

2. Handheld lidar mapping

It is recommended to directly use the virtual machine or Raspberry Pi 5 image we provide, because it may not run due to environmental differences

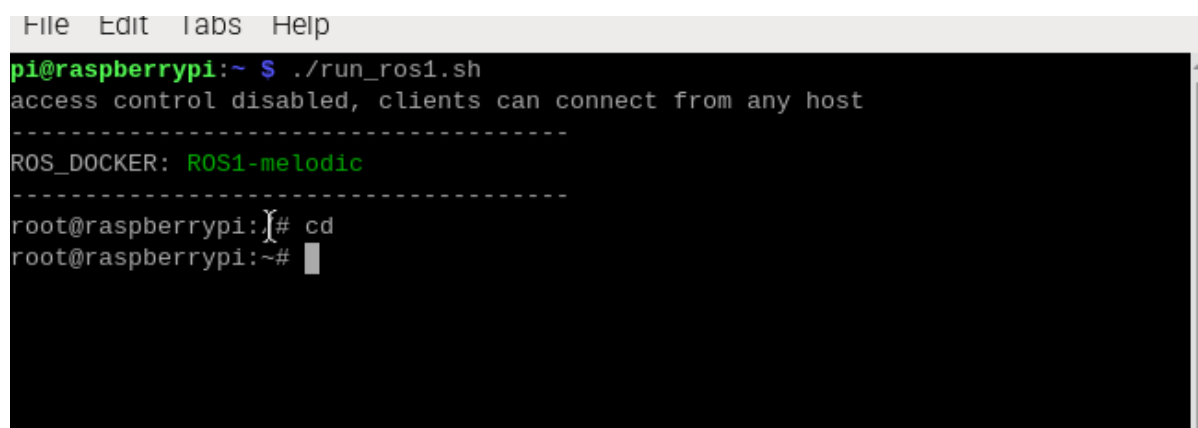
1. Install dependent libraries, take the ros-melodic version as an example, enter in the terminal,

```
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. Start the mapping command

First go into the docker we provide, open a terminal in the Raspberry Pi directory and enter,

```
./run_ros1.sh
```



```
File Edit Tabs Help
pi@raspberrypi:~ $ ./run_ros1.sh
access control disabled, clients can connect from any host
-----
ROS_DOCKER: ROS1-melodic
-----
root@raspberrypi:~# cd
root@raspberrypi:~#
```

Take gmapping mapping as an example, enter in the terminal,

```
roslaunch oradar_lidar test_gmapping.launch
```

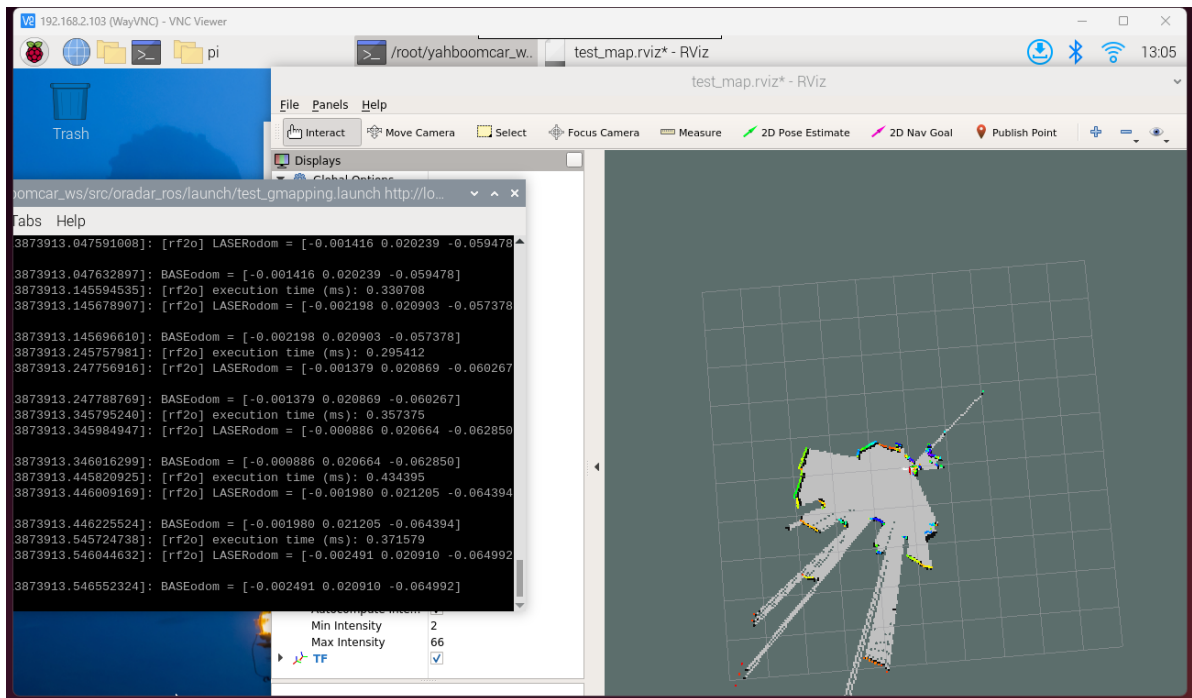
It is normal for this error to appear at the beginning of running the program, because the tf conversion was not converted at the beginning, but was converted later.

```

process [rviz-1] started with pid [5541]
[ERROR] [1712833527.282690671]: "base_link" passed to lookupTransform argument target_frame does not exist.
[YDLIDAR] Lidar successfully connected [/dev/ydlidar:230400]
[YDLIDAR] Lidar running correctly! The health status: good
[YDLIDAR] Baseplate device info
Firmware version: 1.3
Hardware version: 1
Model: Tmini Plus
Serial: 2024000000020063
[ INFO] [1712833528.283775292]: Listening laser scan from topic: /scan
[YDLIDAR] Current scan frequency: 10.00Hz
[YDLIDAR] Lidar init success, Elapsed time 1077 ms

```

The following screen appears, indicating successful operation.



3. Save the map

Taking gmapping as an example, the map saving command is,

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

```

root@raspberrypi:~# roslaunch map_server map_saver -f ~/yahboomcar_ws/src/oradar_ros/maps/map
[ INFO] [1713874088.988990189]: Waiting for the map
[ INFO] [1713874089.1238103646]: Received a 384 X 384 map @ 0.050 m/pix
[ INFO] [1713874089.238264204]: Writing map occupancy data to /root/yahboomcar_ws/src/oradar_ros/maps/map.pgm
[ INFO] [1713874089.243187902]: Writing map occupancy data to /root/yahboomcar_ws/src/oradar_ros/maps/map.yaml
[ INFO] [1713874089.244360067]: Done
root@raspberrypi:~#

```

The map will be saved to the ~/ydlidar_ws/src/oradar_ros/maps/ folder, a pgm image and a yaml file.

map.yaml

```
image: /home/yahboom/yahboomcar_ws/src/oradar_ros/maps/map.pgm
resolution: 0.050000
origin: [-10.000000, -10.000000, 0.000000]
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.

4. View relevant information

View tf tree

```
roslaunch rqt_tf_tree rqt_tf_tree
```

Node communication view

```
roslaunch rqt_graph rqt_graph
```

5. Algorithm information reference website

5.1. Graphing 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/>

5.2. Independent map navigation

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

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

5.3. Save the map

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