WRO Engineering Journal

Team Information:

Team Name: SSTactical

Team ID: FE25-314-01

School: School of Science and Technology

Category: WRO Future Engineers

Team Members: - Chan Jia Sheng Michael ———— (Team Leader)
- Elliott Gee Hsien Zhi ————————- (Coder)

Contents

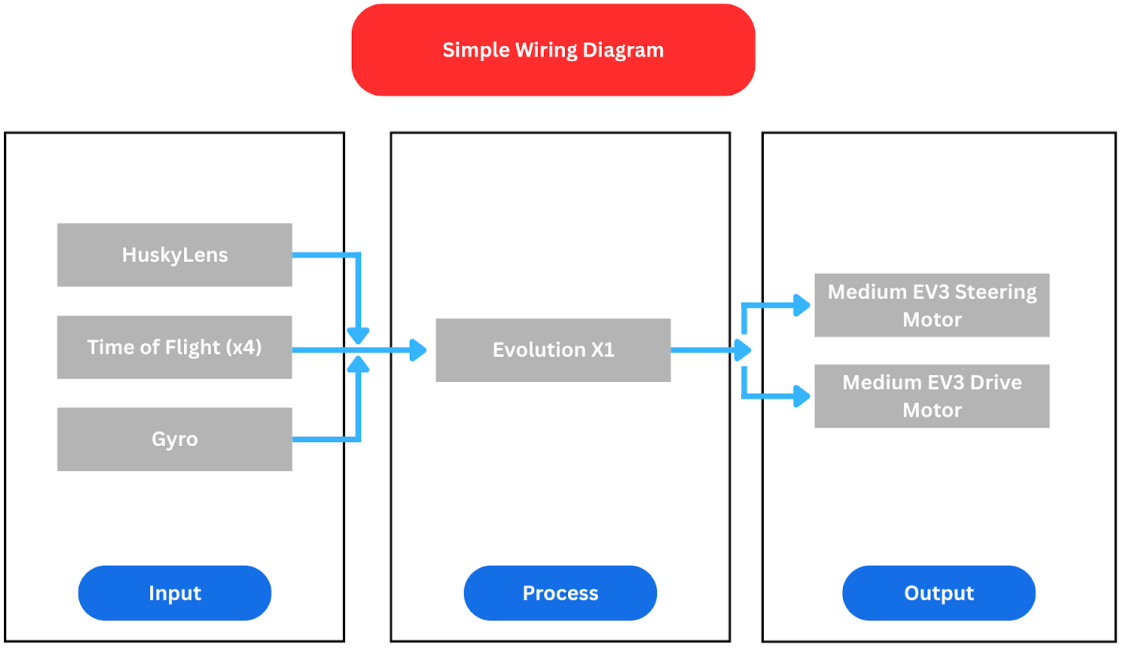
Hardware Overview

Table of Components used by the robot:

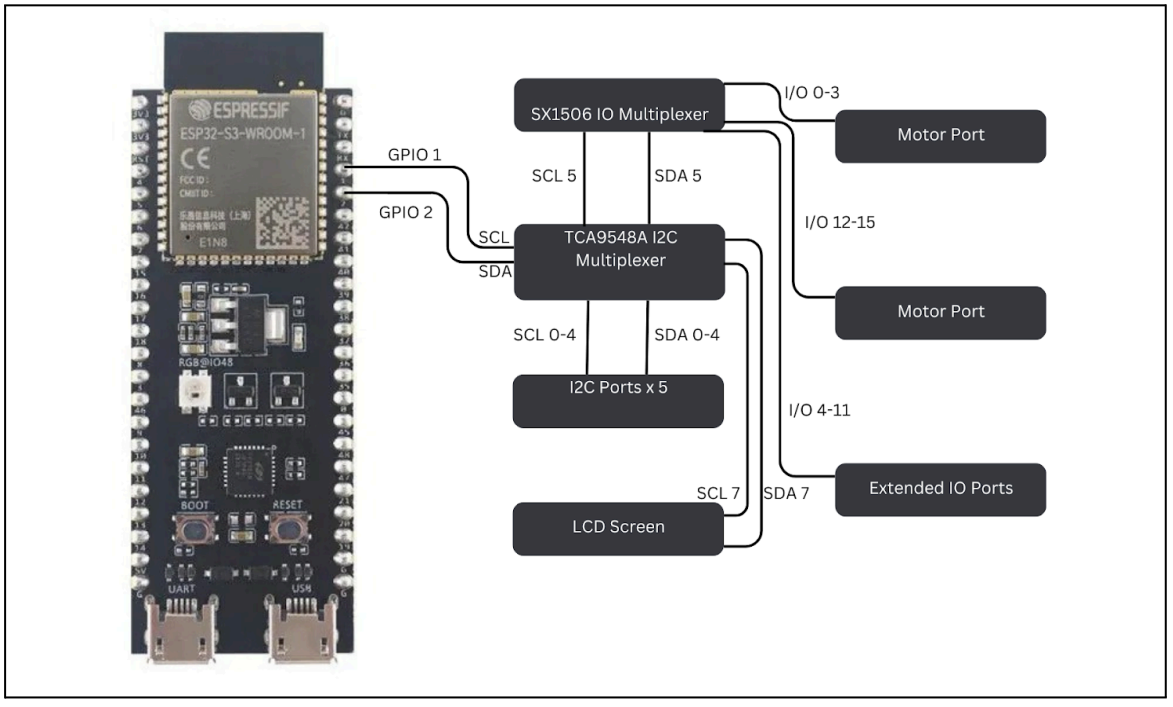
Part Name	**Model**	**Quantity**	**Others**
EV3 Motor	Medium Motor	2	
Time of Flight	VL53L0X	3	
Camera	HuskyLens	1	
Microcontroller	ESP32S3	1	
12C Microplexer	TCA9548A	1	
IO Multiplexer	SX1506	1	
IMU	BNO055	1	
Battery	NCR18650B	2	

