Robot dog seeking ball and position

Quick use

1. DOGZILLA POWER UP

First of all, we switch on the switch power of the robot dog and start the robot dog

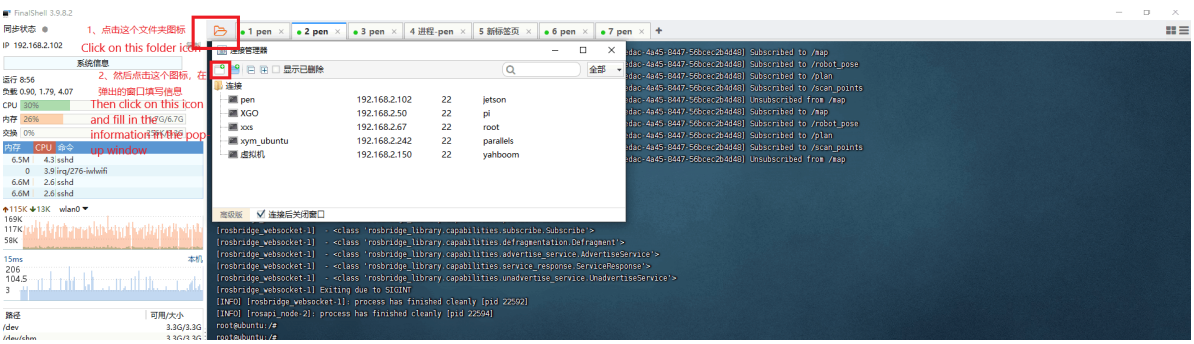
After startup, we can view the IP address on the robot dog's small screen.

2. Open shell to connect to DOGZILLA

Then use the ssh terminal to connect to robot dog.

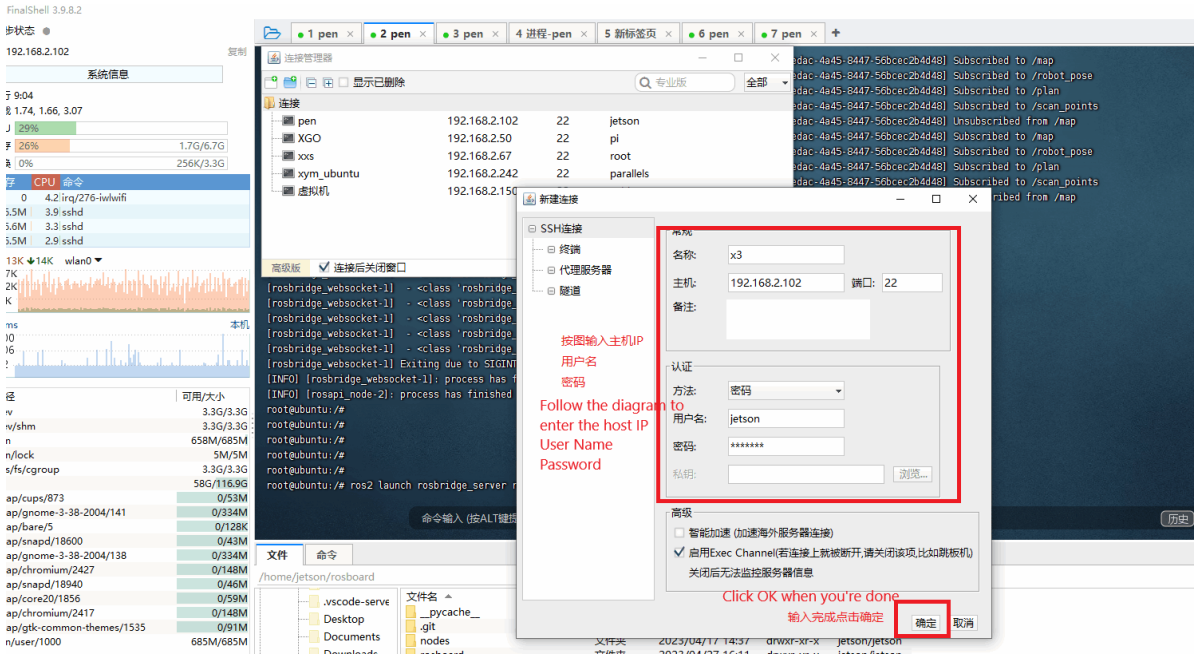
Note: At the time of writing this tutorial, the IP address used is 192.168.2.102 and the username is pi and the password is yahboom, so the actual IP address will prevail.

Open the shell utility, here I use FinalShell, enter the username, password, port, connection name and other information.

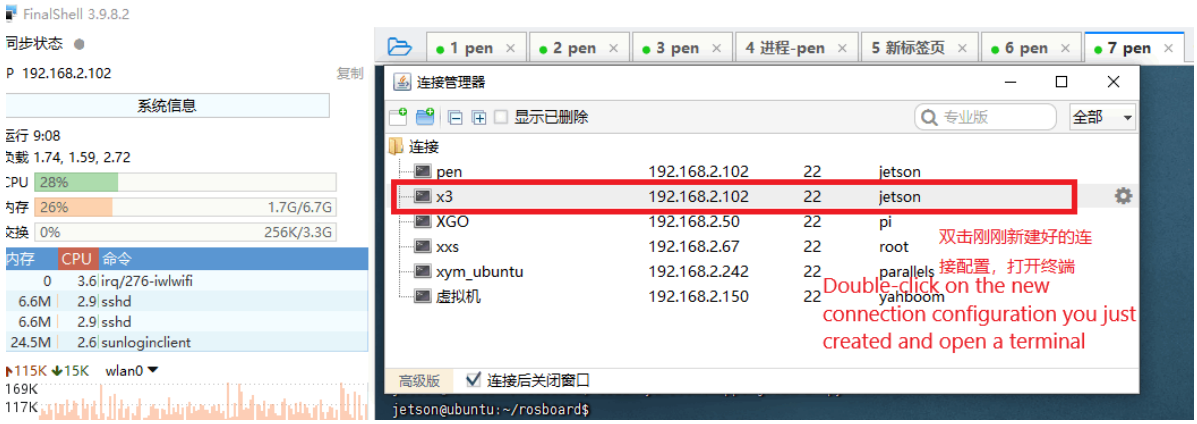


Select ssh connection to create a new ssh connection

Here username fill in pi, password fill in yahboom, ip address fill in the real robot dog's IP address.



Here select the new ssh connection you just created.

