# 7. Voice control multi-point navigation



## 7.1. Description

On the established map, voice control ROSMASTER to navigate to point1, point2, point3. The R2 key on the handle can stop/start this function at any time.

## 7.2. Steps

### 7.2.1. Function package path

```
~/yahboomcar/src/yahboomcar_voice_ctrl/
```

### 7.2.2. Calibrate the target point

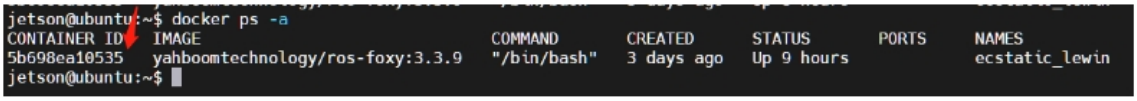
robot side

```
#You need to enter docker first, perform this step more
#If running the script to enter docker fails, please refer to 07.Docker-orin/05,
Enter the robot's docker container
~/run_docker.sh
roslaunch yahboomcar_nav laser_bringup.launch  #laser + yahboomcar
```

**<Open another terminal and 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.

```
roslaunch yahboomcar_nav yahboomcar_navigation.launch use_rviz:=false map:=house
# start navigation, change house to the map name of the map
```

[use_rviz] parameter: whether to open rviz.

[map] Parameters: map name, the map to be loaded.

Open the visual interface (virtual machine side)

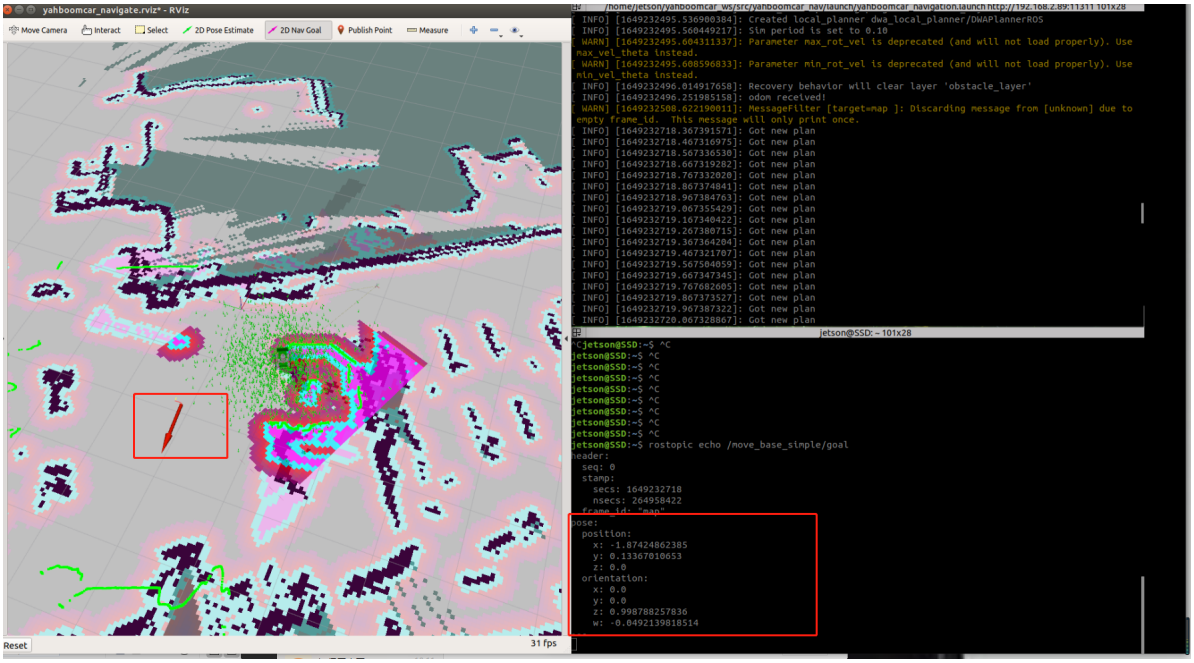
```
roslaunch yahboomcar_nav view_navigate.launch
```

1) In the map rviz, adjust the initial posture of ROSMASTER;

2. terminal input

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

3. Using the 2D Nav Goal tool in rviz, a target point of the car is given in rviz, which is recorded as position 1. At this time, the coordinates of the target point will be printed out in the terminal window that just viewed the data of/move_base_simple/goal. As shown below.




We only need to record the data for that part of the pose. Later, we need to send this part of the content manually through the program.

