COCOROBO

CocoPi

AI for STEM Competition

Asia Pacific STEAM_AI Technology Innovation Challenge

07 Circular movement

Primary school

COCOROBO

目錄

CONTENTS

COCOROBO

CHAPTER

POWER

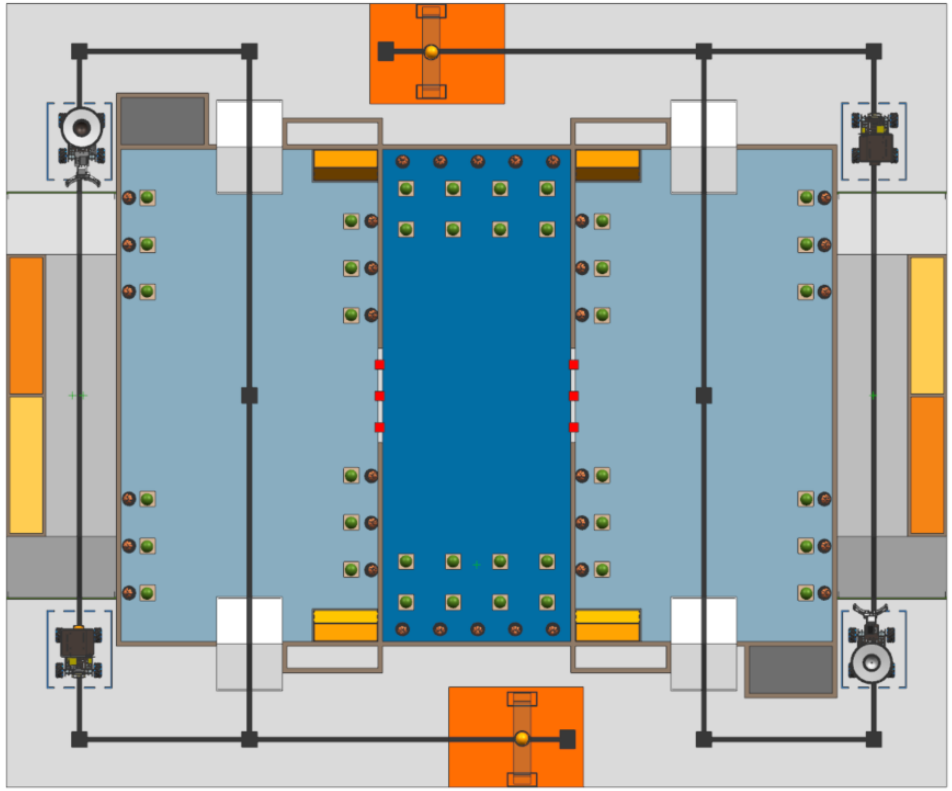
ONE.

Grayscale sensor

CHAPTER

Principle of automatic line following

- For the autonomous vehicle to follow the guided route, it only needs to distinguish between white and black. Is there any way to make the AI car distinguish between the two?



Basic Principles of Automatic Line Following

The principle of line following in intelligent vehicles is based on photoelectric sensors to detect road markings and adjust the vehicle's driving direction through control algorithms, thereby achieving automatic line following.

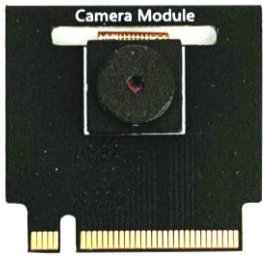
Photoelectric sensors

To accurately detect the path, the sensor must have good responsiveness to the contrast between the path markings and the background colour. Therefore, the reflected light intensity of the path markings should be significantly different from that of the background.

Control algorithms

Motion control can be achieved by adjusting the motor speed to move the car forward, backward, left, or right. The control algorithm can be designed according to actual requirements to achieve more precise motion control.

Common Types of Sensors Used for Line Following



Lens

Colour sensor

Grayscale sensor

Infrared sensor

Grayscale sensor

Photosensitive receiver tubes

Transmitter tubes

A grayscale sensor is a high anti-interference photoelectric sensor that uses a high-brightness white spotlight LED as its light source. The emitted light reflects off different backgrounds and is received by a photosensitive receiver, whose impedance changes based on the intensity of the reflected light (stronger reflection results in lower resistance). A voltage divider and an operational amplifier comparison circuit enable dual digital/analog signal output.
This sensor module excels at distinguishing background differences based on white light reflection intensity, with greater contrast improving accuracy. Compared to standard grayscale sensors, it offers superior anti-interference capabilities.

The connection of the grayscale sensor

S1

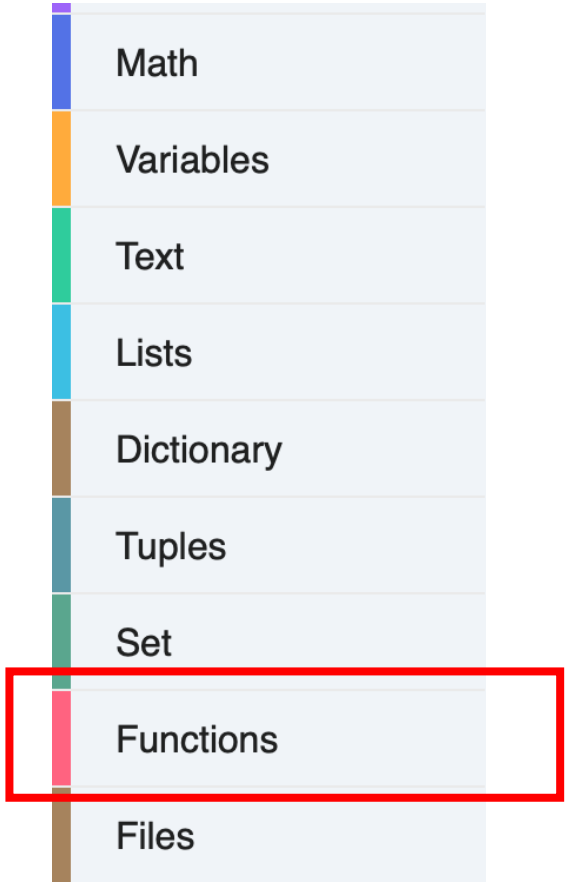
S2

Notes: Flatten the connection wires as shown in the diagram, and then connect them to the S1 and S2 interfaces of CocoPi;

Note: The white wire connects to the S1 port, and the yellow wire connects to the S2 port

(There is only one direction for connection).