Diagrams

Simple Wiring Diagram (Blocks)



Microcontroller System Diagram



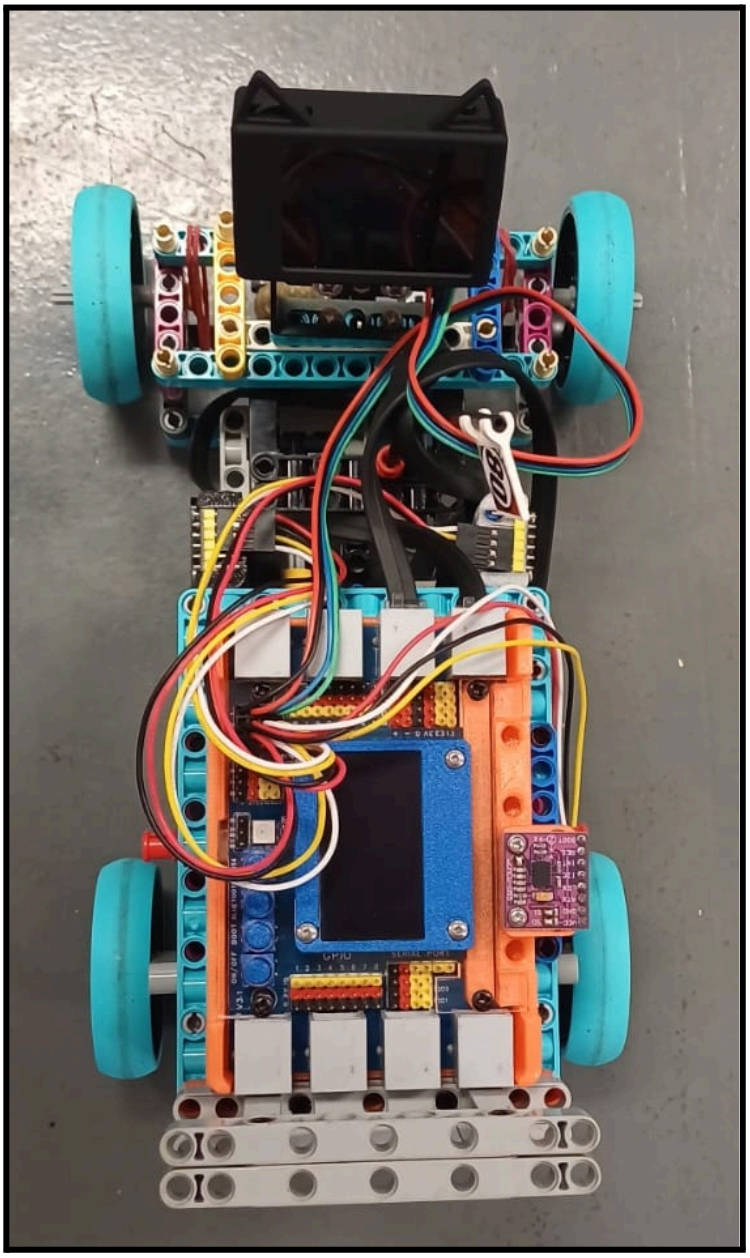
Drivetrain

Chassis Design

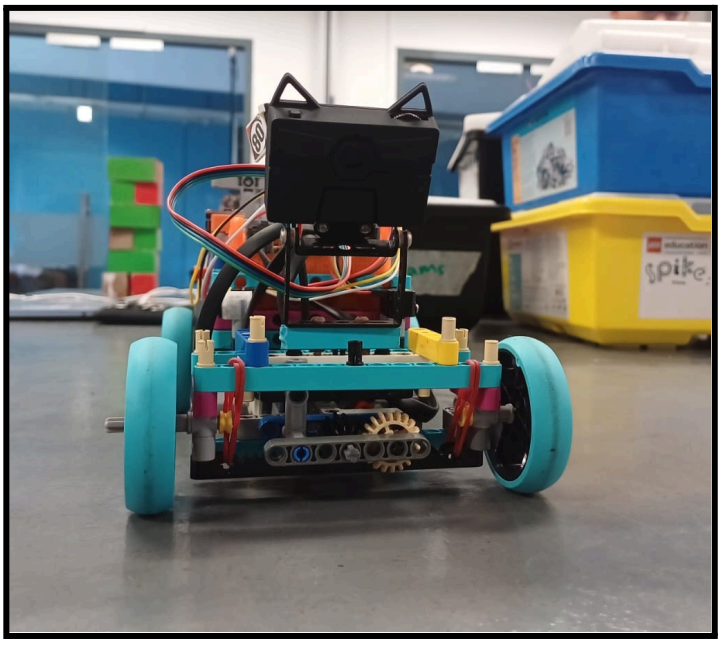
To build our chassis, we used LEGO parts, as they were light yet able to withstand the weight of the EvolutionX1 Brick. LEGO parts made it easy for us to prototype our models, as we built 5 prototype chassis before landing on the final design. We also made use of 3D-modelled holders for our sensors with LEGO compatibility; thus, it is easy to install and replace.

Here are some photos of the vehicle:

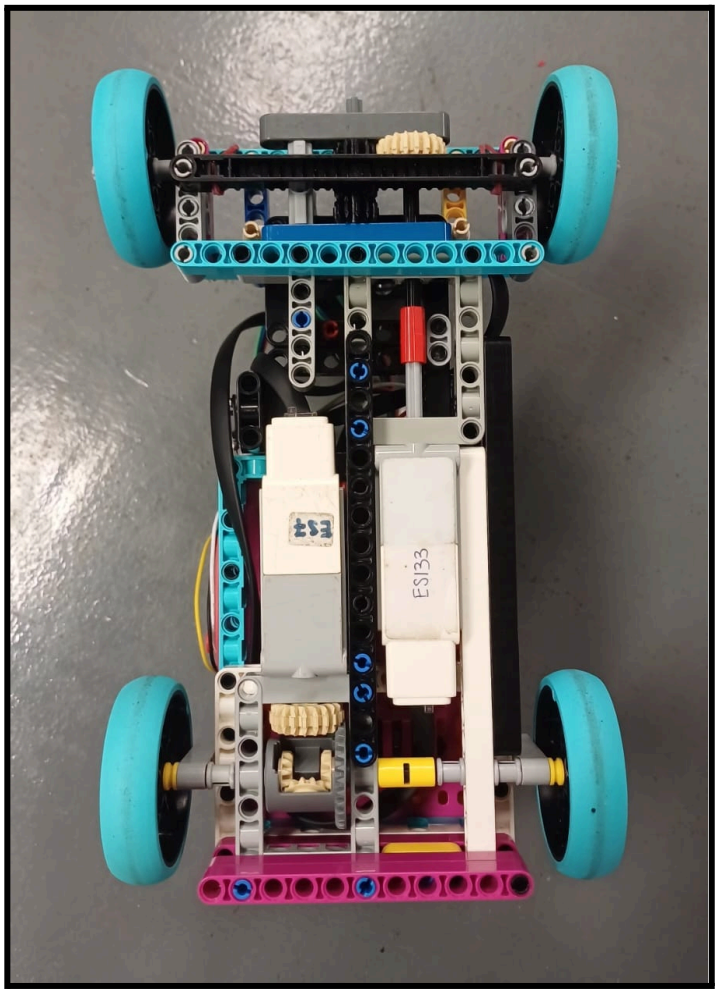
Top View



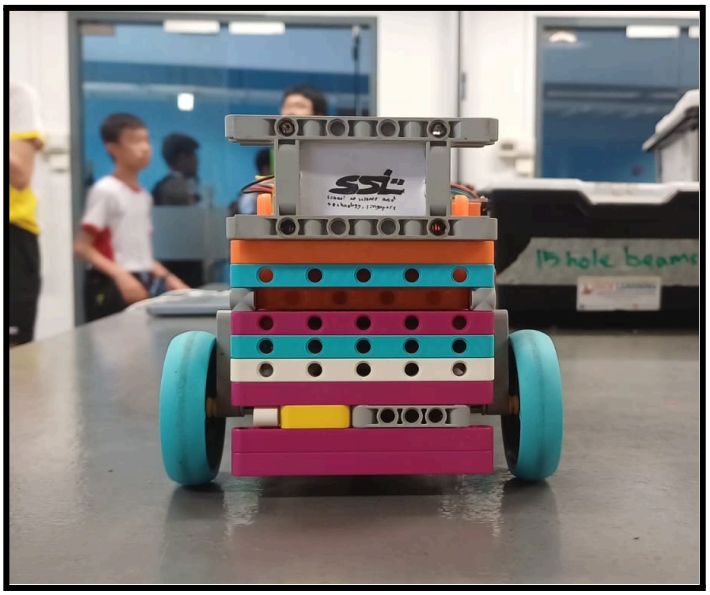
Front View



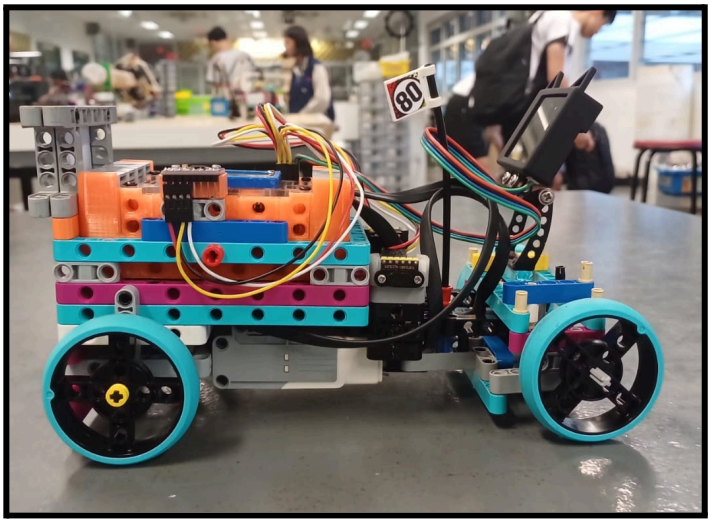
Bottom View



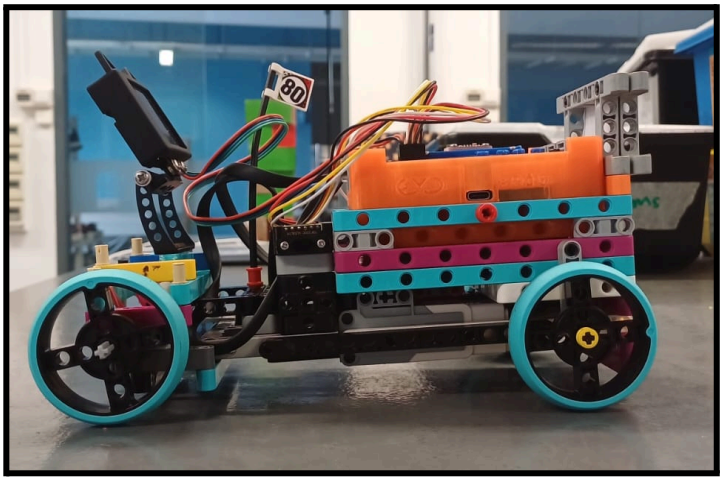
Back View



Right Side View

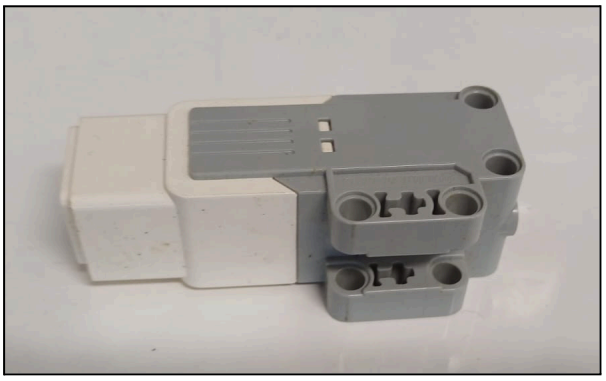


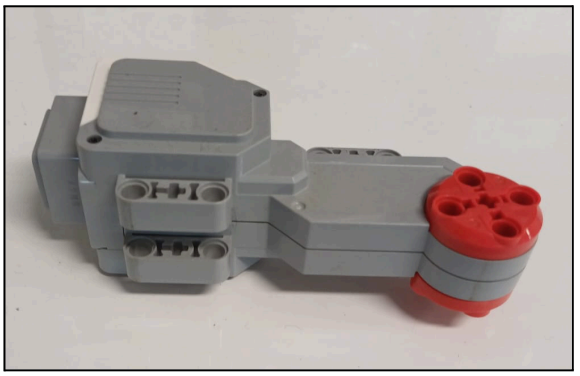
Left Side View



Motors

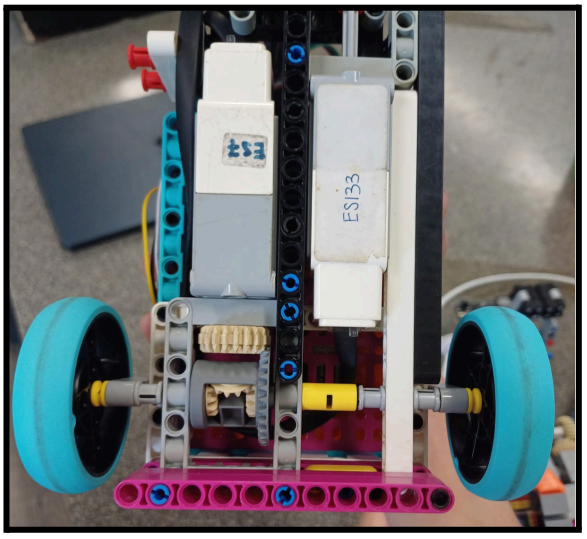
The motors we have chosen to use as a part of our bot is the EV3 Medium Motor, which has a balance of speed and torque compared to the EV3 Big Motor. The other