4. open ~/yahboomcar/src/yahboomcar_voice_ctrl/scripts/voice_Ctrl_send_mark.py, and modify the pose data just recorded 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
```

About the other two points B/C,  use the same steps.

### 7.2.2. Voice navigation

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

1. After calibrating the initial pose in rviz, we can say "Hi Yahboom" to wake up the voice module, until it  replies "Hi, i'm here", indicating that the module has been woken up.
2. We say "Go to the point A", it will reply "OK, I'm going to the point A."

## 7.3. Code analysis

### 7.3.1. For the principle of navigation, please refer to the tutorial "11. Lidar/12. Navigation and Obstacle Avoidance". This course is mainly to judge the results of voice recognition.

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
```

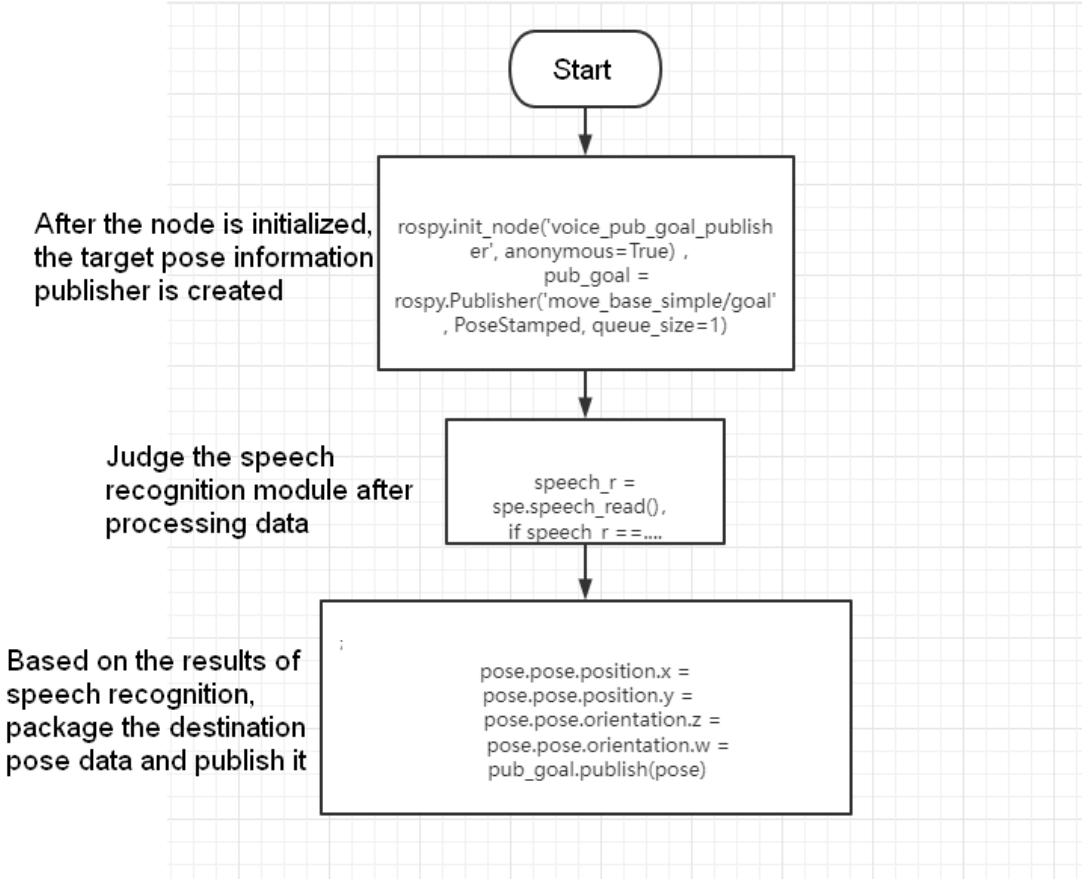
```
            pose.pose.position.y =  -5.02386903763
            pose.pose.orientation.z = 0.726492681307
            pose.pose.orientation.w = 0.687174202082
            pub_goal.publish(pose)
        elif speech_r == 20 :
            print("goal to tow")
            spe.void_write(speech_r)
            pose.pose.position.x = 1.57744419575
            pose.pose.position.y = 4.8174996376
            pose.pose.orientation.z = -0.683335654604
            pose.pose.orientation.w = 0.730104364558
            pub_goal.publish(pose)

        elif speech_r == 21 :
            print("goal to three")
            spe.void_write(speech_r)
            pose.pose.position.x =  -1.08106160164
            pose.pose.position.y = 1.30198049545
            pose.pose.orientation.z =  -0.0132771070267
            pose.pose.orientation.w =  0.99991185533
            pub_goal.publish(pose)

        elif speech_r == 32 :
            print("goal to four")
            spe.void_write(speech_r)
            pose.pose.position.x =  -1.08106160164
            pose.pose.position.y = 1.30198049545
            pose.pose.orientation.z =  -0.0132771070267
            pose.pose.orientation.w =  0.99991185533
            pub_goal.publish(pose)

        elif speech_r == 33 :
            print("goal to Origin")
            spe.void_write(speech_r)
            pose.pose.position.x =  -1.08106160164
            pose.pose.position.y = 1.30198049545
            pose.pose.orientation.z =  -0.0132771070267
            pose.pose.orientation.w =  0.99991185533
            pub_goal.publish(pose)
        elif speech_r == 0 :
            pub_cmdVel.publish(Twist())S
```

### 7.3.2. Flowchart




Code path:

```
~/yahboomcar/src/yahboomcar_voice_ctrl/scripts/voice_Ctrl_send_mark.py
```

### 7.3.3. Voice module communication protocol

| function word | Speech Recognition Module 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. |
| Return to the original place | 33 | OK, I'm return back. |