

Voice controlled mechanical dog navigation and ball hunting

Quick use

1. Power on DOGZILLA

First, we turn on the switching power supply of the mechanical dog and start the mechanical dog



After starting, we can view the IP address on the small screen of the robot dog.

2. Start DOGZILLA chassis

PI4 version steps:

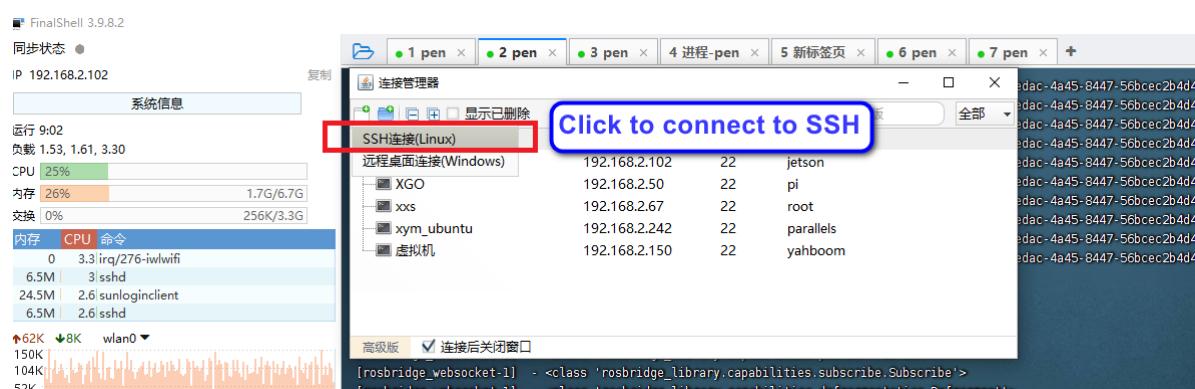
Then use the ssh terminal to connect to the robot dog.

Note: The IP address used when writing this tutorial: 192.168.2.102 User name: pi Password: yahboom The actual IP address shall prevail when used.

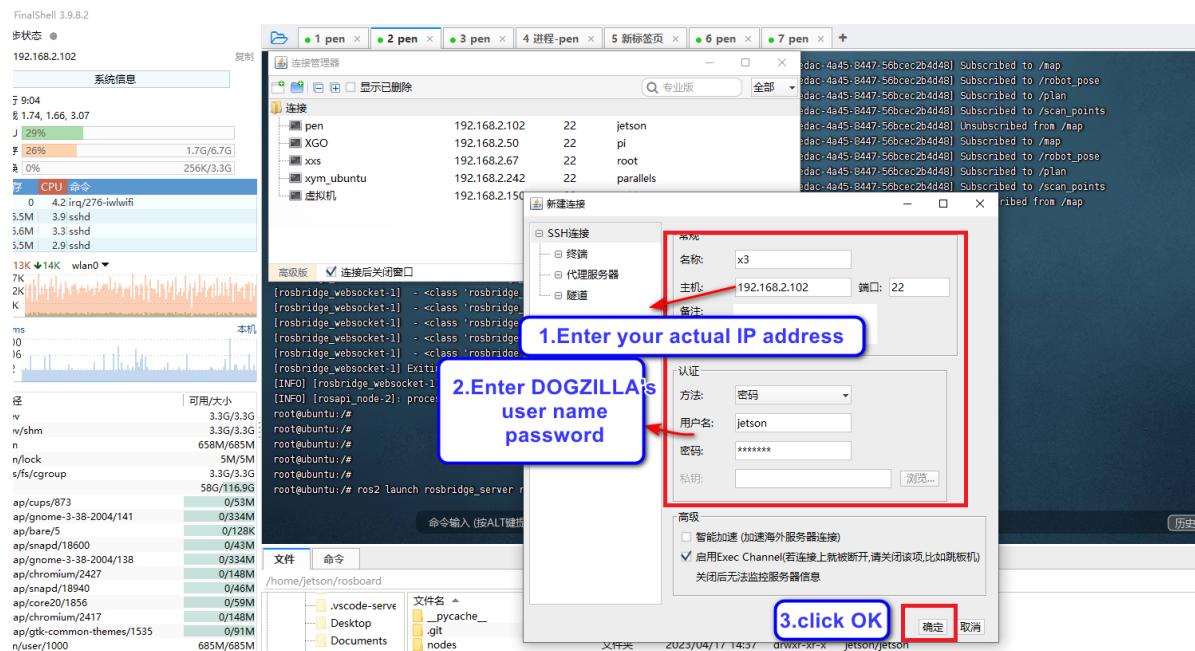
Open the shell tool. The shell tool I use here is FinalShell. Enter username, password, port, connection name and other information.



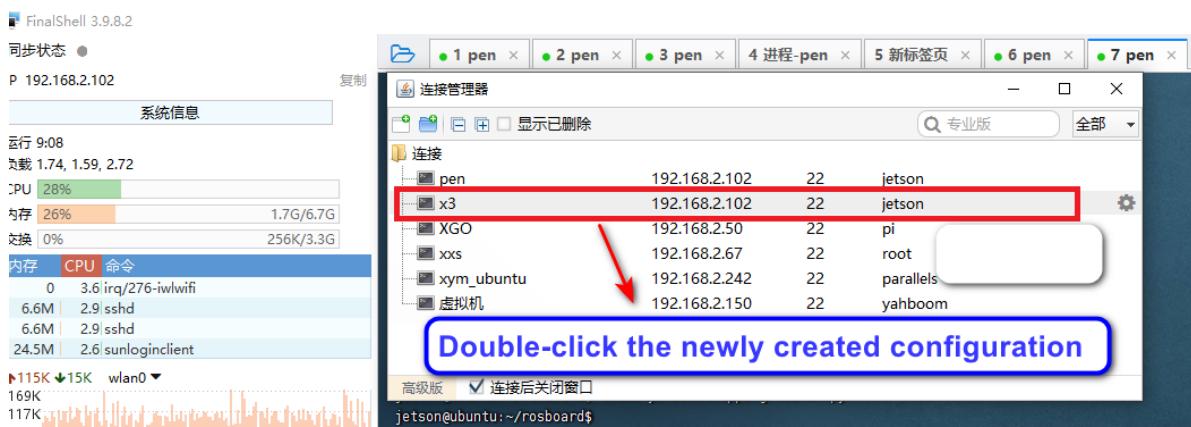
Select ssh connection to create a new ssh connection



Here the username is pi, the password is yahboom, and the ip address is the IP address of the real robot dog.



