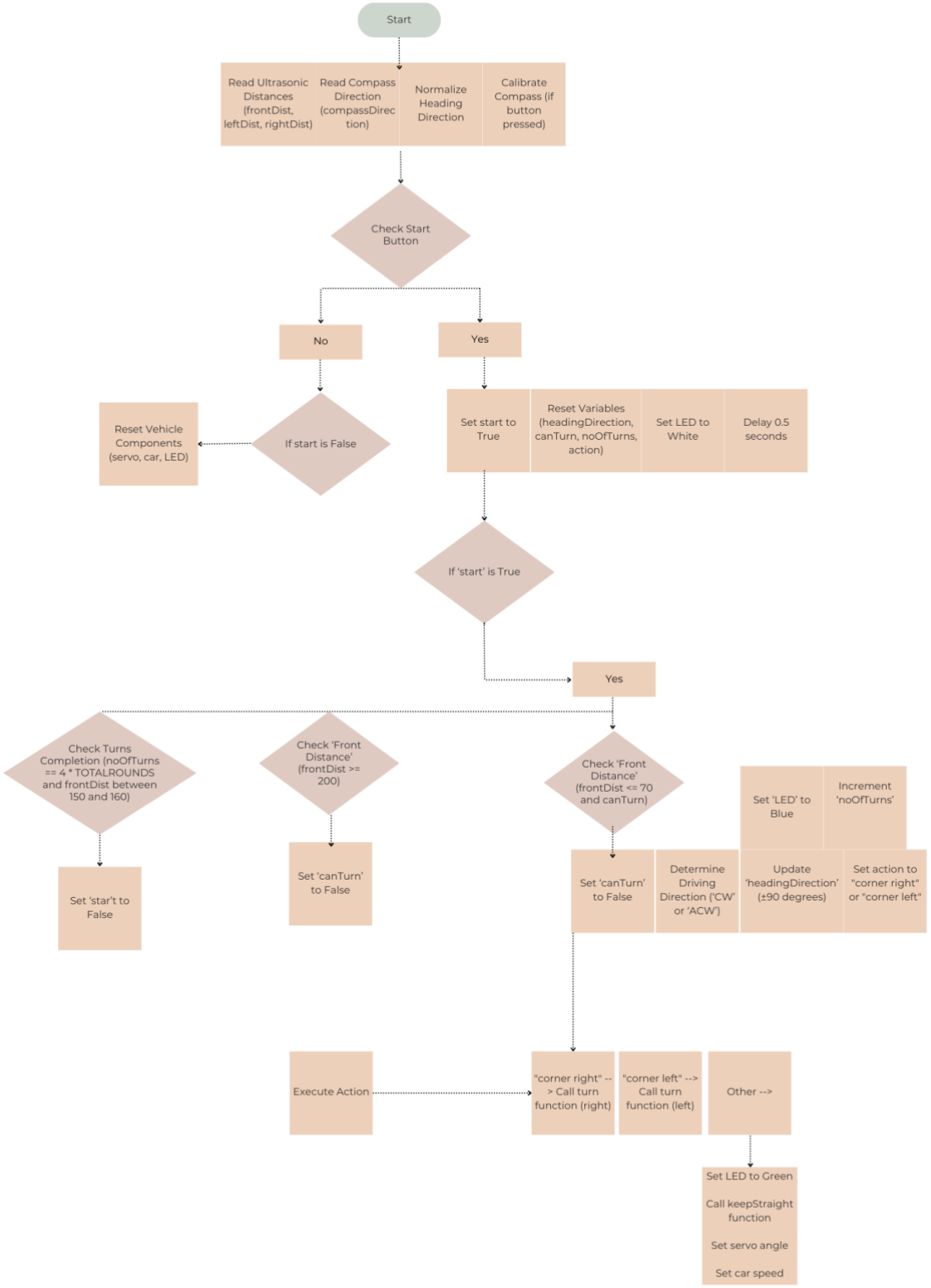
Troubleshooting

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.

