

## 5. Voice-controlled multi-point navigation

This course requires the combination of Rosmaster-X3 car hardware, and only code analysis is done here. First, let's look at the built-in voice commands,

Function words	Speech recognition module results	Voice broadcast content
Navigate to location 1	19	OK, heading to location 1
Navigate to location 2	20	OK, heading to location 2
Navigate to No. 3	21	OK, heading to No. 3
Navigate to No. 4	32	OK, heading to No. 4
Return to origin	33	OK, returning to origin

### 1. Program startup

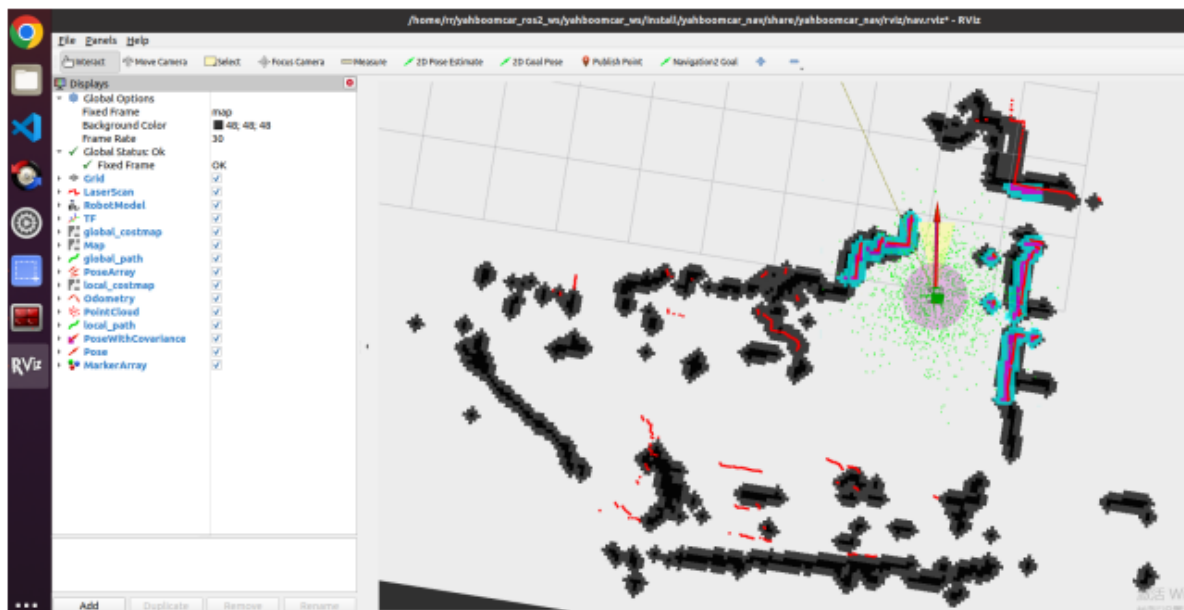
#### 1.1. Calibrate the target point

Terminal input,

```
ros2 launch yahboomcar_nav laser_bringup_launch.py
ros2 launch yahboomcar_nav display_nav_launch.py
ros2 launch yahboomcar_nav navigation_teb_launch.py
```

Click [2D Pose in the rviz interface of the virtual machine Estimate ], and then compare the car's position and mark an initial position on the map;

The display after marking is as follows:



Compare the overlap of the radar scanning point and the obstacle, you can set the initial position of the car multiple times until the radar scanning point and the obstacle roughly overlap;

Terminal input,

```
ros2 topic echo /goal_pose
```

Click [2D Goal Pose], set the first navigation target point, then the car starts navigating, and the terminal will print out the topic data:

```
root@ubuntu:~/yahboomcar_ros2_ws/yahboomcar_ws# ros2 topic echo /goal_pose
header:
  stamp:
    sec: 1682416565
    nanosec: 174762965
  frame_id: map
pose:
  position:
    x: -7.258232593536377
    y: -2.095078229904175
    z: 0.0
  orientation:
    x: 0.0
    y: 0.0
    z: -0.3184907749129588
    w: 0.9479259603446585
```

## 1.2、 Write the target point position program

Edit the voice\_ctrl\_send\_mark.py file, which is located at:

~/driver\_ws/src/yahboomcar\_voice\_ctrl/yahboomcar\_voice\_ctrl/voice\_ctrl\_send\_mark.py

Modify the pose of the first navigation point to the one printed in the terminal:

```
pose.pose.position.x = 2.15381097794
pose.pose.position.y = -5.02386903763
pose.pose.orientation.z = 0.726492681307
pose.pose.orientation.w = 0.687174202082
```

The marking results of the other points are also marked in rviz according to the above steps, and the coordinates of the pose points are recorded. Then modify to the corresponding position.