reasons why we chose this motor were due to the size limit and the challenge of navigation. We felt that having a smaller bot would be easier to navigate through the blocks, so we chose to go for compactness.





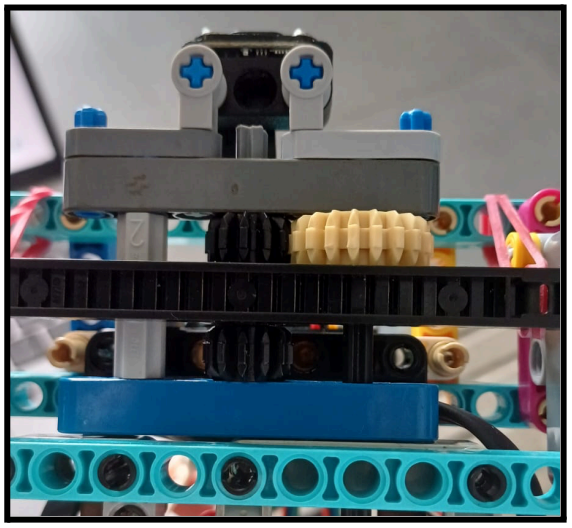
Differential

The drive motor is connected to the differential gearing at the bottom of the bot. Having differential allows the robot's wheel to turn at different speeds, which is crucial to achieving smooth turning.



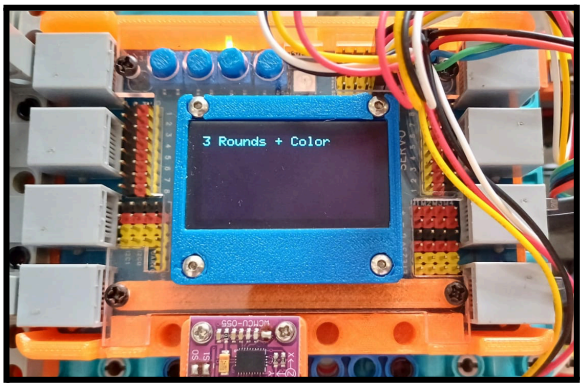
Steering

For our steering, our team has chosen the rack and pinion steering mechanism, which is a more common steering mechanism used by many people in the world, while we also wanted to go for a design of an SUV. Additionally, we used WD-40, which is a lubricant for our steering rack, as it wasn't in the position we liked. Thus, we made an extension to the gears to enable a larger turn angle, as the gear rack would shift up and down the gears, which caused the steering to fail. As a result of these failures, we added extra gears to facilitate the larger turn angle.