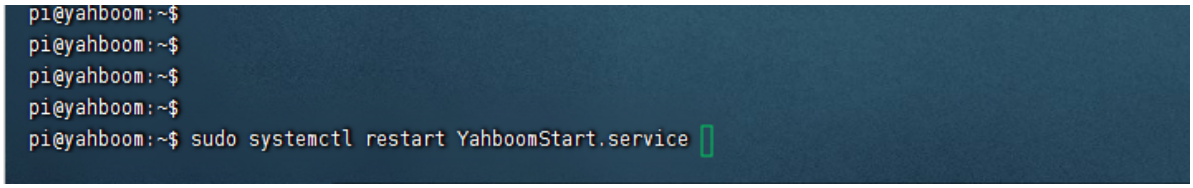


3. Starting the DOGZILLA chassis

Multi-computer communication id modification can refer to the tutorial: 14. Lidar mapping navigation \6. ROS2 environment entity robot dog state acquisition \ ROS2 environment to obtain the real joints of the robot dog data.pdf

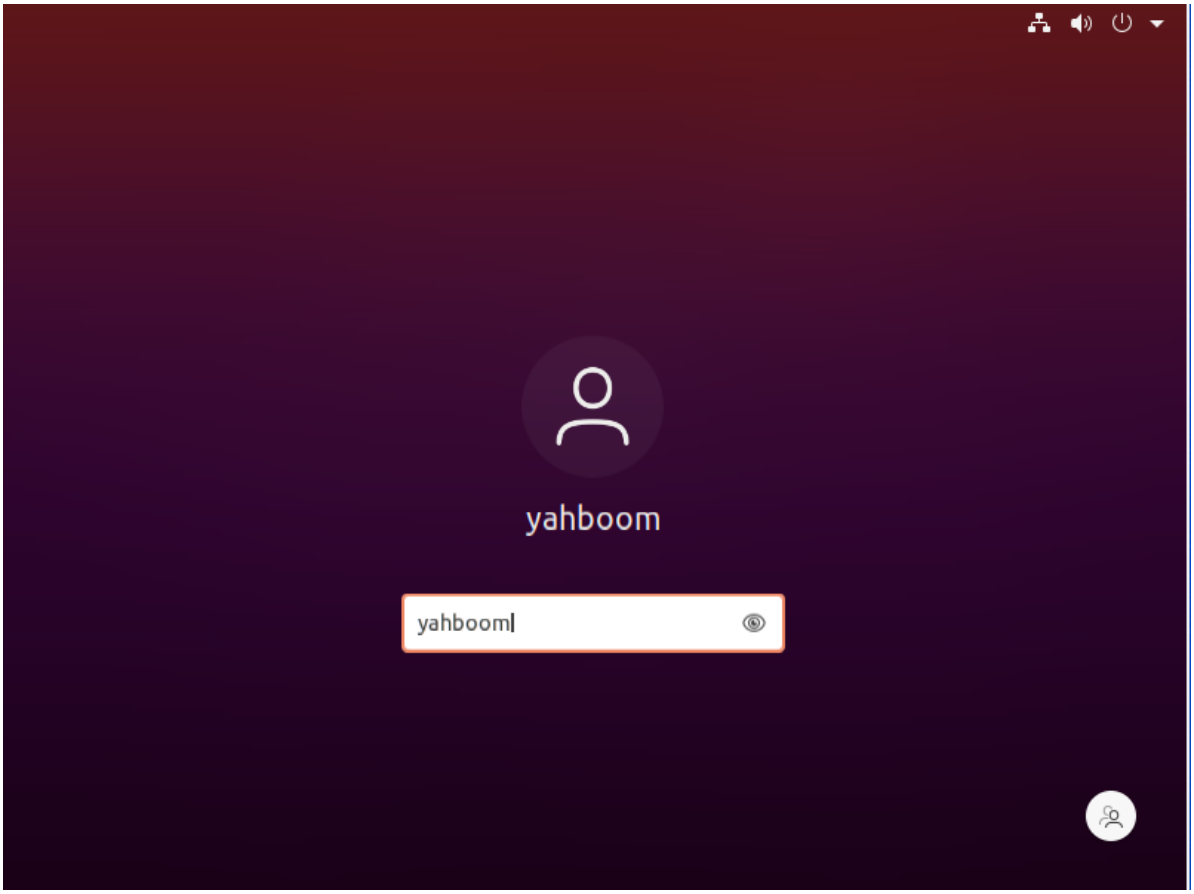
Start the chassis task by entering the command in the terminal.

```
sudo systemctl restart YahboomStart.service
```

