

EAI_X3 Radar ROS2

EAI_X3 Radar ROS2

- 1、Remap the USB serial port
- 2、terminal test
- 3、rviz2 visual test
- 4、gmapping mapping test

ROS 2 official website: <https://docs.ros.org/>

equipment: pc; X3

environment: Ubuntu20.04; ROS2 (Foxy)

new workspace

```
mkdir -p x3_ws/src
```

When using this radar, you need to enter the workspace every time you execute a command

```
cd ~/x3_ws
```

Unzip the [X3_ws_src.zip] function package and put it into the src folder of the X3_ws workspace, and open the terminal in the workspace

```
colcon build          # compile
source install/setup.bash # update environment
```

Note: Every time you modify the code in the function package, you need to [compile] and then [update environment].

Add the workspace to the .bashrc of the environment variable,

```
echo "source ~/x3_ws/install/setup.bash" >> ~/.bashrc
```

1、Remap the USB serial port

Terminal input,

```
cd ~/x3_ws/src/ydlidar_ros2_driver-master/startup
sudo chmod 777 *
sudo ./initenv.sh
```

After binding, unplug the radar again.

Use the following command to view modified remap

```
ll /dev/ydlidar
```

```
-----
yahboom@yahboom-virtual-machine:~$ ll /dev/ydlidar
lrwxrwxrwx 1 root root 7 Mar  4 11:12 /dev/ydlidar -> ttyUSB0
yahboom@yahboom-virtual-machine:~$
```

Note: This is not necessarily ttyUSB0, as long as there is /dev/ydlidar, it means that the binding is successful.

2、terminal test

The first step is to start the corresponding radar

```
ros2 launch ydlidar_ros2_driver ydlidar_launch.py
```

The second step is to view the radar data

```
ros2 topic echo /scan
```

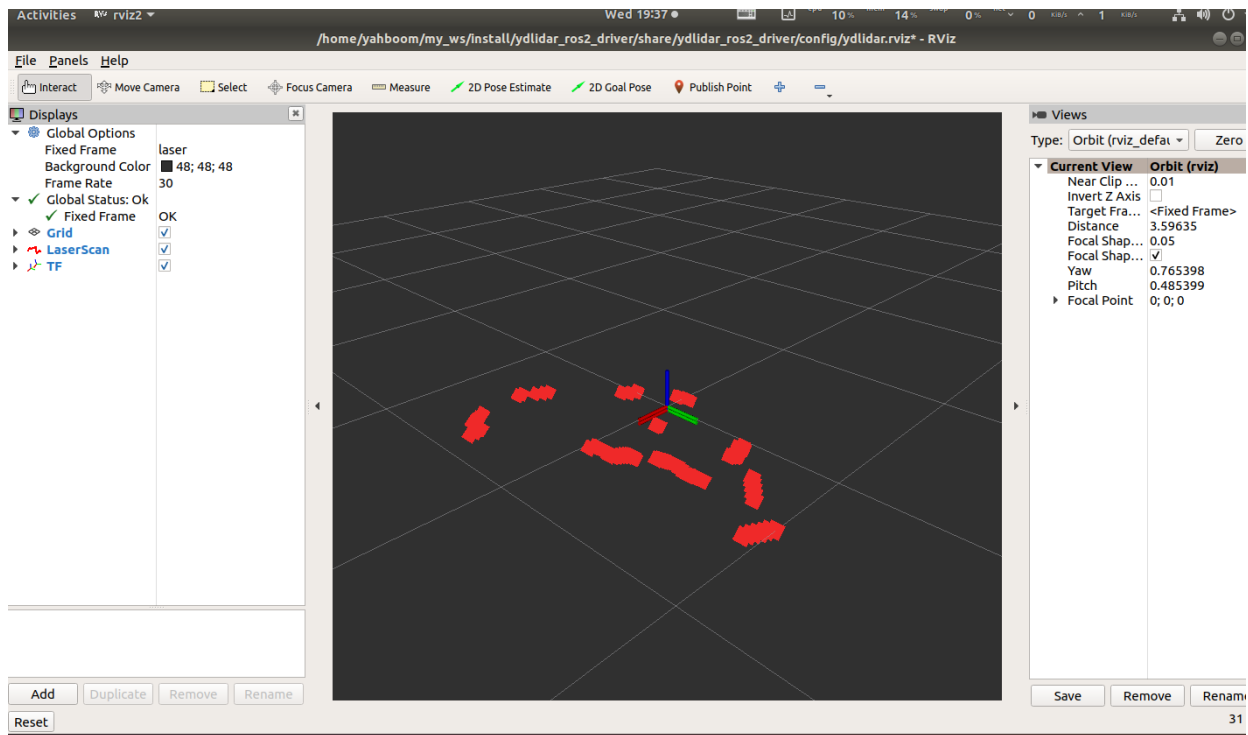
You should see rplidar scan results in the console

```
---
header:
  stamp:
    sec: 1677913201
    nanosec: 333386000
  frame_id: laser
angle_min: -3.1415927410125732
angle_max: 3.1415927410125732
angle_increment: 0.02444819174706936
time_increment: 0.0003327042795717716
scan_time: 0.08550500124692917
range_min: 0.009999999776482582
range_max: 64.0
ranges:
- 0.0
- 0.1237500011920929
- 0.13075000047683716
- 0.0
- 0.13875000178813934
- 0.14775000512599945
- 0.0
- 0.1577499955892563
- 0.17274999618530273
- 0.18774999678134918
- 0.0
- 0.20274999737739563
- 0.22474999725818634
- 0.0
- 0.2527500092983246
- 0.0
- 0.0
- 0.45274999737739563
- 0.0
- 0.0
- 0.7310000061988831
- 0.7170000076293945
- 0.7139999866485596
- 0.6959999799728394
- 0.6997500061988831
- 0.7350000143051147
- 0.7410000037875366
```

