1. Radar basics

- For different models of radar

```
#Raspberry Pi 5 master needs to enter docker first, please perform this step
#If running the script into docker fails, please refer to ROS/07, Docker tutorial
~/run_docker.sh
```

Before using lidar, you need to declare the [RPLIDAR_TYPE] variable in advance in the [.bashrc] file according to different radar models. Open the [.bashrc] file

```
sudo vim ~/.bashrc
```

If there is no sentence below in [.bashrc], you need to add it manually according to the purchased radar model. If there is this sentence, modify the radar model directly. For example: 4ROS lidar

```
export RPLIDAR_TYPE=4ROS # a1, a2, a3, s1, s2，4ROS，X3
```

Note: For rosmaster series cars equipped with X3 or X3Pro radar, the startup methods of the two radars are the same, just change them to X3.

After modification, refresh the environment variables

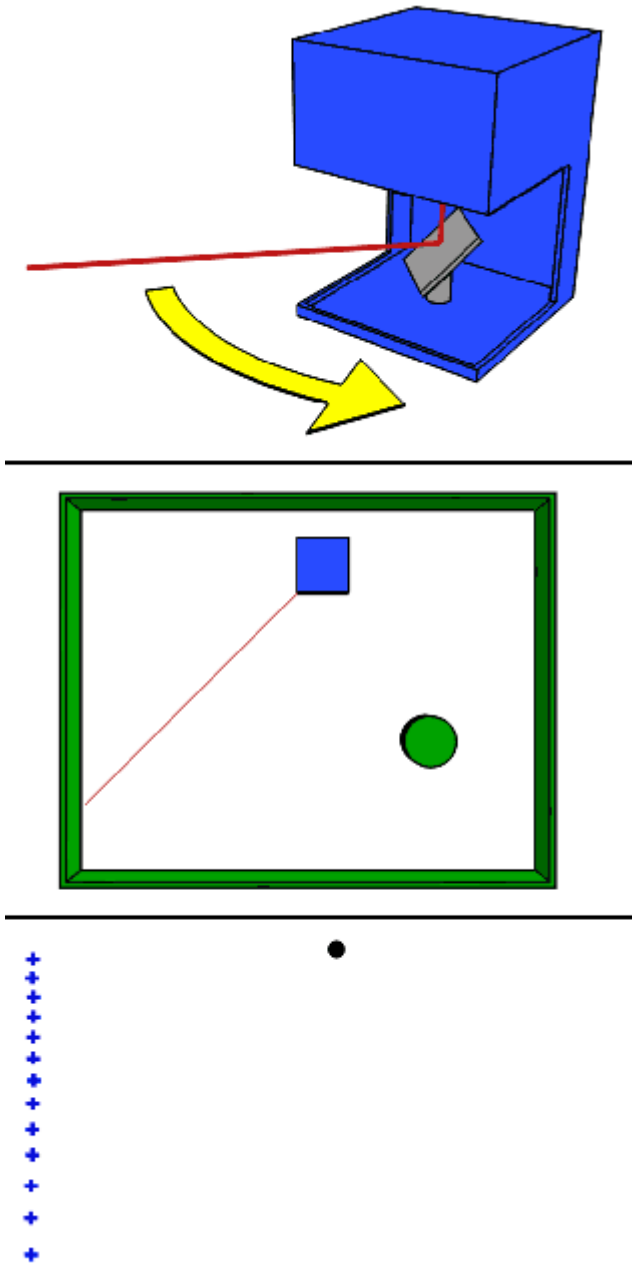
```
source ~/.bashrc
```

1.1. Overview

Single-line lidar refers to a radar whose line beam emitted by the laser source is a single line. It can be divided into triangular ranging and TOF lidar. It is mainly used in the field of robotics. It has fast scanning speed, strong resolution and high reliability. Compared with multi-line lidar, single-line lidar responds faster in angular frequency and sensitivity, so it is more accurate in measuring distance and accuracy of obstacles.