Select the ssh connection you just created here.



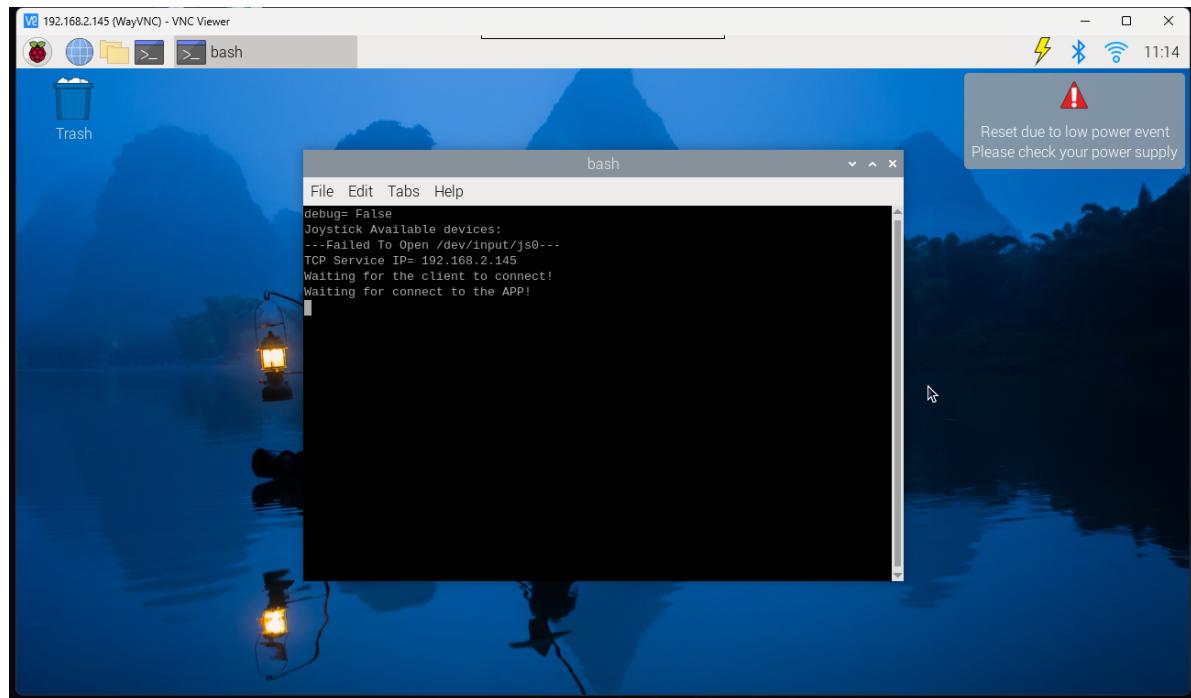
Enter the command in the terminal to start the chassis task.

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

```
pi@yahboom:~$  
pi@yahboom:~$  
pi@yahboom:~$  
pi@yahboom:~$  
pi@yahboom:~$ sudo systemctl restart YahboomStart.service
```

PI5 version steps:

After the mechanical dog is started, use the vnc software to remotely connect to the mechanical dog through the IP address on the OLED (**For specific steps, please see "Remote Login Operation"**).



Then **ctrl+c** closes the large program and enter the following command to enter docker:

```
./run_humble.sh
```

```
TCP Service IP= 192.168.2.145
Waiting for the client to connect!
Waiting for connect to the APP!
^CKeyboardInterrupt
2024-04-28T10:17:27Z
-----program end-----
pi@raspberrypi:~ $ ./run_humble.sh
access control disabled, clients can connect from any host
root@raspberrypi:/#
```

Then enter the following commands in the docker terminal to start the car radar, imu, and mechanical dog joint status nodes.

```
ros2 launch bringup Navigation_bringup.launch.py
```

```
root@raspberrypi: /
```

