

COSC-3070

Programming Autonomous Robot

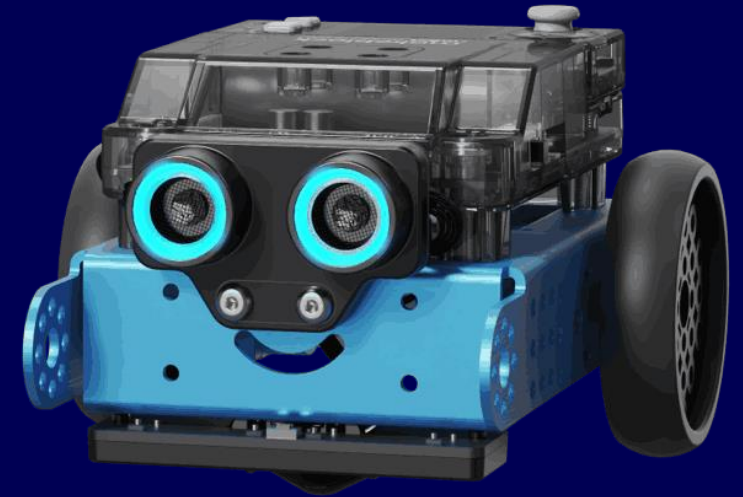
Autonomous Path Following and Obstacle Avoidance With mBot Neo

Lecturer: Ginel Dorleon

Huynh Ngoc Tai (s3978680)

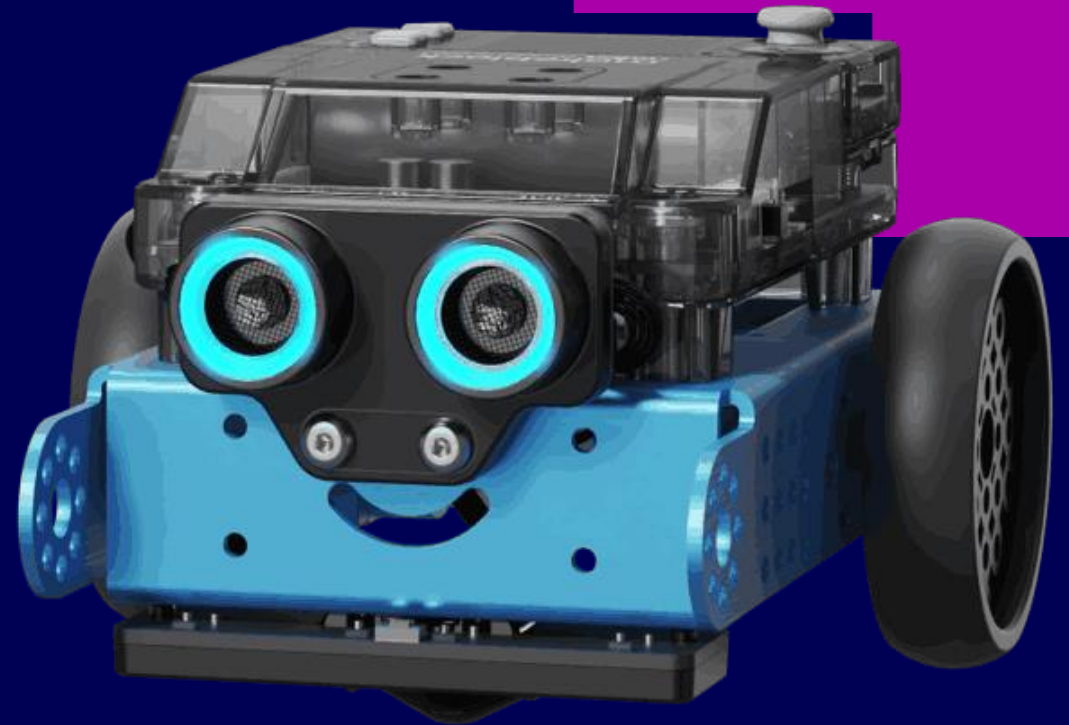
Tran Quang Minh (s3988776)

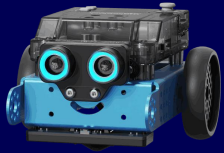
Chau Tung Nguyen (s3976069)



"I declare that in submitting all work for this assessment, I have read, understood and agree to the content and expectations of the assessment declaration".

Problem Overview

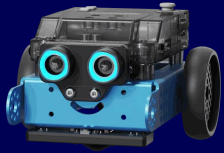




Problem Overview

Autonomous robots are revolutionizing industries like logistics and transportation. They rely on AI, computer vision, and sensors to interpret paths, detect signs, and avoid obstacles.





