

7. Voice control multi-point navigation

7.1. Function description

By interacting with the voice recognition module on ROSMASTER, you can use your voice to let ROSMASTER navigate to point 1/2/3 on an established map. The R2 button on the handle can cancel/enable this function at any time. .

7.2. Start

```
#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
```

7.2.1. Function package path

```
~/yahboomcar_ws/src/yahboomcar_voice_ctrl/
```

7.2.2. Calibrate the target point for voice navigation

robot side

```
roslaunch yahboomcar_nav laser_bringup.launch #laser + yahboomcar
```

<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.

```
roslaunch yahboomcar_nav yahboomcar_navigation.launch use_rviz:=false map:=house
#Open navigation, change house to the map name for mapping
```

[use_rviz] Parameter: whether to open rviz.

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

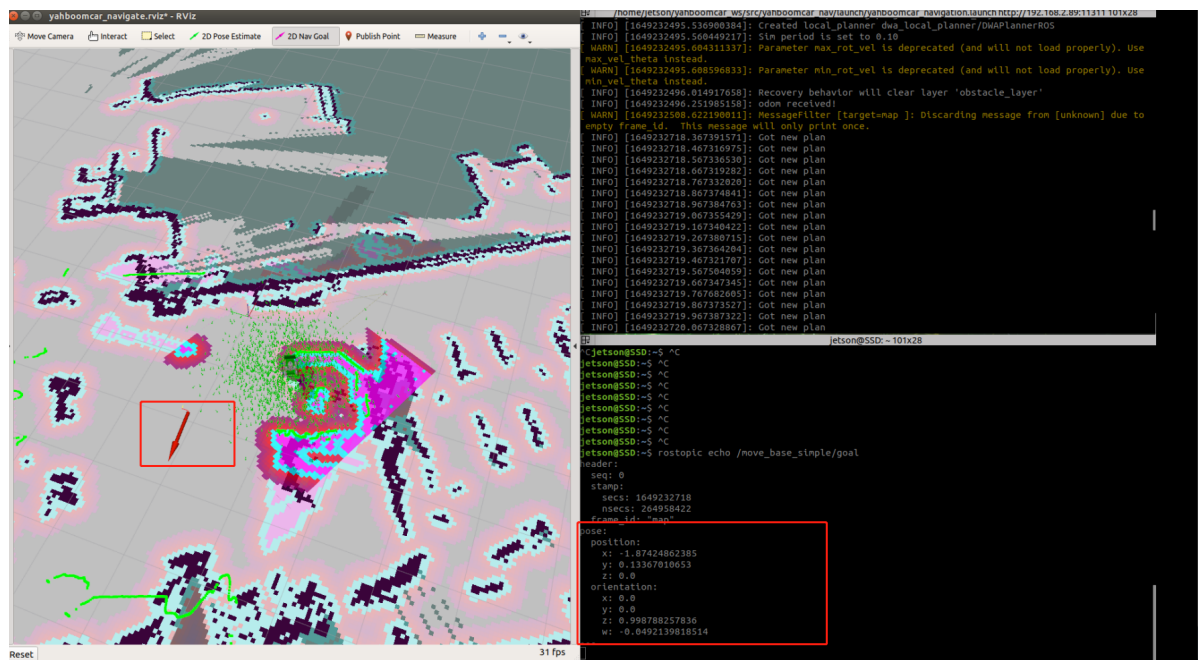
Turn on the visual interface (virtual machine side)

```
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. Use the 2D Nav Goal tool in rviz to give the car a target point in rviz, recorded as position 1. At this time, the coordinates of the target point will be printed out in the terminal window where you just viewed the data of /move_base_simple/goal. As shown below,



Just record the pose part of the data. Later, you need to send this part manually through the program.

4. Open `~/yahboomcar_ws/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
```

In addition to the two BC points, the position is also calibrated first, and then the calibrated value is modified into the function as an actual parameter.