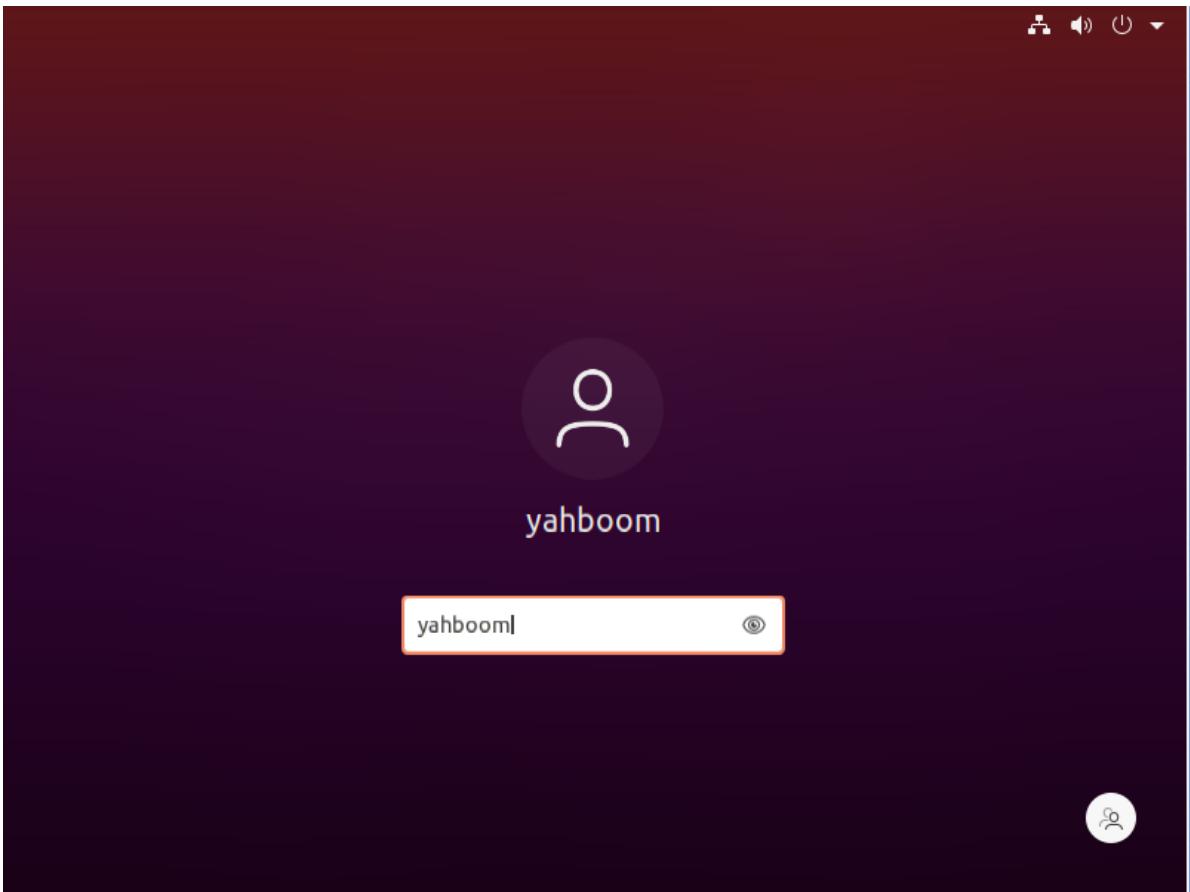
File Edit Tabs Help

```
at 0x7fff363522f0>
[yahboomcar_joint_state-3] [13.16, 45.61, 1.34, 10.1, 44.36, -1.09, 10.1, 51.85,
 2.55, 6.53, 51.22, -0.36]
[yahboomcar_joint_state-3] &&&&&&&&&&&&&&&&&& 0.10927200317382812
[yahboomcar_joint_state-3] #####
[yahboomcar_joint_state-3] [-0.17585449218750002, -0.13996582031250002, -9.72702
63671875, -1.0365853658536586, -0.426829268292683, -0.6097560975609757, 0.010487
360583411322, -0.02726797640323639, 5.983139933268229]
[yahboomcar_joint_state-3] ***** <rclpy.timer.Timer object
at 0x7fff363522f0> ****
[yahboomcar_joint_state-3] [13.16, 45.61, 1.34, 10.1, 44.36, -1.09, 10.1, 51.85,
 2.55, 6.53, 51.22, -0.36]
[yahboomcar_joint_state-3] &&&&&&&&&&&&&&&&& 0.10969948768615723
[yahboomcar_joint_state-3] #####
[yahboomcar_joint_state-3] [-0.14475097656250002, -0.131591796875, -9.7401855468
75, -1.0975609756097562, -0.3658536585365854, -0.6097560975609757, 0.01022947788
9007993, -0.02749979310565525, 5.983139933268229]
[yahboomcar_joint_state-3] ***** <rclpy.timer.Timer object
at 0x7fff363522f0> ****
[yahboomcar_joint_state-3] [13.16, 45.61, 1.34, 10.1, 44.36, -1.09, 10.1, 51.85,
 2.55, 6.53, 51.22, -0.36]
[yahboomcar_joint_state-3] &&&&&&&&&&&&&&&&& 0.10920882225036621
[yahboomcar_joint_state-3] #####
```

