- **Introduction**

Our robot is developed for the WRO Future Engineers – Self-Driving Cars challenge. The vehicle uses a Raspberry Pi as the main controller and is programmed in Python. It is designed to drive autonomously, follow the track, detect and avoid obstacles, interpret the track layout randomized before each round, and perform the required parking manoeuvre during the Obstacle Challenge. The robot's hardware and software follow WRO regulations for vehicle size, drive/steering configuration, and
documentation

- **Electronic Components Used in the robot**
- **Hardware**

1. Motor Driver L298N

2. Motor DC

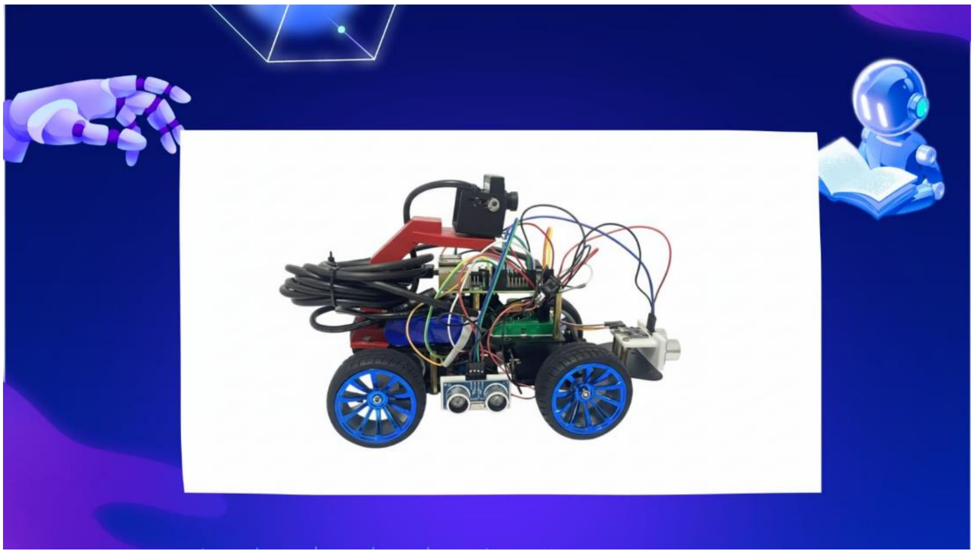
3.Servo Motor

4. Gyroscope MPU6050

5. Ultrasonic

6. Volatage Regulator XL4015

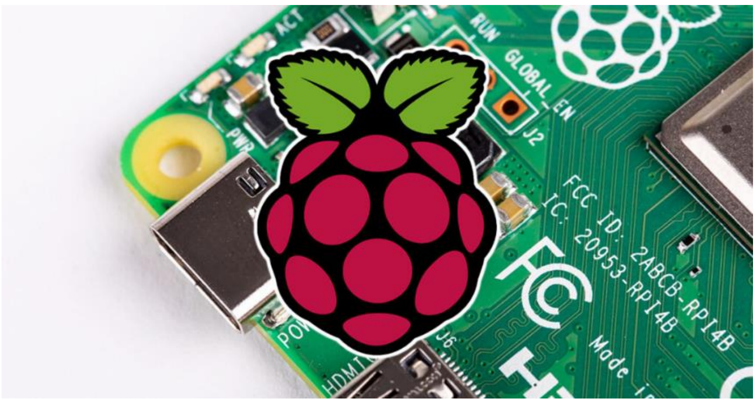
7. Raspberry Pi



- **Software**

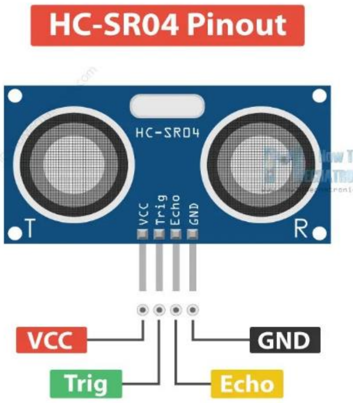
We chose Python as the main programming language for our WRO robot because it is simple, powerful, and widely used in robotics and computer vision. Python allows us to write clear and organized code, making it easier to test, debug, and improve our program One of the main reasons we selected Python is its strong support for the OpenCV library, which we used to process images and recognize objects or patterns. OpenCV gave us the ability to detect lines, colors, and QR codes — features that were essential for our robot's mission Python also works perfectly with Raspberry Pi, which we used as the main controller. Its flexibility allowed us to easily connect sensors and motors, and integrate the results from OpenCV to make real-time decisions.

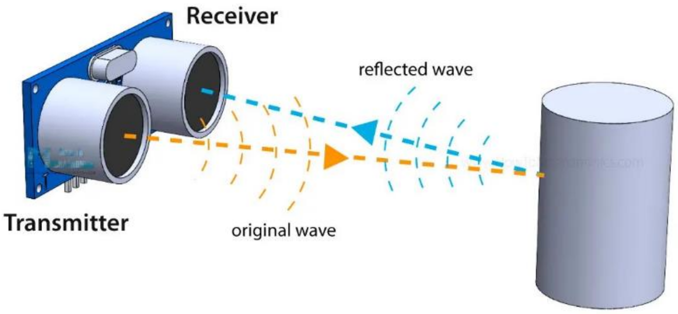
Using Python helped us focus on problem-solving and teamwork, since the language is easy to understand and modify. Overall, Python and OpenCV together provided a powerful combination that made our robot more intelligent and efficient during the WRO challenge.



