

TurtleBot 4 Simulation Environment – Startup Guide (with LLM Navigation)

1. Source ROS 2 and Your Workspace

Open a terminal and source your workspace.

You need to do this in **every new terminal** before running any ROS 2 command.

Command to run:

```
source ~/turtlebot4_ws/install/setup.bash
```

2. Launch the Simulation (Gazebo + Nav2 + RViz)

In the same terminal, launch the full simulation including localization and visualization (but **not SLAM**):

Command to run:

```
ros2 launch turtlebot4_ignition_bringup turtlebot4_ignition.launch.py nav2:=true slam:=false  
localization:=true rviz:=true
```

3. Set the Initial Pose

In a **new terminal**, source the workspace again and run the robot pose initializer:

Commands to run:

```
source ~/turtlebot4_ws/install/setup.bash  
ros2 run llm_nav_interface llm_nav_init
```

This node sends an initial pose to AMCL so the robot knows where it is on the map.

4. Launch the LLM Nav Interface (Main Logic)

In **another new terminal**, launch the main interface that listens for commands, queries the LLM, parses its response, and sends goals to Nav2:

Commands to run:

```
source ~/turtlebot4_ws/install/setup.bash  
ros2 run llm_nav_interface llm_nav_goal_sender
```

This will wait for user input in the terminal (e.g. "Go to the table"), query the LLM API, convert the response into a goal, and send it to Nav2.

5. Operate the Robot via RViz and Gazebo (Optional)

In RViz:

- Use the **2D Pose Estimate** tool if needed to manually adjust the robot's position.
- Use the **2D Nav Goal** tool to manually send goals (alternative to LLM).

In Ignition GUI:

- Use the **Teleop plugin** to drive the robot manually with buttons or keys.

6. To Run The Planner Benchmark Script

Run the commands below:

```
source ~/turtlebot4_ws/install/setup.bash
```

```
cd turtlebot4_ws/src/llm_nav_interface/llm_nav_interface/benchmarking
```

then run one of these:

```
python3 benchmark_planners.py --trials 5 --output results/dwb_results.csv
```

```
python3 benchmark_planners.py --trials 5 --output results/teb_local_planner_results.csv
```

```
python3 benchmark_planners.py --trials 5 --output
```

```
results/regulated_pure_pursuit_controller_results.csv
```

Optional: Manual Teleop with Keyboard

To control the robot manually via your keyboard script:

```
python3 ros2_ws/src/my_first_pubsub/turtlebot4_teleop_custom_key.py
```

Optional: View Robot Model in RViz Only

To view the 3D model of TurtleBot 4 in RViz without launching full simulation:

```
ros2 launch turtlebot4_viz view_model.launch.py
```

Helpful Links

- LLM Nav Integration (Custom): *your repo*
- TurtleBot 4 Navigation Tutorial:
https://turtlebot.github.io/turtlebot4-user-manual/tutorials/turtlebot4_navigator.html#install-tutorial-package
- TurtleBot 4 Driving / Teleop:
<https://turtlebot.github.io/turtlebot4-user-manual/tutorials/driving.html>

For Building The Packages (When building new packages on ROS2):

```
cd turtlebot4_ws
```

```
colcon build --packages-select llm_nav_interface
```