Problem Overview

For the final project of the course, we are given 2 things, which is :

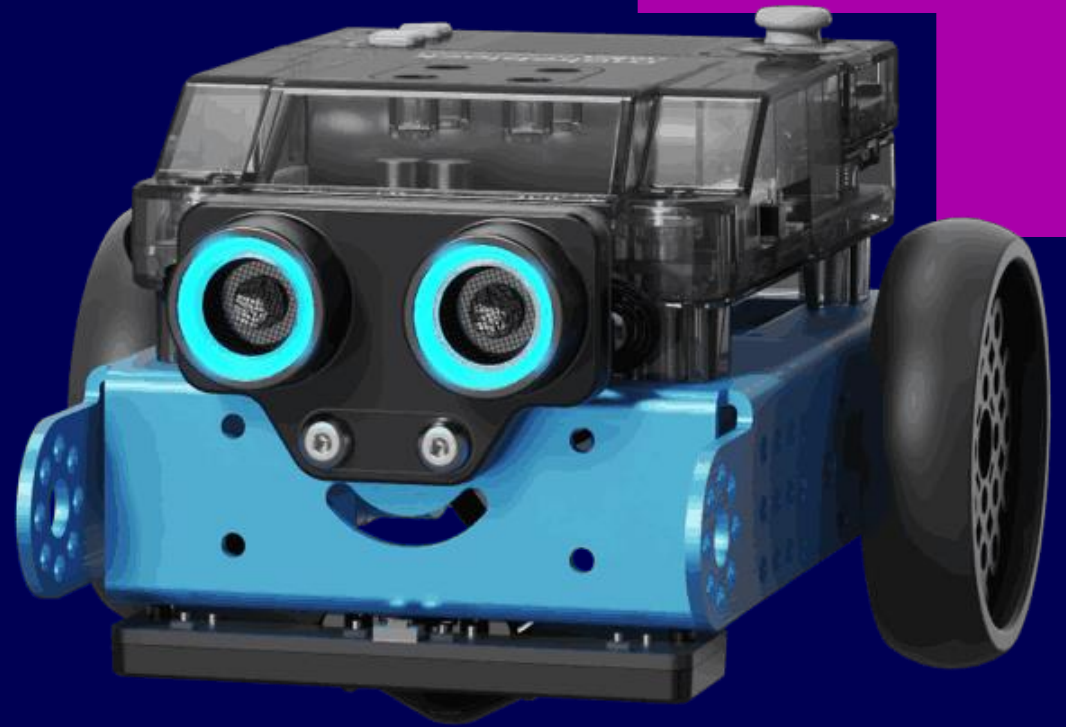
- Cyber Pi is the microcontroller – the brain of the robot.
- mBot2 Neo is the complete robot – includes wheels, motors, sensors.