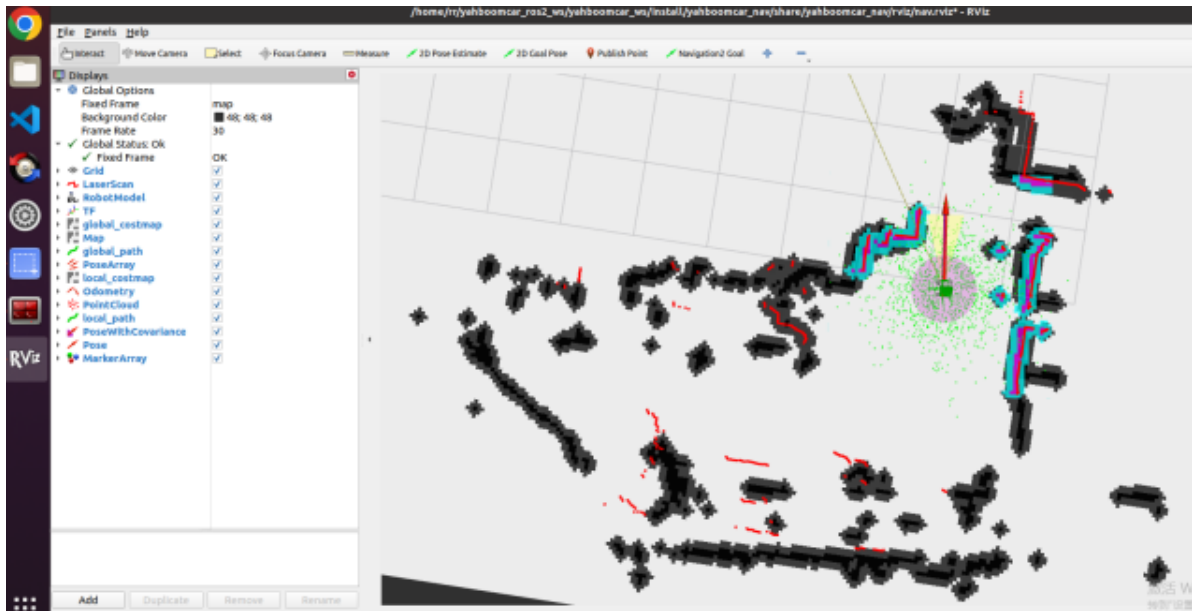
## 1.3, Voice Navigation

Terminal input,

```
ros2 launch yahboomcar_nav laser_bringup_launch.py
ros2 launch yahboomcar_nav display_nav_launch.py
ros2 launch yahboomcar_nav navigation_teb_launch.py
```

At this time, click [2D Pose Estimate] in the rviz interface of the virtual machine, and then compare the position of the car to mark an initial position of the car on the map;

The display after marking is as follows:



Compare the overlap of the radar scanning point and the obstacle, you can set the initial position of the car multiple times until the radar scanning point and the obstacle roughly overlap;

Terminal input,

```
ros2 run yahboomcar_voice_ctrl voice_ctrl_send_mark
```

Say "Hi Yahboom" to the voice module on the car to wake up the voice module. After hearing the voice module's feedback "Yes", continue to say "Navigate to position 1"; the voice module will feedback "OK, going to position 1", and the car will start navigating to position 1. Navigation at other locations can be used in the same way.

## 2. Code analysis

```
#导入语音库 Importing Voice Library
from Speech_Lib import Speech
#创建目标点话题发布者 Create a target topic publisher
self.pub_goal = self.create_publisher(PoseStamped, "/goal_pose", 1)
def voice_pub_goal(self):
    self.pose.header.frame_id = 'map'
    #获取语音指令 Get voice commands
    speech_r = self.spe.speech_read()
    # print("-----speech_r = ",speech_r)
    if speech_r == 19:
        print("goal to one")
        self.spe.void_write(speech_r)
        self.pose.header.stamp = Clock().now().to_msg()
        self.pose.pose.position.x = -7.1171722412109375
        self.pose.pose.position.y = -3.8613715171813965
        self.pose.pose.orientation.z = -0.6484729569092691
        self.pose.pose.orientation.w = 0.7612376922862854
        #发布目标点话题数据 Publish target point topic data
        self.pub_goal.publish(self.pose)
    elif speech_r == 20:
        print("goal to two")
        self.spe.void_write(speech_r)
        self.pose.header.stamp = Clock().now().to_msg()
        self.pose.pose.position.x = -5.434411525726318
        self.pose.pose.position.y = -3.575838088989258
```

```
self.pose.pose.orientation.z = 0.041131907433507836  
self.pose.pose.orientation.w = 0.9991537250047569  
self.pub_goal.publish(self.pose)
```