Misr University for Science and Technology
College of Engineering
Department of Mechatronics Engineering
MTE411

ROBOTICS 2 FINAL PROJECT

Two wheel car robot

RobotBuilders

Index

Topic	Page
Introduction	**4**
The Component	**6**
The Hardware	**12**
The Simulation	**14**
The Calculation	**18**
Conclusion	**21**

RobotBuilders

Members:

ABDELRAHMAN YOUSSEF	95782
AHMED ABDELAZIZ	97908
AMR KHALED	95772
MOHAB REFAT	95787
MOHAMMED HANY SALEM	95776
SARA GOMAA	95963

Supervised by:
Dr. Bahaa Eldin Naser
Dr. Maha Soliman
Dr. Waleed El-badry
Eng. Islam Algendy
Eng. Alaa Ahmed

Introduction

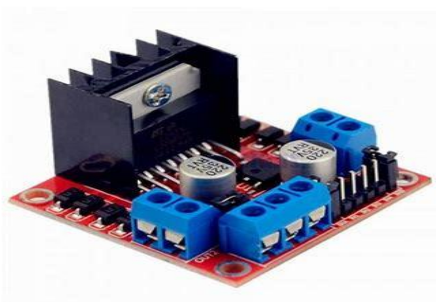
This report details the design, application, and testing of a two-wheel car robot using the (ROS), Gazebo for simulation, and RViz for visualization. The robot is equipped with an IR sensor for obstacle detection, two encoders for wheel movement tracking, and a Bluetooth module for communication.

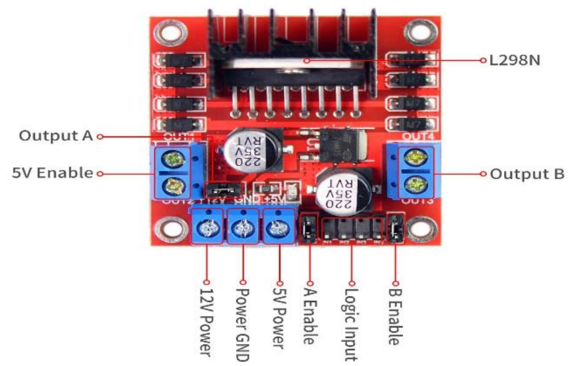
5

The Code

The Component

1-Motor Driver



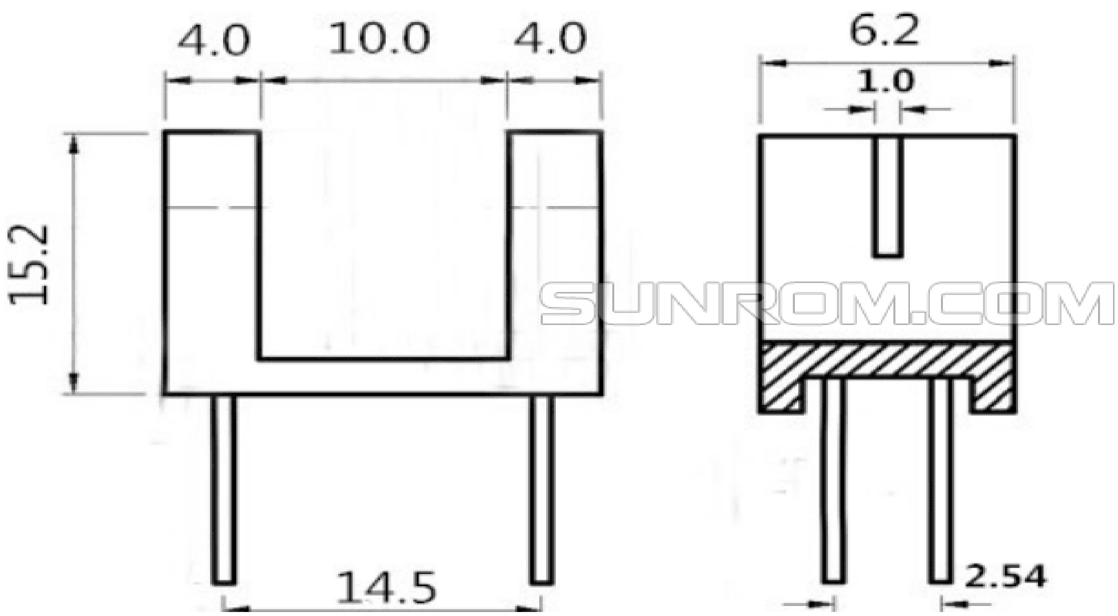


Input Voltage: 3.2V ~ 40Vdc.

-Operating current range: 0 ~ 36mA.

- Maximum power consumption: 20W (when the temperature T = 75 ℃).

2- Encoder h2010



*** IR SLOT GAP SENSOR**

*** 10mm GAP**

3- DC Motor with Gearbox and wheel

-Rated Voltage: 3~6V

-Min. Operating Speed (3V): 90+/- 10% RPM

-Min. Operating Speed (6V): 200+/- 10% RPM

-Torque: 0.15Nm ~0.60Nm

-Gear Ratio: 1:48

4- Bluetooth module

features

- Typical -80dBm sensitivity.

- Up to +4dBm RF transmit power.

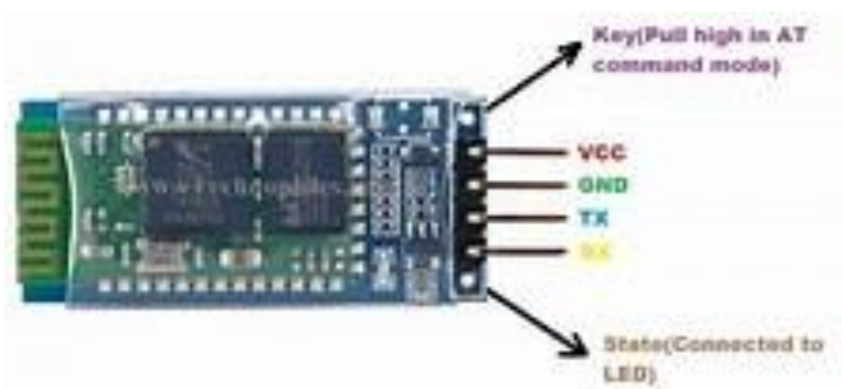
- Low Power 1.8V Operation, 3.3 to 5 V I/O.