1.2. Principle of single-line lidar

The working principle of the single-wire mechanical rotating radar is as shown in the figure below:

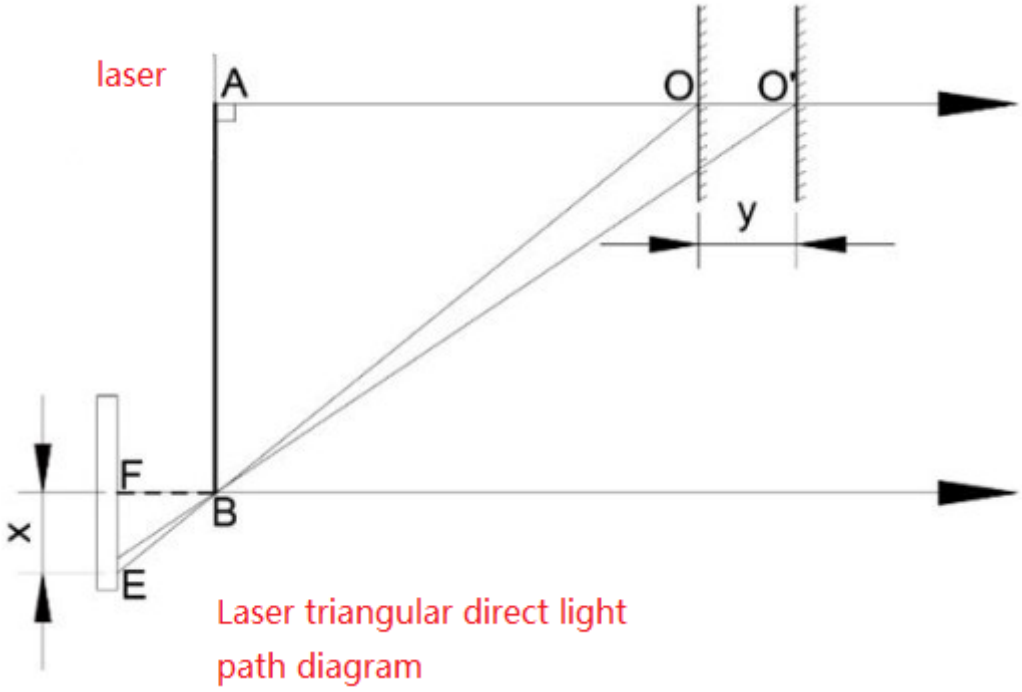


1.2.1. Trigonometric ranging method

The laser triangulation ranging method mainly uses a beam of laser to illuminate the measured target at a certain incident angle. The laser is reflected and scattered on the target surface. At another angle, a lens is used to converge and image the reflected laser. The spot is imaged on the CCD (Charge-coupled Device, photosensitive coupling component) on the position sensor. When the measured object moves along the direction of the laser, the light spot on the position sensor will move, and its displacement corresponds to the movement distance of the measured object. Therefore, the distance between the measured object and the baseline can be calculated from the light spot displacement distance through algorithm design. value. Since the incident light and the reflected light form a triangle, the geometric triangle theorem is used to calculate the spot displacement, so this measurement method is called the laser triangulation ranging method.

According to the angular relationship between the incident beam and the normal line of the surface of the measured object, the laser triangulation ranging method can be divided into two types: oblique type and direct type.

