

2.Handheld lidar mapping

The following workspace contains the entire function package in rplidar_ws.

If you need to transplant it to your own development board, you need to copy all the function packages to the src of the workspace for compilation, and install the relevant environment according to the actual situation.

The approximate steps are as follows, taking ros1-melodic as an example:

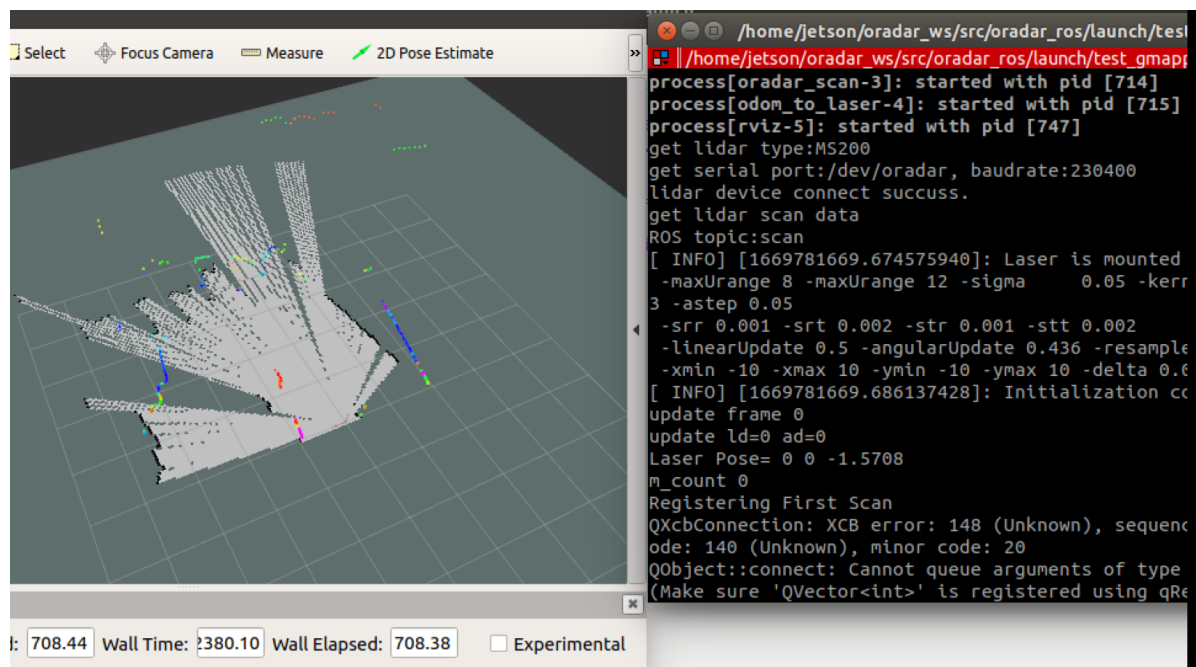
```
sudo apt install ros-melodic-moveit ros-melodic-moveit-visual-tools ros-melodic-kdl-* ros-melodic-joint-state-publisher-gui ros-melodic-trac-ik liborocos-kdl-dev  
ros-melodic-teleop-twist-keyboard ros-melodic-moveit-resources ros-melodic-navigation ros-melodic-gmapping ros-melodic-hector-slam ros-melodic-slam-karto  
ros-melodic-robot-state-publisher ros-melodic-geographic-msgs ros-melodic-libuv-* ros-melodic-rtabmap-ros libavformat-dev libavcodec-dev libswresample-dev  
libswscale-dev libavutil-dev libstd1.2-dev ros-melodic-pointcloud-to-laserscan ros-melodic-mbf-msgs ros-melodic-mbf-costmap-core ros-melodic-costmap-converter  
ros-melodic-bfl ros-melodic-serial ros-melodic-teleop-twist-joy ros-melodic-laser-proc ros-melodic-rosserial-arduino ros-melodic-rosserial-python ros-melodic-rosserial-server  
ros-melodic-rosserial-client ros-melodic-rosserial-msgs ros-melodic-amcl ros-melodic-map-server ros-melodic-urdf ros-melodic-xacro ros-melodic-interactive-markers  
ros-melodic-octomap* ros-melodic-joy* ros-melodic-dwa-local-planner ros-melodic-multirobot-map-merge python-catkin-tools python3-dev python3-catkin-pkg-modules  
python3-numpy python3-yaml build-essential ros-melodic-imu-tools ros-melodic-cartographer*
```

2.1 Start the mapping command

Input following command:

```
roslaunch oradar_lidar test_gmapping.launch
```

If the following screen appears, it means the operation is successful.



2.2 Save map

Input following command:

```
roslaunch map_server map_saver -f ~/oradar_ws/src/oradar_ros/maps/map
```

The map will be saved to the path ~/rplidar_ws/src/oradar_ros/maps/. A pgm picture and a yaml file.

map.yaml

```
image: map.pgm
resolution: 0.05
origin: [-15.4, -12.2, 0.0]
negate: 0
occupied_thresh: 0.65
free_thresh: 0.196
```

Parameter analysis:

- image: The path of the map file, which can be an absolute path or a relative path.
- resolution: resolution of the map, meters/pixel
- Origin: 2D pose (x, y, yaw) in the lower left corner of the map. The yaw here is rotated counterclockwise (yaw=0 means no rotation change). Many parts of the current system ignore the yaw value.
- negate: whether to reverse the meaning of white/black and free/occupied (the interpretation of the threshold is not affected)
- occupied_thresh: Pixels with an occupation probability greater than this threshold will be considered fully occupied.
- free_thresh: Pixels with occupancy probability less than this threshold will be considered completely free.

2.3 View related information

View tf tree

```
roslaunch rqt_tf_tree rqt_tf_tree
```

View node communication

```
roslaunch rqt_graph rqt_graph
```

2.4 Algorithm information reference website

2.4.1 Mapping algorithm

Gmapping: <http://wiki.ros.org/gmapping/>

hector_slam: http://wiki.ros.org/hector_slam

hector_slam/Tutorials: http://wiki.ros.org/hector_slam/Tutorials/SettingUpForYourRobot

hector_mapping: http://wiki.ros.org/hector_mapping

karto: http://wiki.ros.org/slam_karto

Cartographer: <https://google-cartographer.readthedocs.io/en/latest/>

Cartographer ROS: <https://google-cartographer-ros.readthedocs.io/en/latest/>

2.4.2 Autonomous mapping navigation

rrt_exploration: http://wiki.ros.org/rrt_exploration

rrt_exploration/Tutorials: http://wiki.ros.org/rrt_exploration/Tutorials

2.4.3 Save map

map_server: https://wiki.ros.org/map_server