

ReadMe

Install:

1. I have used ROS2 Humble and turtlebot3.
2. Install ROS2 Humble.
3. `sudo apt install ros-humble-desktop`
4. Source the ROS setup.bash script.
5. Install turtlebot3
6. `sudo apt install ros-humble-turtlebot3*`
7. Using rosdep get the dependencies required
8. `rosdep install --from-paths src --ignore-src -y`
9. Then build it
10. `colcon build --symlink-install`

PART1:

Teleop using keyboard

Terminal 1

1. Go to turtle bot workspace and source local setup.bash
2. `export ROS_DOMAIN_ID=1`
3. `source /opt/ros/humble/setup.bash`
4. `export TURTLEBOT3_MODEL=burger`
5. Source gazebo
6. `ros2 launch turtlebot3_gazebo turtlebot3_world.launch.py`
7. The above cmd launches gazebo world

Terminal 2

8. Go to turtle bot workspace and source local setup.bash
9. `export ROS_DOMAIN_ID=1`
10. `source /opt/ros/humble/setup.bash`
11. `export TURTLEBOT3_MODEL=burger`
12. Source gazebo
13. `ros2 run turtlebot3_teleop teleop_keyboard`
14. The above package controls the turtlebot3 using keyboard

PART2:

BUG0 for obstacle avoidance move left

Sensor used: Laser scan

Terminal 1

15. Go to turtle bot workspace and source local setup.bash
16. `export ROS_DOMAIN_ID=1`
17. `source /opt/ros/humble/setup.bash`
18. `export TURTLEBOT3_MODEL=burger`
19. Source gazebo
20. `ros2 launch turtlebot3_gazebo turtlebot3_world.launch.py`
21. The above cmd launches gazebo world

Terminal 2

22. Go to turtle bot workspace and source local setup.bash
23. export ROS_DOMAIN_ID=1
24. source /opt/ros/humble/setup.bash
25. export TURTLEBOT3_MODEL=burger
26. Source gazebo
27. colcon build --packages-select my_package #To build my package
28. Source install/local.bash
29. ros2 run my_package my_node
30. The above cmd runs node my_node for bug0 implementation for obstacle avoidance.

I have attached part1 and part2 mp4 and also my_package and turtlebot3 packages in the zip file.

My_node - BUG0 Code path:

ros2_ws/src/my_package/my_package/my_node.py