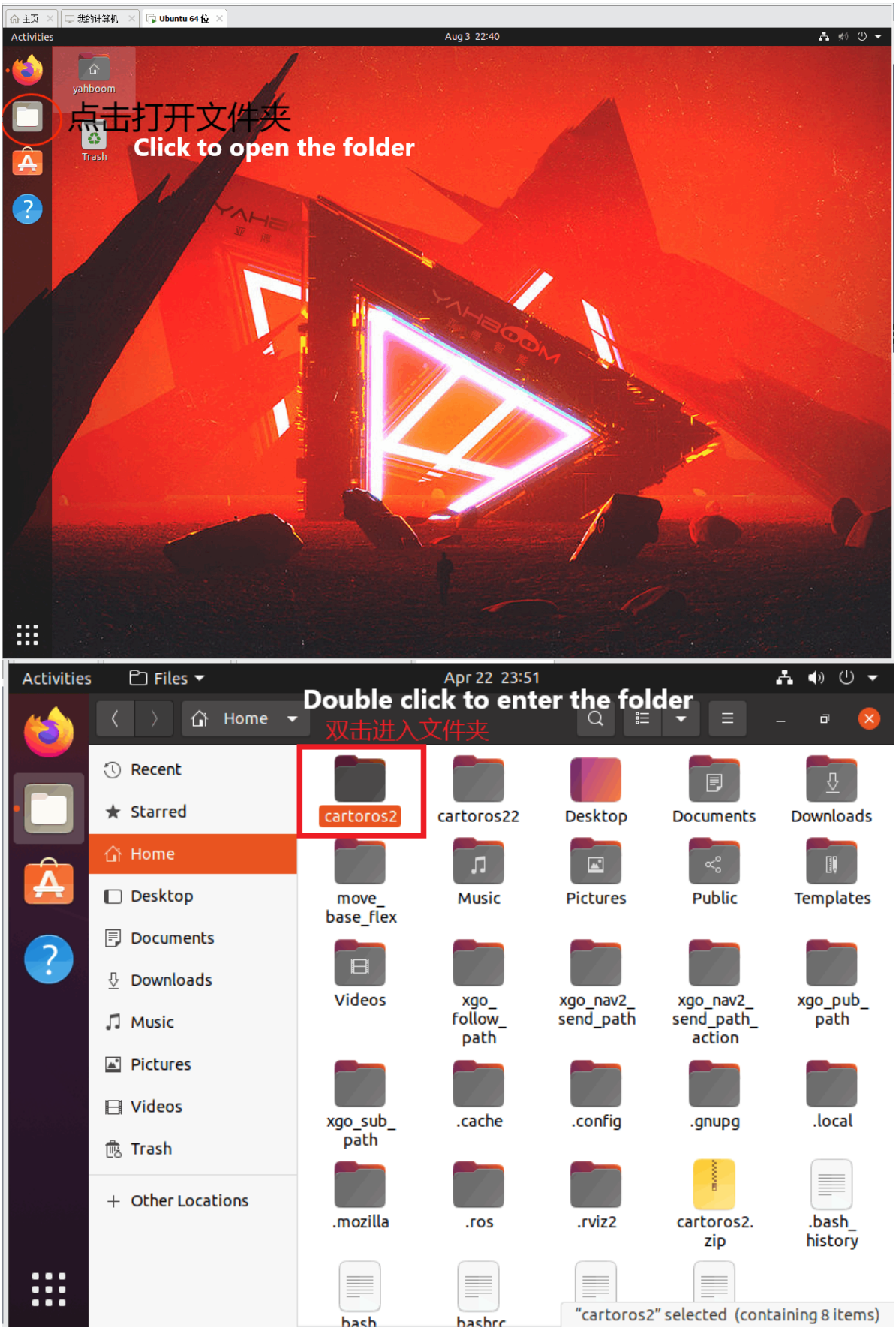


4. Start the navigation programme

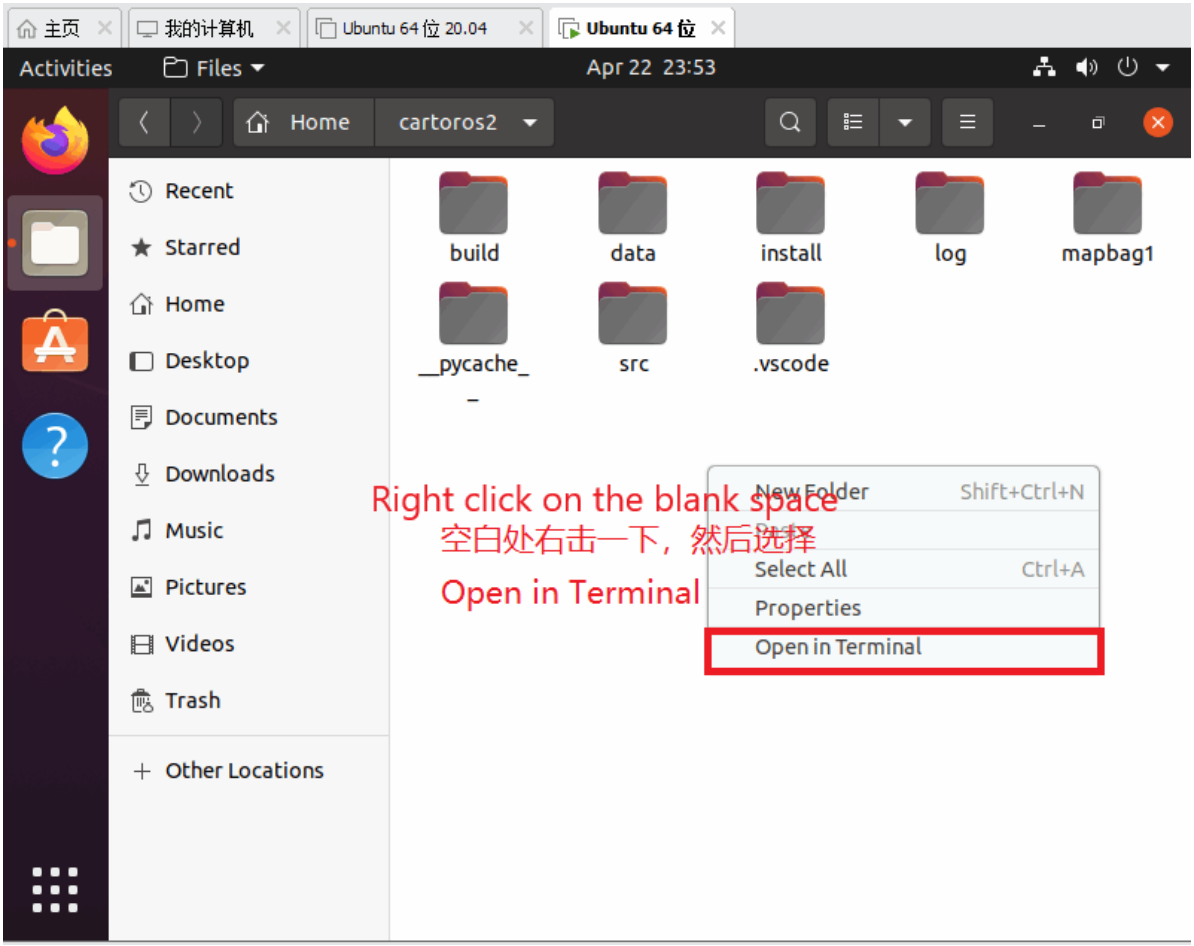
Open the virtual machine and enter the username yahboom, password yahboom.



Click on the folder to open the cartoros2 folder.