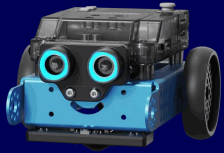


CODE



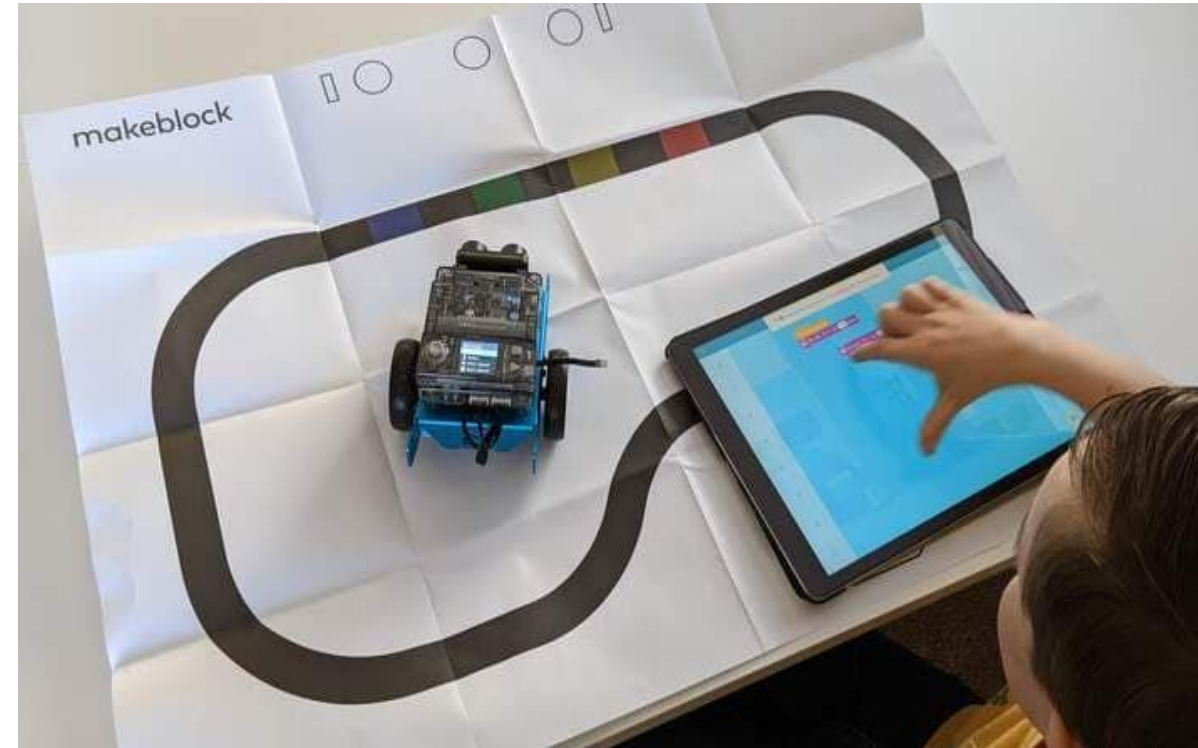
Project Setup

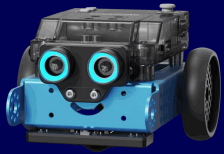




Problem Overview

The aim of the project is to simulate real-world autonomous robot using simple sensors and vision systems, which then the robot will need to successfully navigate a maze-like environment following and act accordingly to the signs it will sensor, or the obstacles along the way like an AI powered robot in the real world.

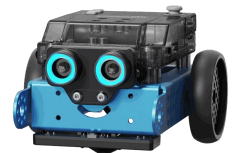




Problem Goal

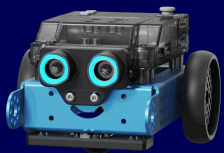
For the demo, the robot must navigate a complex maze using only onboard logic, no remote control or intervention.

It follows traffic rules: stop on red, go on green, slow on yellow, and even handle combinations like yellow-to-green transitions. It must also avoid static obstacles automatically and reroute in T-junctions when needed.