1. Direct type

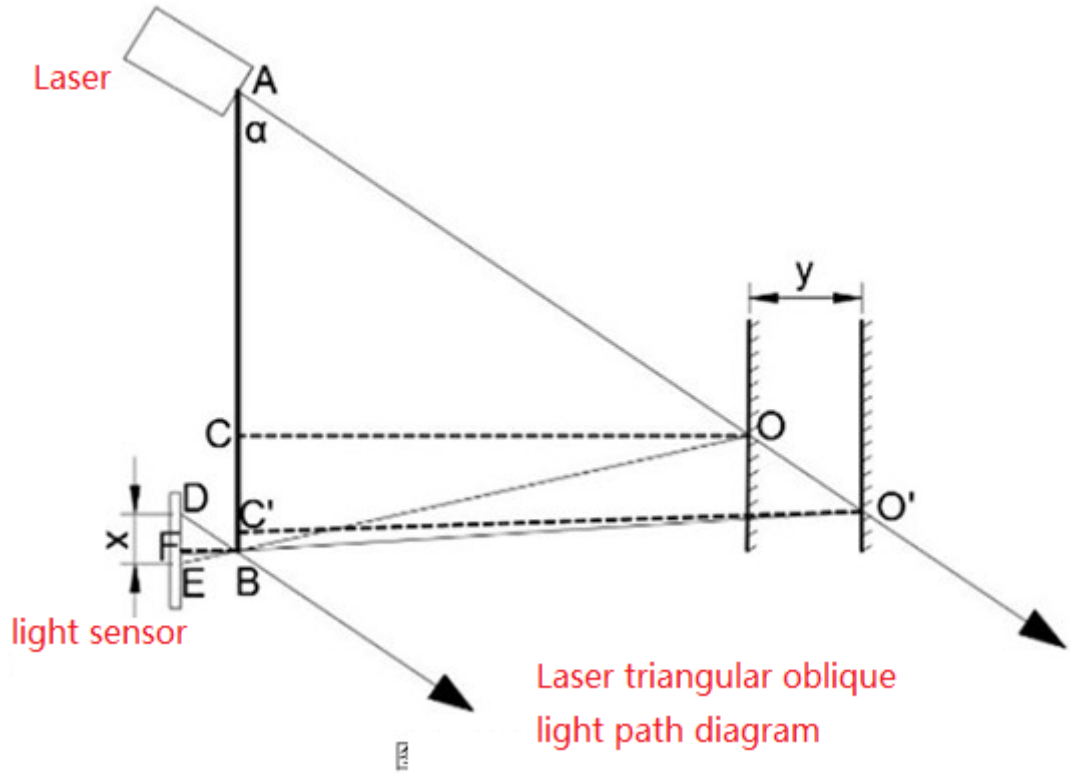


Laser triangular direct light path diagram

As shown in Figure 1, when the laser beam is vertically incident on the surface of the object to be measured, that is, when the incident light is collinear with the normal line of the surface of the object to be measured, it is a direct laser triangulation method.

2. Oblique shot type

When the angle between the incident laser beam and the normal line of the surface of the object being measured is less than 90° in the optical path system, the incident mode is oblique. The optical path diagram shown in Figure 2 is a laser triangulation oblique optical path diagram.



Laser triangular oblique light path diagram

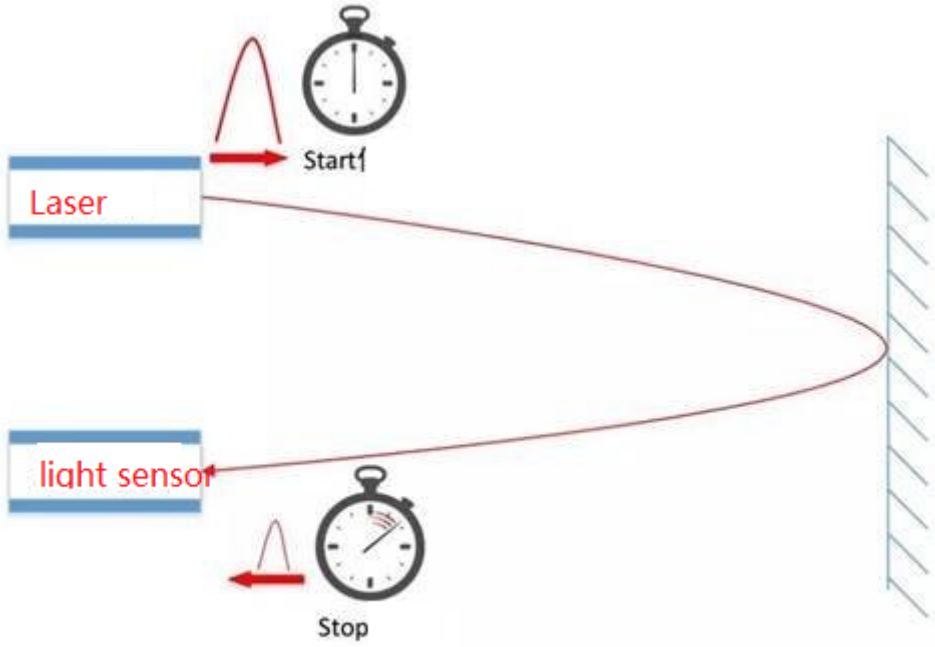
The laser emitted by the laser is incident on the surface of the object being measured at a certain angle with the normal line of the object surface. The reflected (scattered) light is concentrated through the lens at B and is finally collected by the photosensitive unit.

Whether it is the direct or oblique laser triangulation ranging method, it can achieve high-precision, non-contact measurement of the measured object, but the resolution of the direct type is not as high as that of the oblique type.

Silan Technology's RPLIDAR series lidar also uses the oblique laser triangulation ranging method. During each ranging process, the RPLIDAR series lidar will emit a modulated infrared laser signal. The reflection generated by the laser signal after hitting the target object will be received by the RPLIDAR visual acquisition system, and then processed by the DSP embedded inside the RPLIDAR. The device solves the problem in real time, and the distance value between the illuminated target object and the RPLIDAR and the current angle information will be output from the communication interface. Driven by the motor mechanism, the ranging core of RPLIDAR will rotate clockwise, thereby achieving 360-degree all-round scanning and ranging detection of the surrounding environment.

