13. APP Mapping and Navigation

13.1.1. start up

Note: 1. When building a map, the slower the speed, the better the effect (note that the rotation speed should be slower). If the speed is too fast, the effect will be poor.

2. Before building a map, you must first close the large program.

According to different vehicle models, you only need to set the purchased model in [. bashrc], such as X1 (ordinary four-wheel drive), X3 (wheel drive), X3plus (wheel mechanical arm), R2 (Ackermann differential), etc. This section takes X3 as an example

Open the【.bashrc】 file

```
sudo vim .bashrc
```

Find the [ROBOT_TYPE] parameters and modify the corresponding car model

```
export ROBOT_TYPE=X3   # ROBOT_TYPE: X1 X3 X3plus R2 X7
```

Start driver command (robot side),

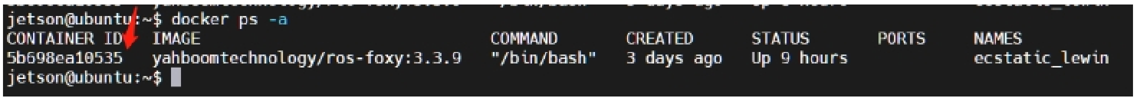
```
#You need to enter docker first, perform this step more
#If running the script to enter docker fails, please refer to 07.Docker-orin/05,
Enter the robot's docker container
~/run_docker.sh
roslaunch yahboomcar_nav laser_bringup.launch          # laser + yahboomcar
```

Start mapping function (robot side)

<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:

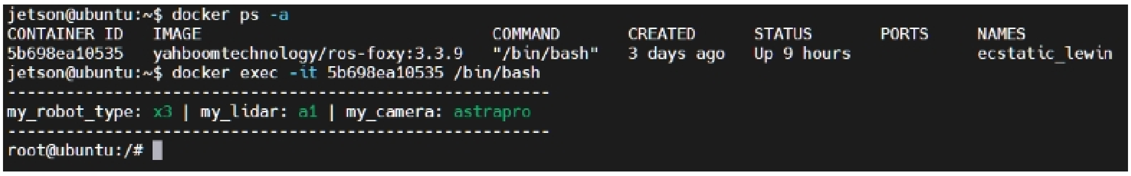
```
docker ps -a
```



```
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_map.launch use_rviz:=false
map_type:=gmapping
```

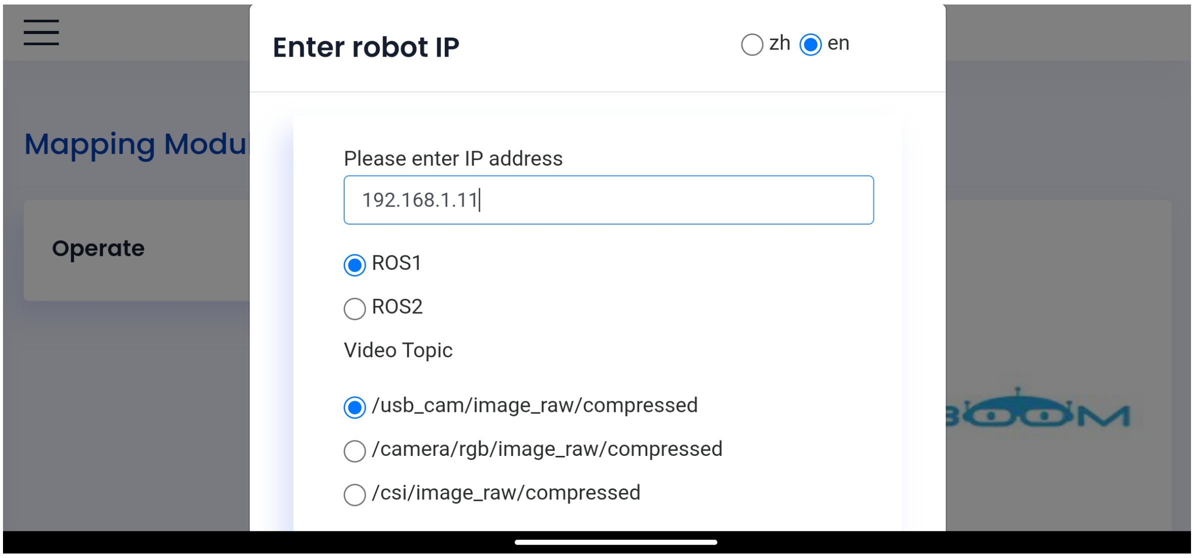
Start the radar data transfer to point cloud nodes and rosbridge and other nodes

```
source /home/jetson/software/laser_app/devel/setup.bash
roslaunch /home/jetson/software/laser_app/laser_app.launch
```