Components

Evolution X1



The Evolution X1 Controller Platform is a versatile and powerful development board designed to seamlessly integrate with the LEGO EV3 Mindstorms ecosystem. Built around the robust ESP32-S3 microcontroller, this platform offers a unique combination of connectivity, performance and flexibility. Below are the key features of Exolution X1, and these are some reasons why we chose it.

Key Features:

- **EV3 Mindstorms Compatibility**
  The controller is equipped with specialised connectors that enable seamless connection and control of the EV3 Mindstorms motors and sensors. This compatibility allows us to utilise the existing EV3 components while taking advantage of the advanced features of the ESP32-S3.

- **Extended GPIO Pinouts**
  In addition to EV3 connectors, the platform offers an extensive set of GPIO pinouts. These pinouts allow us to connect various third-party sensors and actuators, broadening the scope for customisation and experimentation.

- **Powerful ESP32-S3 Microcontroller**
  The platform is powered by the ESP32-S3 microcontroller, which stands out for its great performance, energy efficiency, and versatile features. It offers dual-core processing, built-in Wi-Fi and Bluetooth, and plenty of peripheral options, making it a flexible and reliable choice for all kinds of applications.

- **User-Friendly Development Environment**
  The platform supports popular development environments such as Arduino, MicroPython, and Espressif's IDF, providing a user-friendly and familiar programming experience. It is also backed by comprehensive documentation and strong community support, enhancing the development process.

Time of Flight (ToF)

A Time of Flight (ToF) sensor measures distance by emitting a light signal, typically infrared, and calculating the time it takes for the signal to travel to an object and return. This time is then used to determine the precise distance to the object.

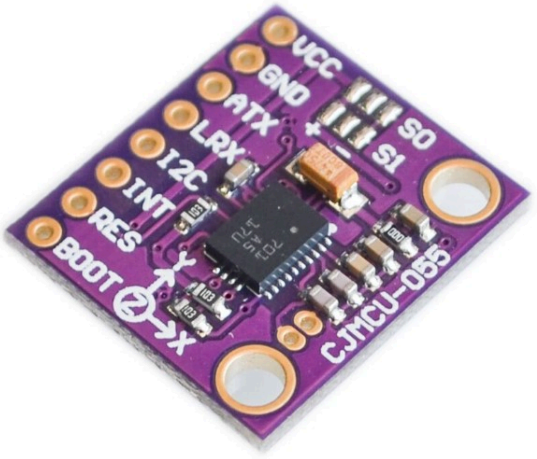
We selected it primarily for its precise distance measurement capabilities. ToF sensors can measure distances with exceptional accuracy—often down to the millimetre—making them ideal for applications that demand precise measurements. They can also measure over a wide range, from just a few centimetres to several metres, depending on the model.

Another advantage is their fast response time. ToF sensors deliver rapid measurements, enabling real-time functions such as gesture recognition, object tracking, and obstacle detection. Since they operate using the speed of light, they can perform distance calculations extremely quickly, providing low-latency data essential in dynamic environments.

Finally, ToF sensors are easy to integrate. They typically use standard communication protocols like I²C or SPI, making them straightforward to connect with microcontrollers and other digital systems. Some models even feature built-in processing, simplifying their use in more complex implementations.

Gyroscope (IMU)

For our gyroscope, we chose the BNO-055. This is a compact 9-axis sensor that combines an accelerometer, gyroscope, magnetometer, and a 32-bit microcontroller with sensor fusion software.

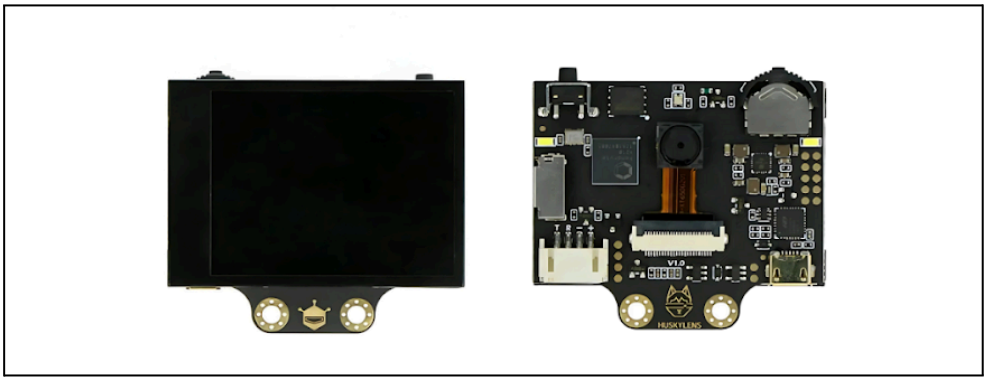


Our main reasons for choosing this component are, firstly, that by integrating an accelerometer, gyroscope, magnetometer, and microcontroller into a single module, it reduces the need for multiple separate sensors, thus reducing the overall data processing requirements and saving memory. Secondly, its compact form also minimises physical space usage, making it ideal for our compactly designed robot.