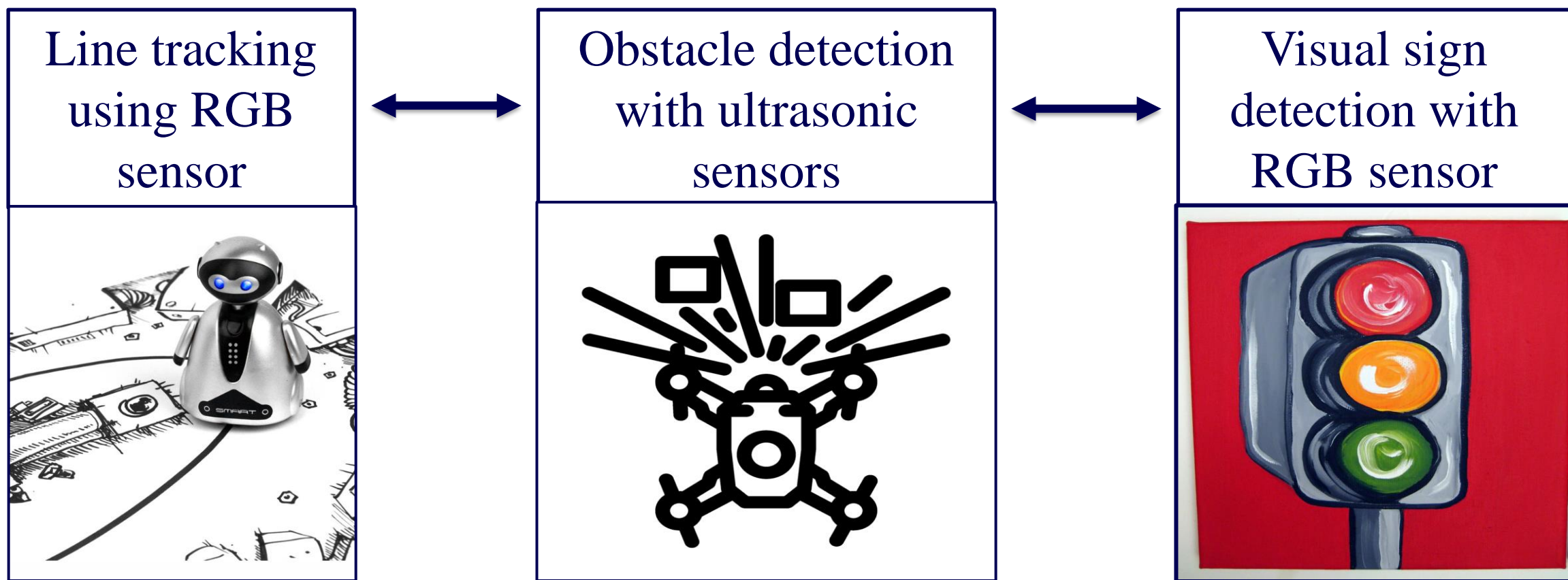
Methodology Implementation

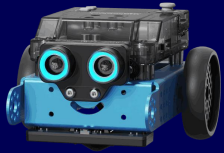




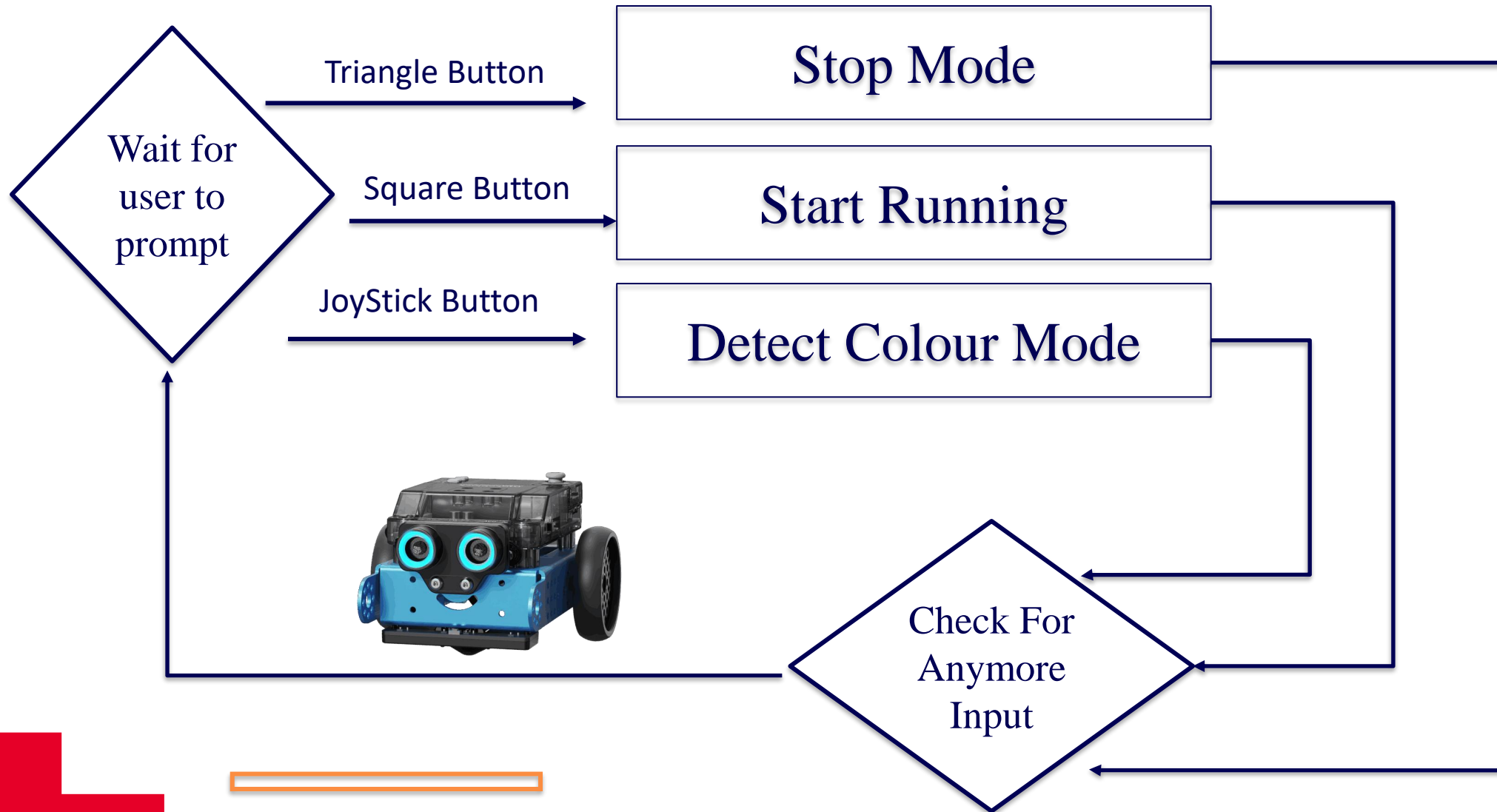
Methodology

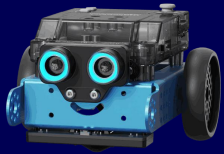
Our method integrates three modules—path tracking, visual detection, and obstacle avoidance—into a unified loop.