PSEUDO CODE

Open Challenge:



CONCLUSION/IMPROVEMENTS

Sensors

1) Instead of Ultrasonic sensors, we can use Lidar sensors instead. Lidar is much more consistent and accurate compared to ultrasonic sensors. Our team encountered some difficulties in the consistency of the Ultrasonic sensor.
2) A different camera can be used. Our current camera does not have very high frames at higher resolutions. This limits our object detection via computer vision.

PCB

1) A push button can be added on the PCB so that we are able to start/stop the robot or give inputs during testing.

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

Programming References:

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

https://forum.arduino.cc/t/multiple-and-simultaneous-ultrasonic-sensor-controlling/266009

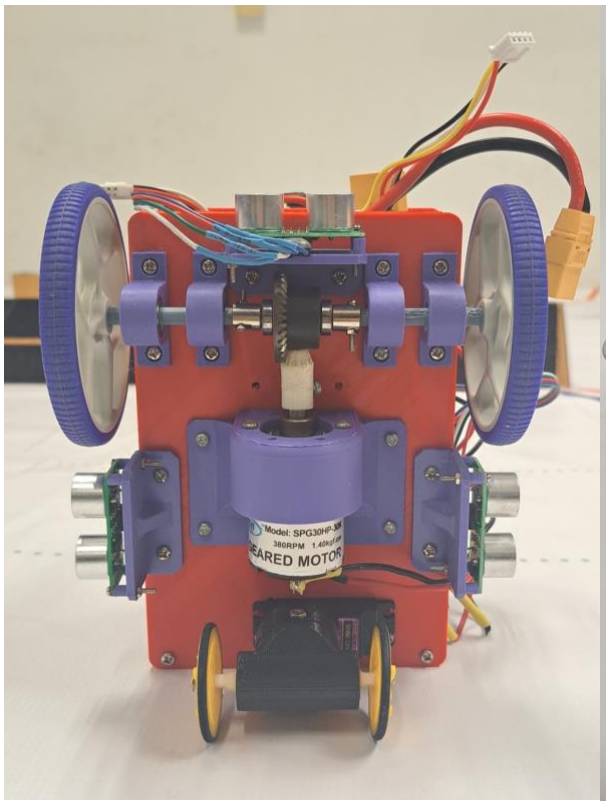
GitHub References:

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

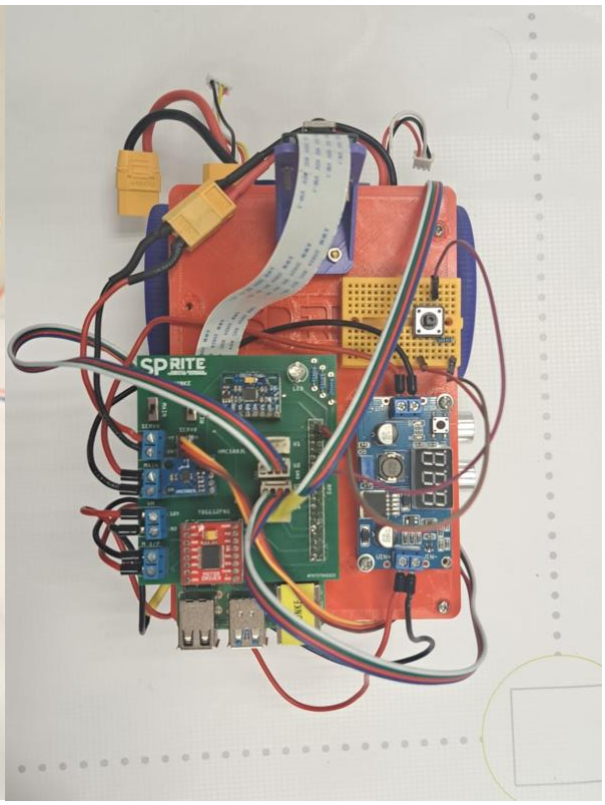
APPENDIX A

Below are the images of our AGV

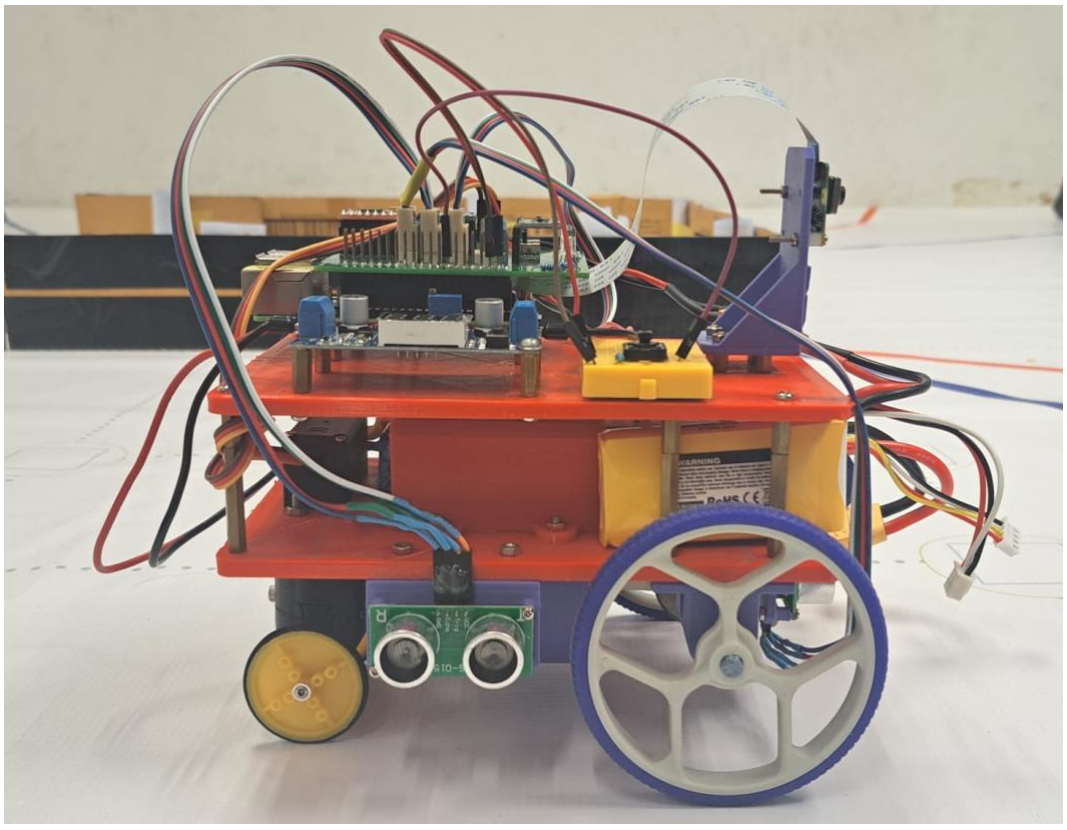
Bottom