13.1.2. Use

After starting the robot, click APP

To ensure that the mobile phone and the robot are on the same network, enter the robot [IP] in the input box. ROS1 is selected by default, zh is selected as the default language, and Video Topic is ignored.



Enter robot IP

○ zh ● en

Please enter IP address

192.168.1.11

● ROS1

○ ROS2

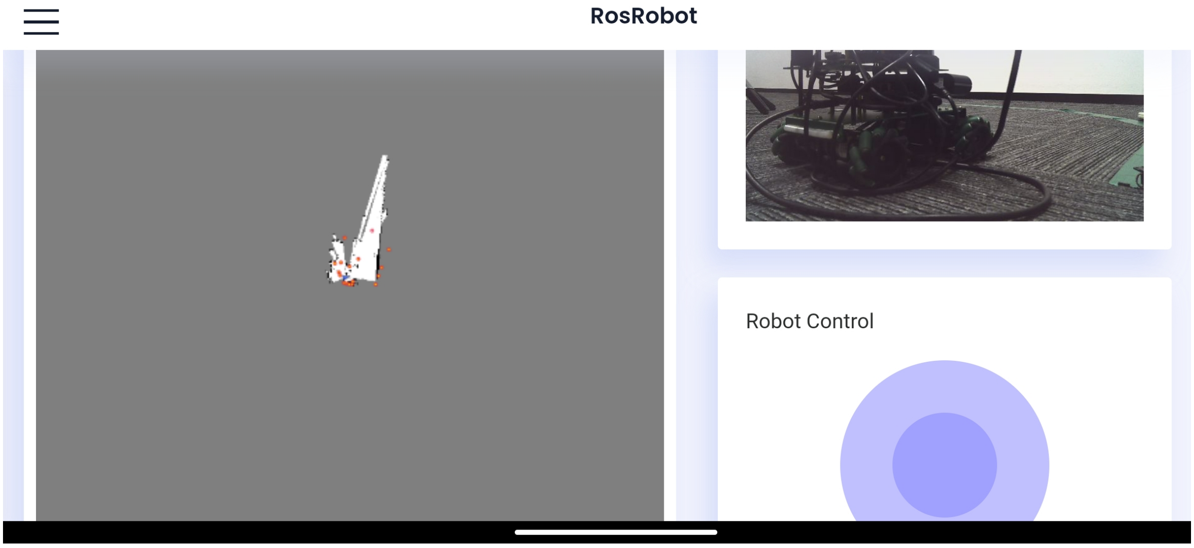
Video Topic

● /usb_cam/image_raw/compressed

○ /camera/rgb/image_raw/compressed

○ /csi/image_raw/compressed

At this point, there is an image in the interface, and the video interface is ignored and not displayed.

