Steps of use turtlebot3 with SLAM approach to create and save a map

Install turlebot3 on ROS

Enter the below command to a terminal

1. **Enter this site**

https://emanual.robotis.com/docs/en/platform/turtlebot3/quick-start/

2. **Download and install ros**
3. **Install package ros**

```
$ sudo apt-get install ros-kinetic-joy ros-kinetic-teleop-twist-joy \
  ros-kinetic-teleop-twist-keyboard ros-kinetic-laser-proc \
  ros-kinetic-rgbd-launch ros-kinetic-depthimage-to-laserscan \
  ros-kinetic-rosserial-arduino ros-kinetic-rosserial-python \
  ros-kinetic-rosserial-server ros-kinetic-rosserial-client \
  ros-kinetic-rosserial-msgs ros-kinetic-amcl ros-kinetic-map-server
\
  ros-kinetic-move-base ros-kinetic-urdf ros-kinetic-xacro \
  ros-kinetic-compressed-image-transport ros-kinetic-rqt* \
  ros-kinetic-gmapping ros-kinetic-navigation ros-kinetic-
interactive-markers
```

4. **Install package turlebot3**

```
$ sudo apt-get install ros-kinetic-dynamixel-sdk
$ sudo apt-get install ros-kinetic-turtlebot3-msgs
$ sudo apt-get install ros-kinetic-turtlebot3
```

5. **Set TurtleBot3 Model Name**

In case of TurtleBot3 Burger

```
$ echo "export TURTLEBOT3_MODEL=burger" >> ~/.bashrc
```

In case of TurtleBot3 Waffle Pi

```
$ echo "export TURTLEBOT3_MODEL=waffle_pi" >> ~/.bashrc
```

Turlebot3 Simulation

Enter this link

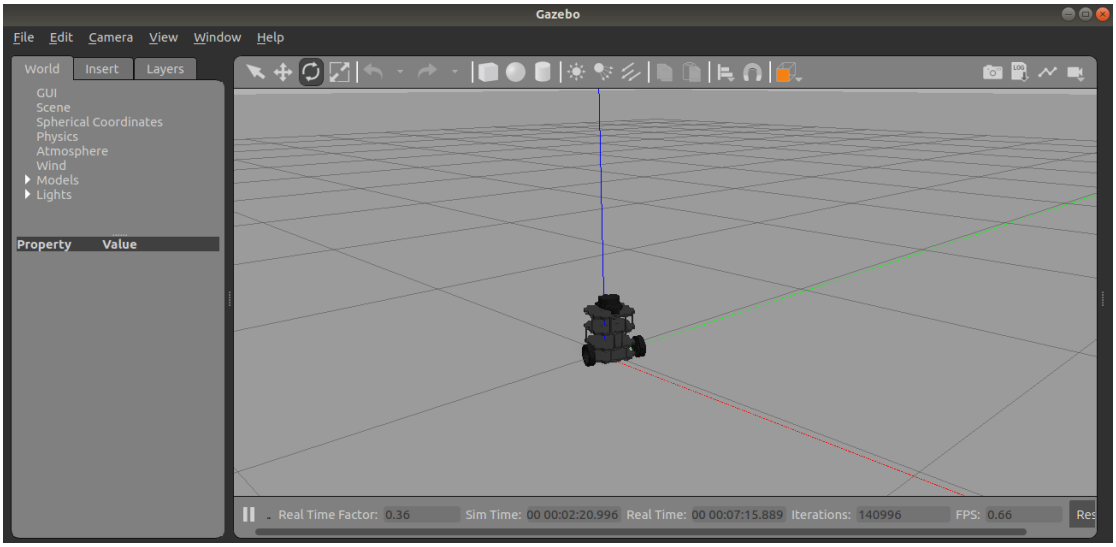
https://emanual.robotis.com/docs/en/platform/turtlebot3/simulation/