Ultrasonic Sensor Working Principle

The ultrasonic sensor measures distance by emitting high-frequency sound waves that are beyond human hearing, typically above 20 kHz, and detecting the echo that bounces back from an object. The sensor consists of a transmitter that sends ultrasonic waves and a receiver that listens for the reflected echo. When the transmitter emits a short burst of sound waves, they travel through the air, hit an object, and return to the sensor as an echo

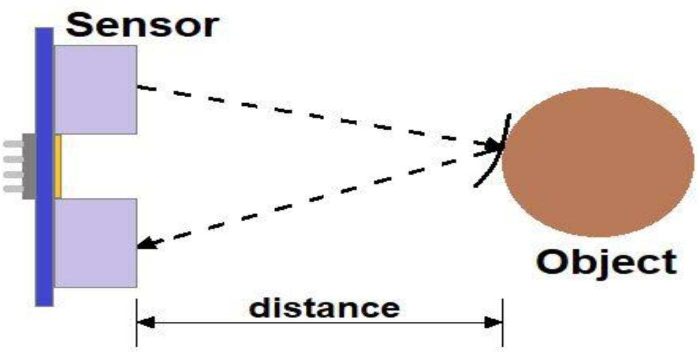




When the transmitter emits a short burst of sound waves, they travel through the air, hit an object, and return to the sensor as an echo. The sensor calculates the distance by measuring the time interval between sending the wave and receiving the echo, using the formula:

Distance = (Speed × Time) / 2

Dividing by two accounts for the round trip of the waves. The speed of sound is approximately 343 meters per second at room temperature, and it can vary with environmental conditions such as temperature and humidity, Ultrasonic sensors are typically accurate within ±3 millimeters and can measure distances ranging from 2 centimeters to 400 centimeters. They usually have a detection angle of 15 to 30 degrees, so objects outside this range may not be detected. These sensors are widely used in robotics for obstacle detection, in cars as parking sensors, in measuring liquid levels, and in various industrial automation applications



$$Distance = \frac{Time\ x\ Speed\ of\ Sound\ in\ Air\ (343m/s)}{2}$$

- **Open Challenge**

- **Role of the Distance Sensor**

To navigate the initial stage intelligently, we developed a dedicated algorithm that empowered the robot to make autonomous directional decisions, Prior to encountering the first turn, the robot was in a state of uncertainty, unsure whether to proceed clockwise or counterclockwise