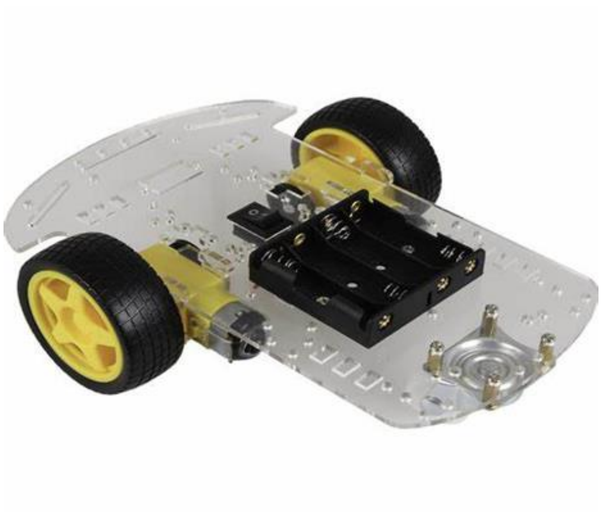
- PIO control.

- UART interface with programmable baud rate.

-With integrated antenna.

- With edge connector.

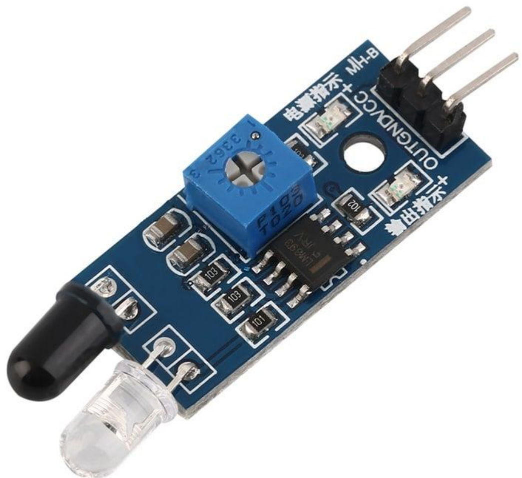


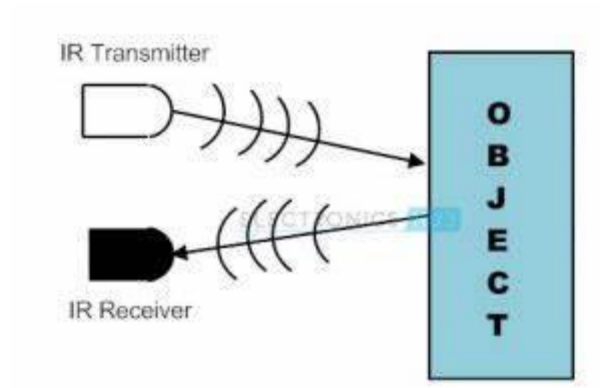


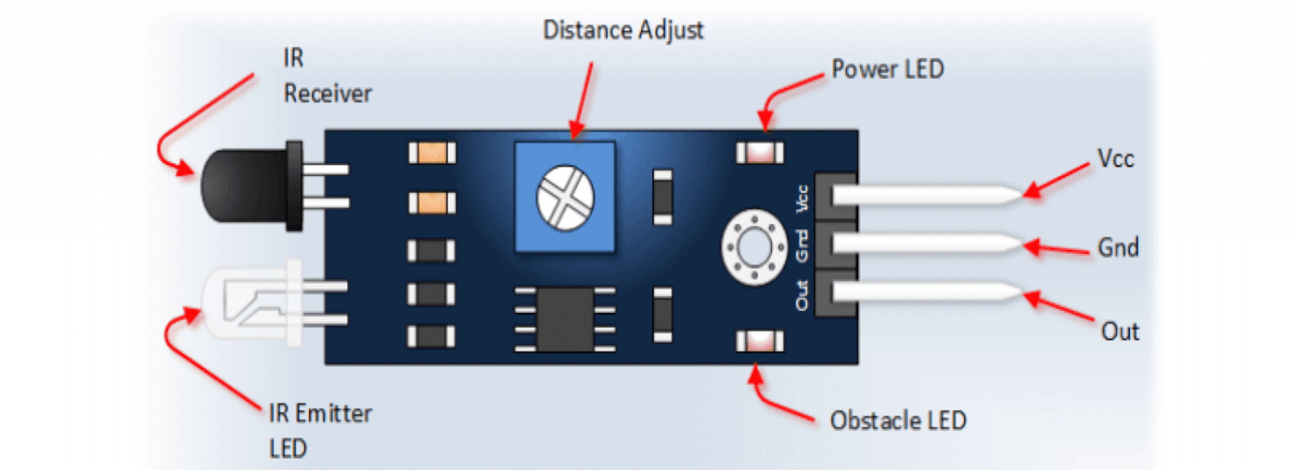
2 wheel Robot Smart Car Chassis Kit BO Motor wheel Kit with Speed Encoder, 2 Wheels, 2 Motor, 1 Caster Wheel and Battery Box for Arduino

The chassis used in this kit is transparent so as to create dynamic handling of the components mounted on your robotic vehicle