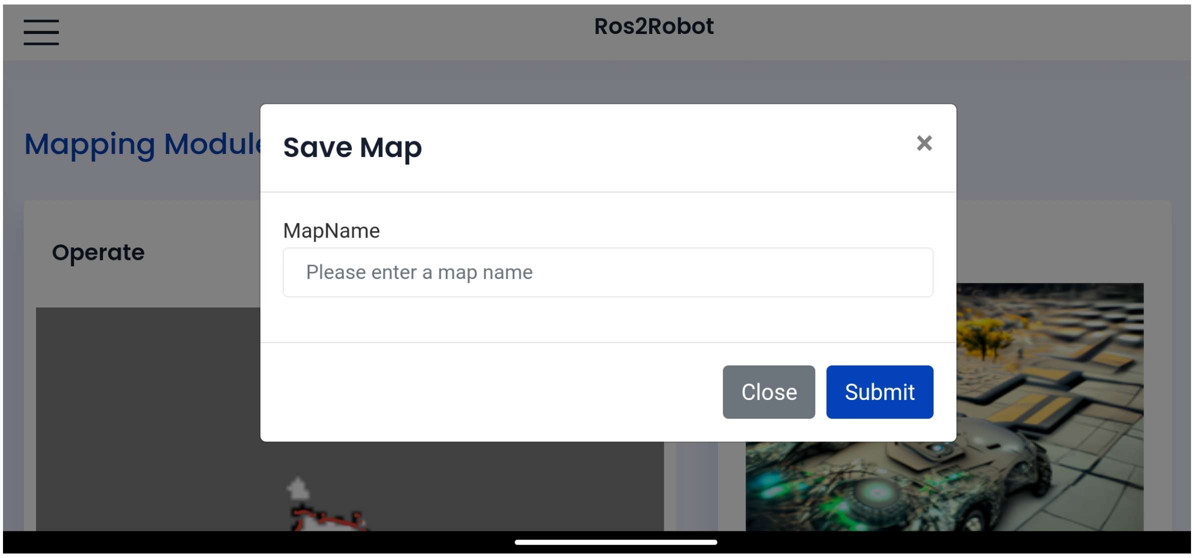


At this time, slide the joystick in the lower left corner. When sliding the joystick, the slower the speed, the better the effect (especially the rotation speed). Control the robot to move until the map is created.

13.1.3. save map

After the map is created, click the Save Map button

Click above the input box, enter the name [house] to save the map, and click [Submit] (only in English, no symbols) to complete saving the map.



The map saving directory is:

```
/home/jetson/yahboomcar_ws/src/yahboomcar_nav/maps
```

You can view the saved map in the specified directory.

13.2. Navigation and obstacle avoidance

Note: [R2] on the remote control handle has the function of canceling the navigation target point! ! ! Do not use joystick control during navigation [all the way]! ! !

13.2.1. start up

Start the driver command (robot side). For convenience of operation,

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

Start the navigation and obstacle avoidance function (robot side)

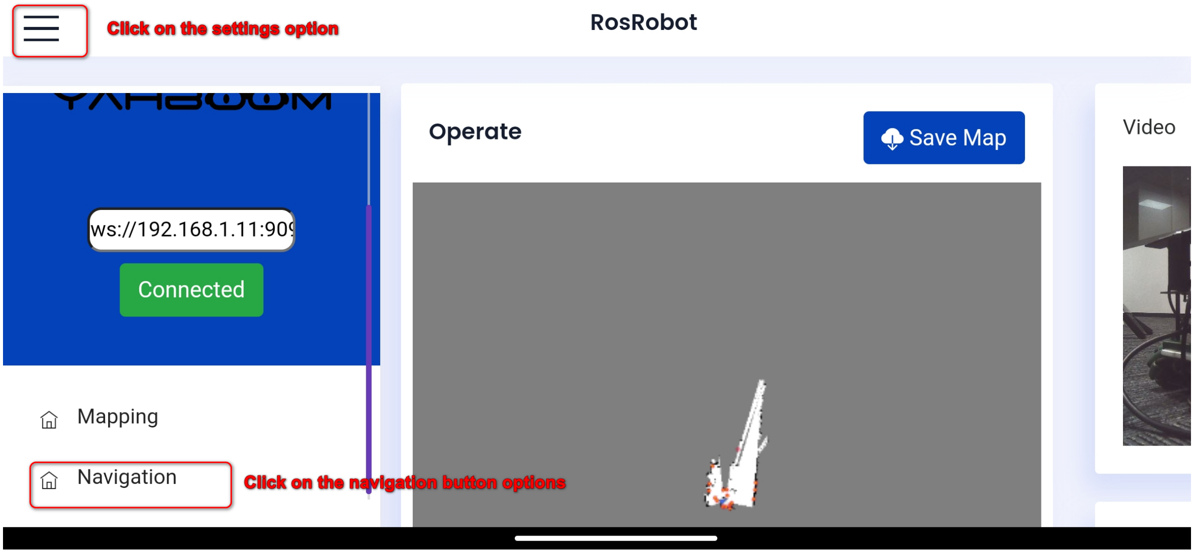
```
roslaunch yahboomcar_nav yahboomcar_navigation.launch use_rviz:=false map:=house
```