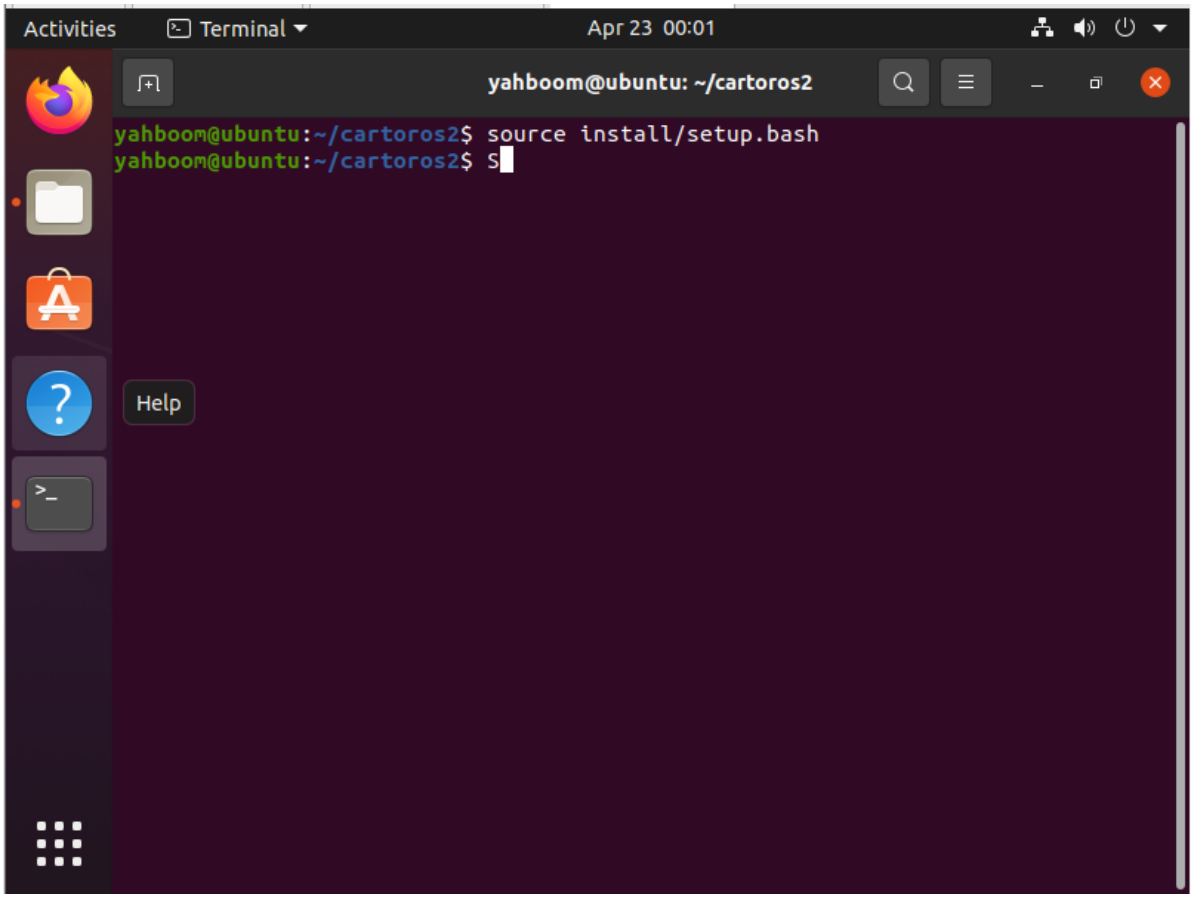


Open a terminal under the folder



Then enter the following command

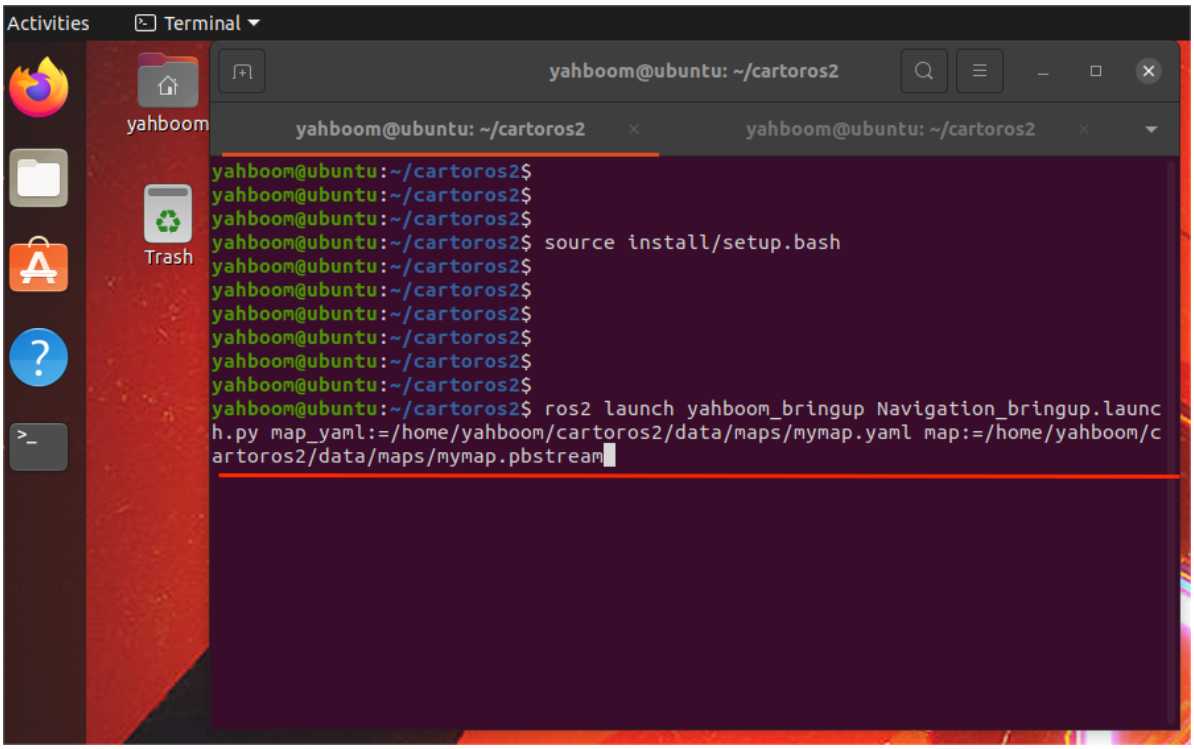
```
source install/setup.bash
```



Then start the navigation programme, this is the first thing to do is to place the robot dog at the build map origin. Then enter the command in the VM terminal:

```
ros2 launch yahboom_bringup Navigation_bringup.launch.py
map_yaml:=/home/yahboom/cartoros2/data/maps/mymap.yaml
map:=/home/yahboom/cartoros2/data/maps/mymap.pbstream
```

Note: The map files xxx.yaml and xxx.pbstream are the two files here that we saved in the previous map building tutorial.

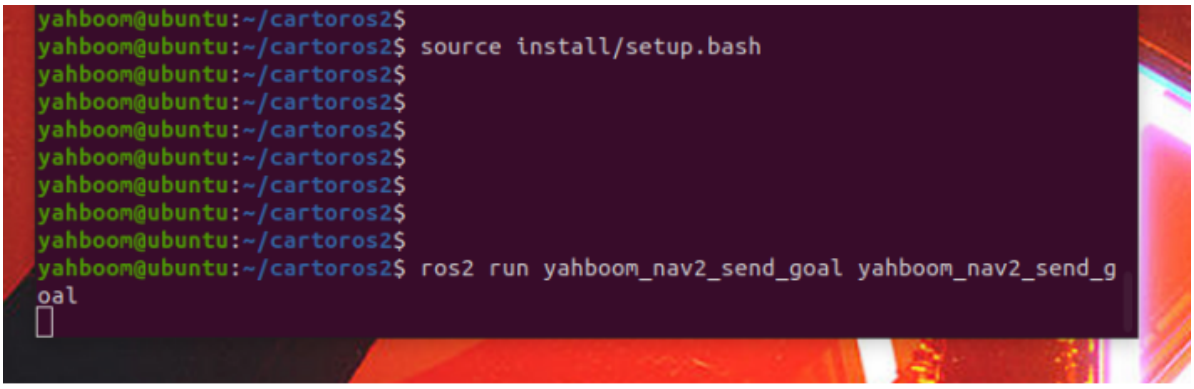


After the navigation module is started, we open another terminal to start the node for multipoint navigation. Enter the command in the newly opened terminal:

```
cd ~/cartoros2
```

```
source install/setup.bash
```

```
ros2 run yahboom_nav2_send_goal yahboom_nav2_send_goal
```