IR sensor

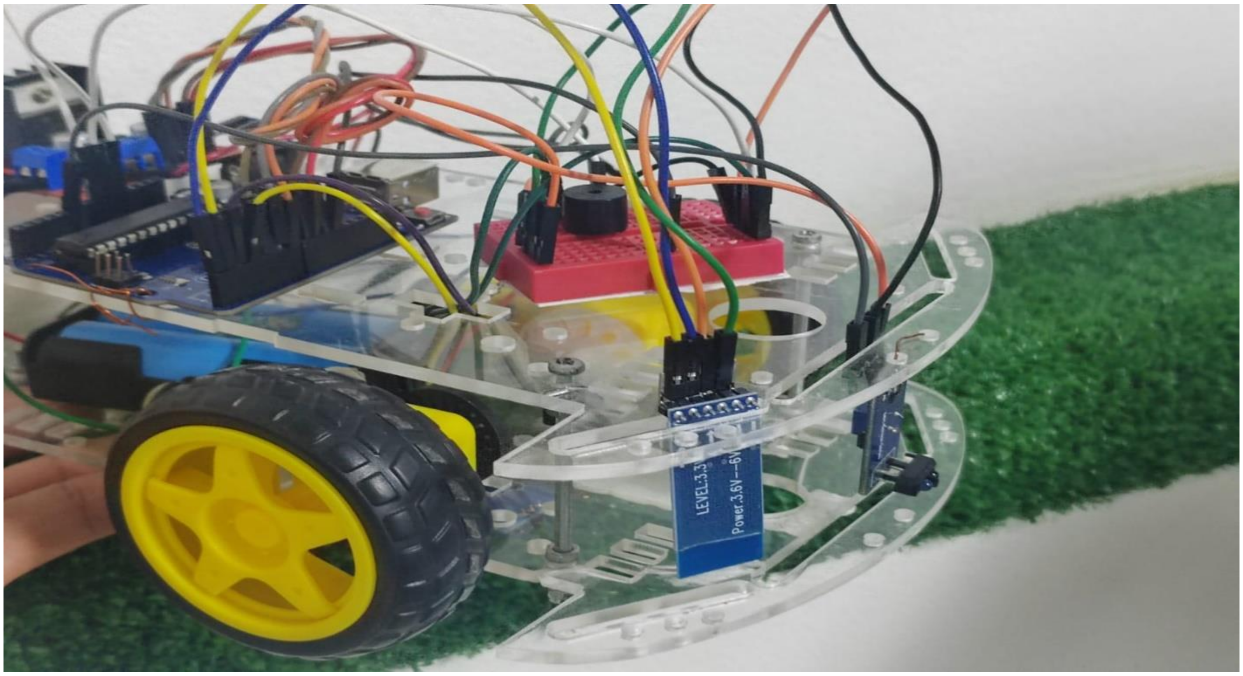


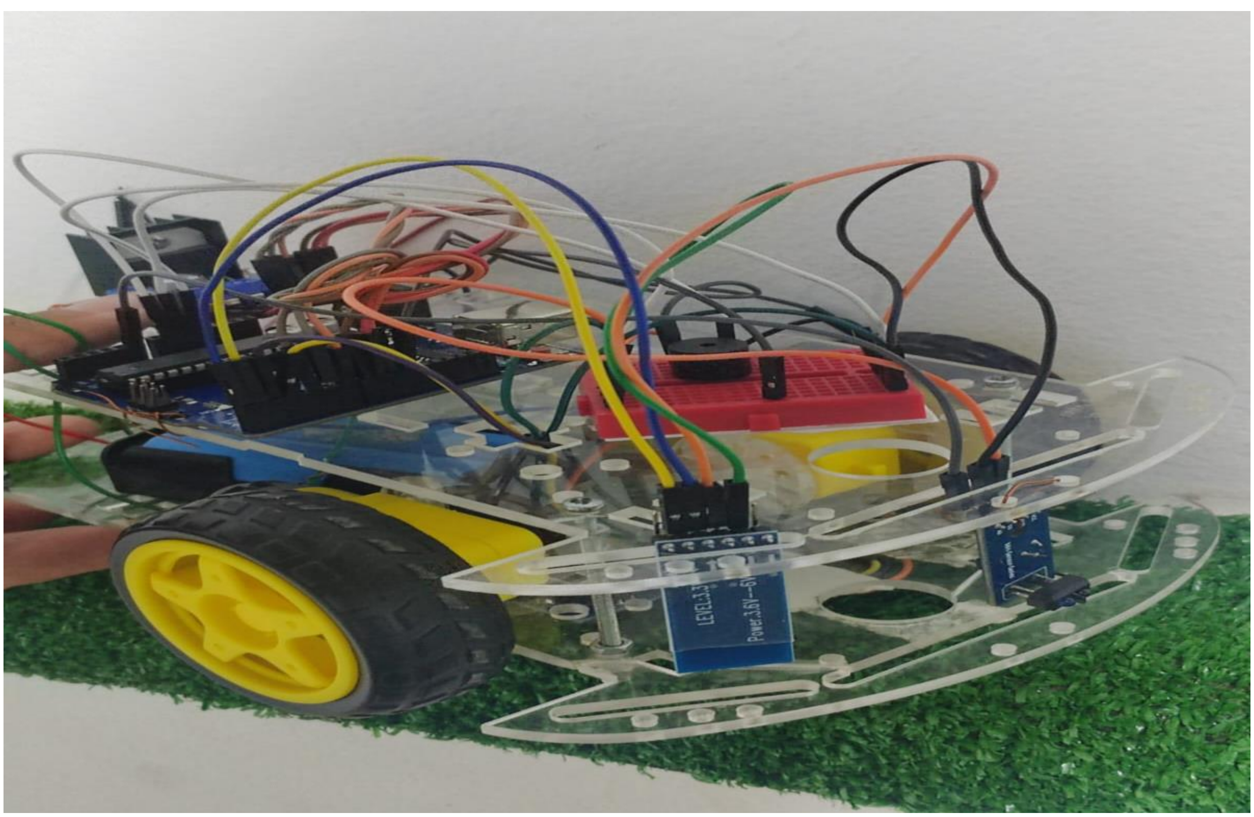


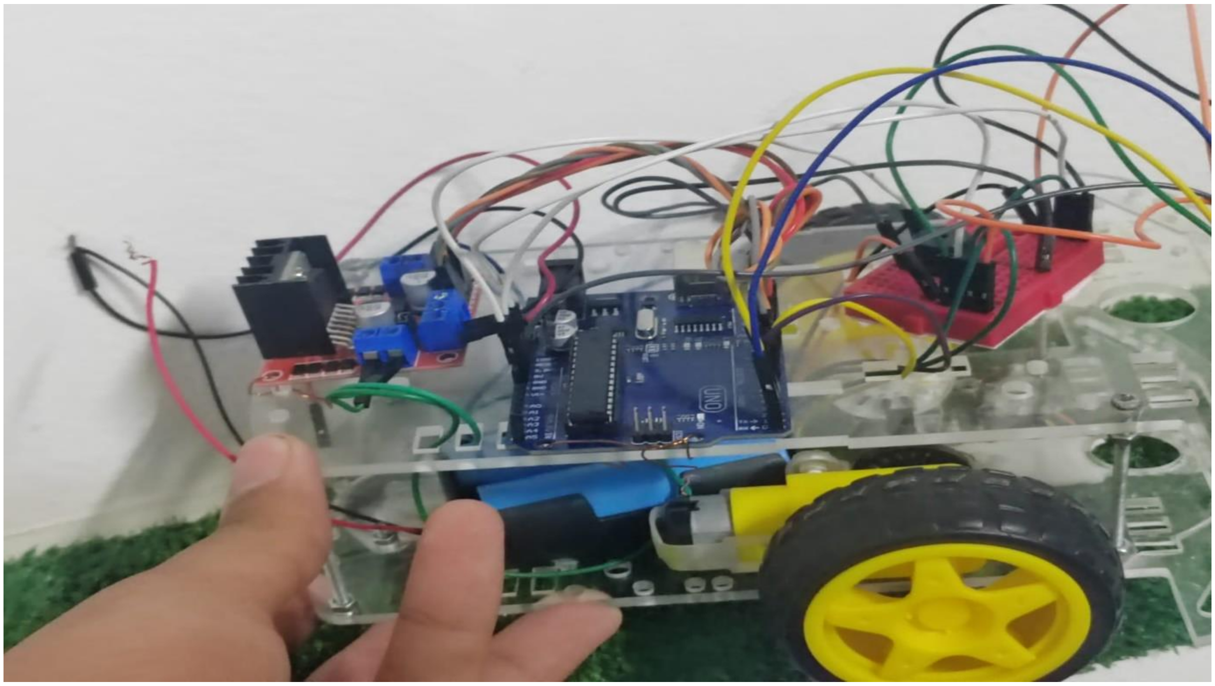


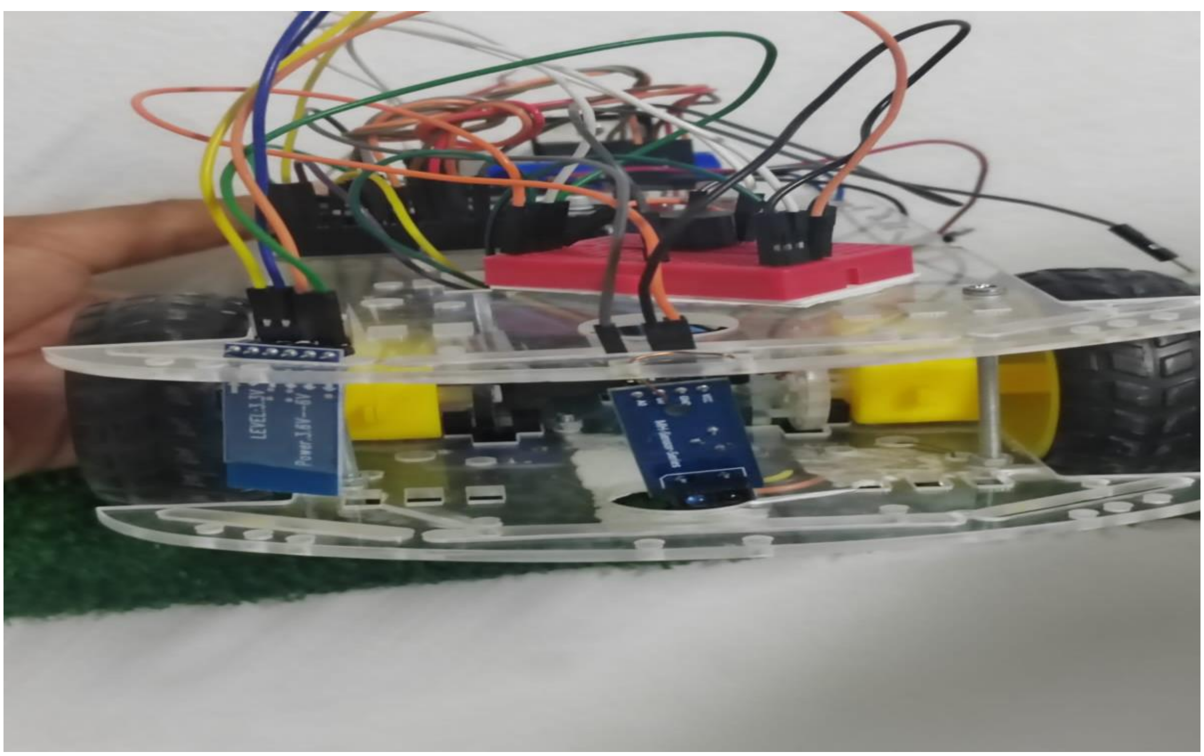
Pin, Control Indicator	Description
Vcc	3.3 to 5 Vdc Supply Input
Gnd	Ground Input
Out	Output that goes low when obstacle is in range
Power LED	Illuminates when power is applied
Obstacle LED	Illuminates when obstacle is detected
Distance Adjust	Adjust detection distance. CCW decreases distance. CW increases distance.
IR Emitter	Infrared emitter LED
IR Receiver	Infrared receiver that receives signal transmitted by Infrared emitter.