We incorporated ultrasonic sensors into the robot's design to provide it with continuous spatial awareness. These sensors measured the distance to nearby obstacles on both the left and right
sides

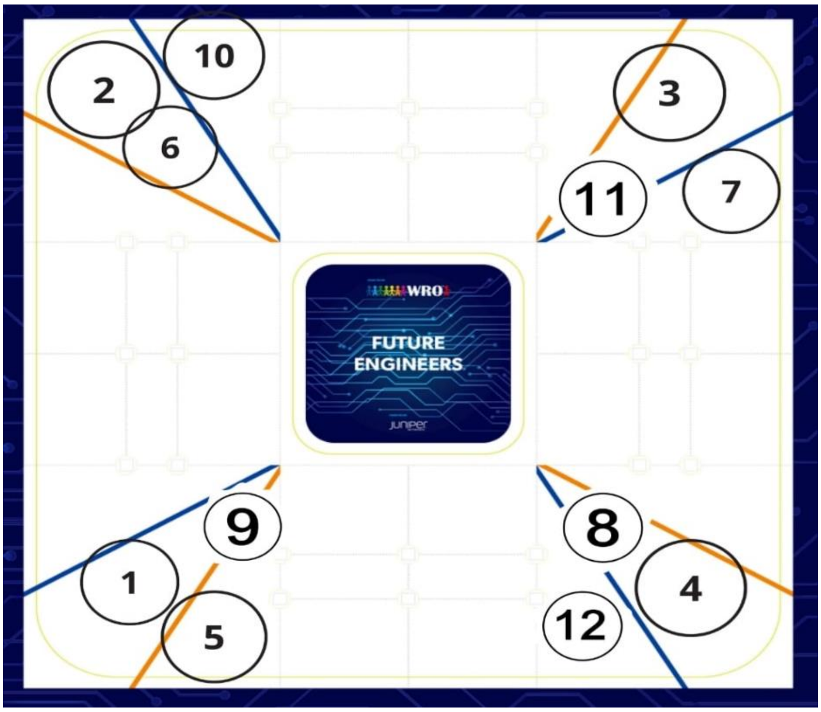
Right Sensor: Conversely, if the right sensor measured a distance beyond 160 cm, the robot opted to move
counterclockwise

Left Sensor: If the left ultrasonic sensor detected a distance exceeding 160 cm (indicating ample space on that side), the robot interpreted this as a signal to move
clockwise

Moreover, while traveling in a straight path the robot maintained a safety buffer of 20 cm

from the inner wall. If this distance was breached, the system automatically adjusted the trajectory to keep the robot stable and balanced This algorithm allowed the robot to dynamically adapt to its surroundings, making intelligent path choices and real-time corrections, embodying a sense of spatial awareness similar to advanced autonomous systems

Every time, the robot will Ultrasonic how many times it has crossed the corner and detected a large distance of around 160 cm. It will count 12 laps (1 round = 4 corners), and after that, the robot will stop at a specific location.



Start counterclockwise

Start clockwise

Gyroscope Working Principle

A gyroscope is a sensor used to measure the rate of rotation around the different axes of an object, allowing it to detect how the body rotates in space. Gyroscopes usually work by measuring the Coriolis force acting on a moving element inside the sensor and converting it into an electrical signal. (Note: Coriolis force is an apparent force that arises when an object moves in a rotating frame of reference.) This signal enables the calculation of the change in angle over time, allowing robots to know their orientation accurately, maintain balance, or move in a straight line. Gyroscopes are widely used in robotics for navigation and motion control

Equation

$\theta = \int \omega \, dt$

Symbols Explanation:

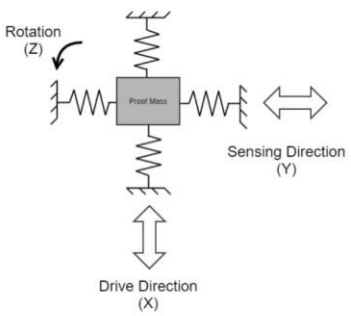
Angle rotated (degrees or radians). : θ

Rotation rate (degrees/s or rad/s). : ω

Time interval between gyroscope readings (seconds): dt

Role of the Gyroscope •

In our robot, the gyroscope helps determine the robot's orientation precisely when facing turns and obstacles. In the Obstacle Challenge, it measures the robot's rotation rate and calculates the change in angle to decide whether to turn left or right, keeping it aligned with the path. In the Open Challenge, the gyroscope works alongside distance sensors to avoid collisions, maintain a fixed distance (~20 cm) from walls or obstacles, and allow the robot to make smart turning decisions without relying solely on the path lines



- **Obstacle Challenge Round**

Strategy:

In this challenge, the robot is required to complete three full laps around a track containing randomly placed red and green traffic signs. Each traffic sign acts as a visual cue, guiding the robot's movement along the track.

Traffic Sign Interpretation

Red Sign: The robot must stay on the right side of the lane

Green Sign: The robot must stay on the left side of the lane

Rules and Constraints:

1. The robot must not move or displace any traffic signs during the run.

2. After completing the three laps, the robot must locate a parking space and perform parallel parking correctly

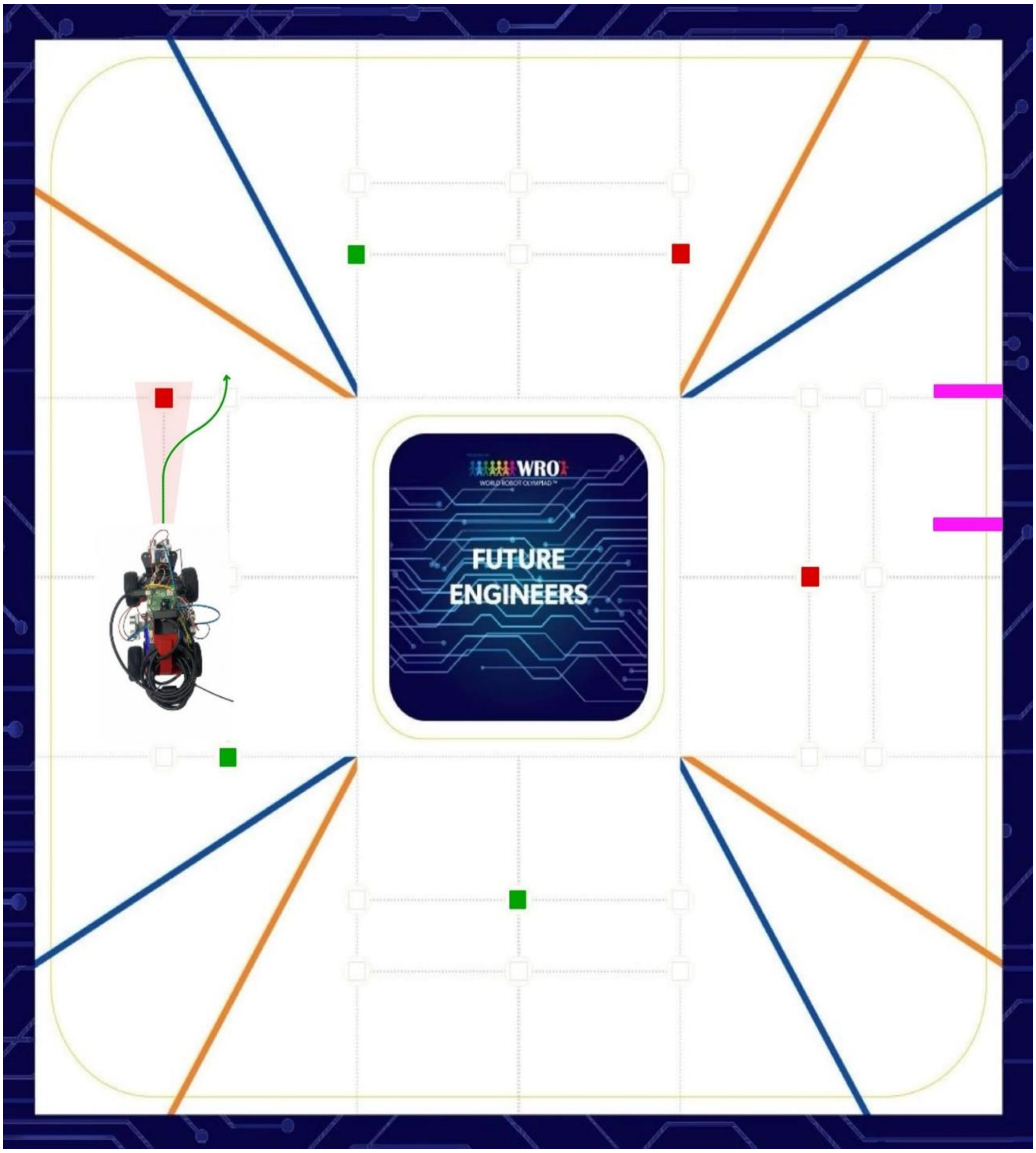
Final Objective:

