

## 5、Voice controlled multi-point navigation

This course needs to be combined with the hardware of the Rosmaster-X3 car, and only code analysis will be done here. Firstly, let's take a look at the built-in voice commands ,

functional word	Speech recognition results	Voice broadcast content
Go to the point A	19	OK, I'm going to the point A.
Go to the point B	20	OK, I'm going to the point B.
Go to the point C	21	OK,I'm going to the point C.
Go to the point D	32	OK, I'm going to the point D.
Back to origianl point	32	OK, I'm return back.

### 1、Program startup

#### 1.1、Calibration target point

Terminal input,

```
roslaunch yahboomcar_nav laser_bringup.launch #laser + yahboomcar
roslaunch yahboomcar_nav yahboomcar_navigation.launch use_rviz:=false map:=house
#Enable navigation, change house to the name of the map where the map was created
```

【use\_rviz】 Parameter: Whether to turn on rviz.

【map】 Parameter: Map name, the map to load.

Enable visualization interface (virtual machine side)

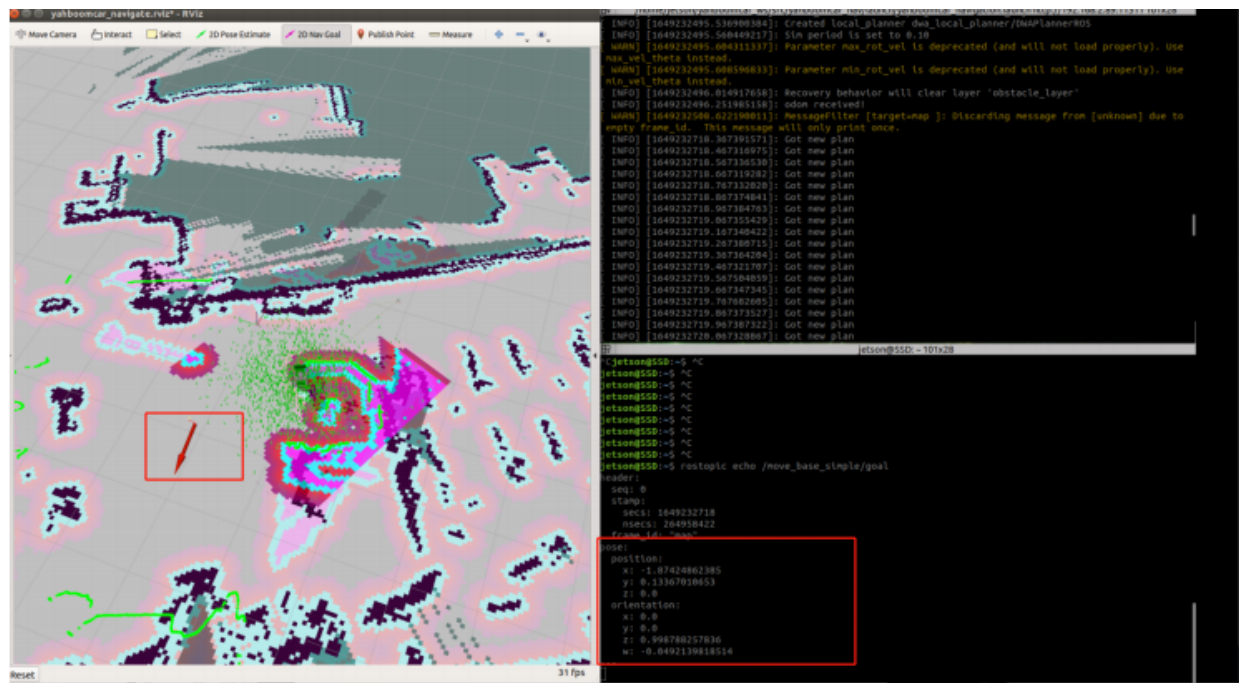
```
roslaunch yahboomcar_nav view_navigate.launch
```

1) 、 Adjust the initial pose of ROSMASTER in the map rviz;

2) 、 Terminal input

```
rostopic echo /move_base_simple/goal
```

3) 、 Using the 2D Nav Goal tool in rviz, give the car a target point in rviz and record it as position 1. At this point, check/move just now\_ Base\_ The terminal window of simple/global data will print the coordinates of the target point, as shown in the following figure,



Just record the post part of the data and manually send it out through the program later.

## 1.2、Target point position writer

Open `~/yahboomcar/src/yahboomcar_Voice_Ctrl/scripts/voice_Ctrl_Send_Mark.py`, modify the previously recorded pose data to the corresponding location,

```
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 several other points are also calibrated in the 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,

```
roslaunch yahboomcar_nav laser_bringup.launch #laser + yahboomcar
roslaunch yahboomcar_nav yahboomcar_navigation.launch use_rviz:=false map:=house
#Enable navigation, change house to the name of the map where the map was created
python ~/yahboomcar_ws/src/yahboomcar_voice_ctrl/scripts/voice_ctrl_send_mark.py
```

After calibrating the initial pose in rviz, say "Hi Yahboom" to ROSMASTER, wake up the voice module, hear the voice module feedback broadcast "Hi, I'm here", and then say "Go to the point A" to ROSMASTER; the voice module will feedback broadcast "OK, I'm going to the point A". The other positions are also the same, as long as the coordinate values of the destination of other points are written in the program.

## 2、Code parsing

code path: ~/driver\_ws/src/yahboomcar\_voice\_ctrl/scripts/voice\_Ctrl\_send\_mark.py

```
#Create target point publisher to publish target pose
pub_goal = rospy.Publisher('move_base_simple/goal', PoseStamped, queue_size=1)
#Packaging target point data
speech_r = spe.speech_read()
if speech_r == 19 :
    print("goal to one")
    spe.void_write(speech_r)
    pose.pose.position.x = 2.15381097794
    pose.pose.position.y = -5.02386903763
    pose.pose.orientation.z = 0.726492681307
    pose.pose.orientation.w = 0.687174202082
#Publish coordinate point pose
pub_goal.publish(pose)
```