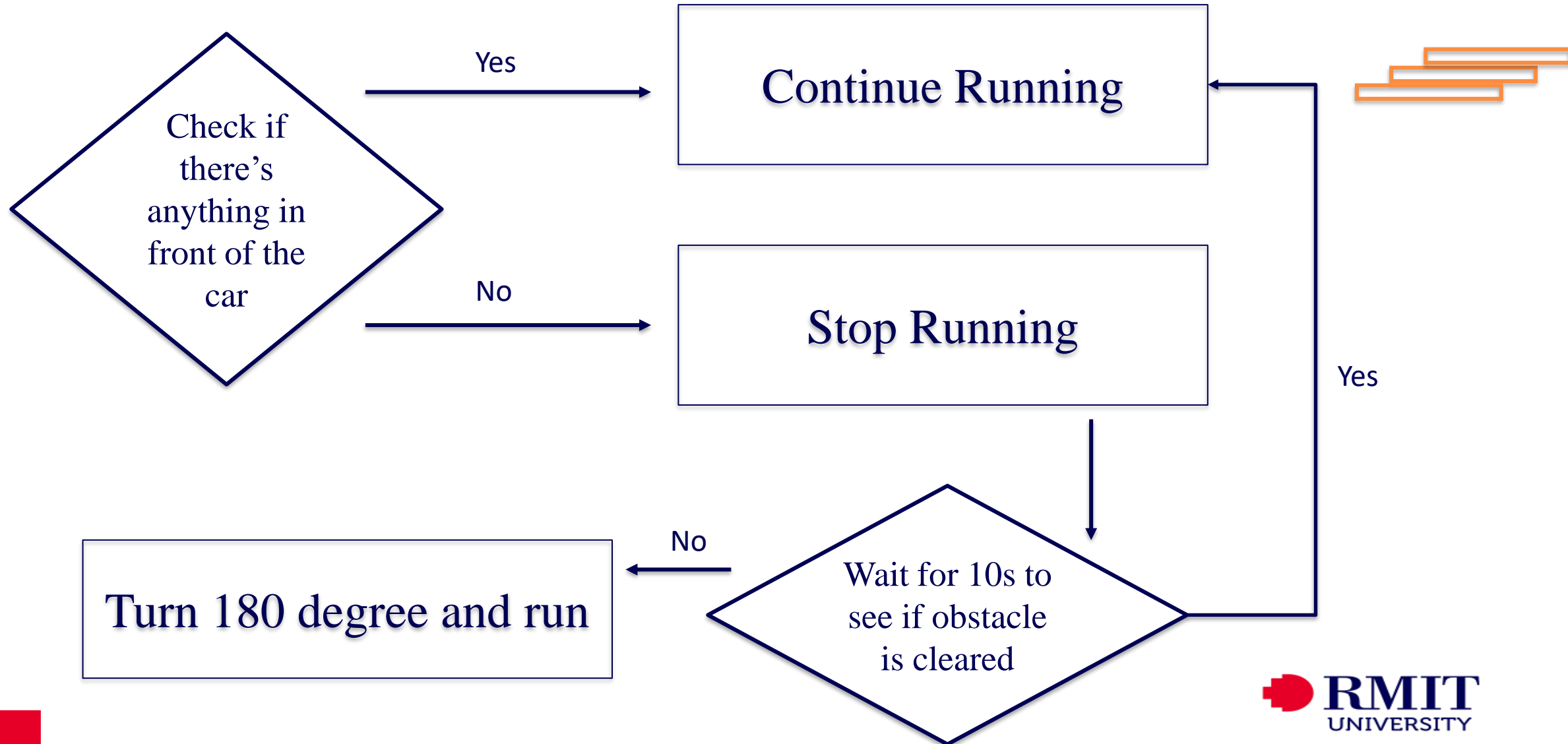


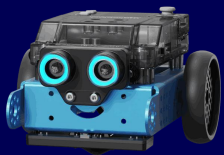
System Initialization and Event Handling





Obstacle detection with ultrasonic sensors





Line tracking using RGB sensor

base_power: the forward speed baseline



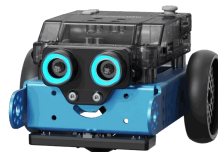
kp: proportional gain to control steering strength



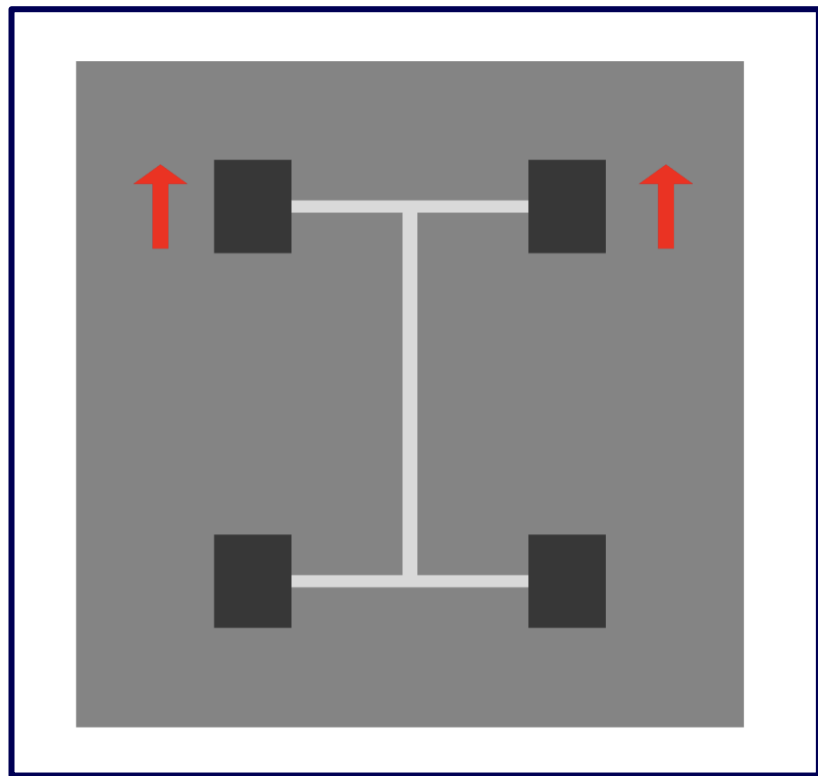
offset: deviation from center line (negative = left, positive = right)