1.2.2. TOF time-of-flight ranging method

TOF lidar is based on measuring the flight time of light to obtain the distance of the target. Its working principle is mainly as follows: a modulated laser signal is emitted through a laser transmitter. The modulated light is received by the laser detector after being reflected by the object being measured. The distance to the target can be calculated by measuring the phase difference between the emitted laser and the received laser. .



Working principle diagram of TOF laser

Under the condition of distant objects, its measurement accuracy remains accurate and stable. At the same time, TOF radar is not inferior in its ability to resist light interference due to its ultra-short light pulse characteristics. It can achieve stable ranging and high-precision mapping even under strong light of 60Klx outdoors.

Generally speaking, triangular ranging lidar and TOF lidar have their own difficulties in implementation. In principle, TOF radar has a longer ranging distance. In some occasions where distance is required, TOF radar is the most common, while The manufacturing cost of triangular ranging lidar is relatively low, and its accuracy can meet most industrial-grade civilian

requirements, so it has also attracted much attention in the industry.

1.3. Use radar

1.3.1, 4ROS radar:

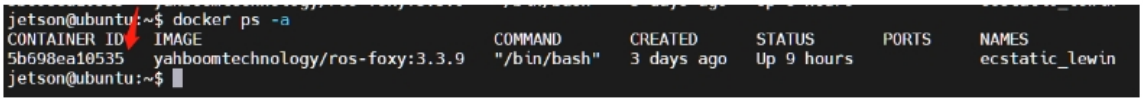
Terminal run

```
roslaunch ydlidar_ros_driver TG.launch
```

<PI5 needs to open another terminal to enter the same docker container

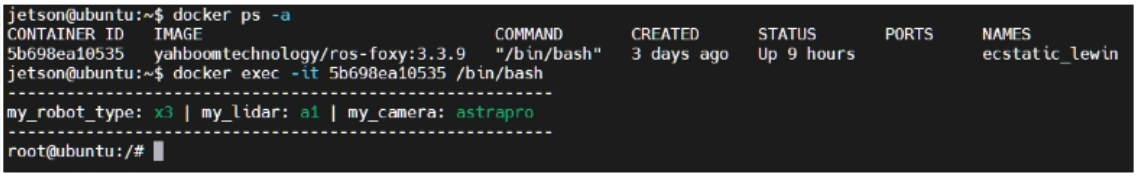
1. In the above steps, a docker container has been opened. You can open another terminal on the host (car) to view:

```
docker ps -a
```



2. Now enter the docker container in the newly opened terminal:

```
docker exec -it 5b698ea10535 /bin/bash
```



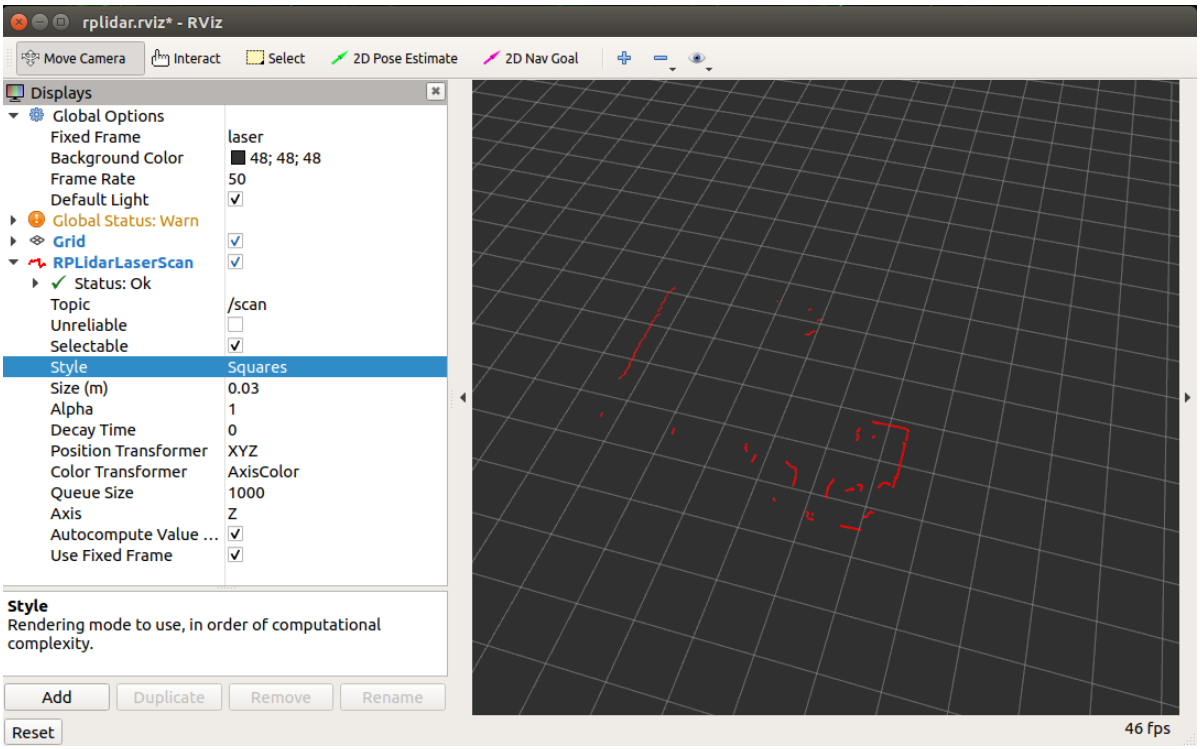
After successfully entering the container, you can open countless terminals to enter the container.