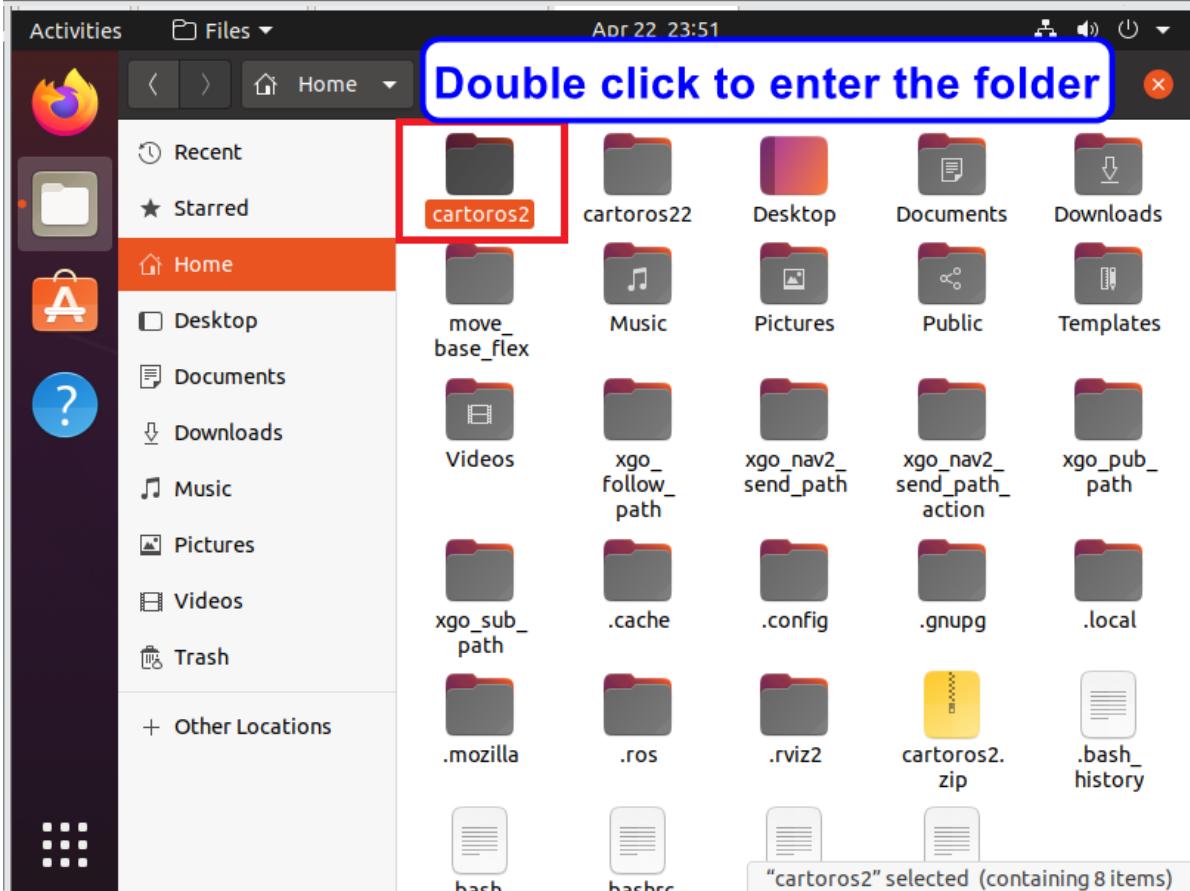
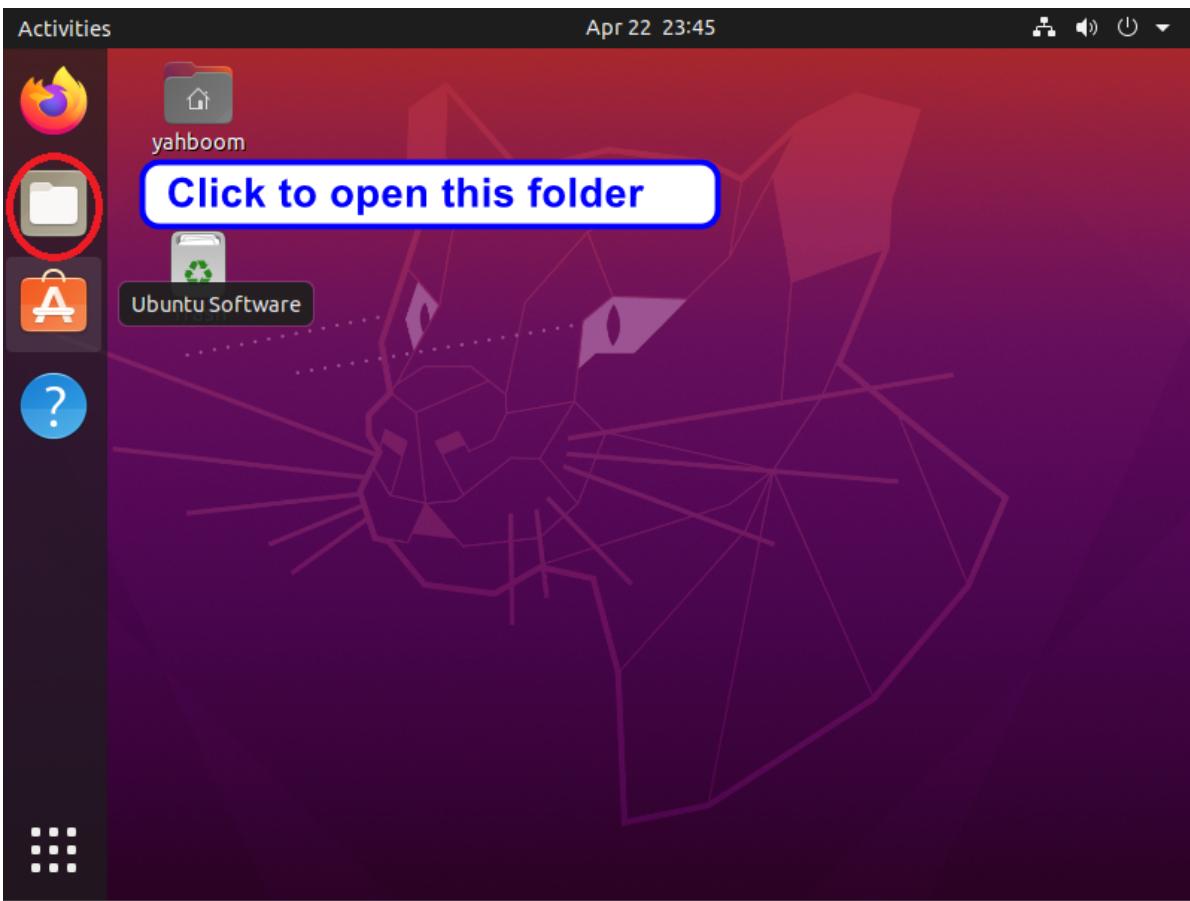
3. Start the navigation program

PI4 version steps:

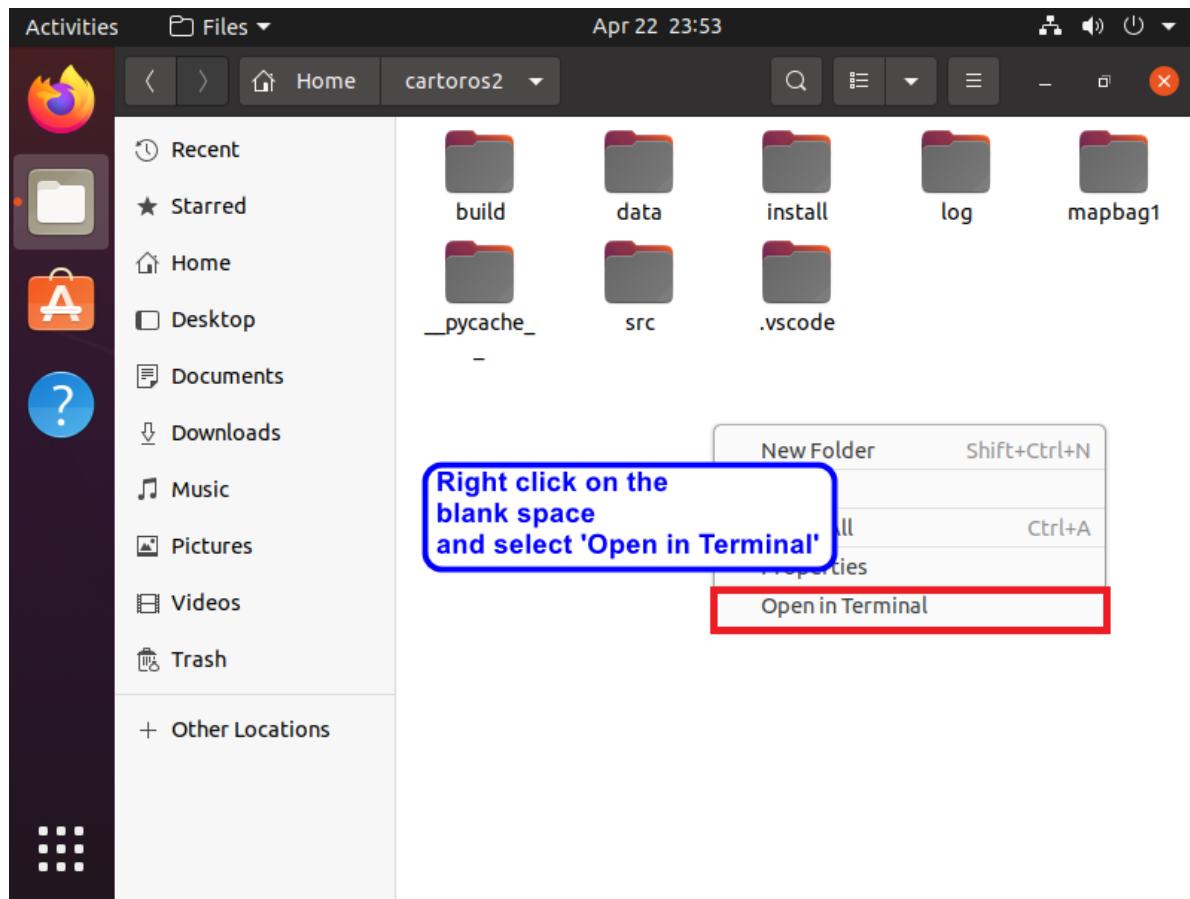
Open the virtual machine and enter the user name yahboom and the password yahboom.