Grayscale Sensor

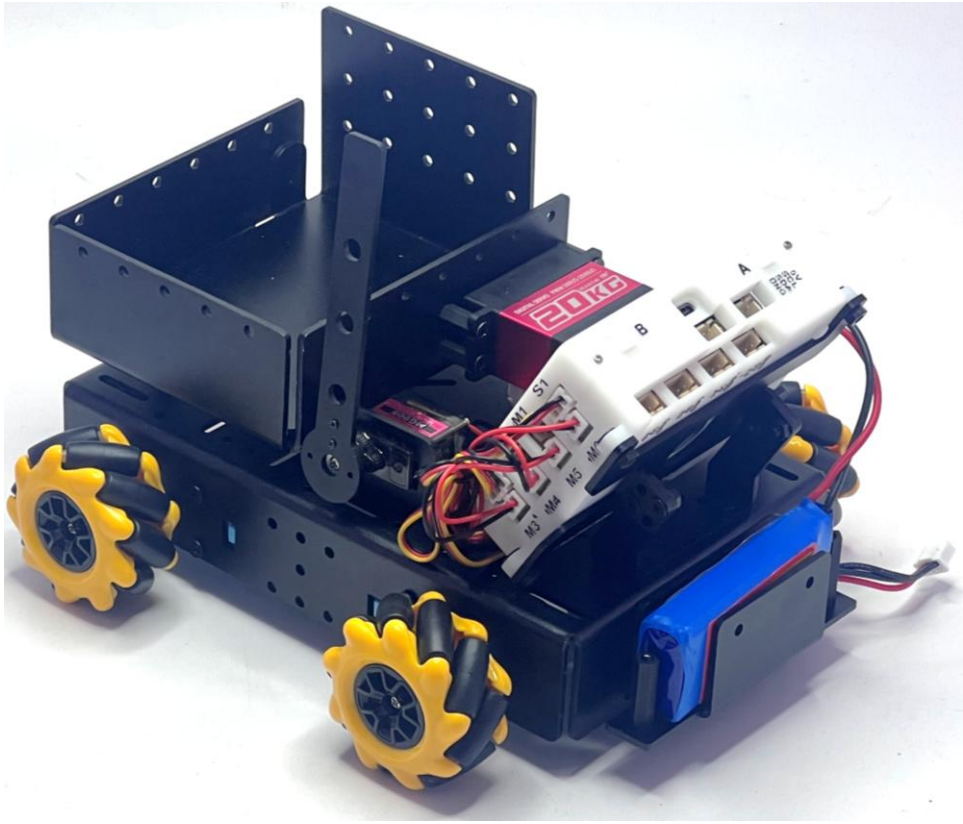


Corresponding pins for connection

Grayscale Sensor

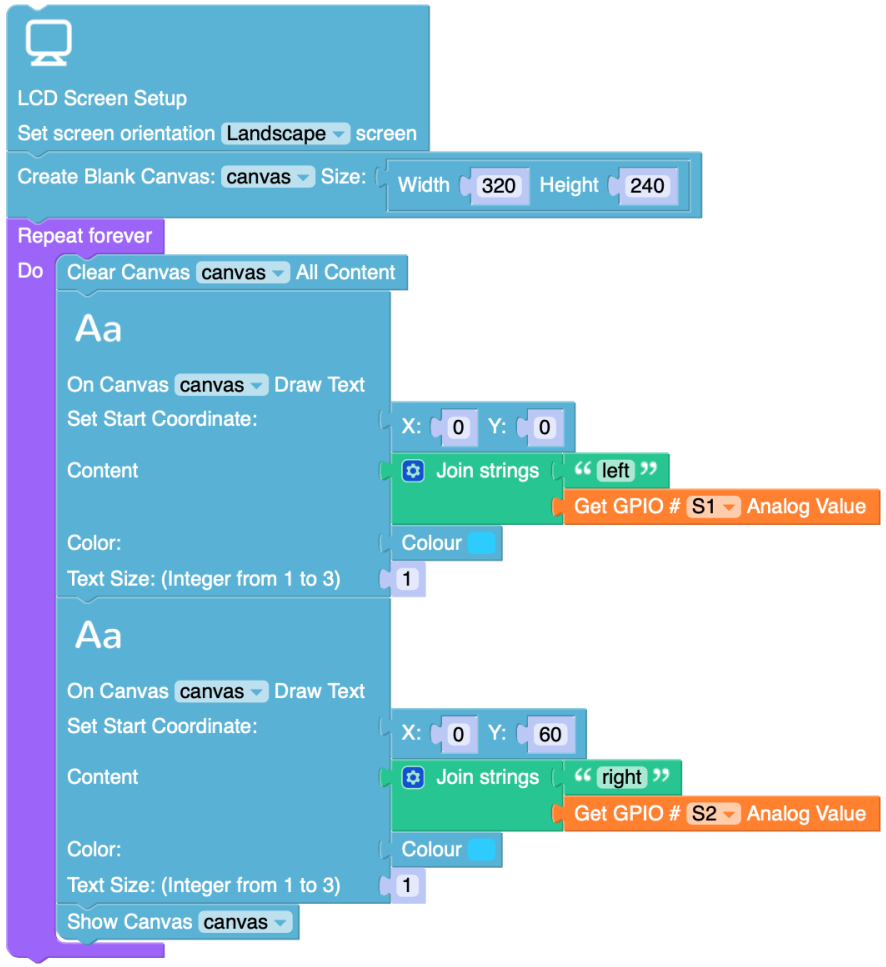
Task1 :

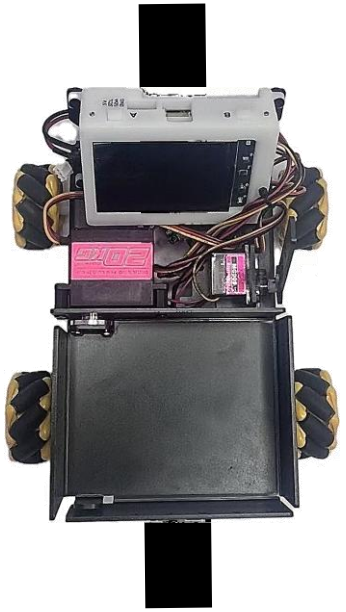
Place the car on the black-and-white paper, and observe the output on the screen when the grayscale sensor identifies the black and white areas.



Grayscale Sensor

Reference Program





Place the car on the black-and-white paper for testing, and observe the pattern of value changes on the screen.

COCOROBO

CHAPTER TWO

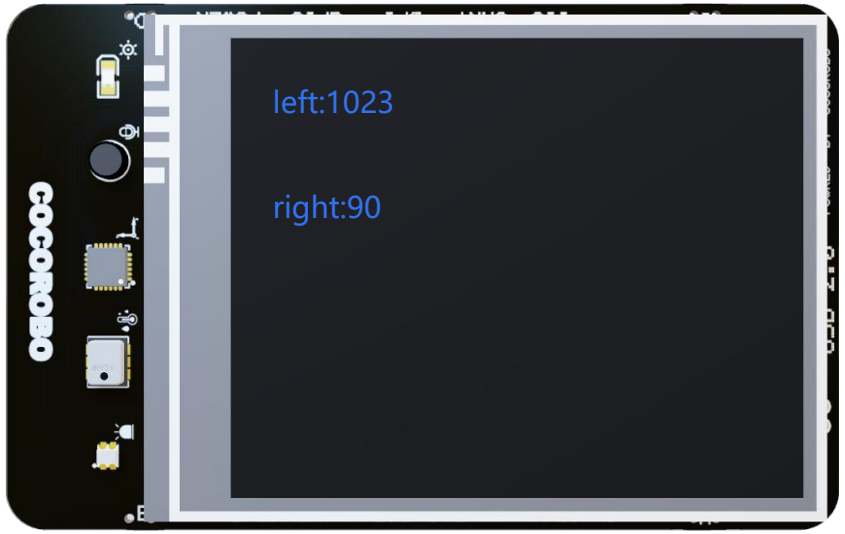
TWO.

Automatic Line Following

CHAPTER TWO

Line Following Principle

Place the car as shown in the diagram and observe the data displayed on the screen.

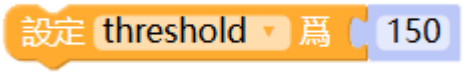


Note: The analog signal value falls within a specific range, not a fixed value. The data value can vary depending on the environment and power supply.

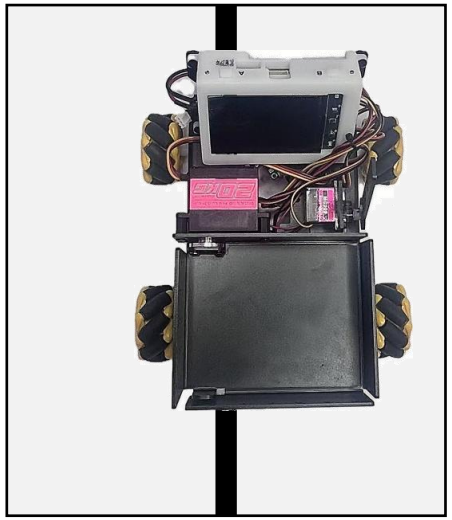
When white is detected, the output analog signal value is 1023, indicating that white is recognized, and the output analog signal value is higher.

When black is detected, the output analog signal value is 90, indicating that black is recognized, and the output analog signal value is lower.

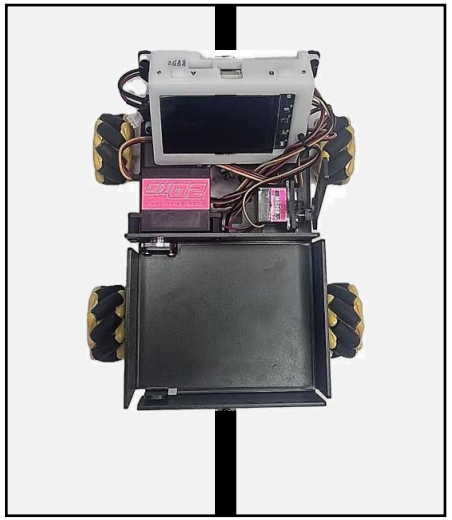
Line Following Principle



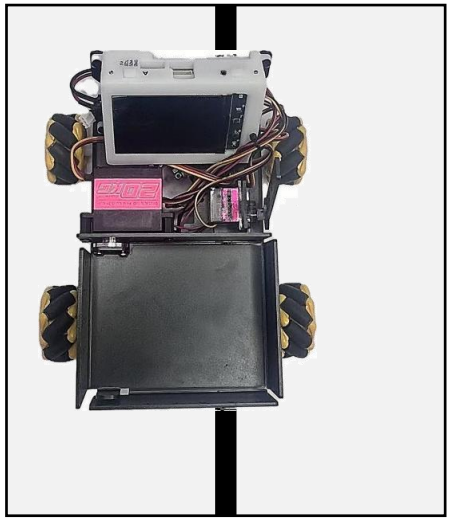
The reference threshold can be adjusted based on actual testing conditions.



Left side pressure line



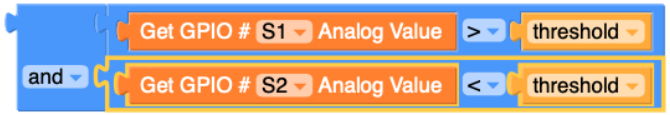
There are no pressure lines on both sides