The Hardware



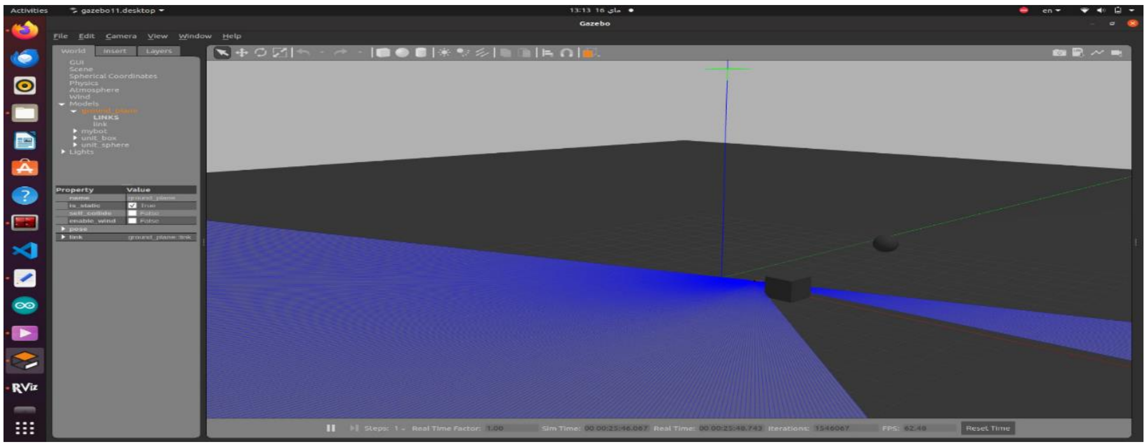


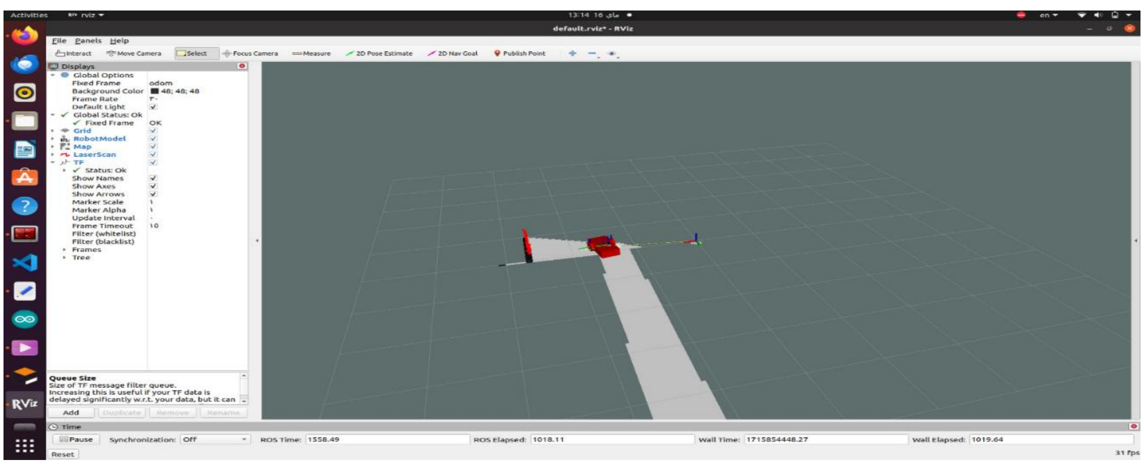

UNO

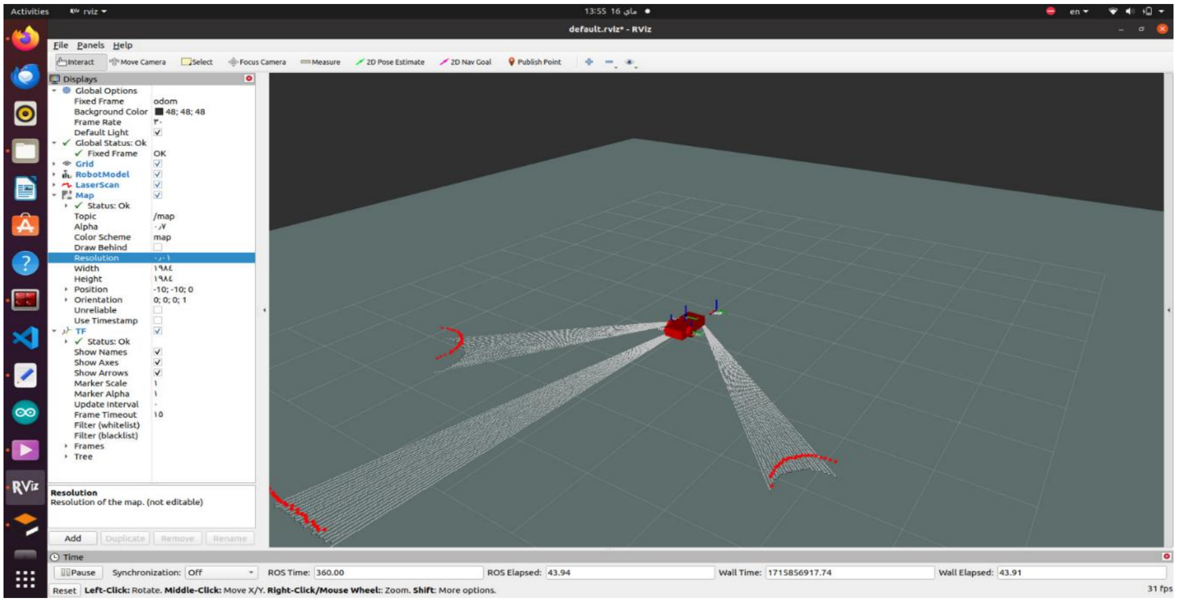


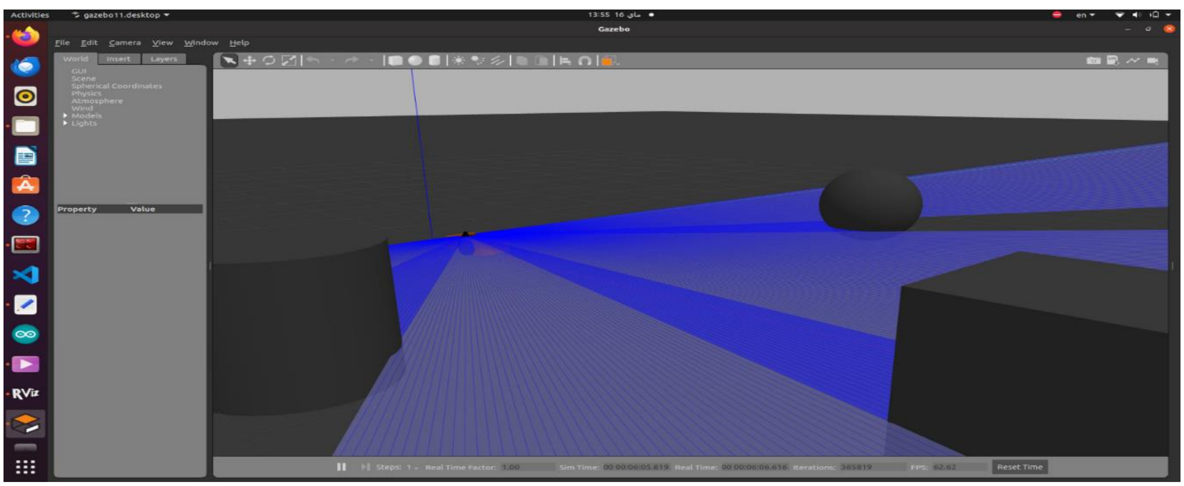
14

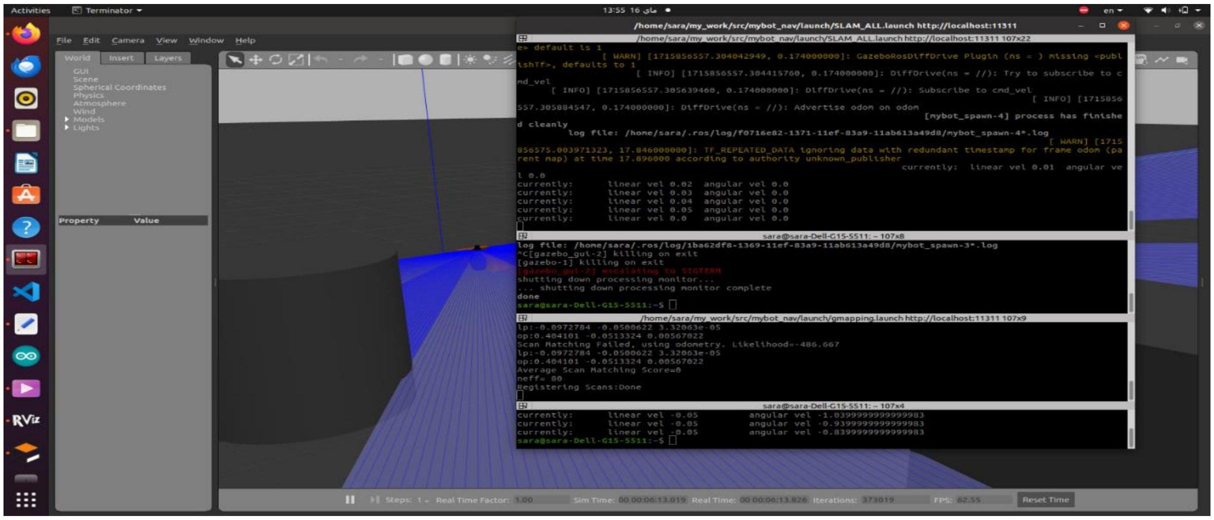
The Simulation

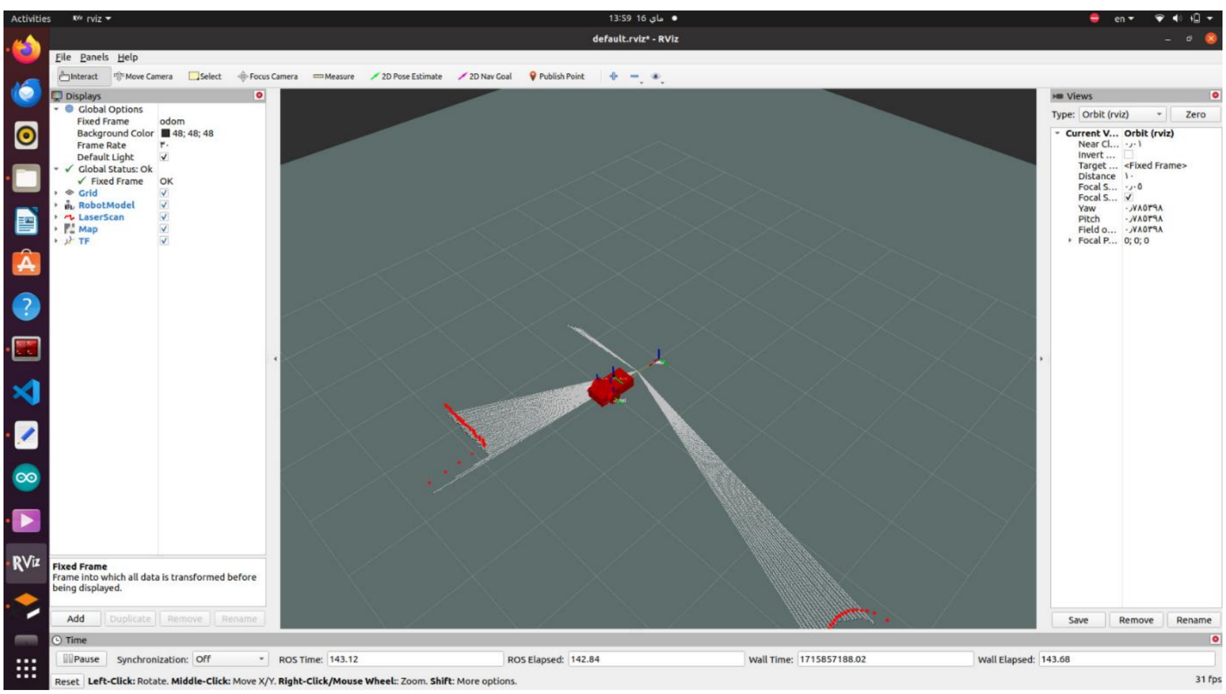


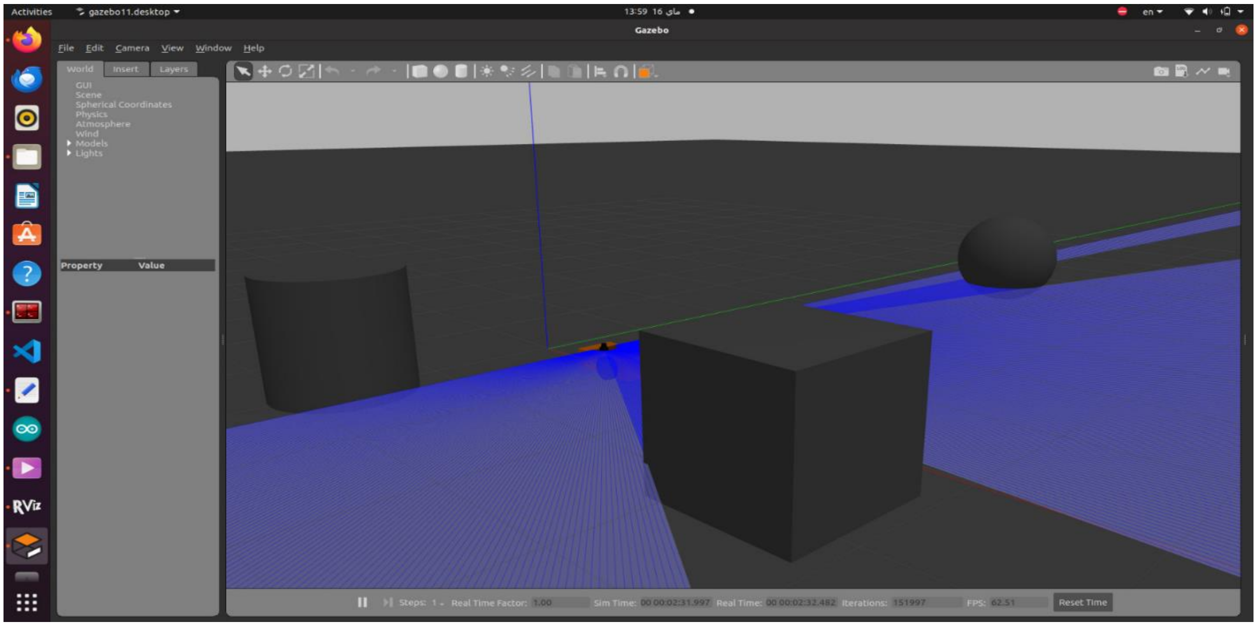



default.rviz* - RViz
File Panels Help
Interact
Move Camera
Select
Focus Camera
Measure
2D Pose Estimate
2D Nav Goal
Publish Point
Displays
Global Options
Fixed Frame odom
Background Color 48; 48; 48