Gazebo simulation

1. **Install simulation package**

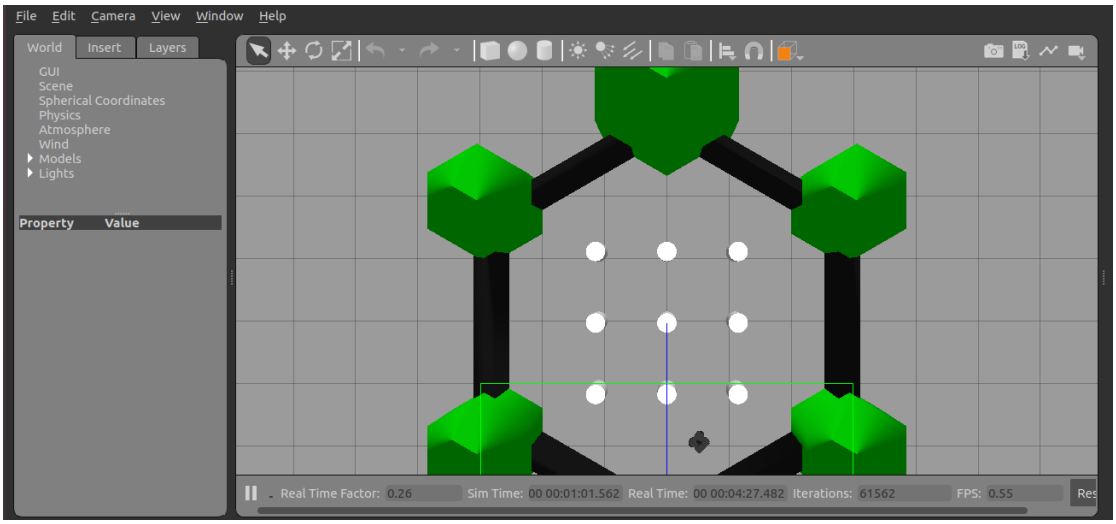
```
$ cd ~/catkin_ws/src/
$ git clone -b kinetic-devel https://github.com/ROBOTIS-
GIT/turtlebot3_simulations.git
$ cd ~/catkin_ws && catkin_make
```

2. **Launch simulation world**
    a. **Empty World**



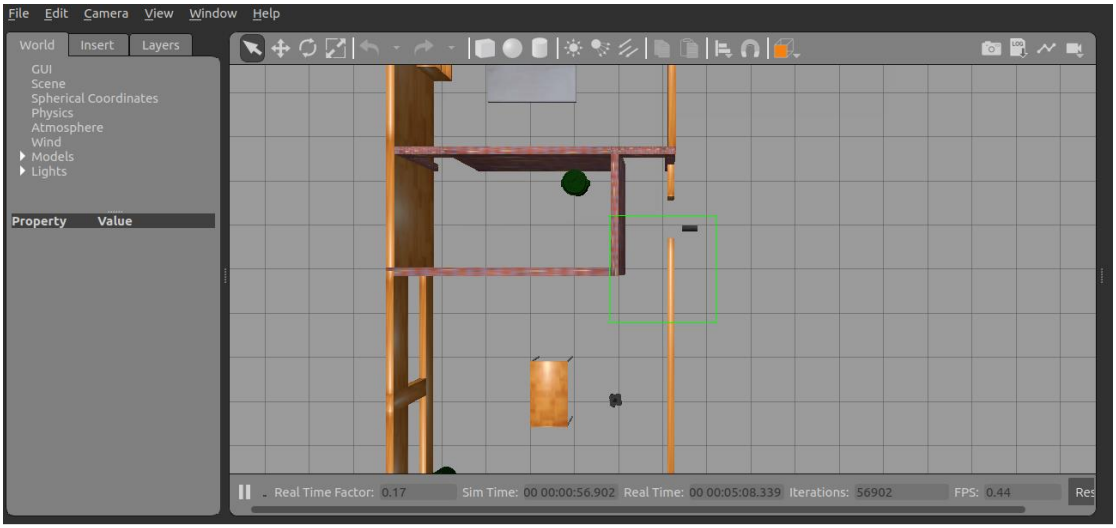

```
$ export TURTLEBOT3_MODEL=burger
$ roslaunch turtlebot3_gazebo turtlebot3_empty_world.launch
```

b. TurtleBot3 World




```
$ export TURTLEBOT3_MODEL=waffle
$ roslaunch turtlebot3_gazebo turtlebot3_world.launch
```

c. TurtleBot3 House




```
$ export TURTLEBOT3_MODEL=waffle_pi
$ roslaunch turtlebot3_gazebo turtlebot3_house.launch
```