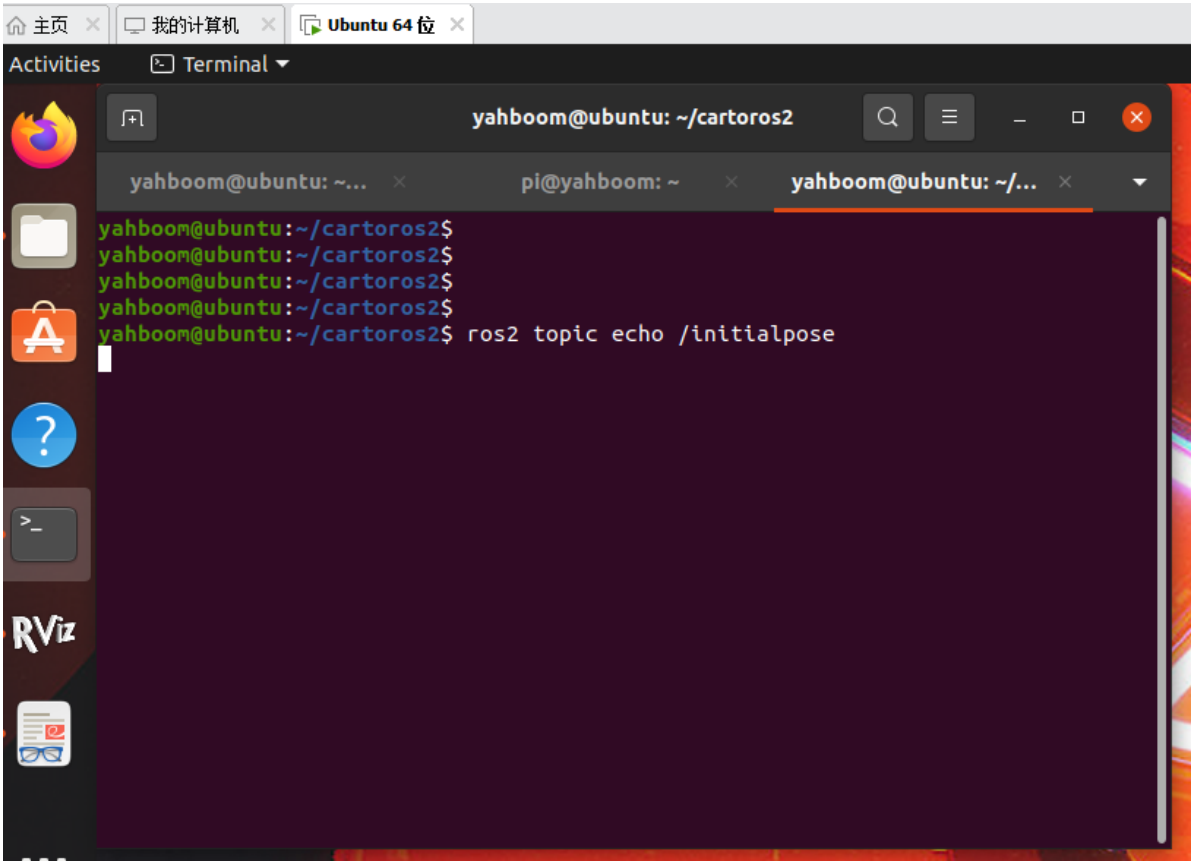
Open another terminal and start the node for multi point navigation. Enter the command in the newly opened terminal:

```
cd ~/cartoros2
source install/setup.bash
ros2 launch rosbridge_server rosbridge_websocket_launch.xml
```

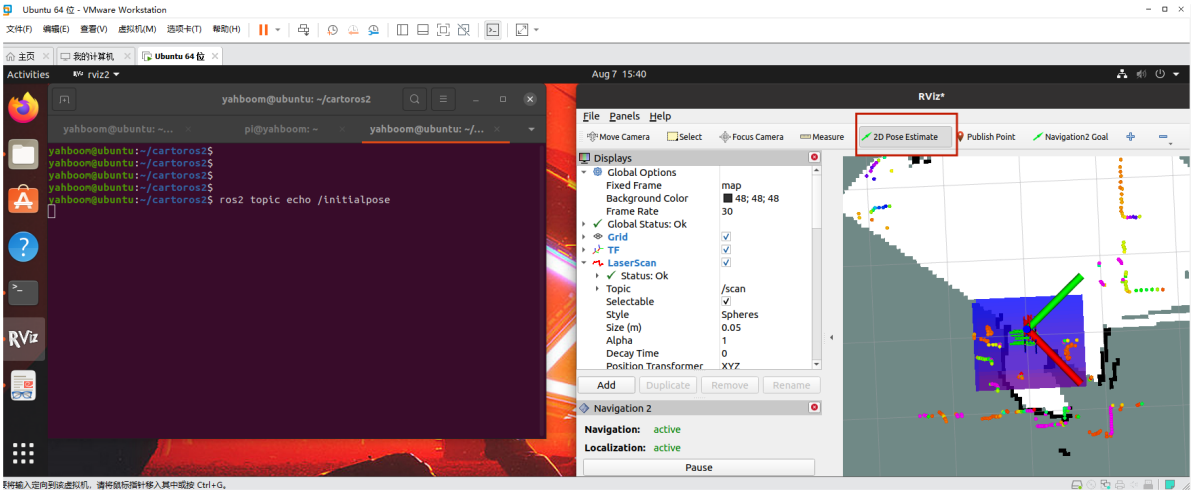
4. Getting the position of the navigation point