Default Light
Global Status: Ok
Fixed Frame OK
Grid
RobotModel
LaserScan
Map
Status: Ok
Topic /map
Color Scheme map
Draw Behind
Resolution
Width
Height
Position -10; -10; 0
Orientation 0; 0; 0; 1
Unreliable
Use Timestamp
TF
Status: Ok
Show Names
Show Axes
Show Arrows
Marker Scale
Marker Alpha
Update Interval
Frame Timeout
Filter (whitelist)
Filter (blacklist)
Frames
Tree
Resolution
Resolution of the map. (not editable)
Add
Duplicate
Remove
Rename
Time
Pause
Synchronization: Off
ROS Time: 360.00
ROS Elapsed: 43.94
Wall Time: 1715856917.74
Wall Elapsed: 43.91
Reset
Left-Click: Rotate. Middle-Click: Move X/Y. Right-Click/Mouse Wheel:: Zoom. Shift: More options.
31 fps

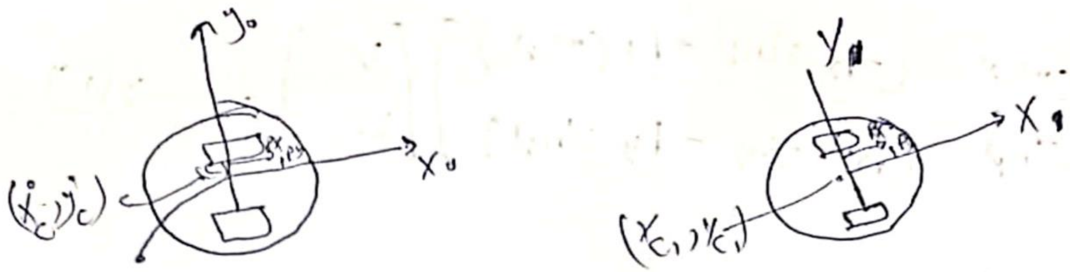

Gazebo
File Edit Camera View Window Help
World Insert Layers
GUI
Scene
Spherical Coordinates
Physics
Atmosphere
Wind
Models
Lights
Property Value
Reset Time


Activities
Terminator
File Edit Camera View Window Help
World Insert Layers
GUI
Scene
Spherical Coordinates
Physics
Atmosphere
Wind
Models
Lights
Property Value
Reset Time
RViz