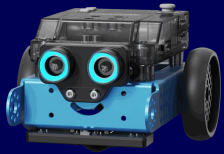


Motor speed formulas:

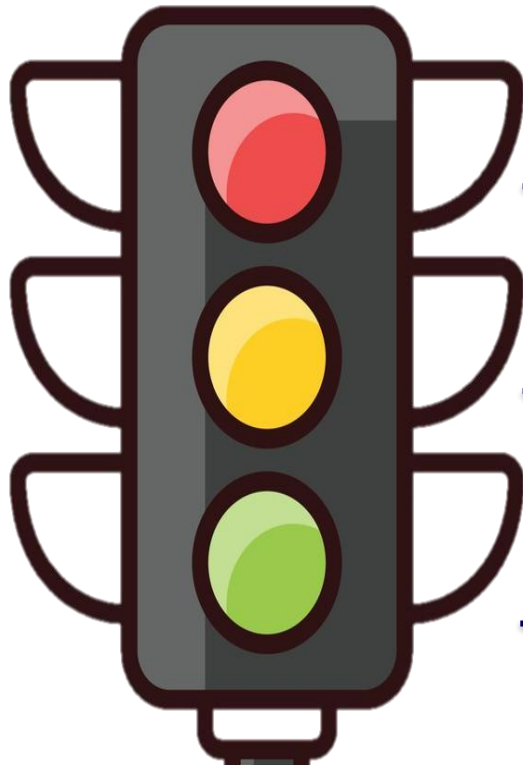


- $\text{right_power} = \text{base_power} - \text{kp} \times \text{offset}$
- $\text{left_power} = -1 \times (\text{base_power} + \text{kp} \times \text{offset})$





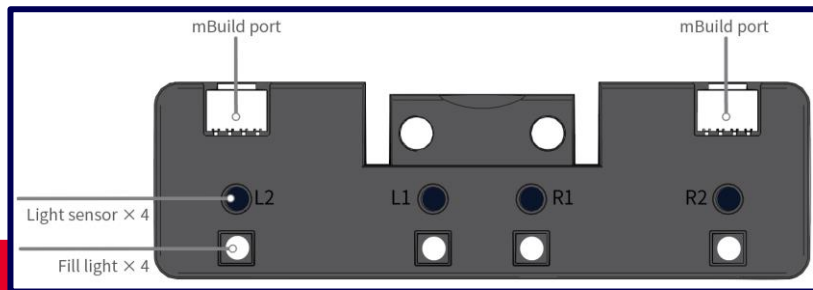
Visual sign detection with RGB sensor



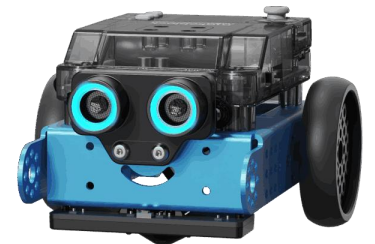
Stop the car, will wait for 10 seconds
then turn back