Right-hand side

Turn left

Go straight

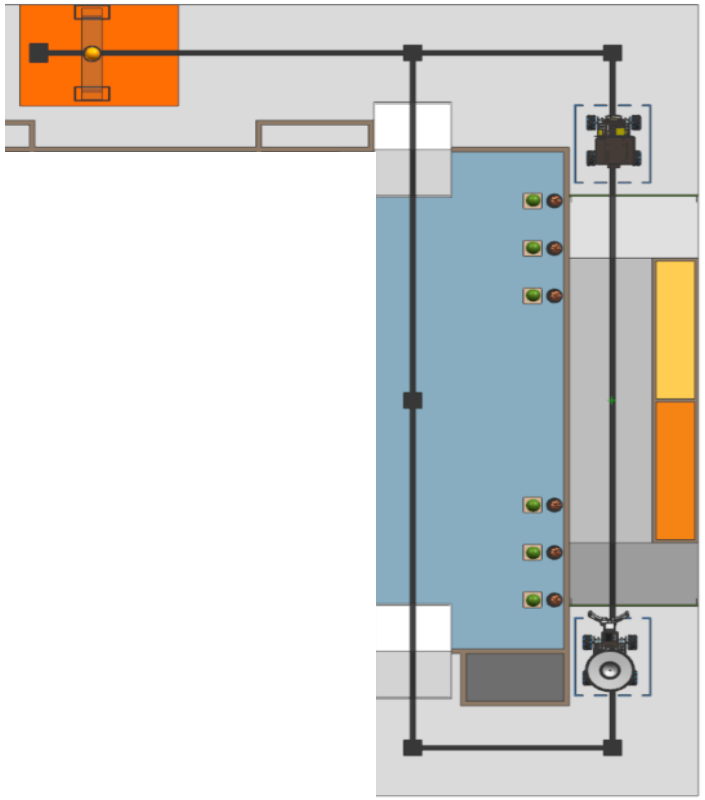


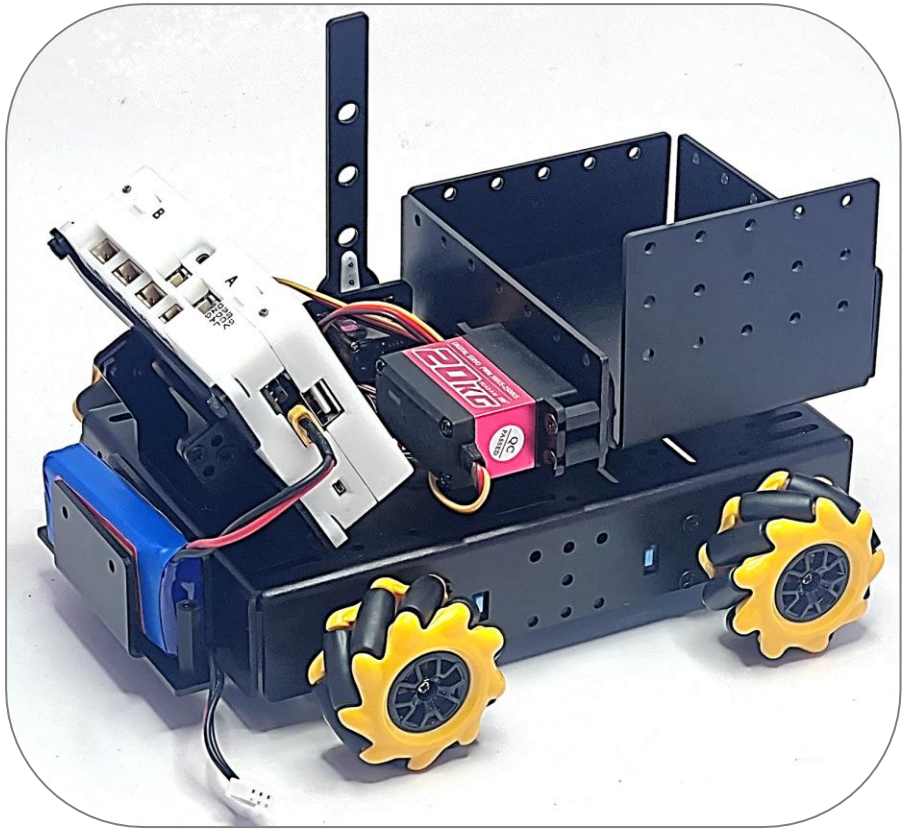
Turn right

Line Following Operation

Task 2:

Based on the conclusion from Task 1, write a program that allows the car to automatically follow the line.

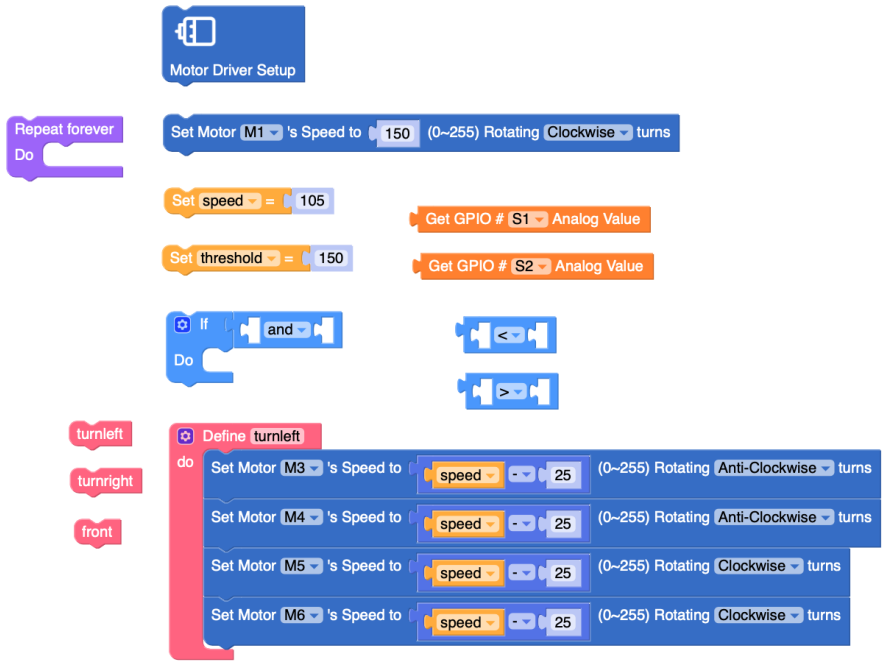


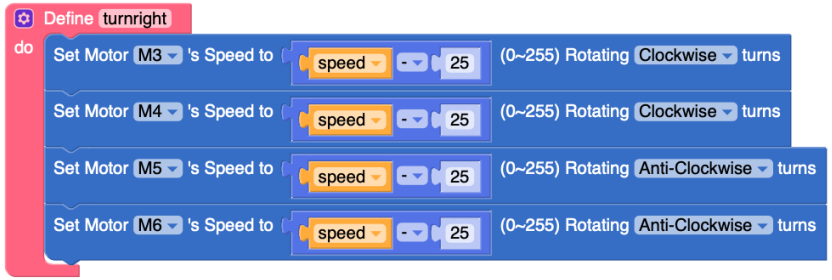


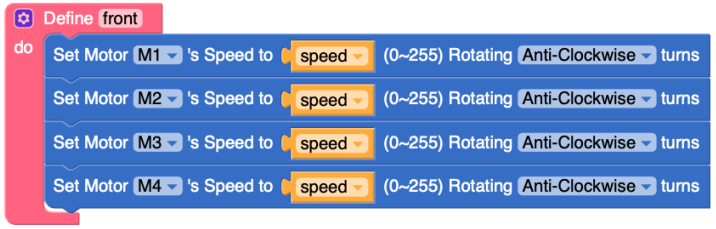
Line Following Operation

Task 2: Based on the conclusion from Task 1, write a program that allows the car to automatically follow the line.

Using Blocks



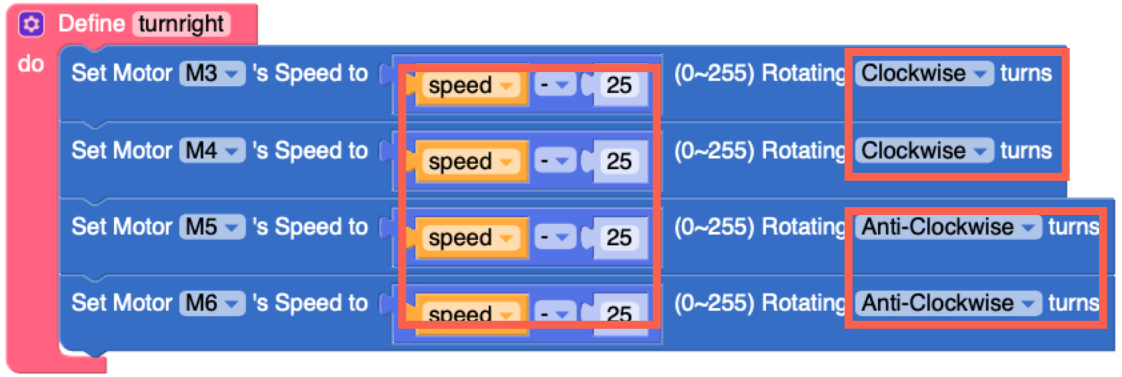




Line Following Operation

Task 2: Based on the conclusion from Task 1, write a program that allows the car to automatically follow the line.

Block Explanation



*Due to differences in factory settings, even if the same speed value is given, different motors may have different rotation speeds. It is necessary to perform actual tests to determine the optimal speed for your car.