Demonstrate the robot's ability to detect signs and make accurate decisions in real time.

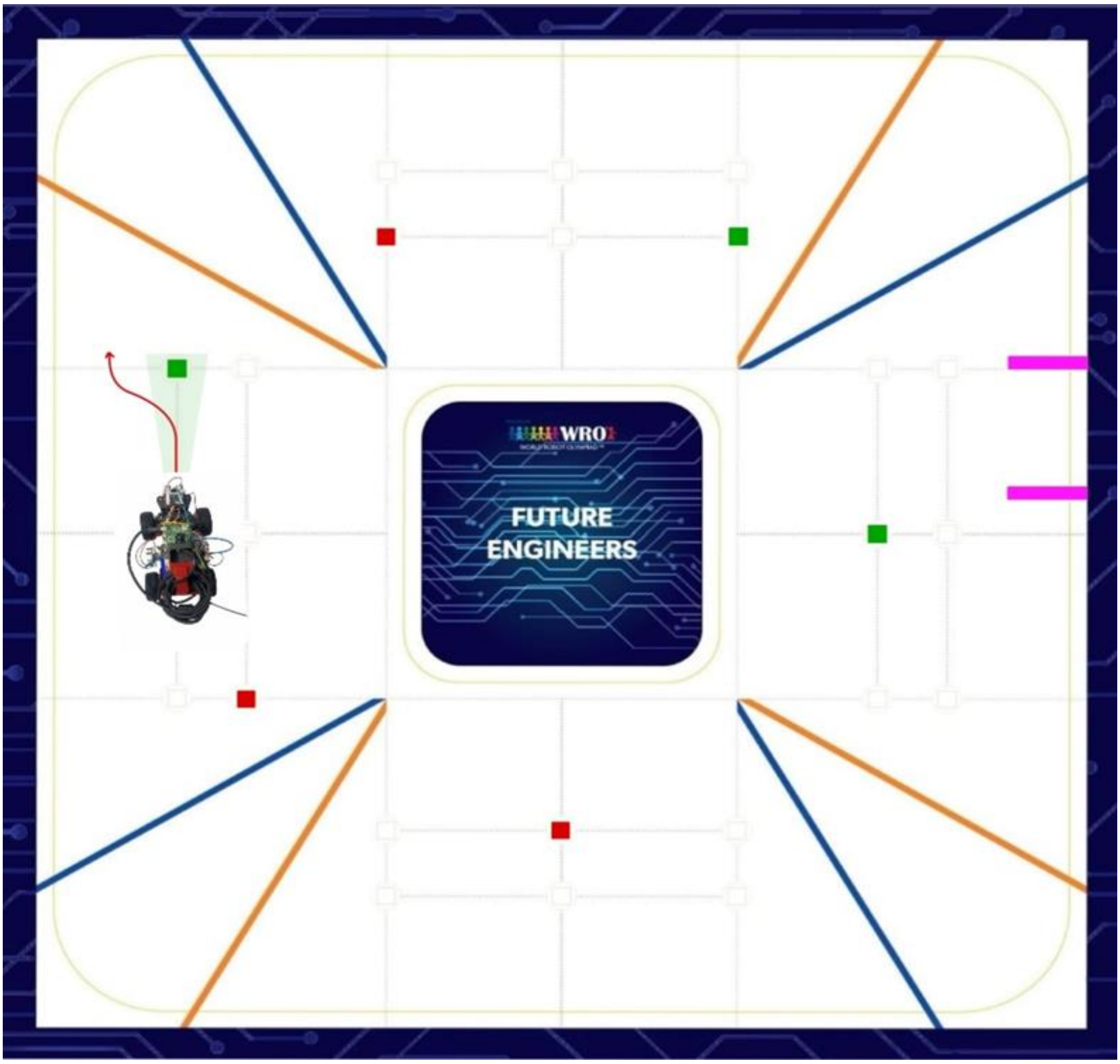
Maintain stable and precise movement while following all track rules.