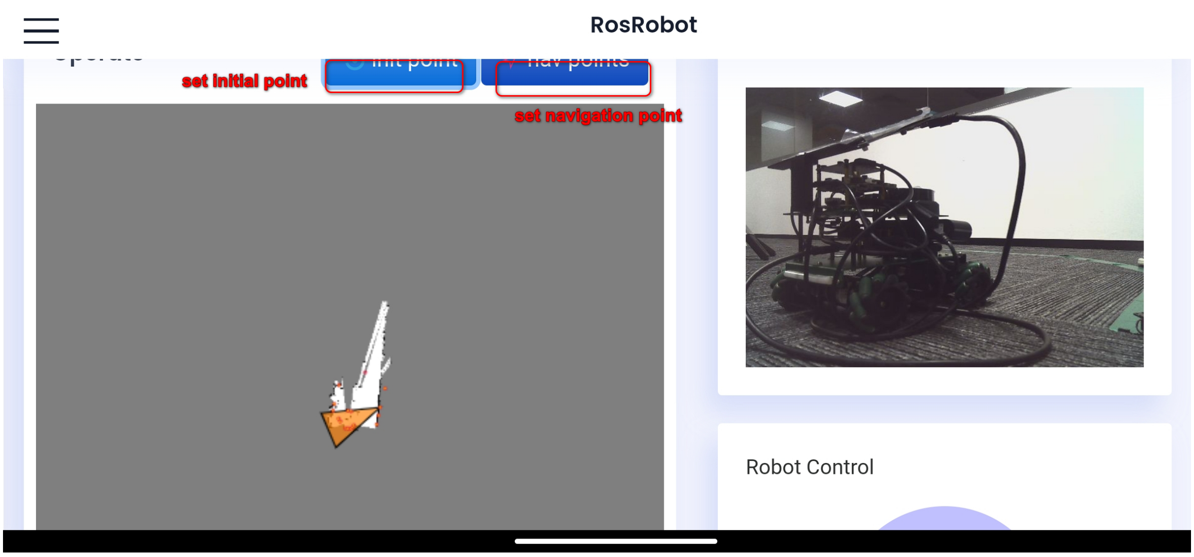
- use_rviz parameter: whether to open rviz.
- map： map name, load the map file named [house].

Start the radar data transfer to point cloud nodes and rosbridge and other nodes

```
source /home/jetson/software/laser_app/devel/setup.bash
roslaunch /home/jetson/software/laser_app/laser_app.launch
```