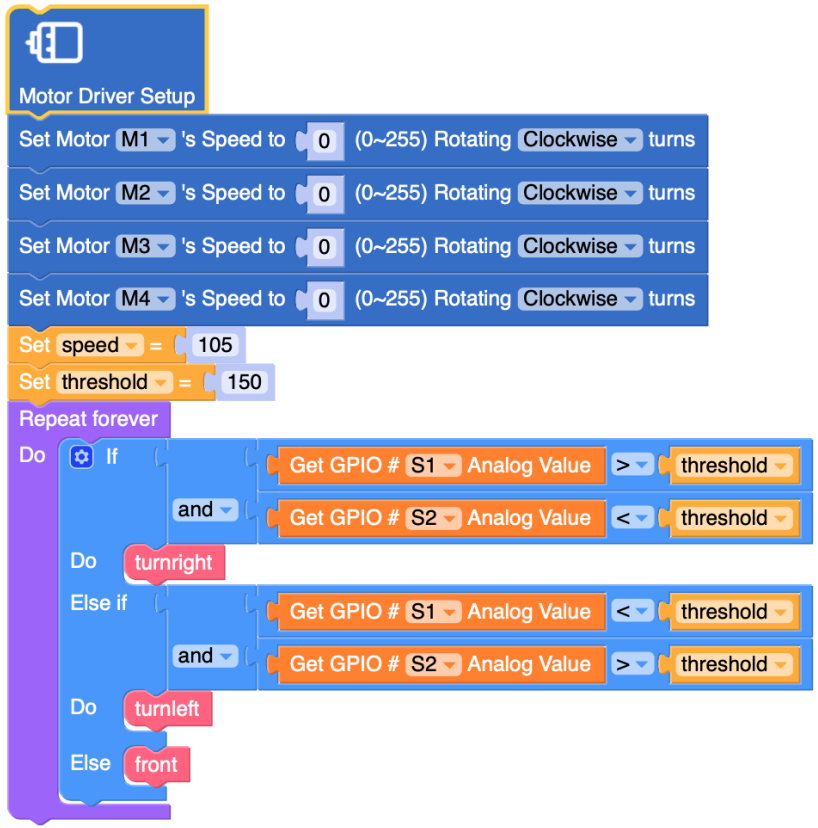
Line Following Operation

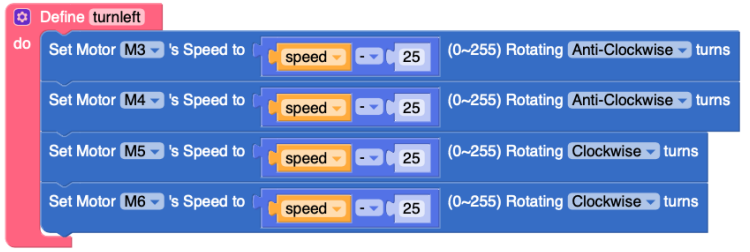
Task 2:

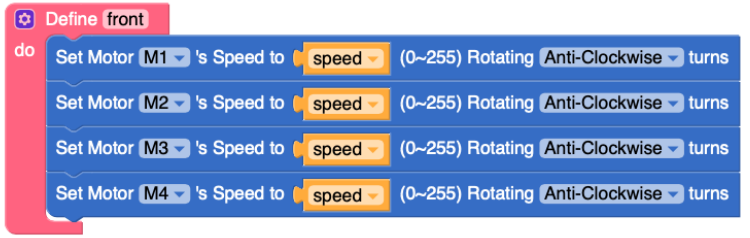
(Test) Based on the conclusion from Task 1, write a program that allows the car to automatically follow the line.

Reference Program

Go ahead and test your program on the map to see if the car can follow the line automatically.



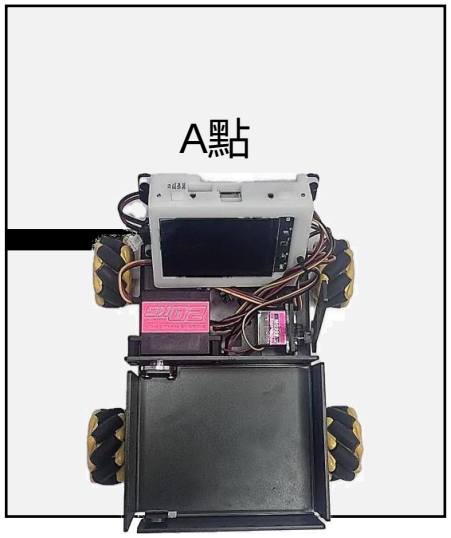




Line Following Operation

Task 2:

Based on the conclusion from Task 1, write a program that allows the car to automatically follow the line.



Identifying Issues

The car can move straight, but it cannot turn.

This happens because when the car reaches point A, the output of the grayscale sensor is:

According to the previous program design, the car will go straight at this point. Therefore, the design approach here should be: when both the left and right ends of the grayscale sensor detect black, the car should continue straight for a short distance (about 100-300 milliseconds) before making a left turn.

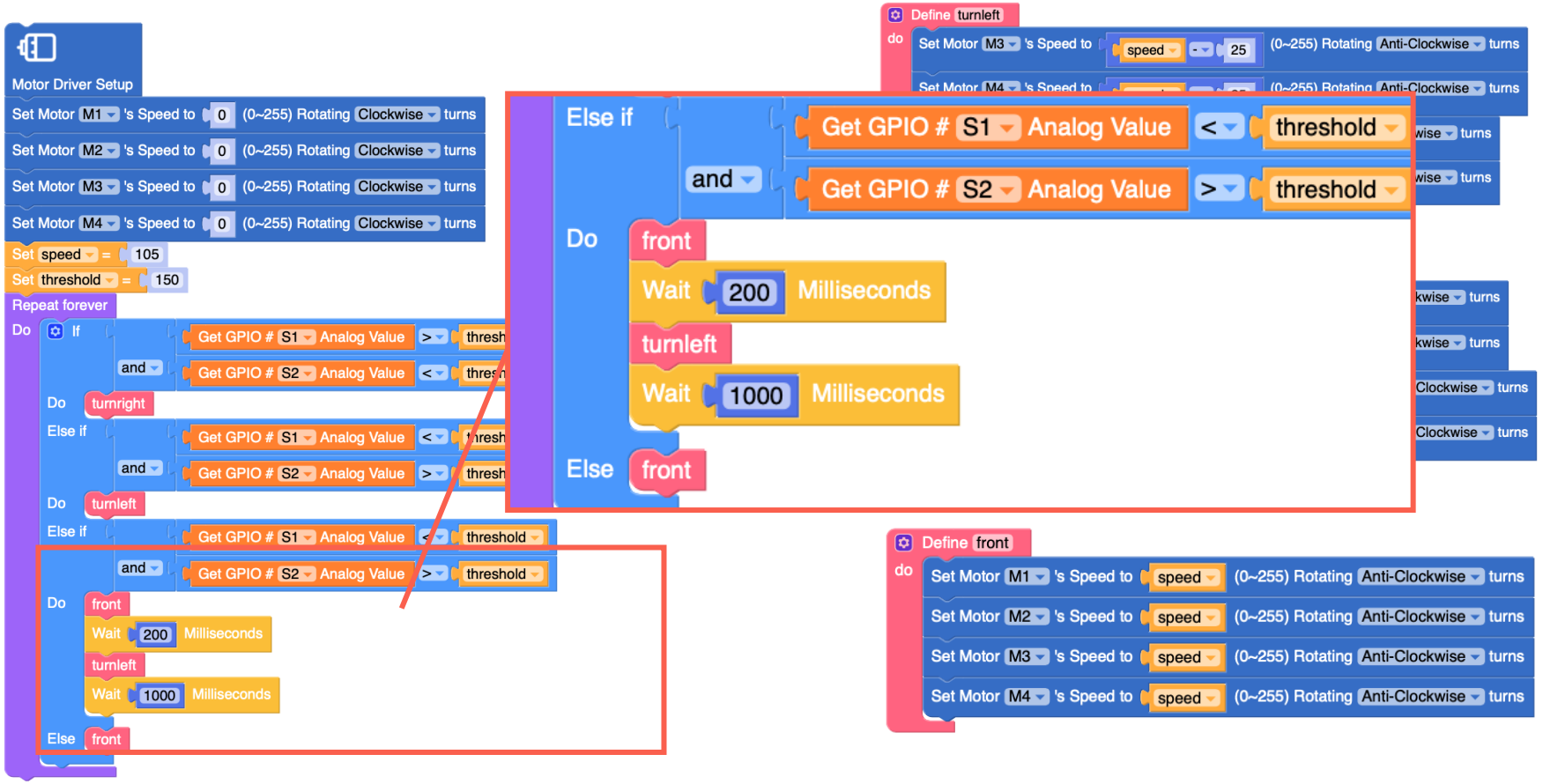
Line Following Operation

Task 2:

Based on the conclusion from Task 1, write a program that allows the car to automatically follow the line.

Go ahead and test your program on the map to see if the car can automatically follow the line.

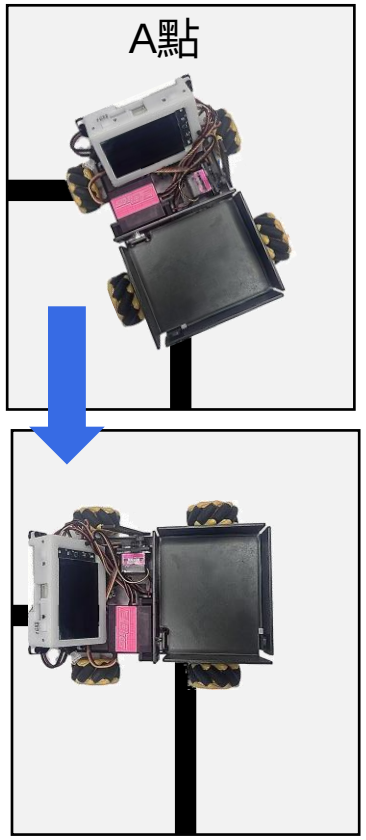
Reference Program



Line Following Operation

Task 2:

Based on the conclusion from Task 1, write a program that allows the car to automatically follow the line.