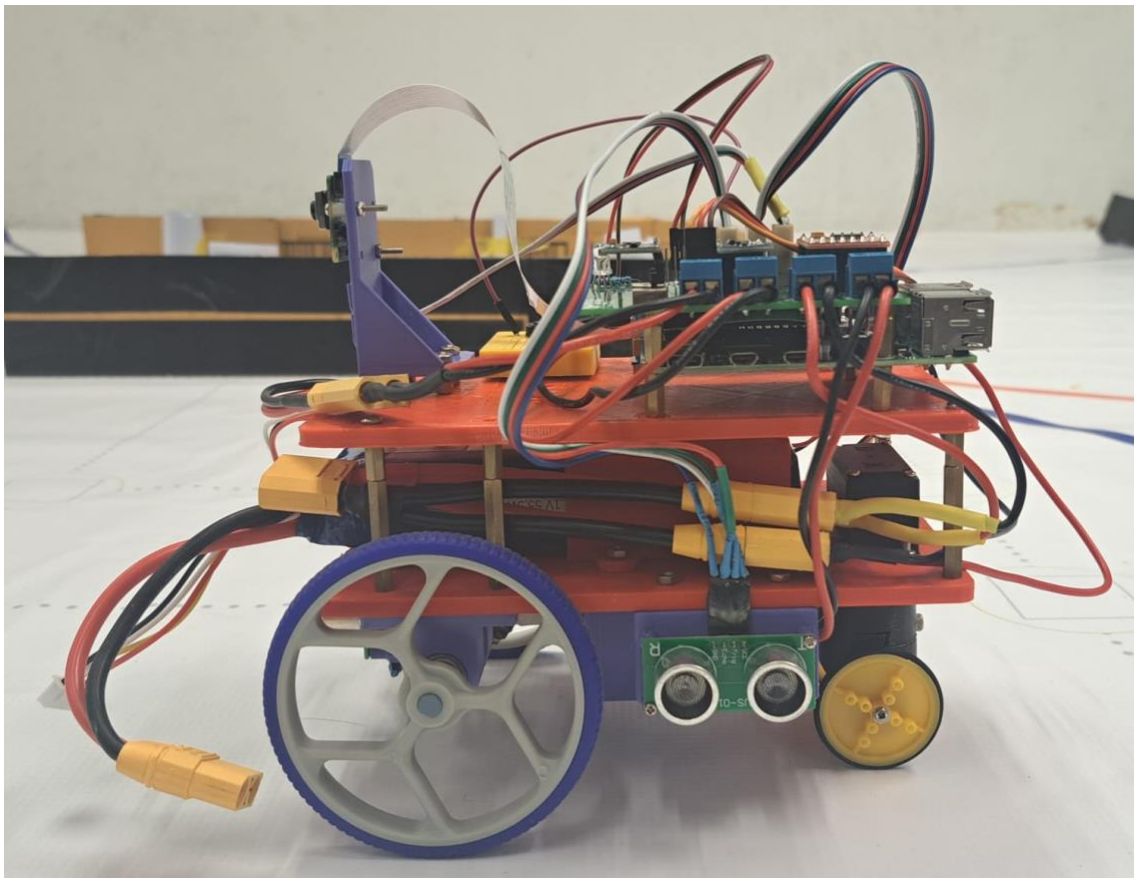
Top



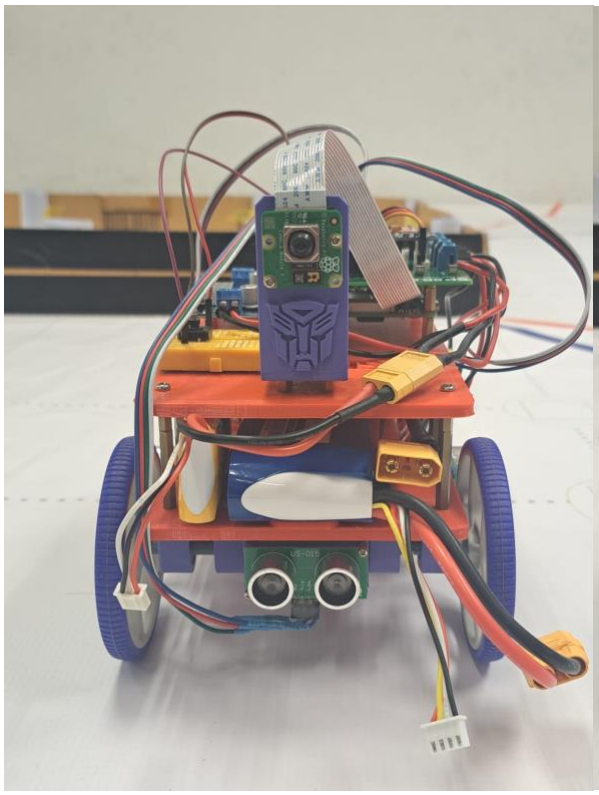
Right



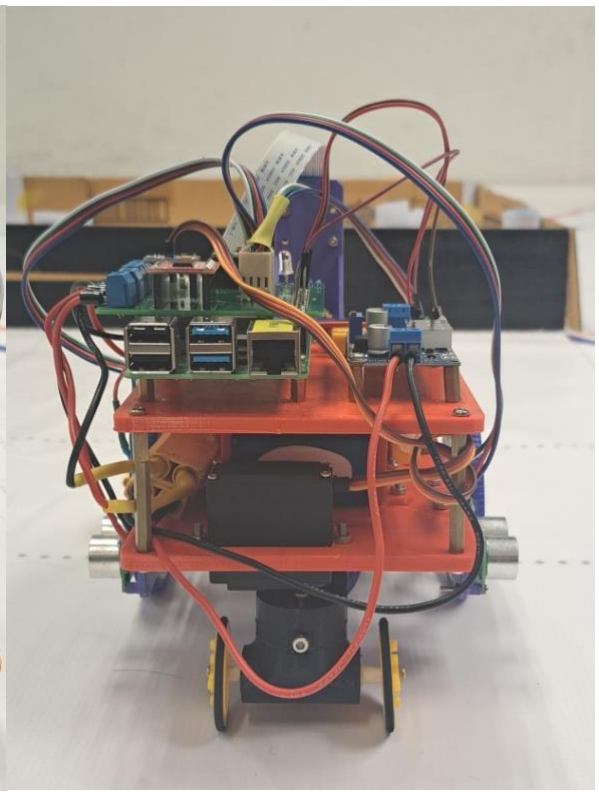
Left



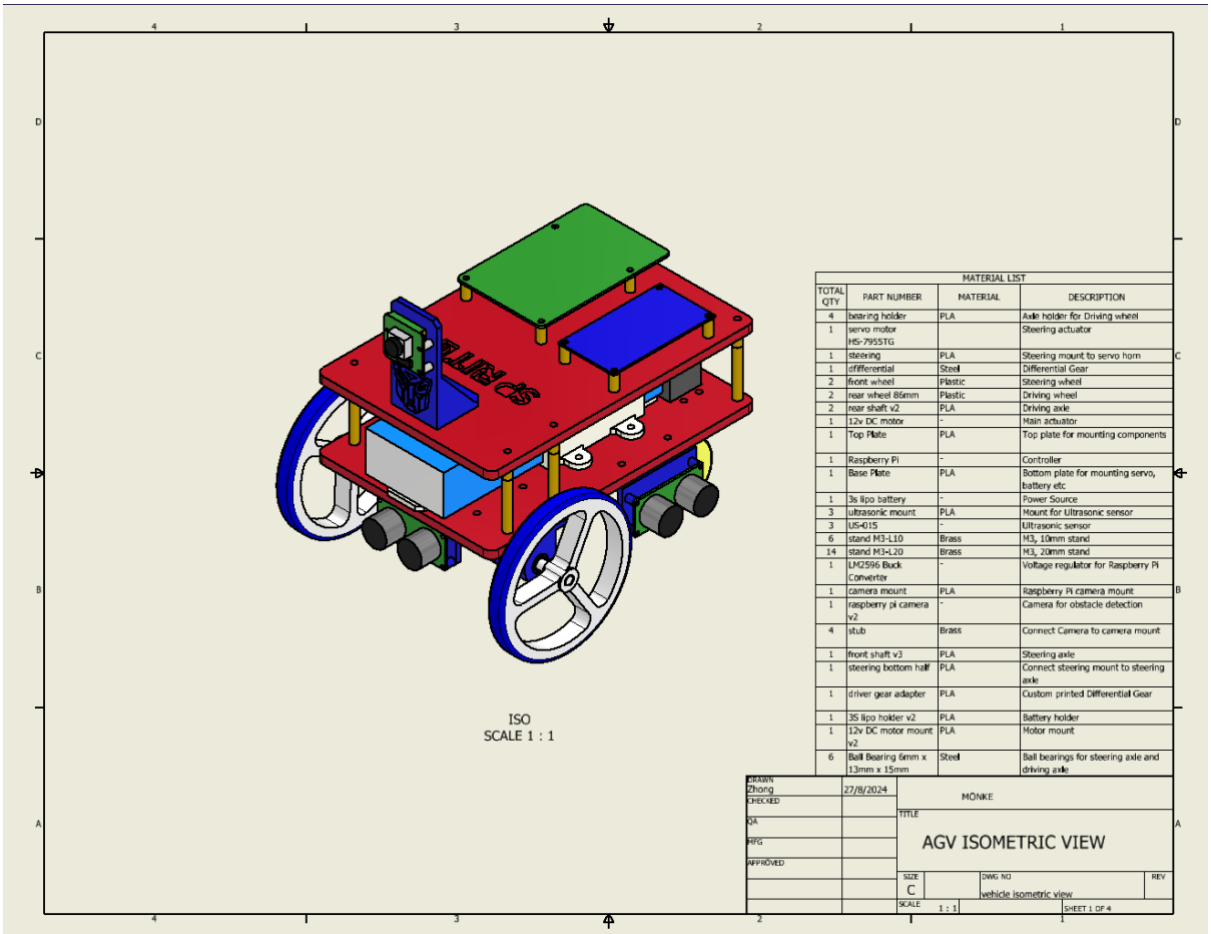
Front



Back



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



MATERIAL LIST

TOTAL QTY	PART NUMBER	MATERIAL	DESCRIPTION
4	bearing holder	PLA	Axle holder for Driving wheel
1	servo motor MS-7955TG		Steering actuator
1	steering	PLA	Steering mount to servo horn
1	differential	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

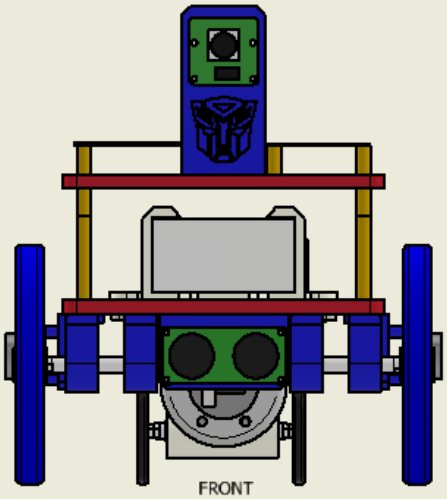
DRAWN	Zhong	27/8/2024
CHECKED		
QA		
MFG		
APPROVED		