We can print the topic data through the terminal to check whether the radar starts normally, enter the terminal,

```
rostopic echo /scan
```

If you want to view the scan results in RVIZ, enter in the terminal,

```
roslaunch ydlidar_ros_driver lidar_view.launch
```



1.3.2, X3/X3Pro radar:

Terminal run

```
roslaunch ydlidar_ros_driver X2.launch
```

<PI5 needs to open another terminal to enter the same docker container

1. In the above steps, a docker container has been opened. You can open another terminal on the host (car) to view:

```
docker ps -a
```

```
jetson@ubuntu:~$ docker ps -a
CONTAINER ID   IMAGE                                COMMAND       CREATED      STATUS       PORTS     NAMES
5b698ea10535   yahboomtechnology/ros-foxy:3.3.9   "/bin/bash"   3 days ago   Up 9 hours             ecstatic_lewin
jetson@ubuntu:~$
```

2. Now enter the docker container in the newly opened terminal:

```
docker exec -it 5b698ea10535 /bin/bash
```

```
jetson@ubuntu:~$ docker ps -a
CONTAINER ID   IMAGE                                COMMAND       CREATED      STATUS       PORTS     NAMES
5b698ea10535   yahboomtechnology/ros-foxy:3.3.9   "/bin/bash"   3 days ago   Up 9 hours             ecstatic_lewin
jetson@ubuntu:~$ docker exec -it 5b698ea10535 /bin/bash
-------------------------------------------------------
my_robot_type: x3 | my_lidar: a1 | my_camera: astrapro
-------------------------------------------------------
root@ubuntu:/#
```