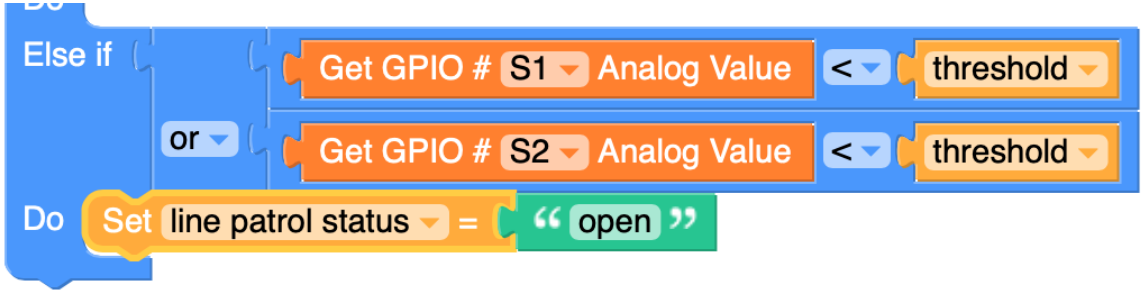


Identifying Issues

The car can now turn, but how can we precisely control when it stops turning?

- In the original program, the car continues following the line while turning (since factors such as lighting or the distance between the grayscale sensor and the ground can affect the car's control).
- Therefore, when the car has already executed the turn command, we need to disable the line following mode.

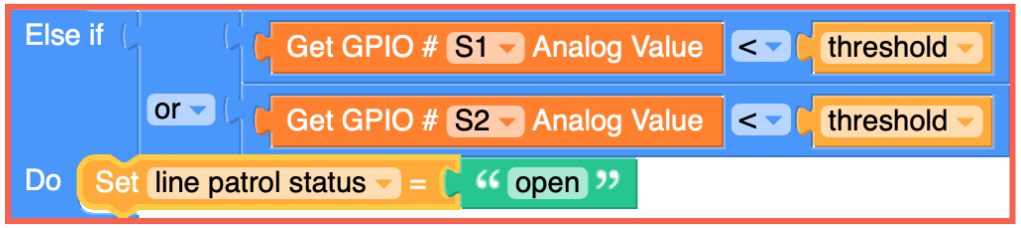
  Once the car completes the turn and the grayscale sensor detects the black line on either the left or right side, we can re-enable the line following mode.



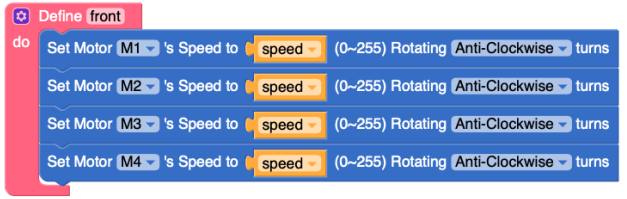
Line Following Operation

Task 2:

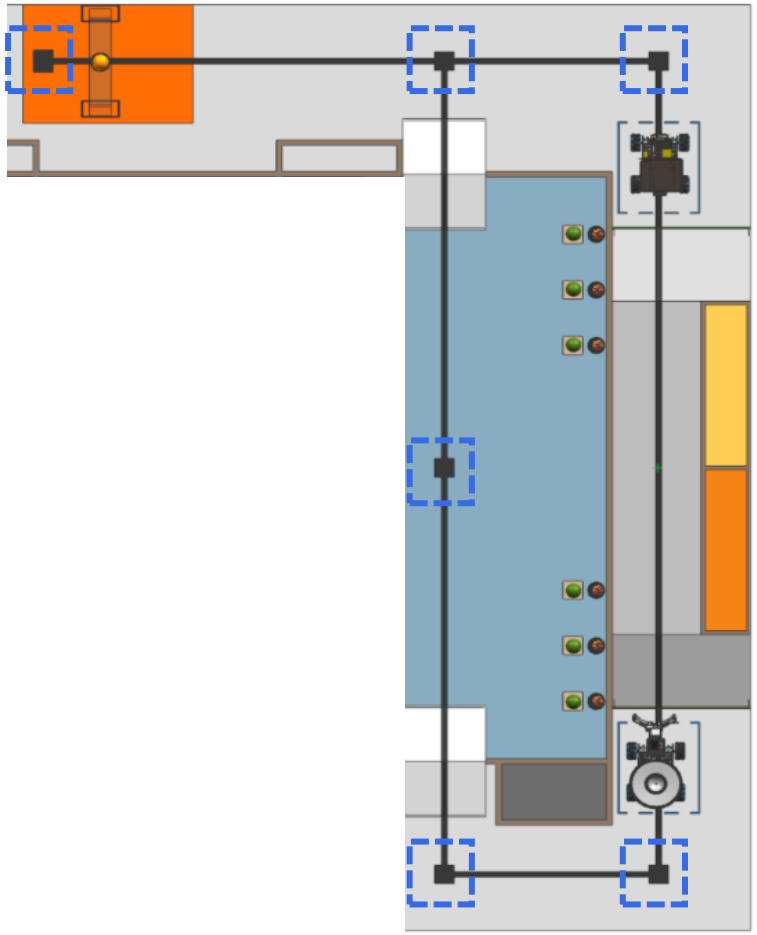
(Optimization) Based on the conclusion from Task 1, write a program that allows the car to automatically follow the line.



*When both the left and right sensors detect black, it means the car has reached the turning point. At this point, disable the line-following mode. When one side detects the black line again, re-enable the line-following mode.

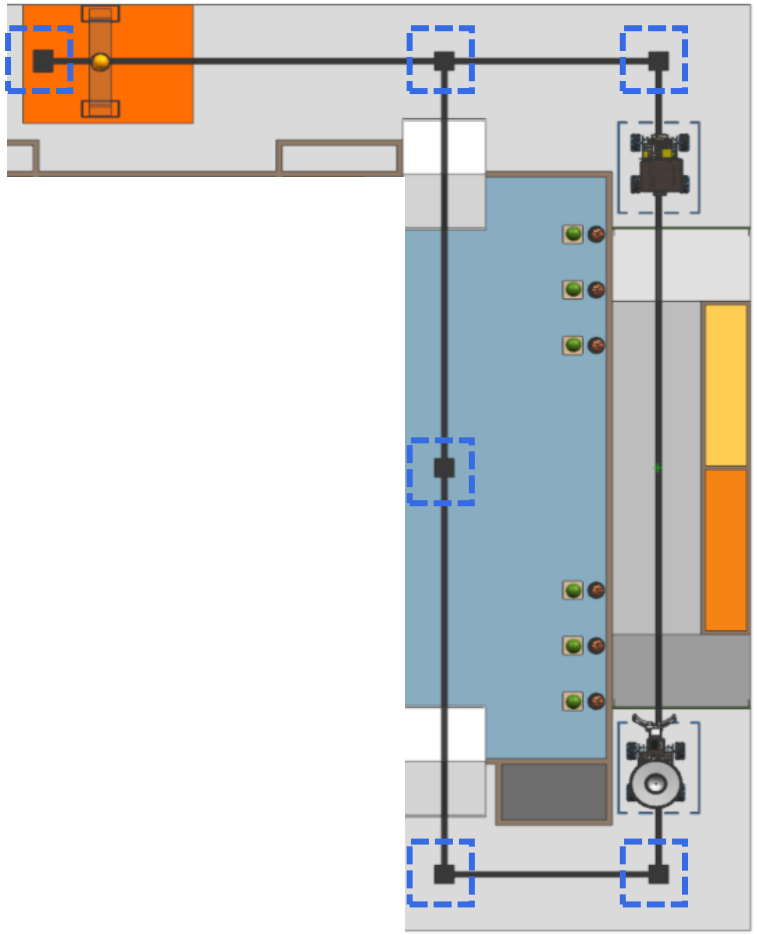


Line Following Operation



How should the car complete the corresponding task when it reaches these positions? How does it know where it is?

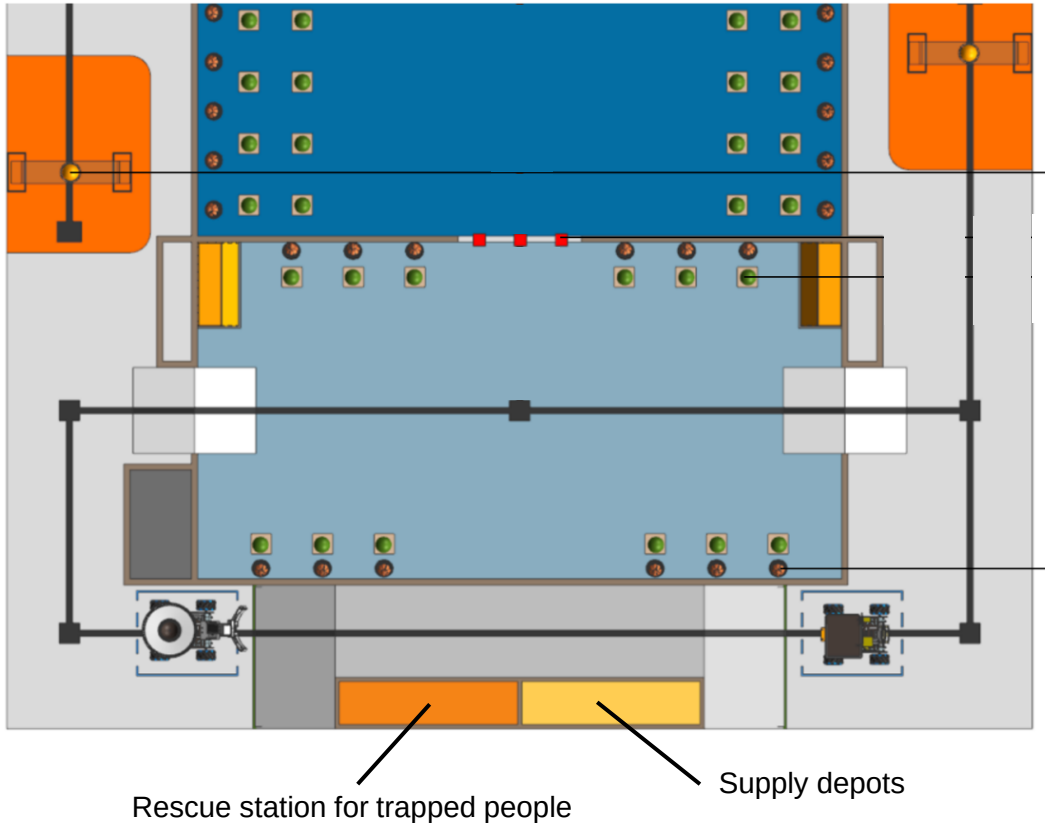
Line Following Operation



Carefully observe: when the car reaches these positions, both the left and right infrared sensors at the bottom will detect the black colour.