Click on the folder to open the cartoros2 folder.

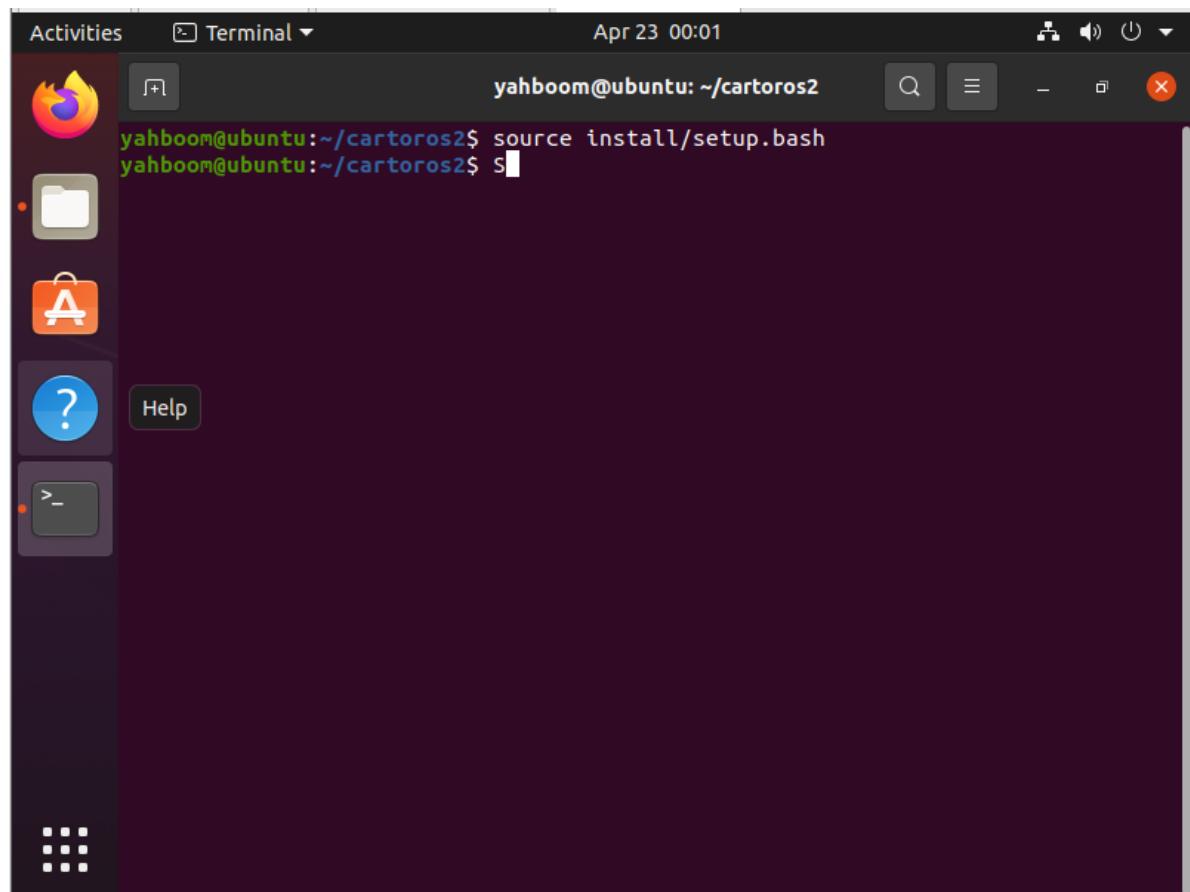


Open the terminal under the folder



Then enter the following command

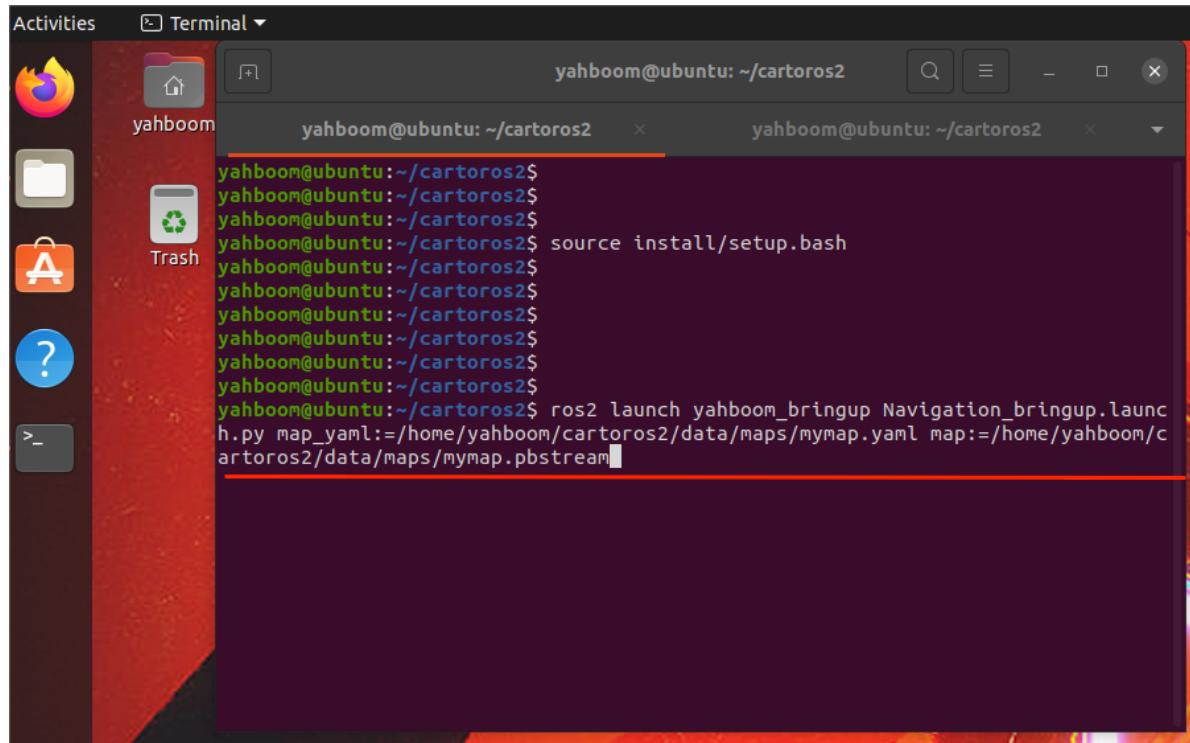
```
source install/setup.bash
```



Then start the navigation program, which is to first place the mechanical dog at the origin of the mapping. Then enter the command in the virtual machine terminal:

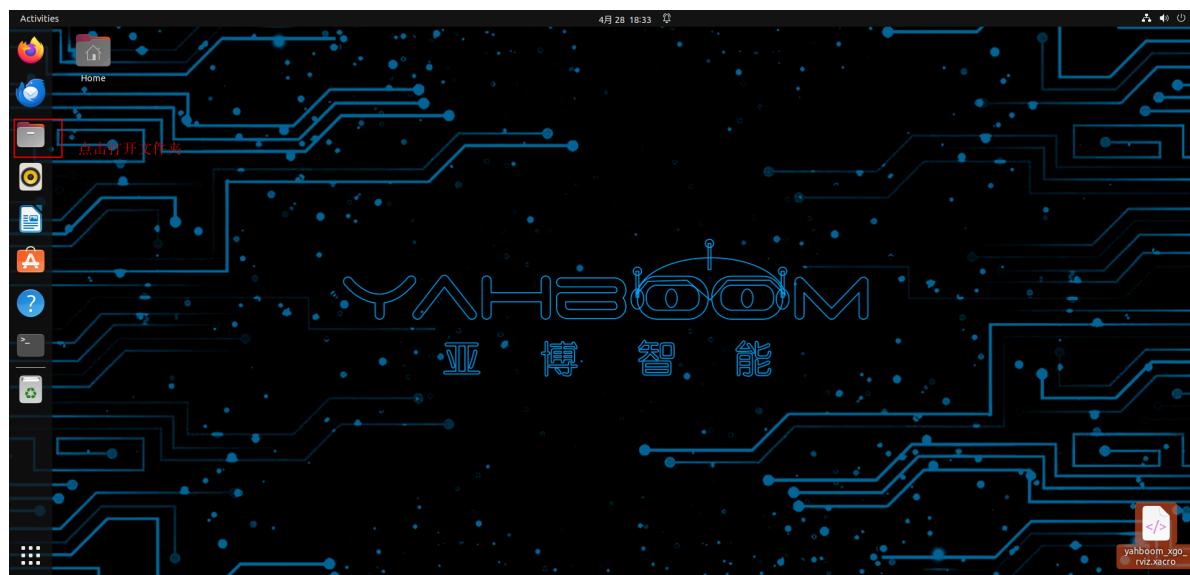
```
ros2 launch yahboom_bringup Navigation_bringup.launch.py  
map_yaml:=/home/yahboom/cartoros2/data/maps/mymap.yaml  
map:=/home/yahboom/cartoros2/data/maps/mymap.pbstream
```

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

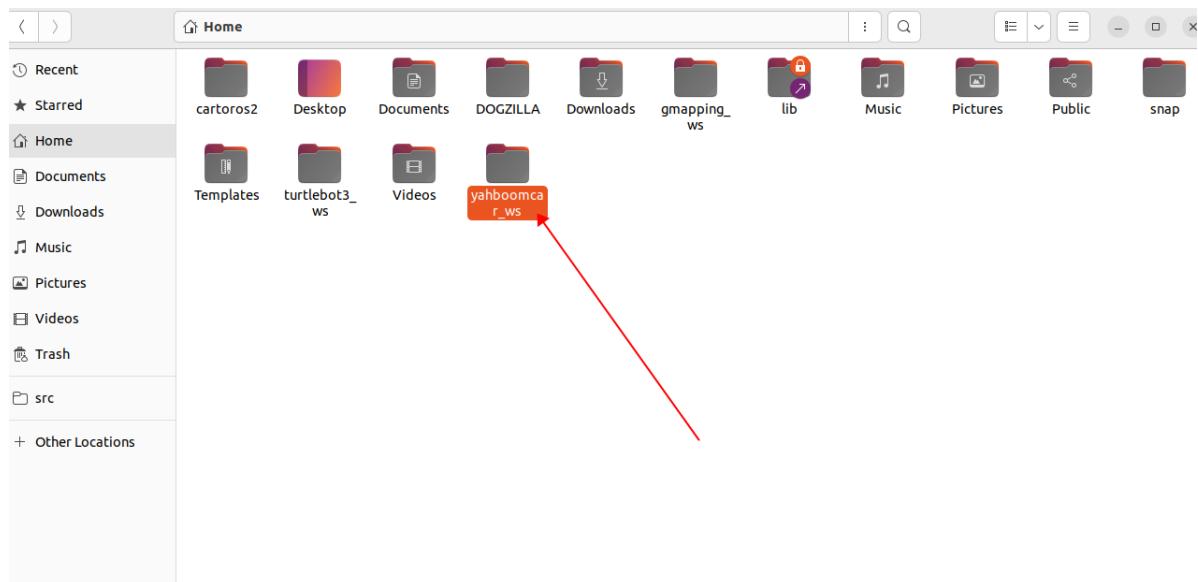


PI5 version steps:

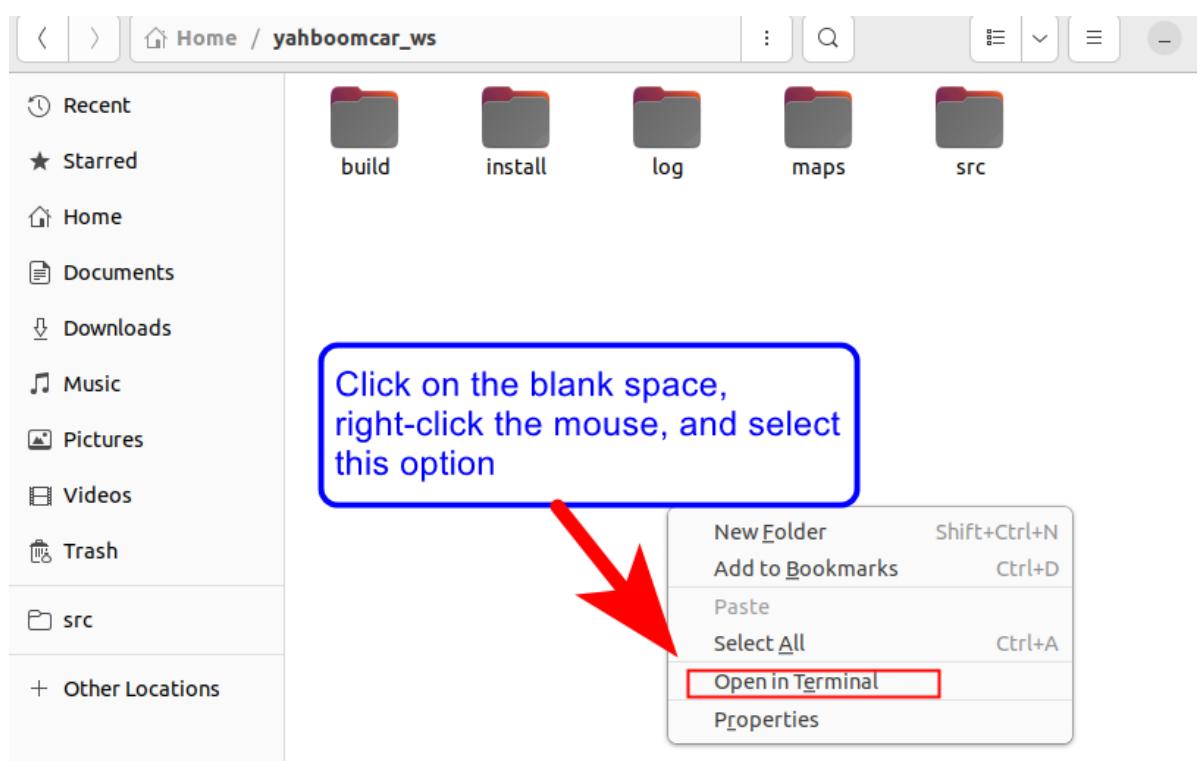
Enter the desktop system and open the folder.



Double-click to open the yahboomcar_ws folder



Then right-click in an empty space of the folder and select Open in Terminal



Then enter the following command in the terminal to activate the environment

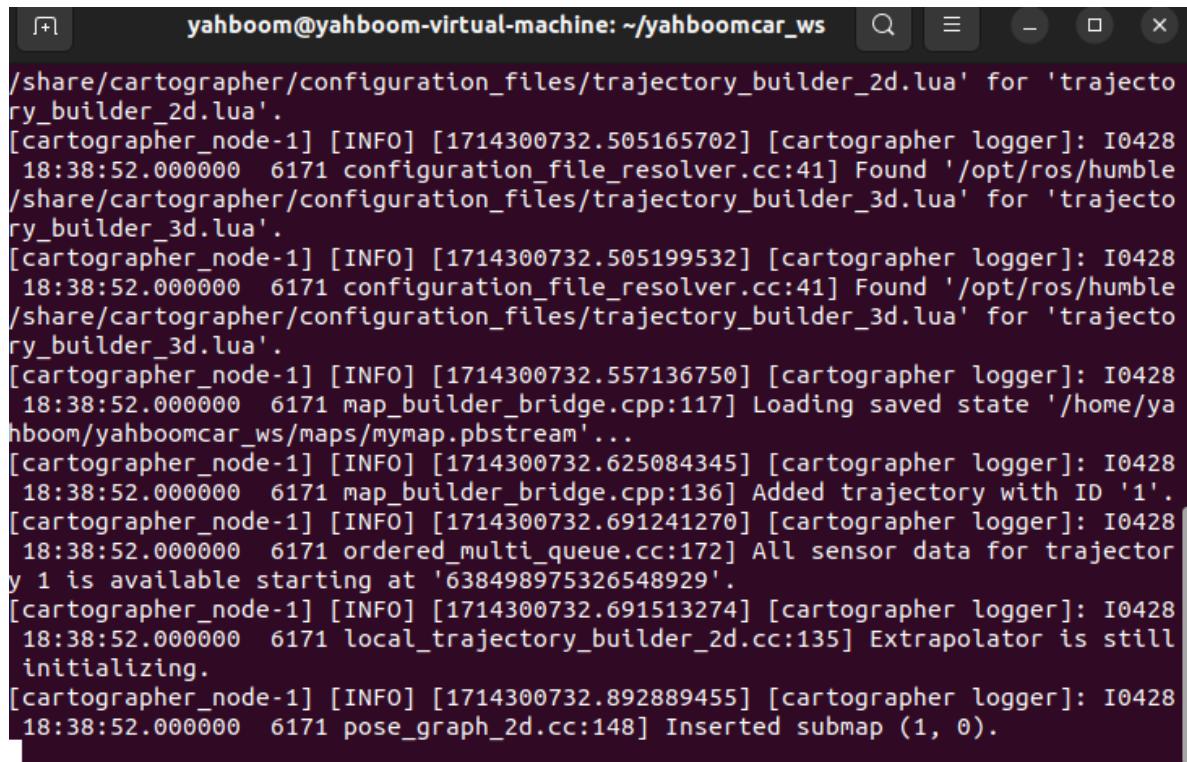
```
source install/setup.bash
```

After completing the input, press Enter.

```
yahboom@yahboom-virtual-machine:~/yahboomcar_ws$ source install/setup.bash
yahboom@yahboom-virtual-machine:~/yahboomcar_ws$
```