Activities
rviz
default.rviz* - RViz
File Panels Help
Interact
Move Camera
Select
Focus Camera
Measure
2D Pose Estimate
2D Nav Goal
Publish Point
Displays
Global Options
Fixed Frame
odom
Background Color
48; 48; 48
Frame Rate
Default Light
Global Status: Ok
Fixed Frame
OK
Grid
RobotModel
LaserScan
Map
TF
Fixed Frame
Frame into which all data is transformed before being displayed.
Add
Duplicate
Remove
Rename
Views
Type: Orbit (rviz)
Zero
Current V...
Orbit (rviz)
Target ...
<Fixed Frame>
Save
Remove
Rename
Time
Pause
Synchronization:
Off
ROS Time:
143.12
ROS Elapsed:
142.84
Wall Time:
1715857188.02
Wall Elapsed:
143.68
Reset
Left-Click: Rotate. Middle-Click: Move X/Y. Right-Click/Mouse Wheel:: Zoom. Shift: More options.
31 fps


Activities
gazebo11.desktop
Gazebo
File
Edit
Camera
View
Window
Help
World
Insert
Layers
GUI
Scene
Spherical Coordinates
Physics
Atmosphere
Wind
Models
Lights
Property
Value
RViz
Steps:
Real Time Factor:
Sim Time:
Real Time:
Iterations:
FPS:
Reset Time

The Calculation



$$\dot{P} = {}^{1}_{0}R \cdot \dot{P} \longrightarrow (1)$$

$$\dot{C} = {}^{0}_{1}R \cdot \dot{C} \longrightarrow (2)$$

$${}^{0}P - {}^{0}C = {}^{0}_{1}R \, ({}^{1}P - {}^{1}C)$$

$$\begin{bmatrix} X_P - X_C \\ Y_P - Y_C \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} X_P - X_C \\ Y_P - Y_C \end{bmatrix}$$

$$\begin{bmatrix} X_P \\ Y_P \end{bmatrix} = \begin{bmatrix} X_C \\ Y_C \end{bmatrix} + \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} P_X \\ P_Y \end{bmatrix}$$

$$\begin{bmatrix} X_C \\ Y_C \end{bmatrix} = \begin{bmatrix} X_P \\ Y_P \end{bmatrix} - \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} P_X \\ P_Y \end{bmatrix}$$

~~Differtable~~ Differention

$$\begin{bmatrix} \dot{X}_P \\ \dot{Y}_P \end{bmatrix} = \begin{bmatrix} \dot{X}_C \\ \dot{Y}_C \end{bmatrix} + \begin{bmatrix} -\sin\theta \cdot \dot{\theta} & -\cos\theta \cdot \dot{\theta} \\ \cos\theta \cdot \dot{\theta} & -\sin\theta \cdot \dot{\theta} \end{bmatrix} \begin{bmatrix} P_X \\ P_Y \end{bmatrix}$$

$$\therefore \begin{bmatrix} \dot{X}_P \\ \dot{Y}_P \end{bmatrix} = \begin{bmatrix} V\cos\theta \\ V\sin\theta \end{bmatrix} + \begin{bmatrix} -\sin\theta\, w & -\cos\theta\, w \\ \cos\theta\, w & -\sin\theta\, w \end{bmatrix} \begin{bmatrix} P_X \\ P_Y \end{bmatrix}$$

$$\begin{bmatrix} \dot{X}_P \\ \dot{Y}_P \end{bmatrix} = \begin{bmatrix} V\cos\theta - \sin\theta \; wP_X - \cos\theta \; wP_Y \\ V\sin\theta + \cos\theta \; wP_X - \sin\theta \; wP_Y \end{bmatrix}$$

$$\begin{bmatrix} \dot{x}_p \\ \dot{y}_p \end{bmatrix} = \begin{bmatrix} \cos\theta & (-P_x \sin\theta - P_y \cos\theta) \\ \sin\theta & (P_x \cos\theta - P_y \sin\theta) \end{bmatrix} \begin{bmatrix} V \\ W \end{bmatrix} \longrightarrow (1)$$

$$\left.\begin{aligned} \dot{x}_p &= K_{px} (x_{ref} - x_p) \\ \dot{y}_p &= K_{py} (y_{ref} - y_p) \end{aligned}\right\} \longrightarrow (2)$$

From 1, 2

$$\begin{bmatrix} K_{px} (x_{ref} - x_p) \\ K_{py} (y_{ref} - y_p) \end{bmatrix} = \begin{bmatrix} \cos\theta & (-P_x \sin\theta - P_y \cos\theta) \\ \sin\theta & (P_x \cos\theta - P_y \sin\theta) \end{bmatrix} \begin{bmatrix} V \\ W \end{bmatrix}$$

$$\begin{bmatrix} V \\ W \end{bmatrix} = \begin{bmatrix} \cos - \left(\frac{P_y}{P_x}\right)\sin\theta & \sin\theta + \left(\frac{P_y}{P_x}\right)\cos\theta \\ -\left(\frac{1}{P_x}\right)\sin\theta & -\left(\frac{1}{P_x}\right)\cos\theta \end{bmatrix} \begin{bmatrix} K_{px} (x_{ref} - x_p) \\ K_{py} (y_{ref} - y_p) \end{bmatrix}$$

عزم الجسم $\Leftarrow$ $I = \frac{1}{12} m L^2$

$L = 17 \times 10^{-2} + 8 \times 10^{-2} = ,25\ m$

واحنا عندنا إتنين شاسيه يعني

$L = 2 \times ,25 = ,5\ m$

كلي $I = \frac{1}{12} \times (1) \times (,5)^2 = ,020\ kg.m^2$

عزم العجل الواحدة $I = \frac{1}{2} m r^2$

$= \frac{1}{2} \times (30 \times 10^{-2}) \times (2.5 \times 10^{-2}) = 3.75 \times 10^{-4}\ kg.m^2$

كليتين $I = 2 \times 3.75 \times 10^{-4} = 7.5 \times 10^{-4}\ kg.m^2$

$I = 7.5 \times 10^{-4} + ,020 = ,02075\ kg.m^2$

Conclusion

The two-wheel car robot project demonstrates the integration of ROS, Gazebo, and RViz for developing and testing a mobile robot. The use of IR sensors, encoders, and Bluetooth for communication enhances the robot's capabilities in navigation and control. The successful implementation and testing in a simulated environment validate the effectiveness of the proposed system architecture.