Competition Content

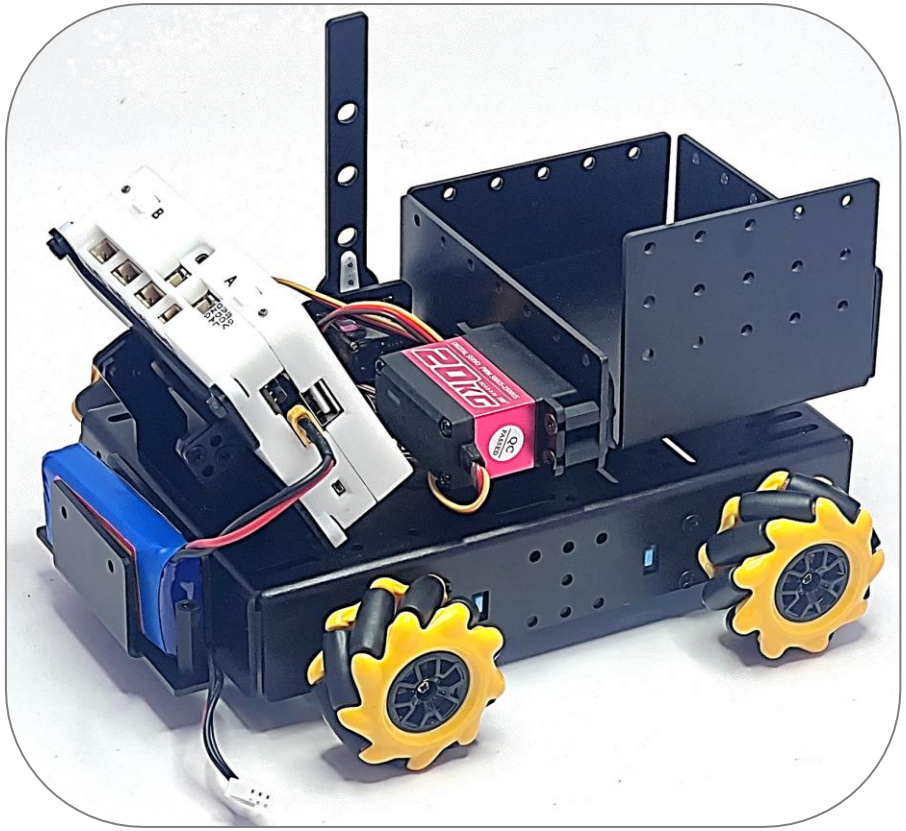
Transport Car



The transport car must start from the starting area.

- It will follow the line to the supply station; upon reaching the supply station, it will activate a mechanism to collect supplies (supplies will be replenished indefinitely, and small balls will drop into the car's rear frame, with a maximum of 3 supplies per collection). Once the supplies are collected, the car will transport them to its own supply station.
- After the mechanical vehicle drains the flooded area, it will rescue trapped individuals and place them on the transport car. The transport car must then transport the trapped individuals to the rescue station. (The mechanical vehicle can control the transport car's operation by providing the transport car with a specific tag for identification.)
- Once all the floodwater in the own A-level flooded area is drained, the mechanical vehicle will move the obstacle blocks to the AI transport car, which will then transport the obstacle blocks to the obstacle clearing area. (This task is only for the middle school group to perform.)

Transport Car Path Analysis



1

Materials

Collecting supplies, transporting supplies, unloading supplies

2

Rescue

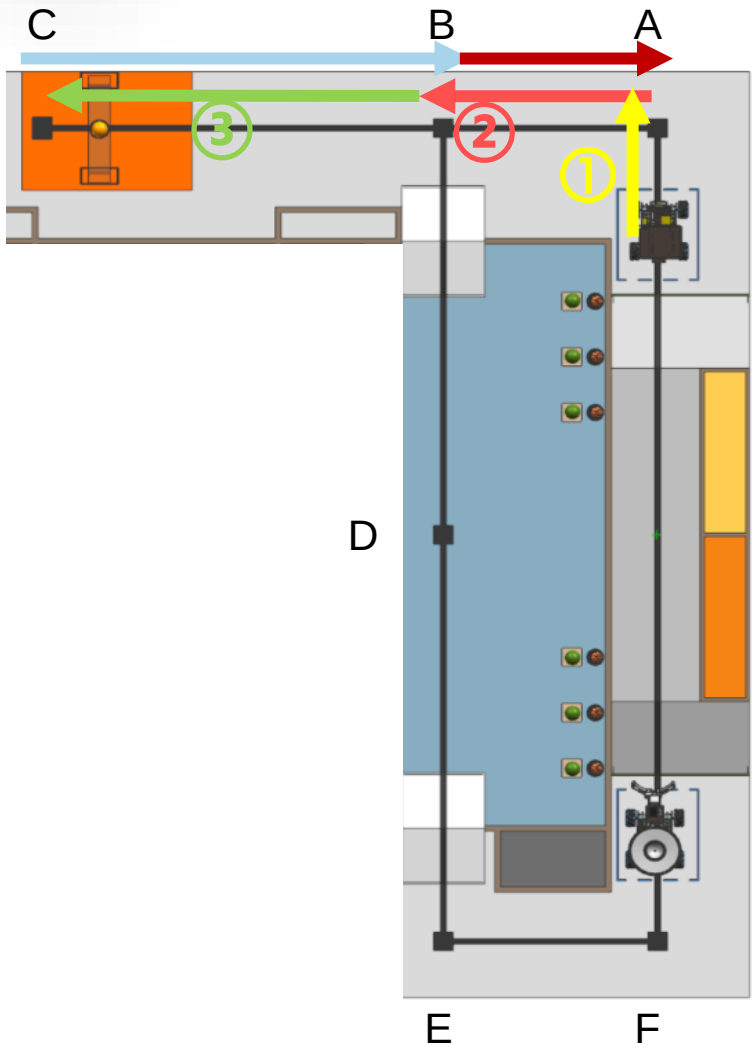
Transporting trapped individuals to the rescue station

3

Obstacles

The mechanical vehicle will move the obstacle blocks to the AI transport car, which will then transport them to the obstacle clearing area.

Line Following Driving



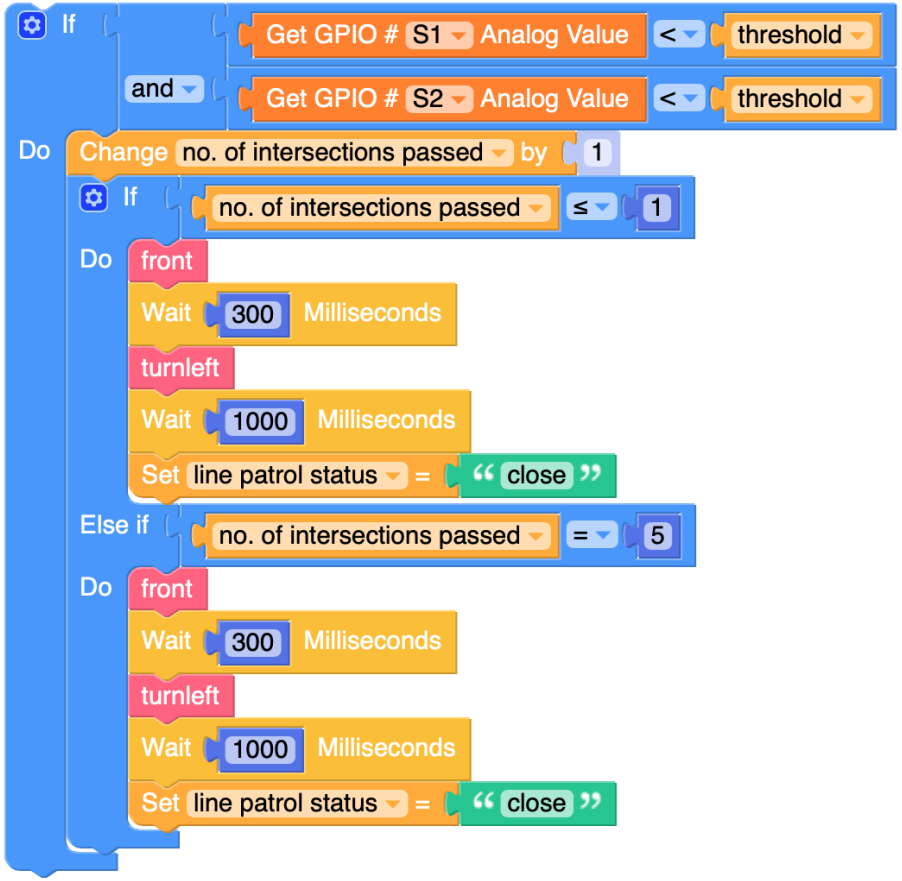
Task 3: Make the car follow the "Supply" path automatically.