3. Operate TurtleBot3

```
$ roslaunch turtlebot3_teleop turtlebot3_teleop_key.launch
```

SLAM Simulation

1. Launch Simulation World

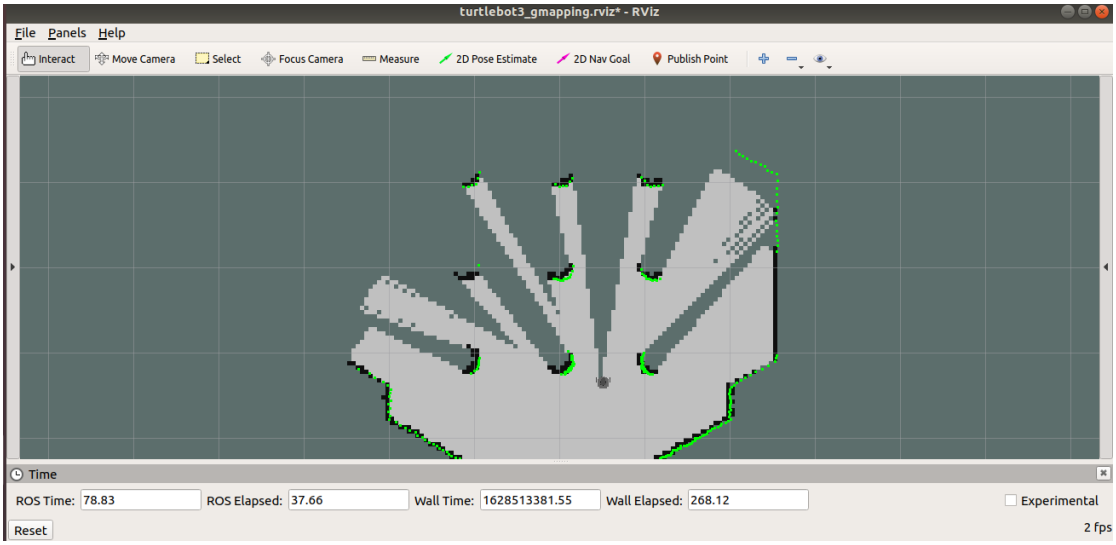
```
$ export TURTLEBOT3_MODEL=burger
$ roslaunch turtlebot3_gazebo turtlebot3_world.launch
```

2. Run SLAM Node

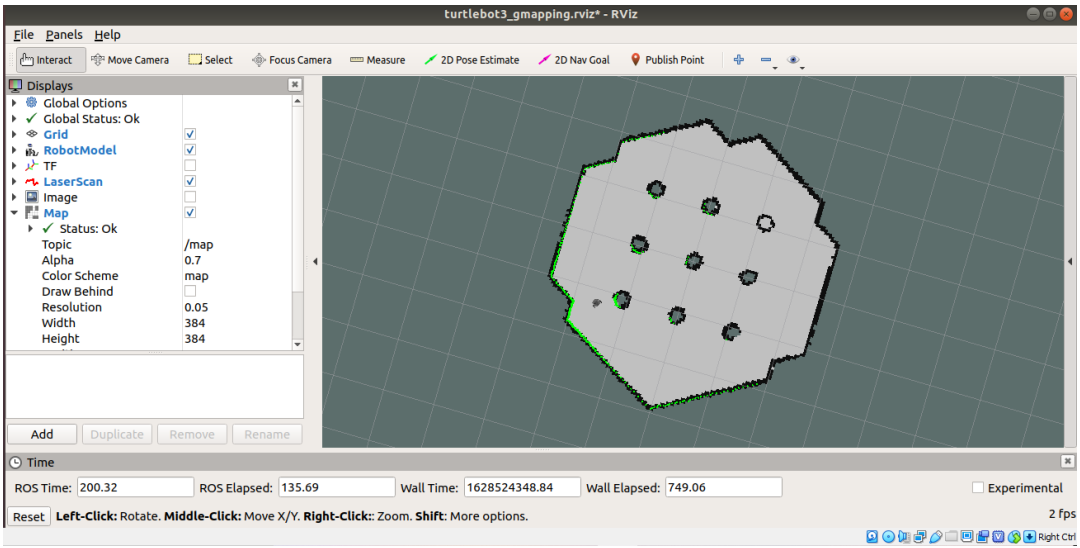
```
$ export TURTLEBOT3_MODEL=burger
$ roslaunch turtlebot3_slam turtlebot3_slam.launch
slam_methods:=gmapping
```

3. Run Teleperation Node

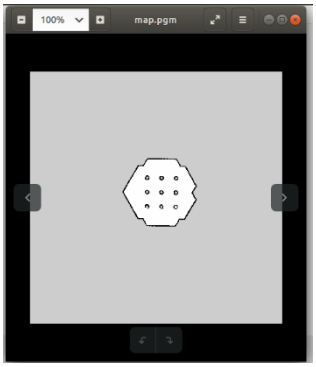
```
$ export TURTLEBOT3_MODEL=burger
$ roslaunch turtlebot3_teleop turtlebot3_teleop_key.launch
```



4. Save Map




```
jamal@jamal:~/catkin_ws$ rosrun map_server map_saver -f ~/map
[ INFO] [1628513789.072654151]: Waiting for the map
[ INFO] [1628513789.257595674]: Received a 384 X 384 map @ 0.050 m/pix
[ INFO] [1628513789.258707787]: Writing map occupancy data to /home/jamal/map.pgm
[ INFO] [1628513789.341260097, 146.206000000]: Writing map occupancy data to /home/jamal/map.yaml
[ INFO] [1628513789.352883523, 146.206000000]: Done
```




```
rosrun map_server map_saver -f ~/map
```