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
```
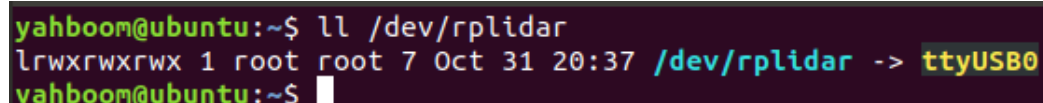
Copy the following content to the end of the file.

```
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 lidar wiring and enter the command ll /dev/rplidar in the terminal.

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 lidar and display it in rviz.

```
#a1 lidar
roslaunch rplidar_ros view_rplidar_a1.launch
#a2 lidar
roslaunch rplidar_ros view_rplidar_a2m12.launch
#a3 lidar
roslaunch rplidar_ros view_rplidar_a3.launch
#s2 lidar/s2l lidar
roslaunch rplidar_ros view_rplidar_s2.launch
#c1 lidar
roslaunch rplidar_ros view_rplidar_c1.launch
#s3
roslaunch rplidar_ros view_rplidar_s3.launch
```
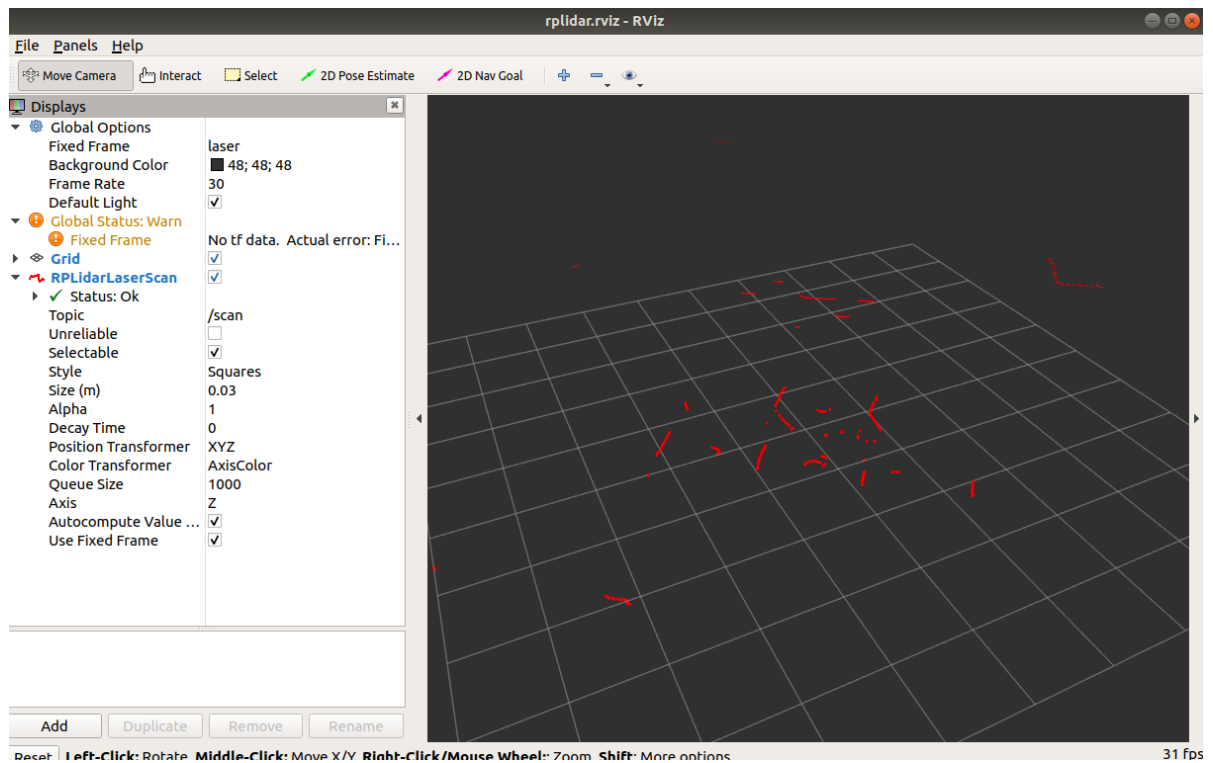

When the above screen appears, it means that all preparations have been completed.