Create a path state called "Supply" to indicate that the car has completed the supply task.

- The car will start at the starting point and pass through points A, B, and C, then return.
- We will use the "Number of intersections passed" to record the points the car has passed.
- Each time the car passes a point, we can make the "Number of intersections passed" increase by 1 in the program design.

① Start → A: Left turn
② A → B: Go straight
③ B → C: Stop, wait for loading, turn around
④ C → B: Go straight
⑤ B → A: Right turn

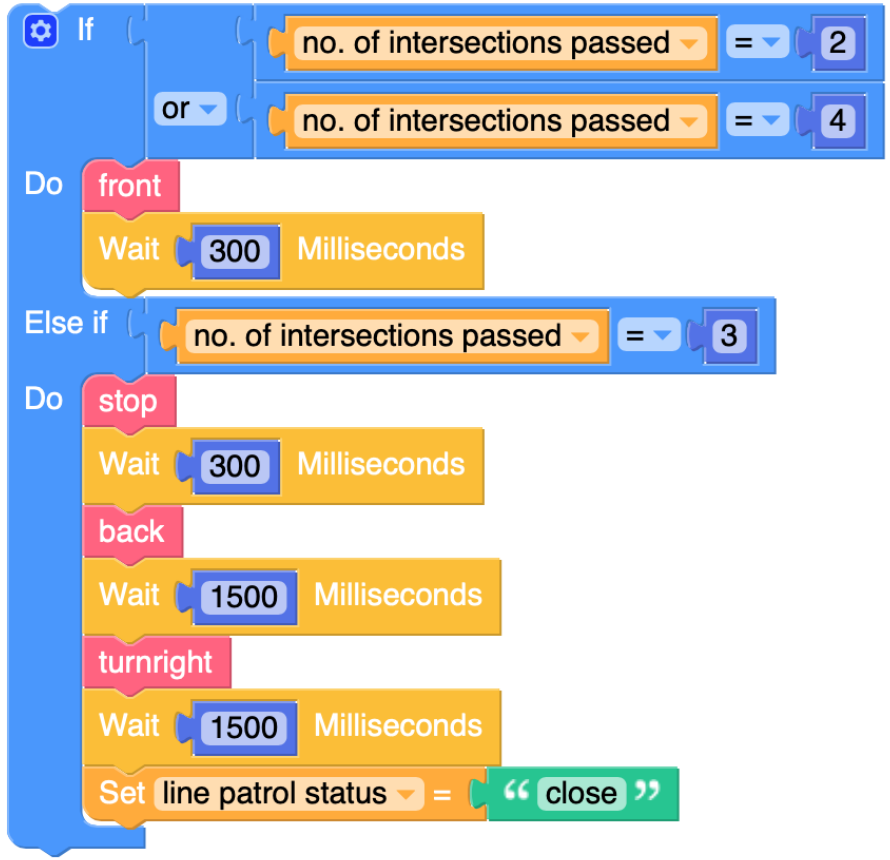
Line Following Driving



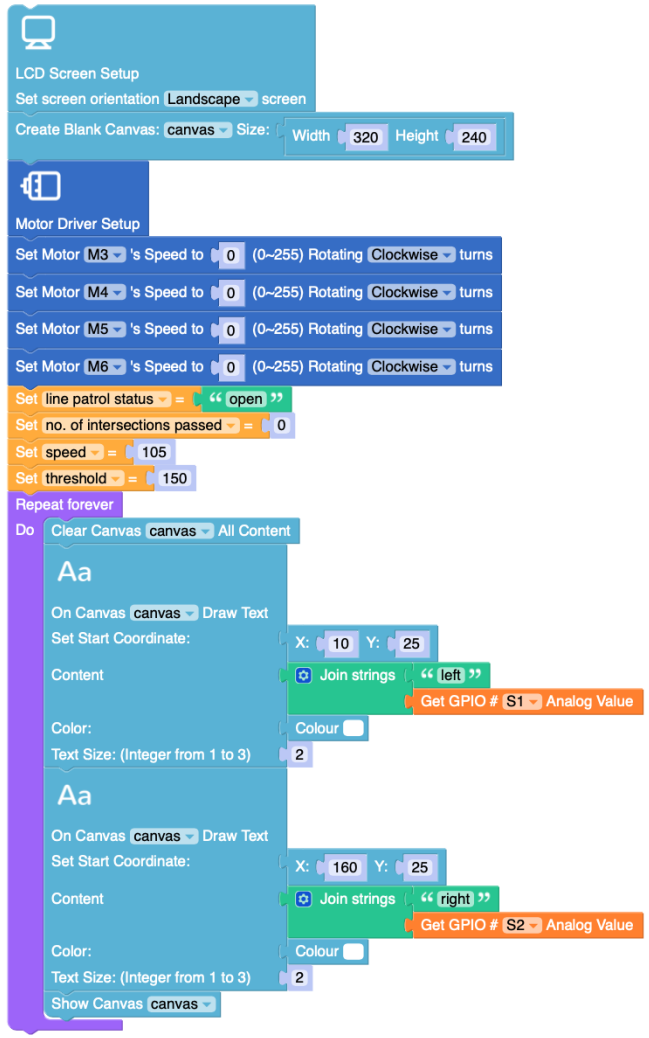
1

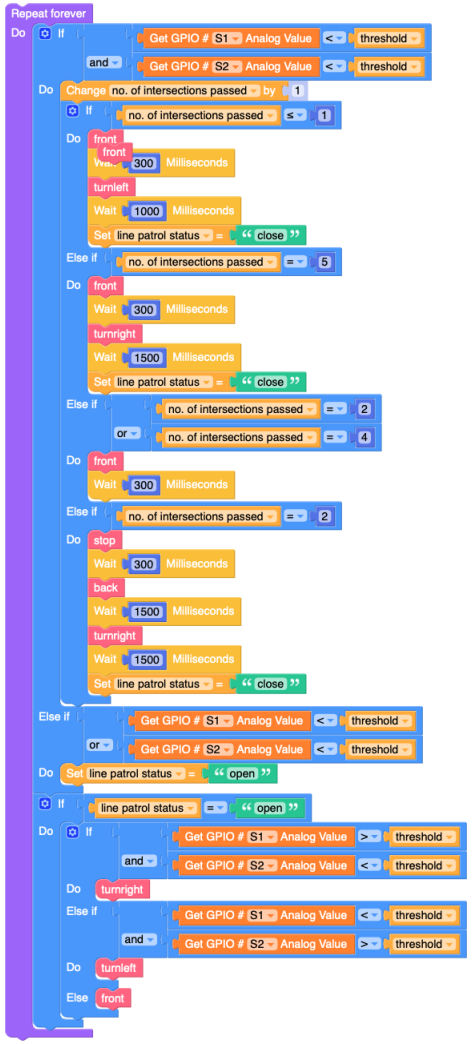
Materials

Collecting supplies, transporting supplies, unloading supplies



Line Following Driving

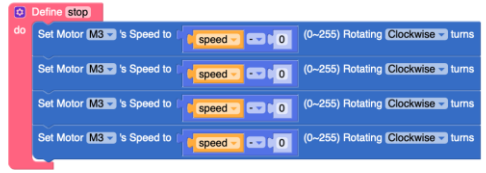




1

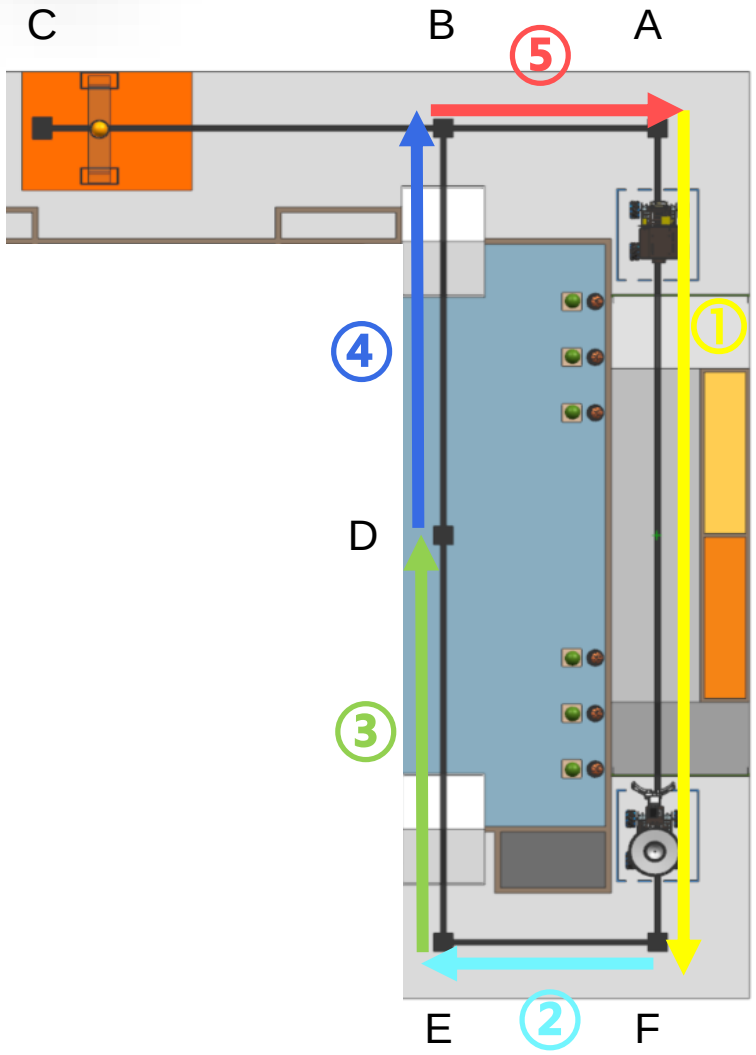
Materials

Collecting supplies, transporting supplies, unloading supplies



Line Following Driving

Task 4: Make the car follow the "Rescue" path automatically.