MONKE

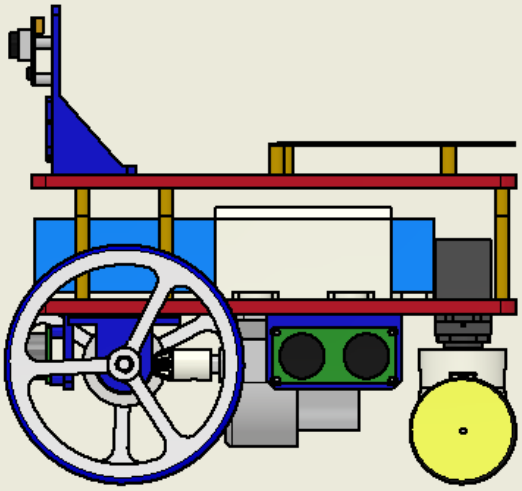
TITLE: AGV ISOMETRIC VIEW

SIZE: C | DWG NO: vehicle isometric view | REV

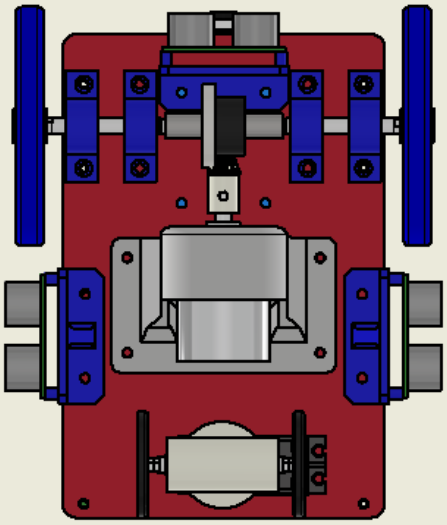
SCALE: 1 : 1 | SHEET 1 OF 4



FRONT
SCALE 1 : 1



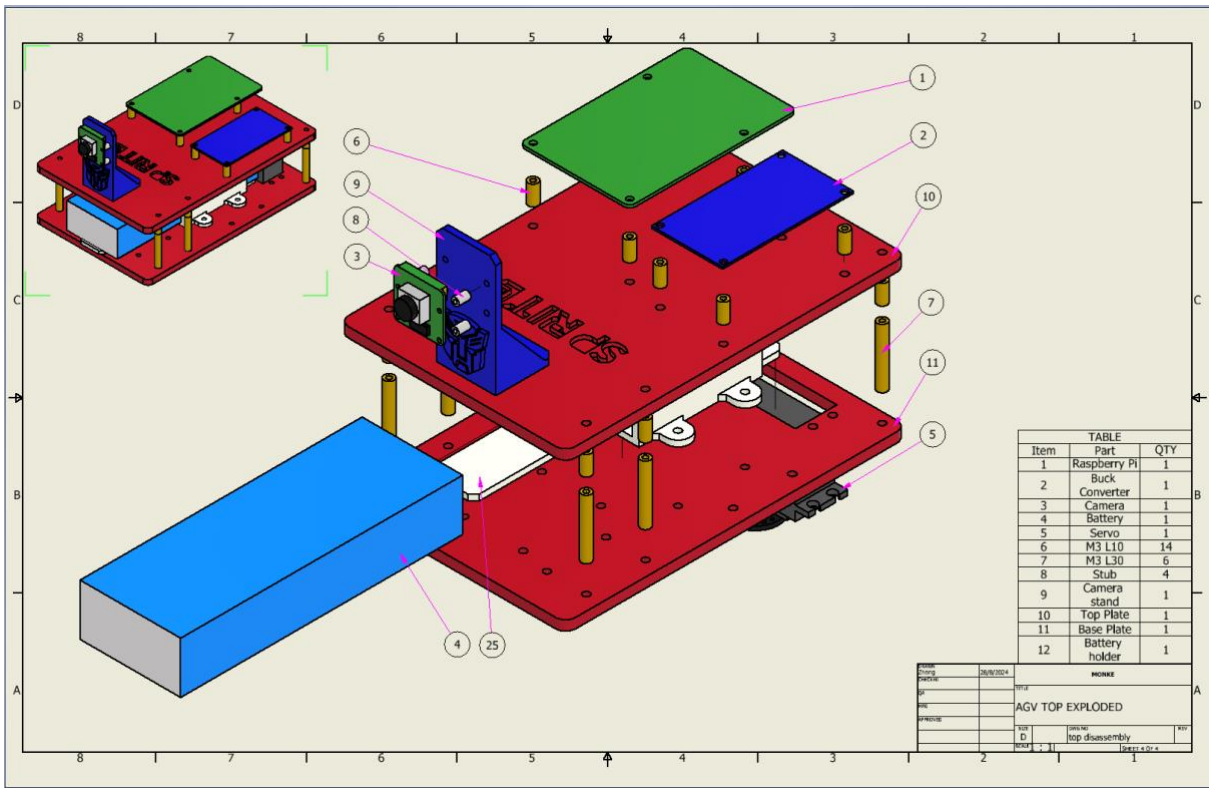
RIGHT
SCALE 1 : 1



BOTTOM
SCALE 1 : 1

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

Top part expanded



Bottom part expanded