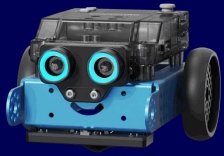
Slow the car

Will tell the car to run after being stopped
by red light

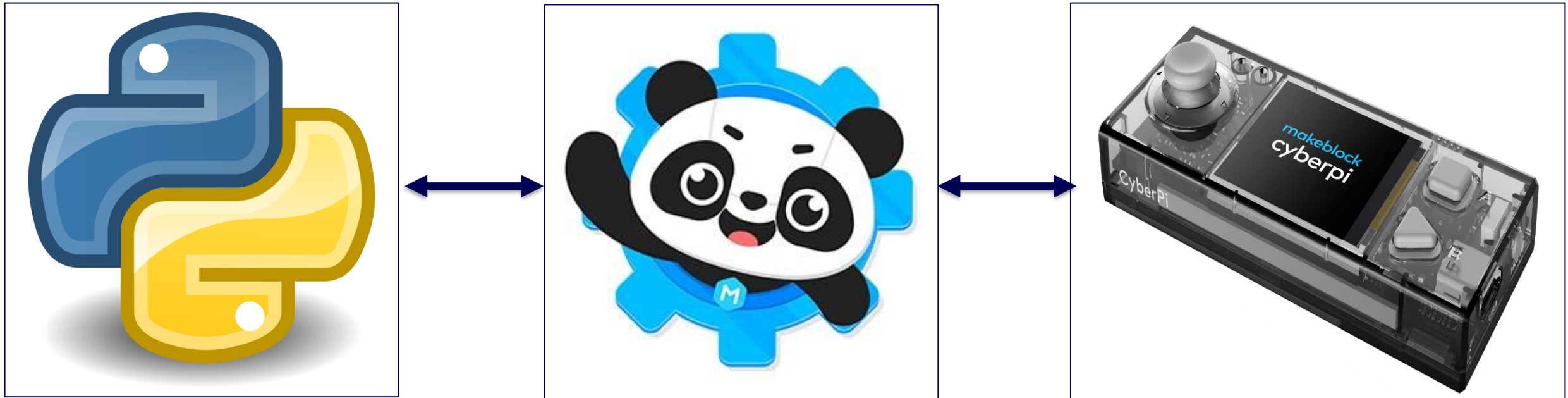


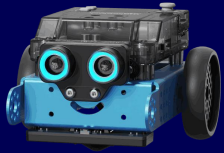
Quad RGB sensor will check the
colour of the sign under it to
perform action accordingly





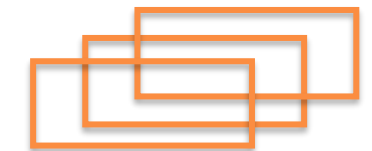
Implementation

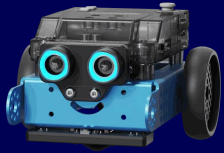




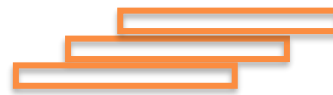
In-Detailed Values for Implementation

Parameter	Value	Purpose
base_power	30% PWM	Normal speed (~0.25 m/s)
base_power slow	25% PWM	Caution zone speed (~0.20 m/s)
k_p	0.3	Proportional gain for smooth steering
Obstacle threshold	15 cm	Safe stopping distance
Timeout duration	10 seconds	Before rerouting
Turn angle	190°	Prevents re-entering same blocked path





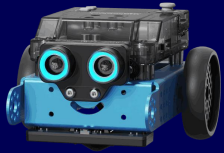
Creative Addon



Color Detected	Action	LED Feedback	Sound
Red	Stop completely, wait for green	Red	261 Hz tone
Green	Resume motion at 30% PWM	Green	659 Hz tone
Yellow	Slow to 25% PWM for 5 seconds	Yellow	294 Hz tone
White	Continue line tracking at base speed	White	No tone

Testing

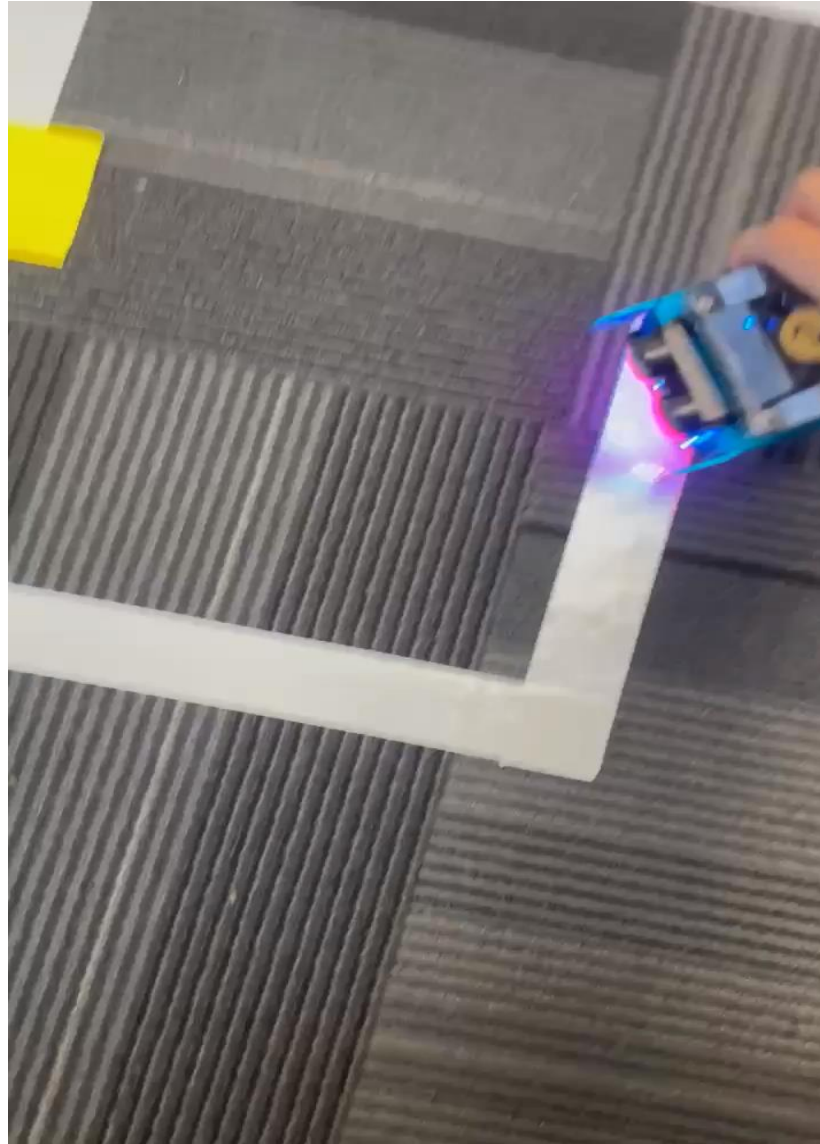
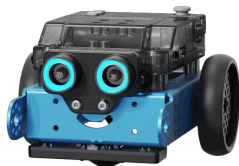