Create a path state called "Rescue" to indicate that the car has completed the rescue task.

- The car will start at the starting point and pass through points F, E, D, B, and A.
- We will use the "Number of intersections passed" to record the points the car has passed. Each time the car passes a point, we can make the "Number of intersections passed" increase by 1 in the program design.

① Starting point → F: Right turn
② From F → E: Right turn
③ Arrive at D: Stop and wait
④ From D → B: Right turn
⑤ From B → A: Right turn

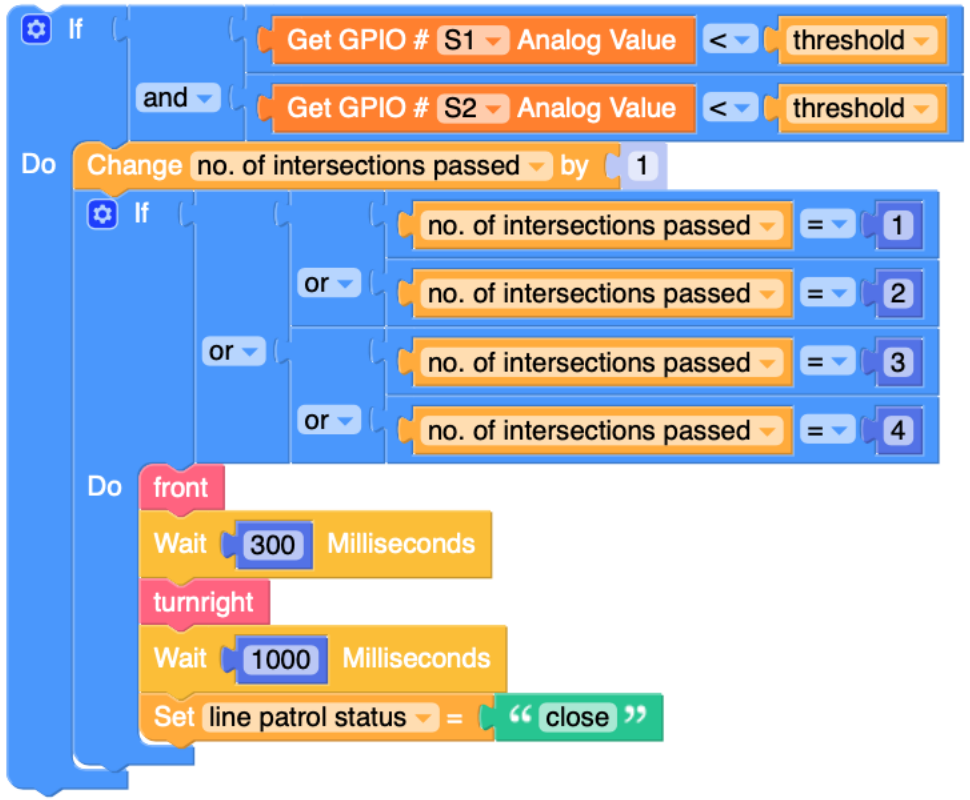
Line Following Driving

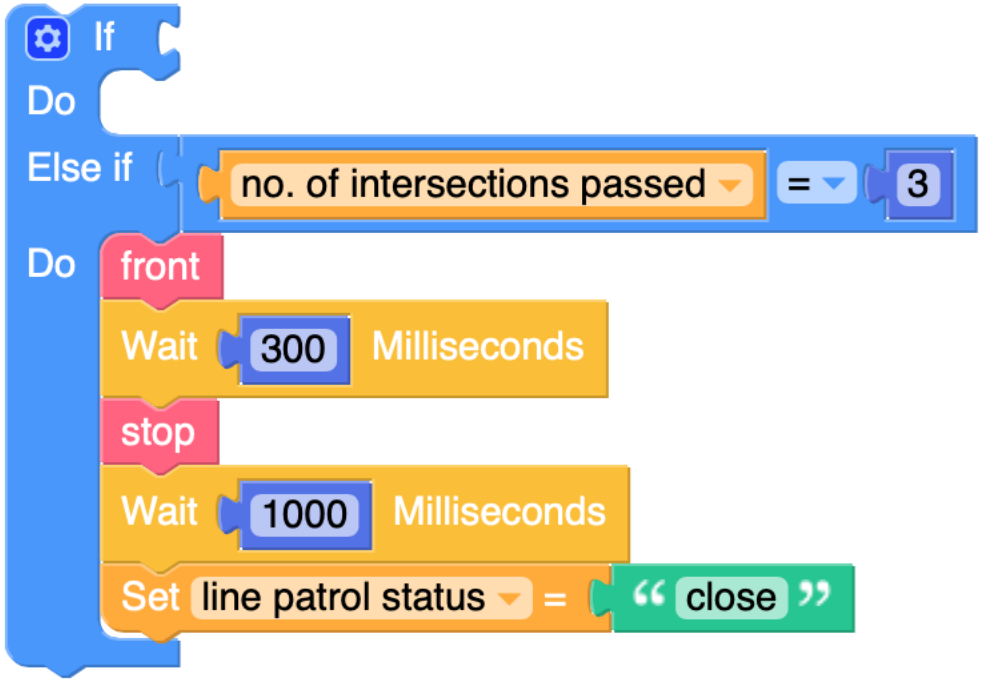
Task 4: Make the car follow the "Rescue" path automatically.

2

Rescue

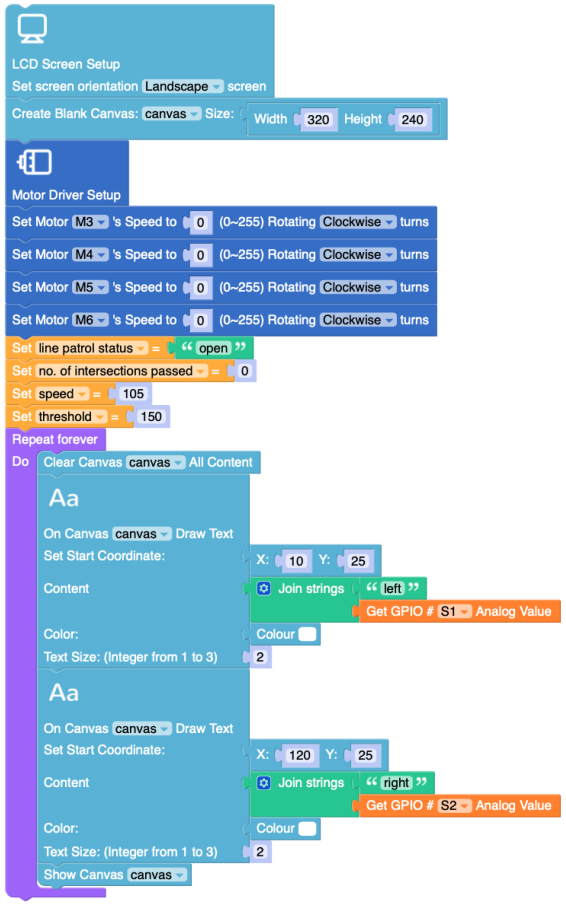
Transporting trapped individuals to the rescue station

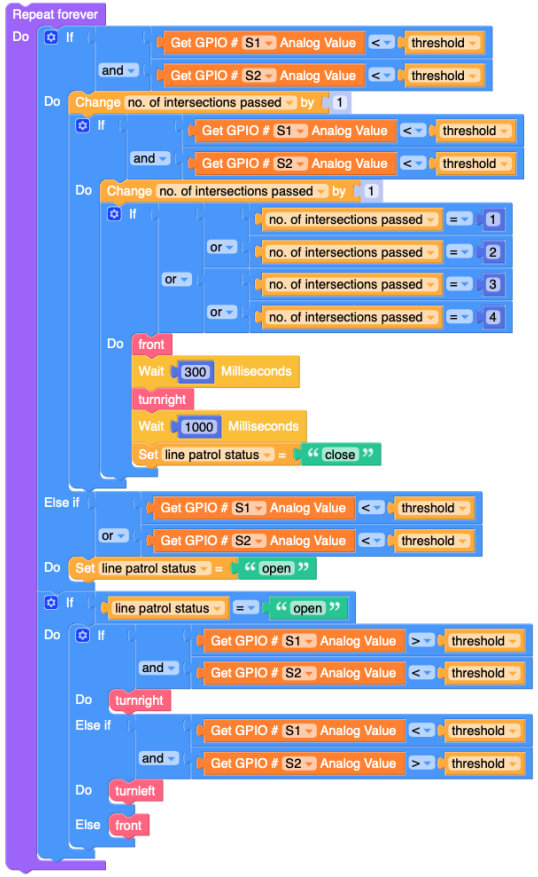




Line Following Driving

Task 4: Make the car follow the "Rescue" path automatically.





2

Rescue

Transporting trapped individuals to the rescue station

See you in the next class!

THANK YOU

COCOROBO