7.2.2. Voice navigation

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

After calibrating the initial pose in rviz, say "Hello, Xiaoya" to ROSMASTER to wake up the voice module. After hearing the voice module's feedback broadcast of "is", say to ROSMASTER "Navigate to position 1"; the voice module There will be a feedback message saying "Okay, heading to location 1". The same goes for other locations, as long as the coordinate values of the destinations of other points are written in the program.

7.3. Core code analysis

7.3.1. For the principle of navigation, please refer to the tutorial " 11. Lidar \ 12. Navigation and Obstacle Avoidance " The content of this section is mainly about judging the speech recognition results and packaging and sending

target point data,

```
#Pack 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 show")
    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)

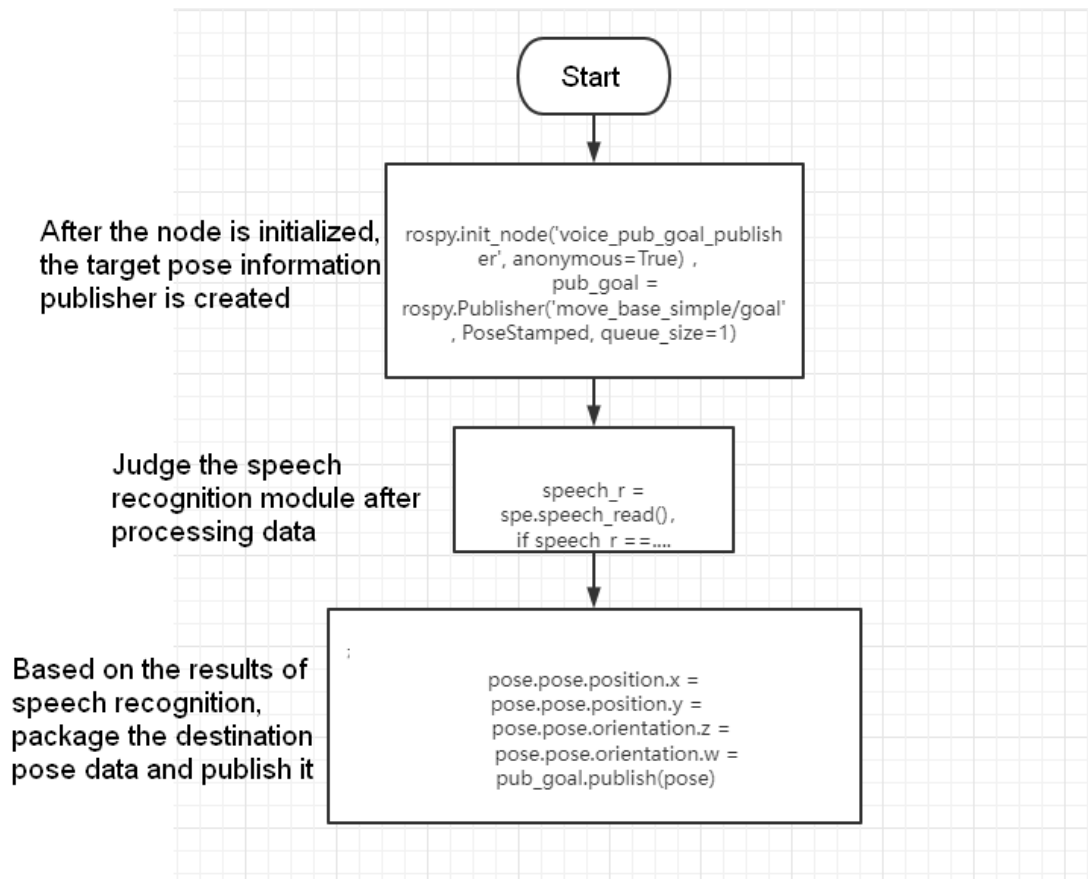
elif speech_r == 33 :
    print("goal to Origin")
```

```

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. Program flow chart



For complete code, please refer to:

[~/yahboomcar_ws/src/yahboomcar_voice_ctrl/scripts/voice_Ctrl_send_mark.py](#)

7.3.3. Function module communication table

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