

Preparation

1. Create a new workspace and compile function packages

Take the creation name `rplidar_ws` as an example, input following command:

```
mkdir rplidar_ws
cd rplidar_ws
mkdir src
cd src
catkin_init_workspace
```

Copy the decompressed `rplidar_ros` function package to the `rplidar_ws/src` directory.

Then, in the directory of `rplidar_ws`, use `catkin_make` to compile.

```
cd ~/rplidar_ws
catkin_make
```

After the compilation is passed, add the path of the workspace to `.bashrc`.

```
sudo gedit ~/.bashrc
```

Copy the following content to the end of the file.

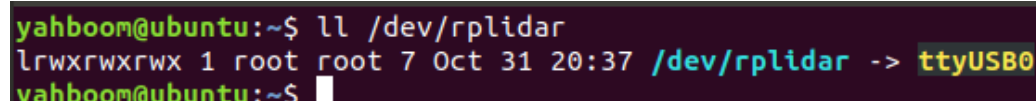
```
source ~/rplidar_ws/devel/setup.bash --extend
```

2. Bind lidar port name

Open the terminal, input the following command, and copy the `rplidar.rules` file under the function package to `/etc/udev/rules.d`.

```
cd ~/rplidar_ws/src/rplidar_ros/scripts
sudo cp rplidar.rules /etc/udev/rules.d/
```

Then, re-plug the radar wiring and enter the command `ll /dev/rplidar` in the terminal.



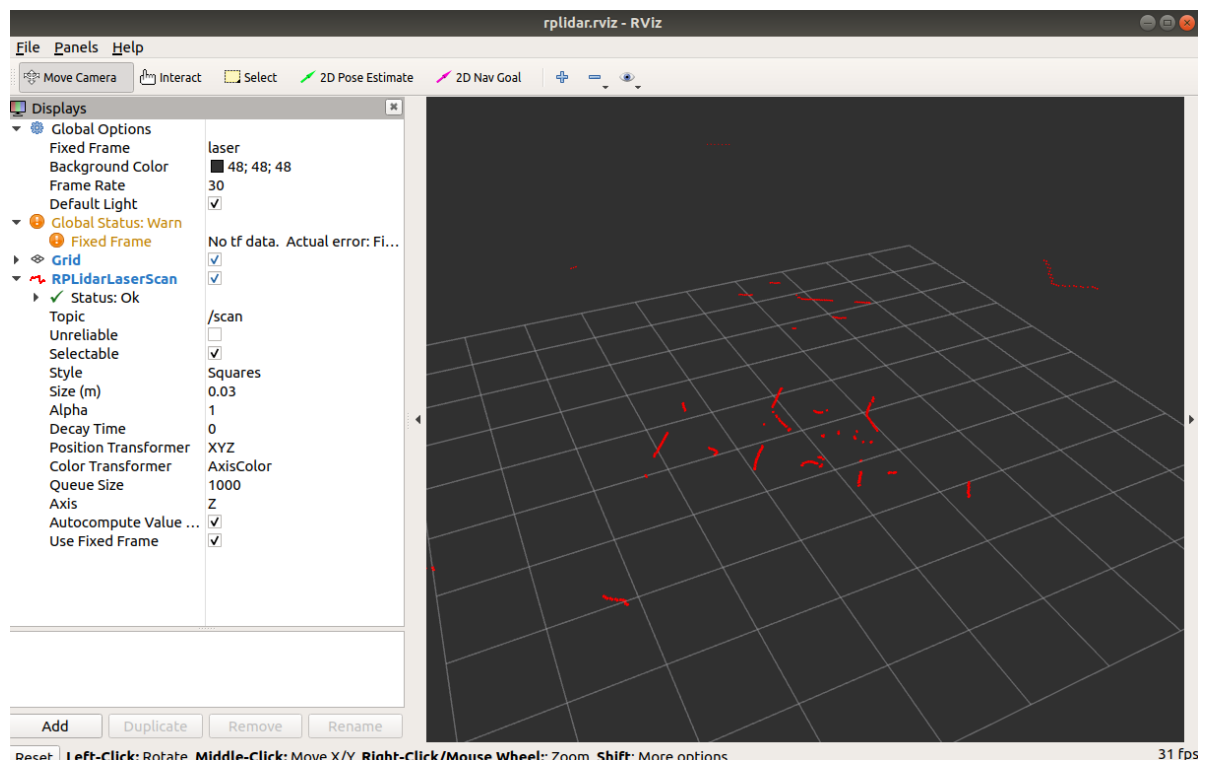
```
yahboom@ubuntu:~$ ll /dev/rplidar
lrwxrwxrwx 1 root root 7 Oct 31 20:37 /dev/rplidar -> ttyUSB0
yahboom@ubuntu:~$
```

The above content indicates that the binding is successful. The end is not necessarily 0 and changes according to the order in which the devices are inserted.

3. Test Lidar

Reopen a terminal, input following command to open the radar and display it in `rviz`.

```
roslaunch rplidar_ros view_rplidar_c1.launch
```



When the above screen appears, it means that all preparations have been completed.