Testing

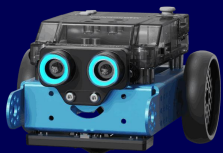
As per the requirement from the assignment, we decided to create a maze which satisfy most of the objectives the robot need to be able to do:

- Avoid obstacle
- Perform sign actions
- Navigate when it meets T turn
-



Result, Evaluation and Conclusion





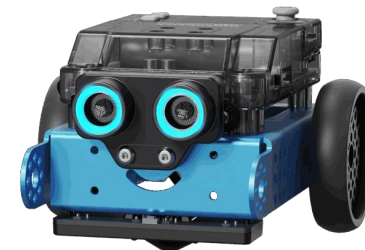
Result, Evaluation and Conclusion

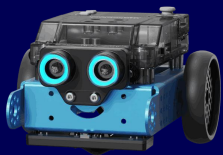
Successfully completed tasks with decent-performing models.

The robot responded correctly to all sign-based instructions

Obstacle avoidance worked reliably under varied layouts.

Minor challenges like lighting and alignment drift were resolved through recalibration and logic tuning.



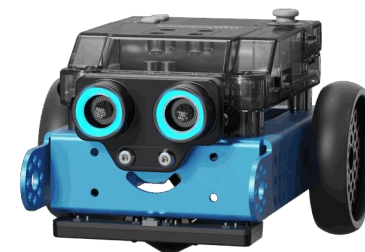


Challenges

While yellow slows the robot and green resumes it, the logic for maintaining speed for the first GREEN transition then speeding up at the second GREEN was not explicitly coded as we only have 5 seconds for the slowdown time.

If an obstacle partially blocks the path, the robot does not drift laterally to rejoin; it instead waits and turns

The robot does not yet evaluate both directions at a T intersection to select an available path, it will perform a single pivot if blocked

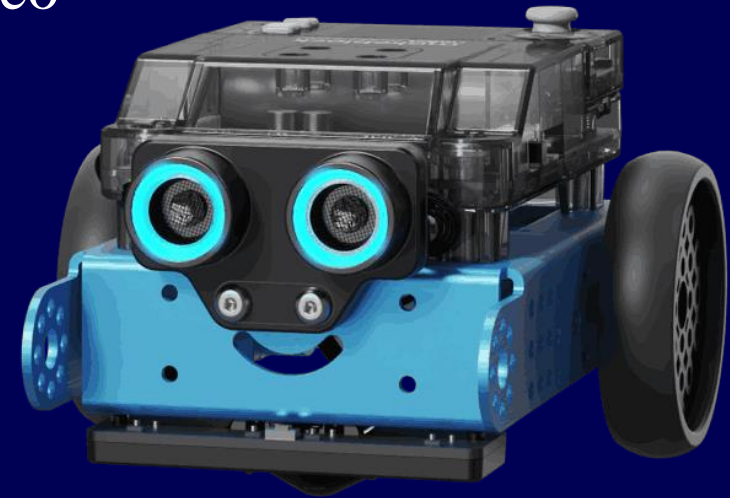


COSC-3070

Programming Autonomous Robot

Autonomous Path Following and Obstacle Avoidance With mBot Neo

**THANK YOU FOR LISTENING TO OUR
PRESENTATION AND WISH YOU A NICE DAY**



"I declare that in submitting all work for this assessment, I have read, understood and agree to the content and expectations of the assessment declaration".