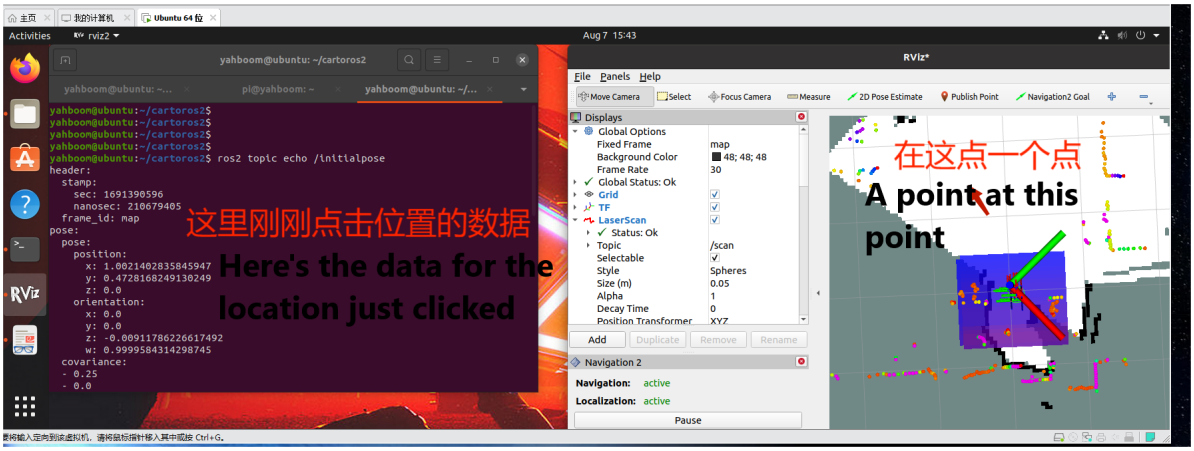
Inside the VM terminal, press the shortcut key: ctrl + alt + T to reopen a terminal, and subscribe to the /initialpose topic in the terminal to get, the point we set in rviz. The command is as follows:

```
ros2 topic echo /initialpose
```



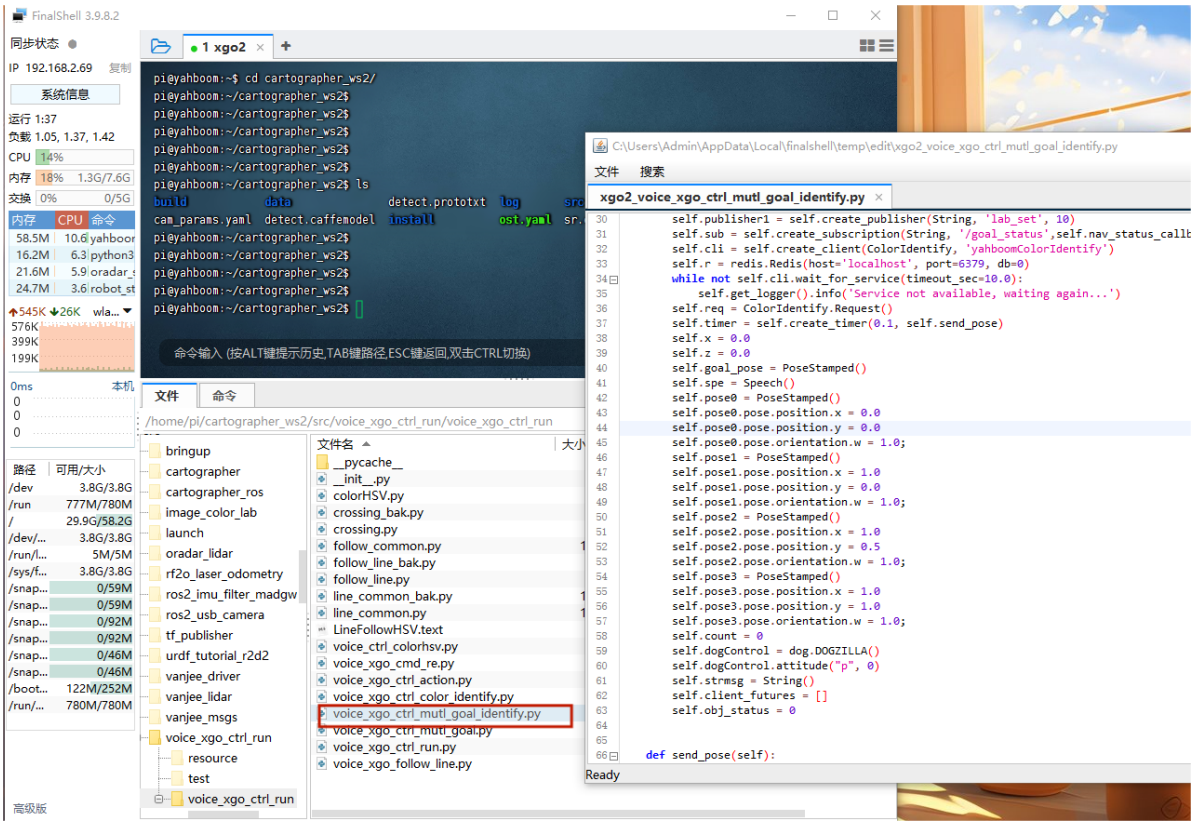
Click the button "2D Pose Estimate" in rviz to click a point on the map where you can walk, and the terminal will print out the position of the point you just clicked.





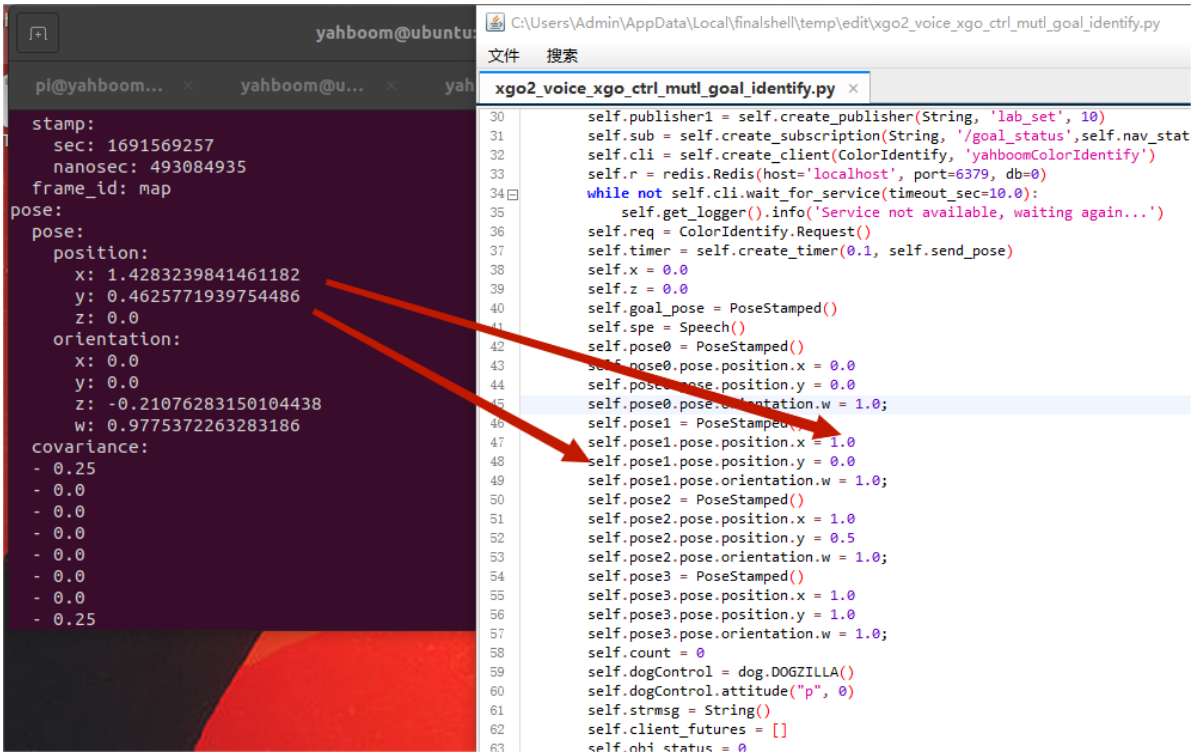
According to the second step, open a new shell terminal, connect to robot dog and we open the file with the following path.

```
/home/pi/cartographer_ws2/install/voice_xgo_ctrl_run/lib/python3.8/site-packages/voice_xgo_ctrl_run
```