In this round, the robot must complete three laps on a track with randomly placed green and red traffic signs

- Red Obstacle: The robot must keep to the right side of the lane

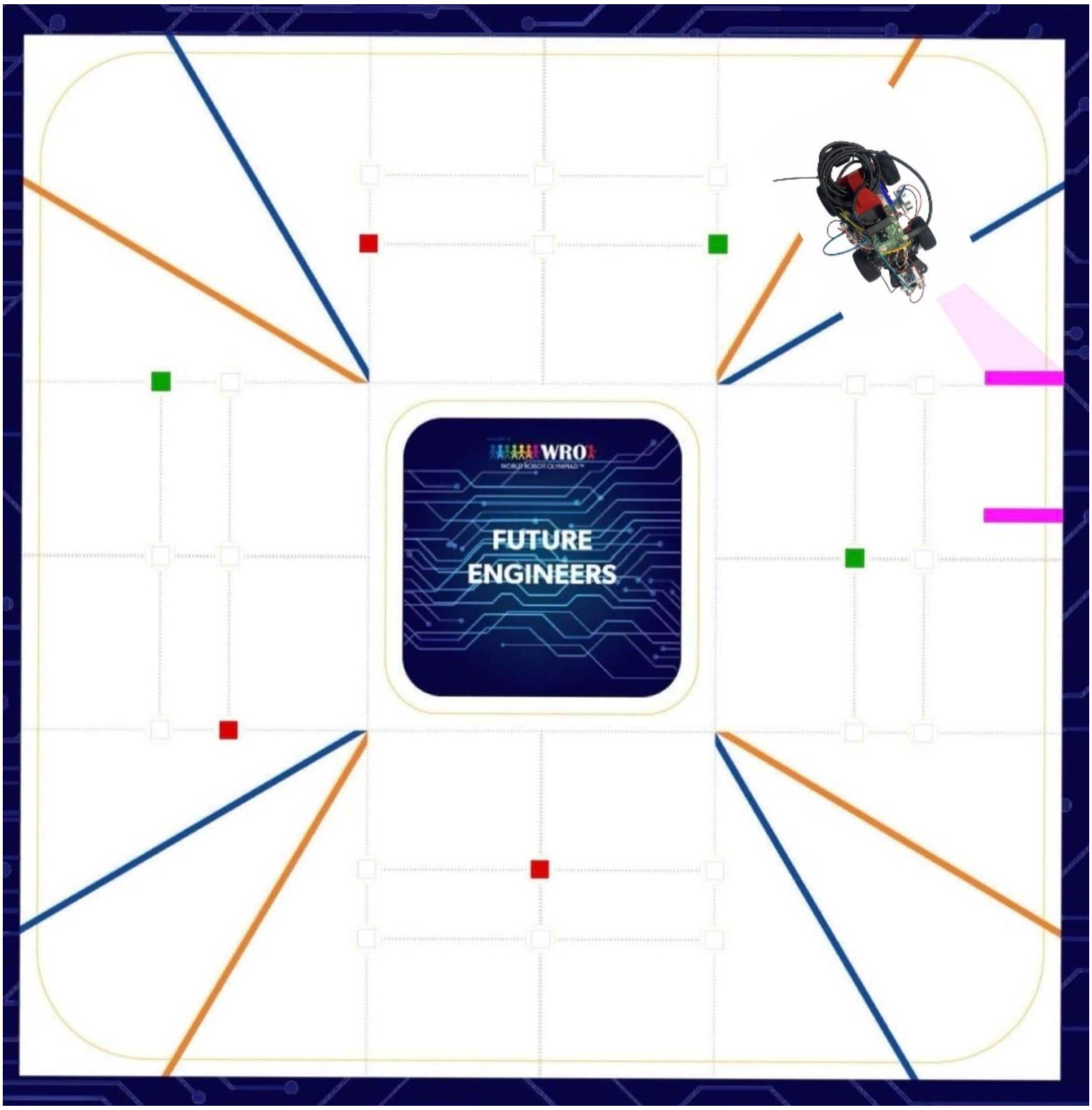


- Green Obstacle: The robot must keep to the left side of the lane



- The robot must not move or pass any of the traffic signs.

- After finishing the three laps, the robot must find a parking lot and perform parallel parking



- **Camera Specifications and Essential Details (ELP 720p H.264 USB)**

This compact ELP camera model is an ideal choice for robotic systems, serving as the robot's primary "eye." The camera utilizes a CMOS sensor and delivers 720p HD resolution at 30 frames per second (fps), ensuring a smooth, real-time visual data stream. The most critical feature is its use of H.264 compression via a standard USB (UVC) interface. This setup allows the transmission of high-quality, color-accurate images with minimal load on the main processing unit, which is vital in autonomous robotics. Its streamlined and compact design, combined with support for low-light conditions, makes it easy to integrate neatly into any robot chassis.

- **Working Principle of Camera**

Application in the WRO Future Engineers Competition (Vision and Obstacle Avoidance)

This camera proved to be a pivotal component in our robot, designed for the WRO Future Engineers category—a challenge focused on building a self-driving robotic car capable of autonomous navigation and smart decision-making.

Computer Vision Powered by Raspberry Pi

The camera was connected to a Raspberry Pi, which functioned as the robot's "brain" for image processing. The camera was instrumental in executing key tasks on the competition track:

Green Obstacle: The robot must keep to the left side of the lane

Red Obstacle: The robot must keep to the right side of the lane

In addition, the robot uses a camera to recognize the parking area and navigate directly to it.

Cube Recognition (Target Identification): Utilizing Computer Vision libraries (such as OpenCV) on the Raspberry Pi, the robot was able to analyze video frames to accurately determine the shape, color, and precise location of the target cubes on the field with high speed.

Dynamic Obstacle Avoidance: Beyond target identification, the camera was used for continuous environmental scanning to detect sudden obstacles or course changes. Thanks to the fast 30 fps frame rate, the Raspberry Pi system could process the data and make movement decisions (such as steering, stopping, or re-routing) within milliseconds. This rapid decision-making ensured reliable obstacle avoidance and successful mission completion under the randomized conditions of the competition