HuskyLens Camera

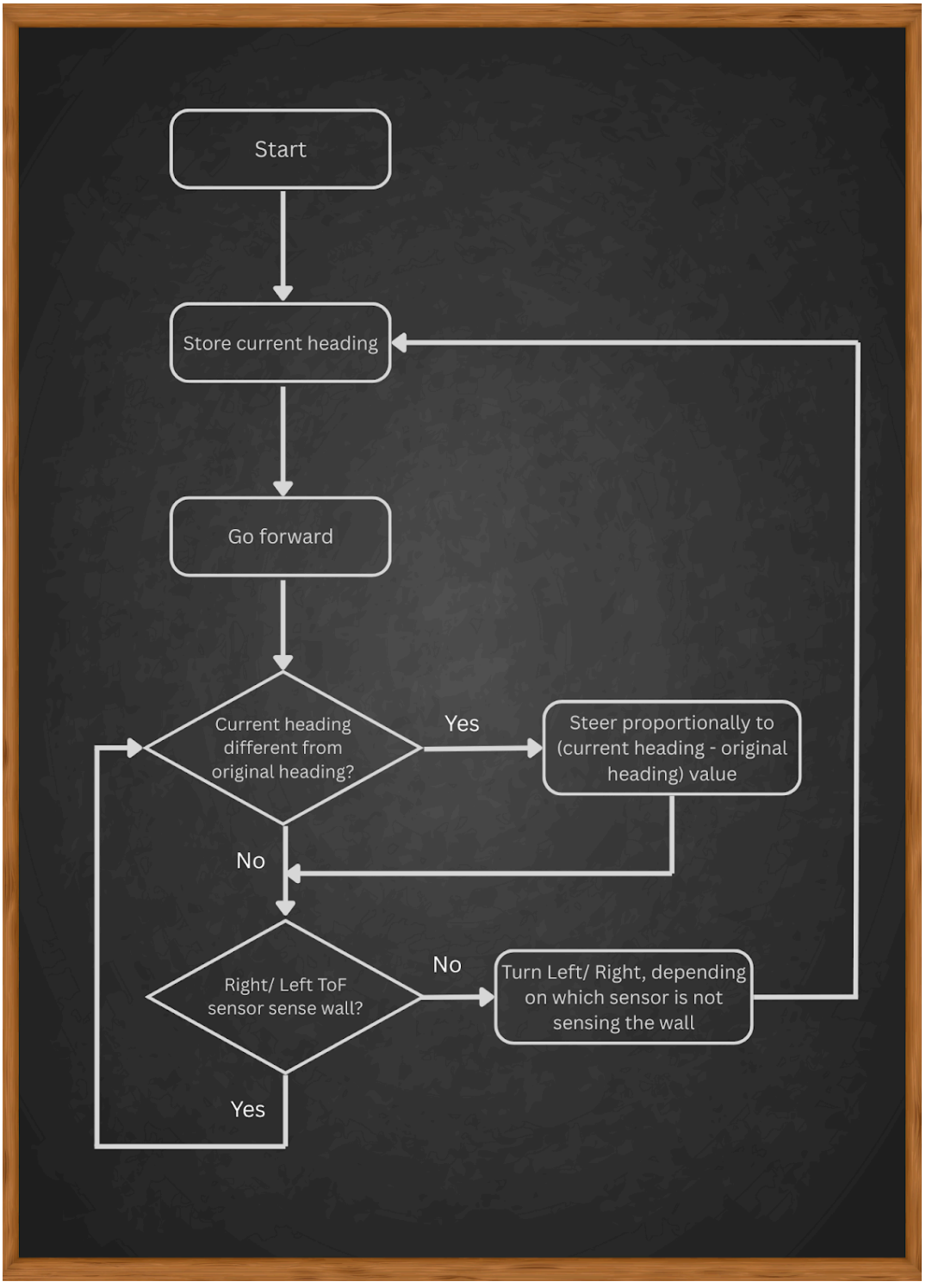
The HuskyLens is an AI vision sensor capable of detecting objects, colours, and more. It comes with various AI algorithms, including object tracking and colour recognition.