Modify the corresponding position.x, position.y values according to the data subscribed to in rviz. This is shown in the following figure:

```
self.pose1.pose.position.x = 1.0
self.pose1.pose.position.y = 0.0
```



We can see that there are 4 points, but we only need to modify pose1 to pose3 according to the above method, because pose0 is the origin position, so we don't need to modify it.

5. Scene simulation

The 3 positions we set up in the above step are the ball placement positions, we can place the ball to any position. When the robot dog starts the ball searching procedure, the robot dog will start from the origin, navigate to the above 3 positions in turn, and start the colour recognition mode to search for the red ball at each position. If there is a red ball, the robot dog will return to the origin and print a message in the terminal.

6. Start the voice-controlled robot dog to look for the ball.

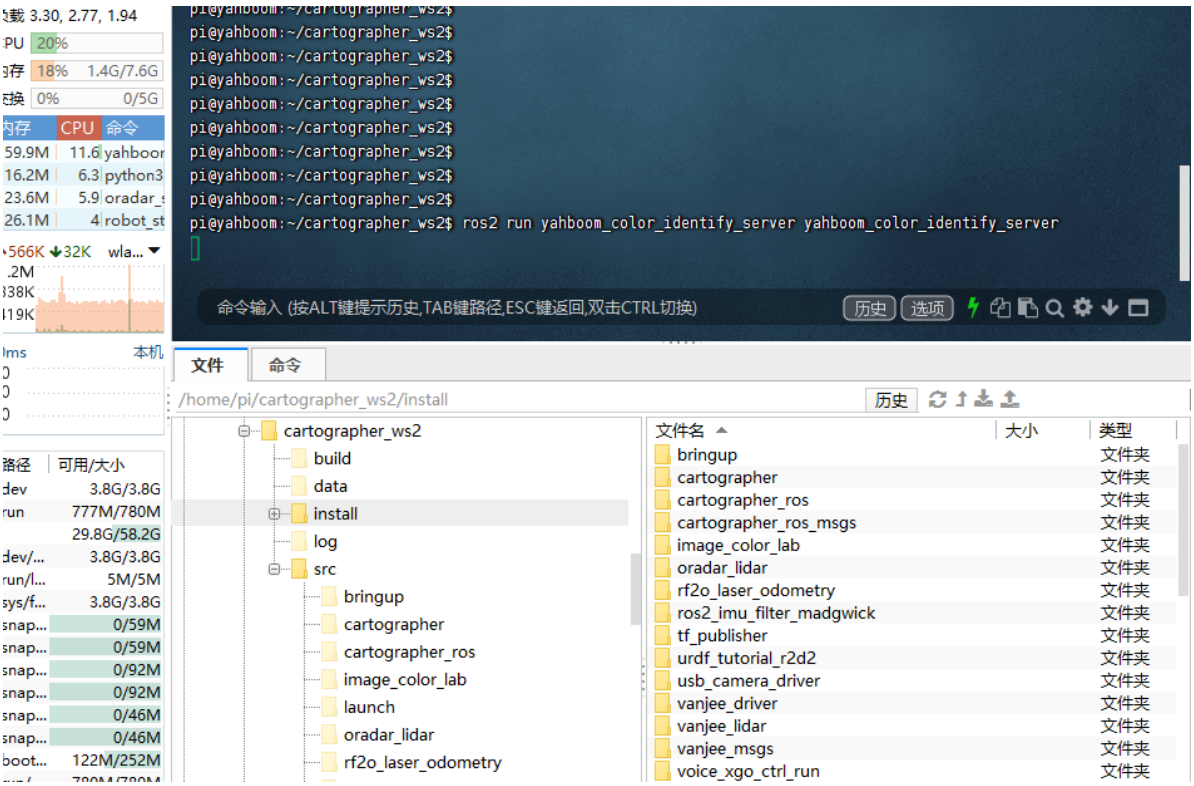
Open another shell terminal and enter the following commands in the terminal:

Note:This terminal is the terminal that opens the remote connection to the robot dog.

```
cd ~/cartographer_ws2/
```

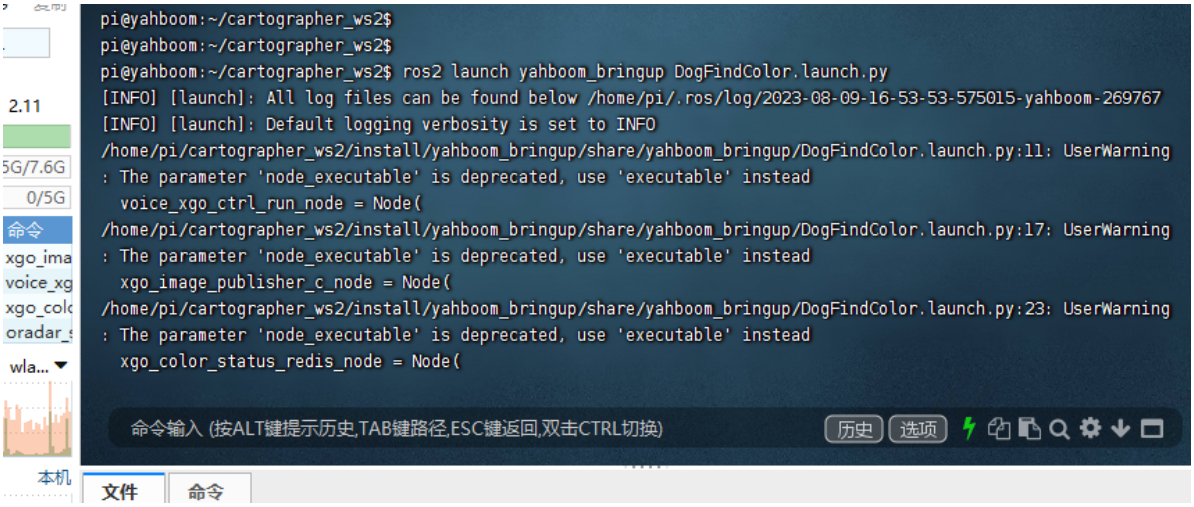
```
source install/setup.bash
```

```
ros2 run yahboom_color_identify_server yahboom_color_identify_server
```