3、rviz2 visual test

```
ros2 launch ydlidar_ros2_driver ydlidar_launch_view.py
```

You should see the radar scan results in rviz2



Note: The topic of [Fixed Frame] coordinate system and [LaserScan] should be consistent with the published ones.

4、gmapping mapping test

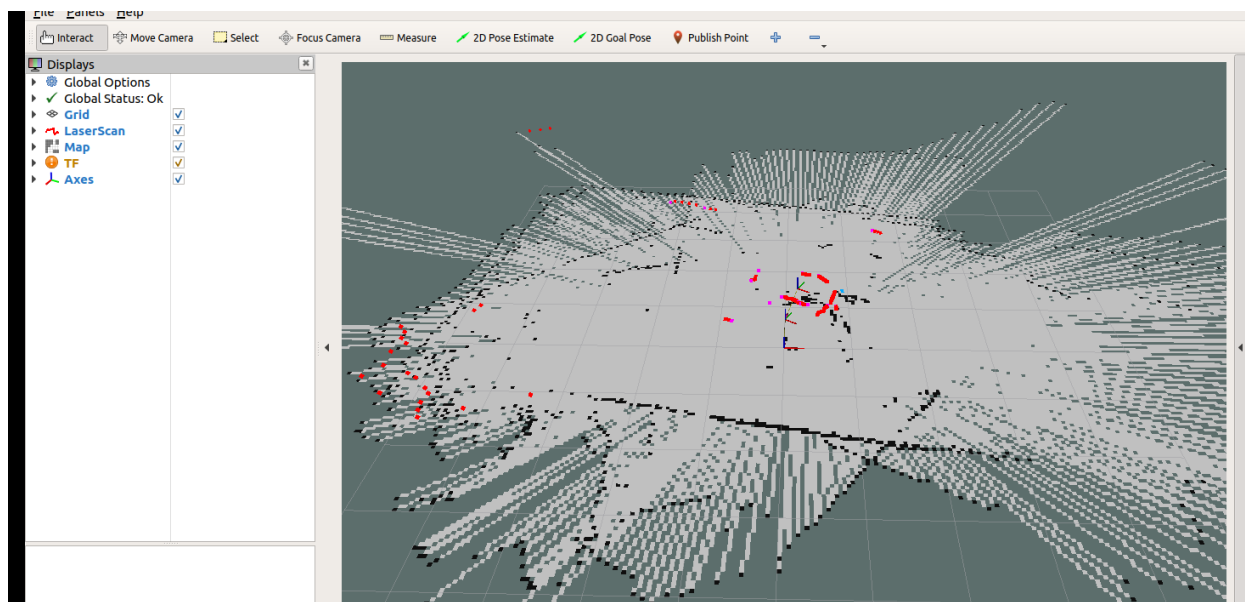
The first step is to start the corresponding radar

```
ros2 launch ydlidar_ros2_driver ydlidar_launch.py
```

The second step is to start gmapping to build a map

```
ros2 launch sllidar_ros2 gmapping_launch.py
ros2 run slam_gmapping slam_gmapping
```

You should see the mapping result in rviz2



Note: [Fixed Frame] coordinate system, [LaserScan] topic, and [map] topic should be consistent with the published ones.