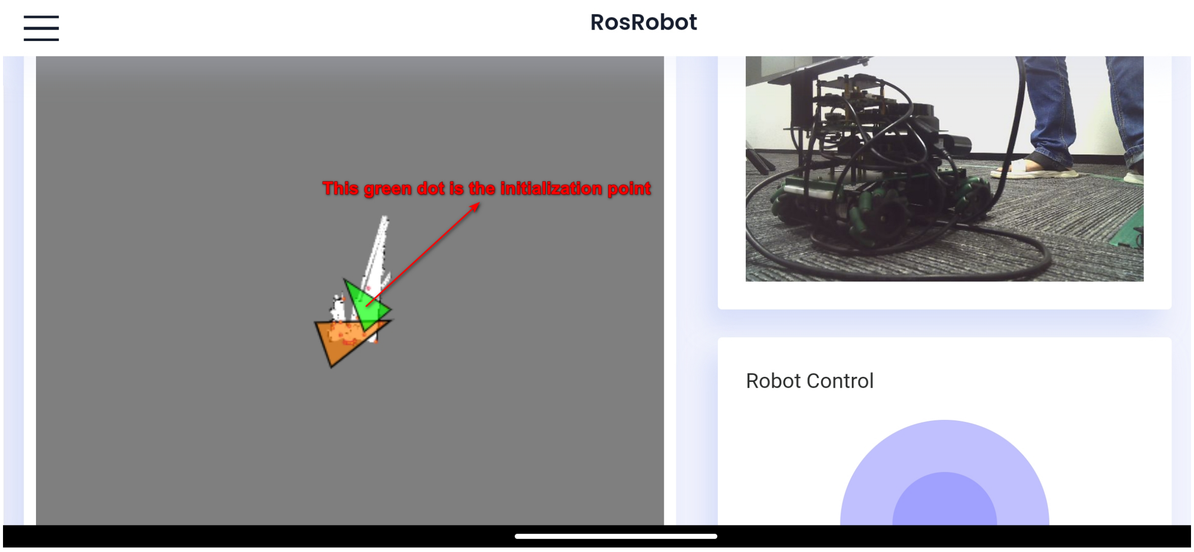
The login method is the same as above. After logging in, you will directly enter the mapping interface. At this time, you need to click the Settings button to open the menu bar and select Navigation. Click the Navigation button to enter the navigation interface.



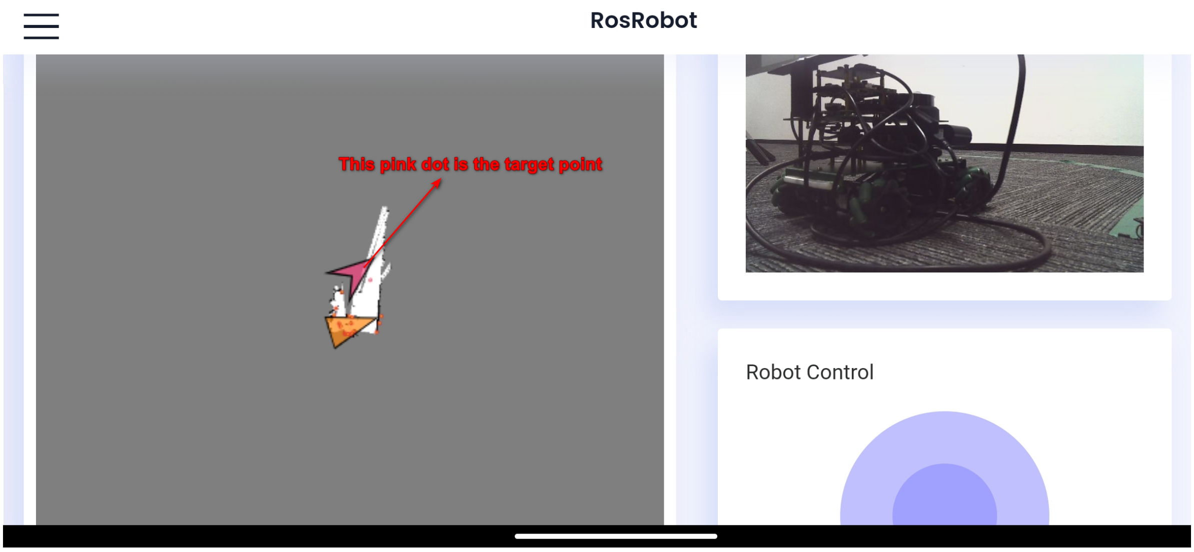


Click Set Initialization Point to enter the interface for setting initialization points.

As you can see in the picture above, the robot is at the origin when building the map, but we have not placed the robot at the origin, so the scanned points do not coincide with the map. At this time, we need to set the initial pose of the robot. First, click the Set Initialization Point button, select the approximate position of the robot on the map, do not let go, and continue sliding towards the approximate pose of the robot. Try to make the scanned points coincide with the map as much as possible



Click the Set Navigation Point button to enter the navigation interface.



Select the robot's target position on the map, don't let go, and continue sliding toward the posture you want the robot to maintain. After releasing, the robot will drive toward the target posture.