Open a new shell terminal and enter the command:

```
cd ~/cartographer_ws2/
source install/setup.bash
ros2 launch yahboom_bringup DogFindColor.launch.py
```

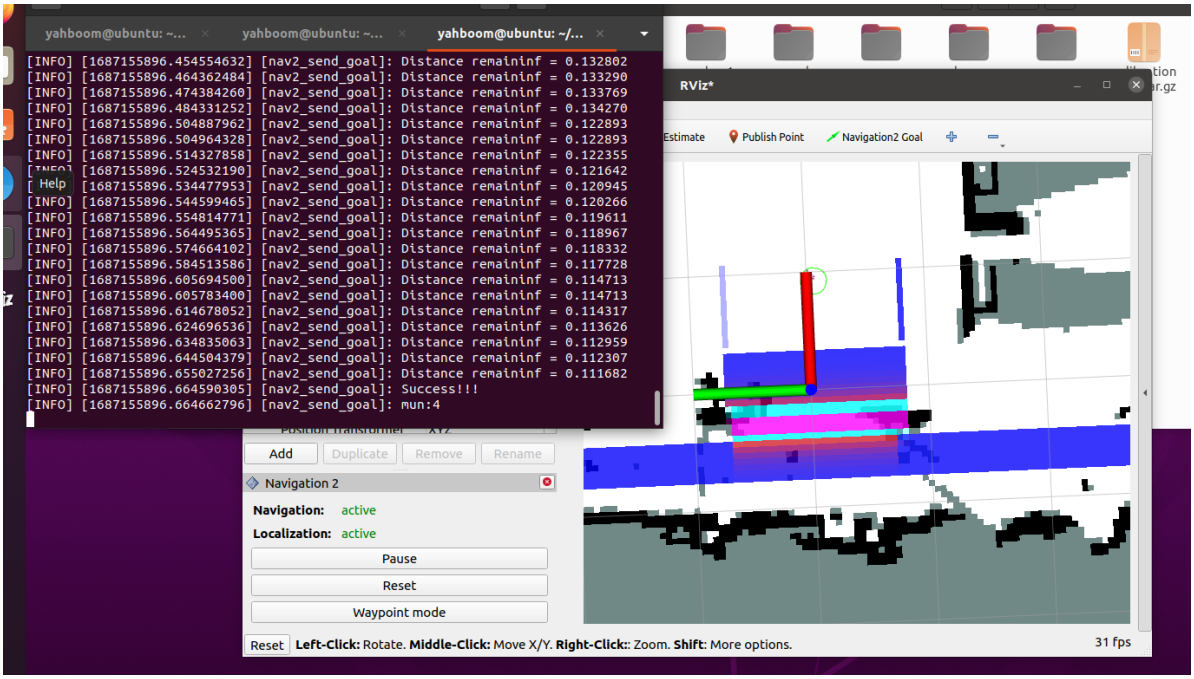


Then to the robot dog he says, "Hi, Yahboom"

The robot dog replies, "Hi, I am here"

Then say to the robot dog, "Go to the point A"

As shown in the picture below, the robot dog will automatically navigate to point A.


yahboom@ubuntu: ~...
yahboom@ubuntu: ~...
yahboom@ubuntu: ~/...
[INFO] [1687155896.454554632] [nav2_send_goal]: Distance remaininf = 0.132802
[INFO] [1687155896.464362484] [nav2_send_goal]: Distance remaininf = 0.133290
[INFO] [1687155896.474384260] [nav2_send_goal]: Distance remaininf = 0.133769
[INFO] [1687155896.484331252] [nav2_send_goal]: Distance remaininf = 0.134270
[INFO] [1687155896.504887962] [nav2_send_goal]: Distance remaininf = 0.122893
[INFO] [1687155896.504964328] [nav2_send_goal]: Distance remaininf = 0.122893
[INFO] [1687155896.514327858] [nav2_send_goal]: Distance remaininf = 0.122355
[INFO] [1687155896.524532190] [nav2_send_goal]: Distance remaininf = 0.121642
[INFO] [1687155896.534477953] [nav2_send_goal]: Distance remaininf = 0.120945
[INFO] [1687155896.544599465] [nav2_send_goal]: Distance remaininf = 0.120266
[INFO] [1687155896.554814771] [nav2_send_goal]: Distance remaininf = 0.119611
[INFO] [1687155896.564495365] [nav2_send_goal]: Distance remaininf = 0.118967
[INFO] [1687155896.574664102] [nav2_send_goal]: Distance remaininf = 0.118332
[INFO] [1687155896.584513586] [nav2_send_goal]: Distance remaininf = 0.117728
[INFO] [1687155896.605694500] [nav2_send_goal]: Distance remaininf = 0.114713
[INFO] [1687155896.605783400] [nav2_send_goal]: Distance remaininf = 0.114713
[INFO] [1687155896.614678052] [nav2_send_goal]: Distance remaininf = 0.114317
[INFO] [1687155896.624696536] [nav2_send_goal]: Distance remaininf = 0.113626
[INFO] [1687155896.634835063] [nav2_send_goal]: Distance remaininf = 0.112959
[INFO] [1687155896.644504379] [nav2_send_goal]: Distance remaininf = 0.112307
[INFO] [1687155896.655027256] [nav2_send_goal]: Distance remaininf = 0.111682
[INFO] [1687155896.664590305] [nav2_send_goal]: Success!!!
[INFO] [1687155896.664662796] [nav2_send_goal]: mun:4
Help
RViz*
Estimate
Publish Point
Navigation2 Goal
Add
Duplicate
Remove
Rename
Navigation 2
Navigation: active
Localization: active
Pause
Reset
Waypoint mode
Reset
Left-Click: Rotate. Middle-Click: Move X/Y. Right-Click:: Zoom. Shift: More options.
31 fps