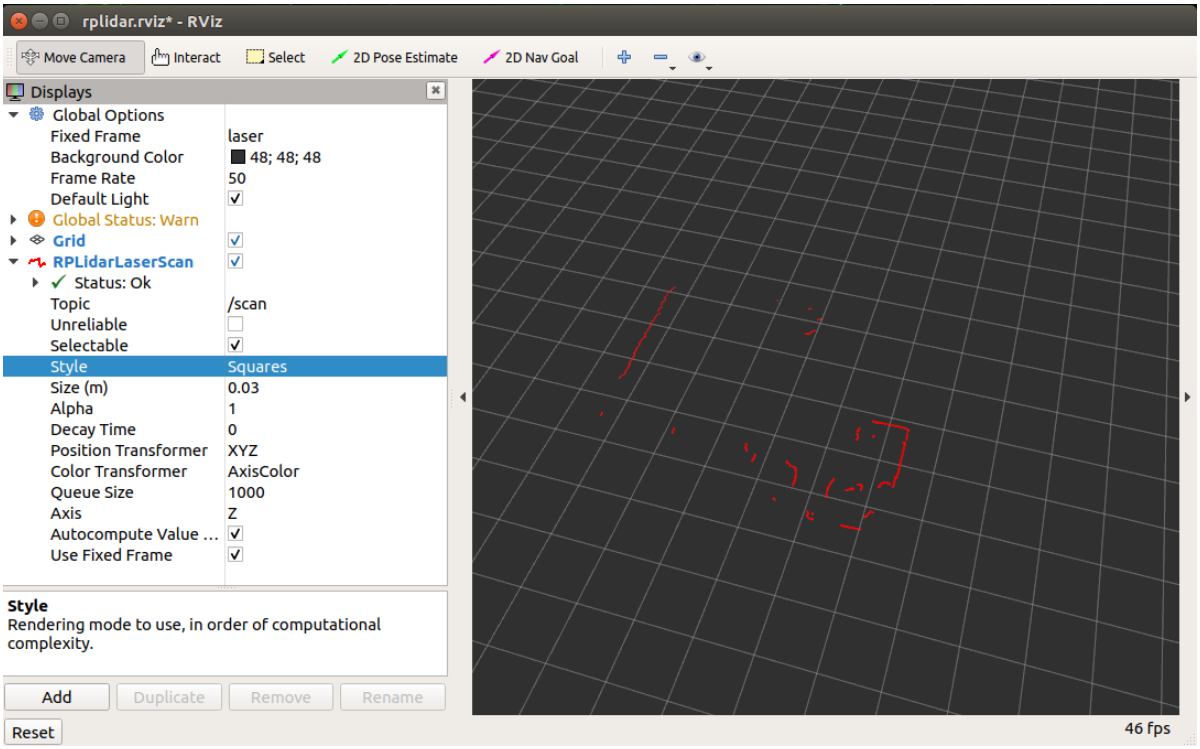
After successfully entering the container, you can open countless terminals to enter the container.

We can print the topic data through the terminal to check whether the radar starts normally, enter the terminal,

```
rostopic echo /scan
```

If you want to view the scan results in RVIZ, enter in the terminal,

```
roslaunch ydlidar_ros_driver lidar_view.launch
```



1.4, launch analysis

Path: ~/software/library_ws/src/ydlidar_ros_driver-master/launch

TG.launch file

```
<launch>
   <arg name="frame_id" default="laser"/>
   <node name="ydlidar_lidar_publisher" pkg="ydlidar_ros_driver"
type="ydlidar_ros_driver_node" output="screen" respawn="false" >
     <!-- string property -->
     <param name="port" type="string" value="/dev/ydlidar"/>
     <param name="frame_id" type="string" value="$(arg frame_id)"/>
     <!--param name="ignore_array" type="string" value="-90,90"/-->
     <param name="ignore_array" type="string" value=""/>
     <!--remap from="scan" to="scan_raw"/-->

     <!-- int property -->
     <param name="baudrate" type="int" value="512000"/>
     <!-- 0:TYPE_TOF, 1:TYPE_TRIANGLE, 2:TYPE_TOF_NET -->
     <param name="lidar_type" type="int" value="0"/>
     <!-- 0:YDLIDAR_TYPE_SERIAL, 1:YDLIDAR_TYPE_TCP -->
     <param name="device_type" type="int" value="0"/>
     <param name="sample_rate" type="int" value="20"/>
     <param name="abnormal_check_count" type="int" value="4"/>

     <!-- bool property -->
     <param name="resolution_fixed" type="bool" value="true"/>
     <param name="auto_reconnect" type="bool" value="true"/>
     <param name="reversion" type="bool" value="true"/>
     <param name="inverted" type="bool" value="true"/>
     <param name="isSingleChannel" type="bool" value="false"/>
```

```
    <param name="intensity" type="bool" value="false"/>
    <param name="support_motor_dtr" type="bool" value="false"/>
    <param name="invalid_range_is_inf" type="bool" value="true"/>
    <param name="point_cloud_preservative" type="bool" value="false"/>

    <!-- float property -->
    <param name="angle_min" type="double" value="-90" />
    <param name="angle_max" type="double" value="90" />
    <param name="range_min" type="double" value="0.01" />
    <param name="range_max" type="double" value="50.0" />
    <param name="frequency" type="double" value="10.0"/>
  </node>
  <!--node pkg="tf" type="static_transform_publisher" name="base_link_to_laser4"
    args="0.0 0.0 0.2 3.14 0.0 0.0 /base_footprint /laser 40" /-->
</launch>
```

Main debugging parameters:

- angle_min parameter: radar left angle
- angle_max parameter: radar right angle

For other parameters, please refer to the official documentation.

ydlidar_ros_driver/README.md at master · YDLIDAR/ydlidar_ros_driver· GitHub