We selected the HuskyLens primarily because it features built-in object and colour recognition, which aids in obstacle detection during the challenges. Additionally, its rear-facing screen makes testing more convenient. Lastly, its compact and portable design ensures easy installation on our robot.

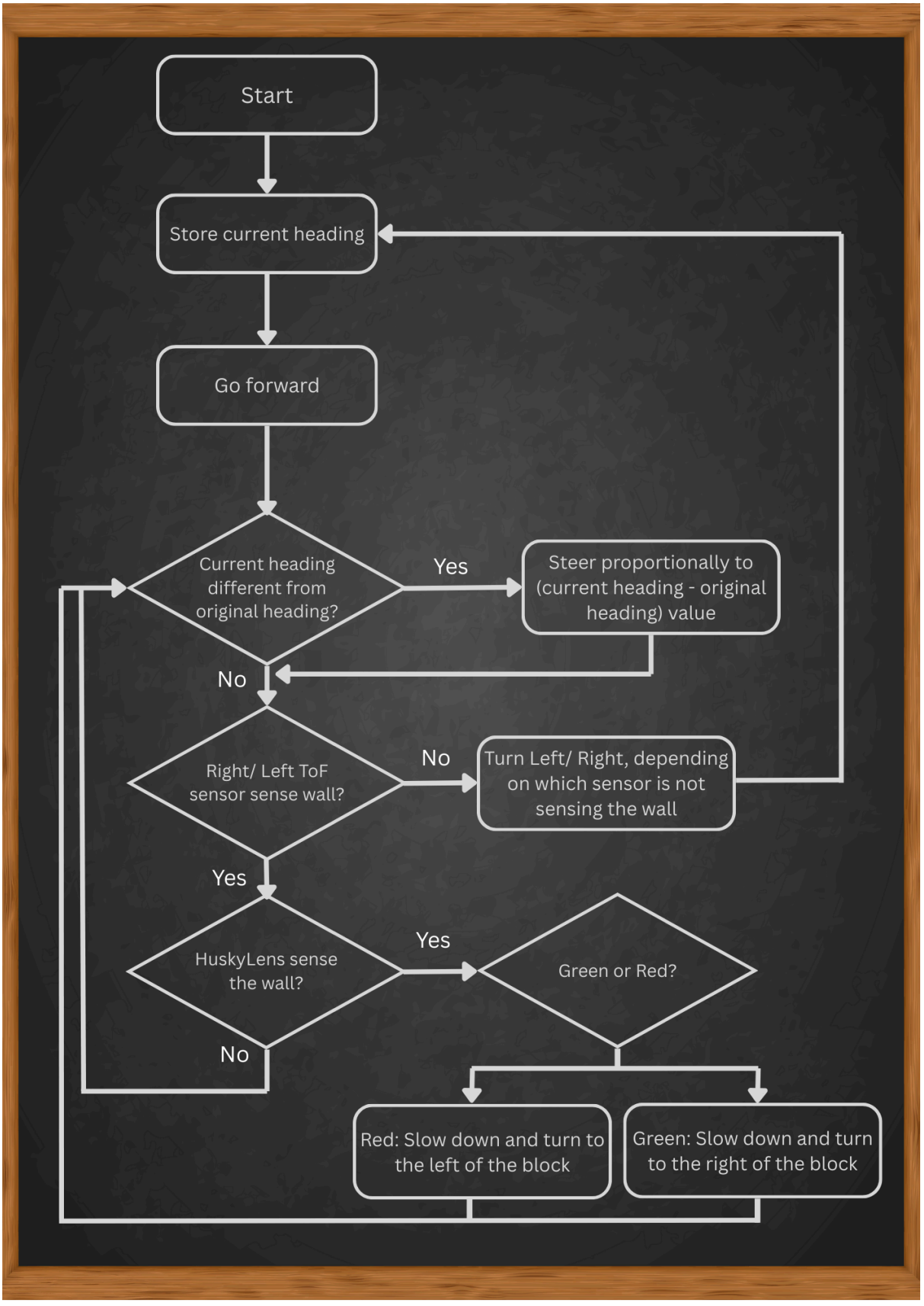


Strategy

Preliminary Round Flowchart



Obstacle Round Flowchart



Code

In order to make sure everyone on the team could understand the code, we focused on keeping it readable and easy to follow. Even though the code for the self-driving robot is complex, we broke the code into smaller parts and added clear labels.

We kept the code clean by doing the following:

- Wrote comments at important parts of the code to explain what each section does.
- Used libraries and separate classes to keep the main code simple and less crowded.
- Gave variables and functions clear and meaningful names so it's easy to know what they do.

- END -