Then enter the command to start the relocation function

```
ros2 launch yahboom_dog_cartographer localization_imu_odom.launch.py  
load_state_filename:=~/home/yahboom/yahboomcar_ws/maps/mymap.pbstream
```

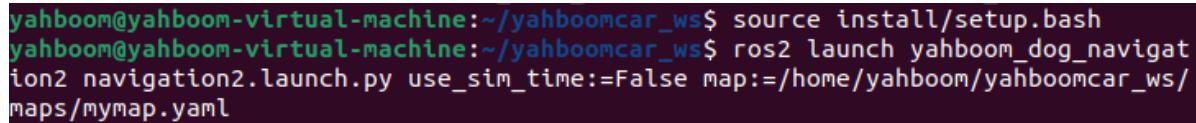


A terminal window titled "yahboom@yahboom-virtual-machine: ~/yahboomcar_ws". The log output shows the Cartographer node loading configuration files, finding trajectory builder scripts, loading a saved state from a pbstream file, adding a trajectory with ID '1', and inserting a submap (1, 0). The log entries are timestamped at 18:38:52.000000.

```
/share/cartographer/configuration_files/trajectory_builder_2d.lua' for 'trajectory_builder_2d.lua'.
[cartographer_node-1] [INFO] [1714300732.505165702] [cartographer logger]: I0428
18:38:52.000000 6171 configuration_file_resolver.cc:41] Found '/opt/ros/humble
/share/cartographer/configuration_files/trajectory_builder_3d.lua' for 'trajectory_builder_3d.lua'.
[cartographer_node-1] [INFO] [1714300732.505199532] [cartographer logger]: I0428
18:38:52.000000 6171 configuration_file_resolver.cc:41] Found '/opt/ros/humble
/share/cartographer/configuration_files/trajectory_builder_3d.lua' for 'trajectory_builder_3d.lua'.
[cartographer_node-1] [INFO] [1714300732.557136750] [cartographer logger]: I0428
18:38:52.000000 6171 map_builder_bridge.cpp:117] Loading saved state '/home/yahboom/yahboomcar_ws/maps/mymap.pbstream'...
[cartographer_node-1] [INFO] [1714300732.625084345] [cartographer logger]: I0428
18:38:52.000000 6171 map_builder_bridge.cpp:136] Added trajectory with ID '1'.
[cartographer_node-1] [INFO] [1714300732.691241270] [cartographer logger]: I0428
18:38:52.000000 6171 ordered_multi_queue.cc:172] All sensor data for trajectory 1 is available starting at '638498975326548929'.
[cartographer_node-1] [INFO] [1714300732.691513274] [cartographer logger]: I0428
18:38:52.000000 6171 local_trajectory_builder_2d.cc:135] Extrapolator is still
initializing.
[cartographer_node-1] [INFO] [1714300732.892889455] [cartographer logger]: I0428
18:38:52.000000 6171 pose_graph_2d.cc:148] Inserted submap (1, 0).
```

Repeat the above steps to open the terminal, reopen a terminal and enter navigation commands.

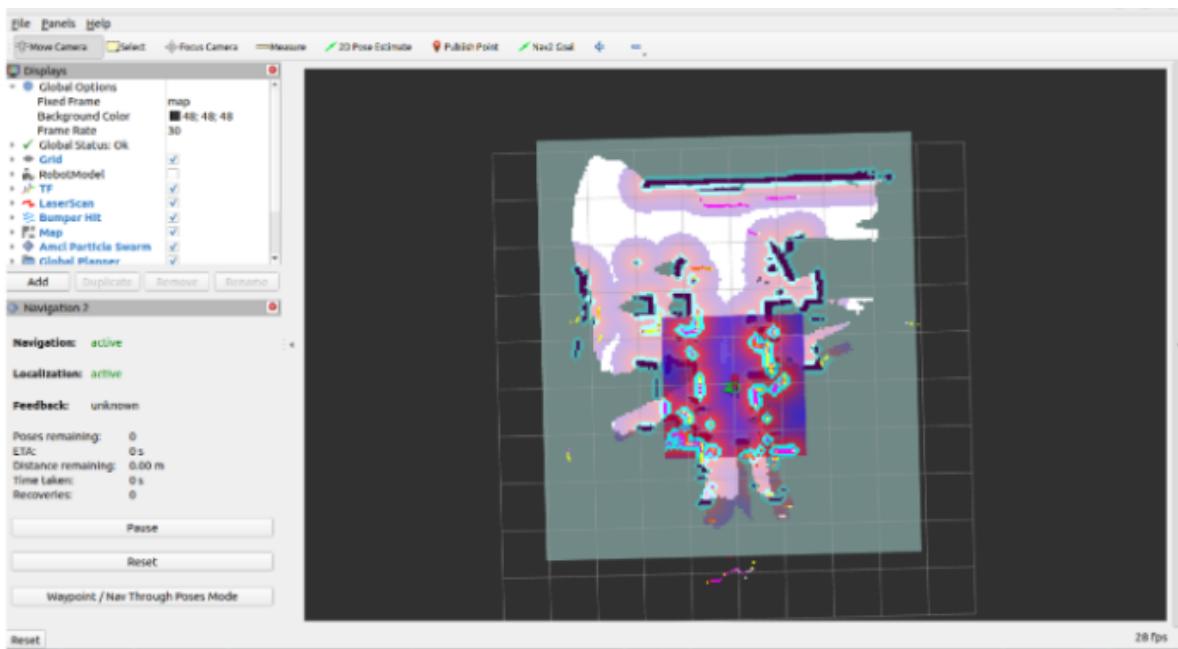
```
ros2 launch yahboom_dog_navigation2 navigation2.launch.py use_sim_time:=False  
map:=~/home/yahboom/yahboomcar_ws/maps/mymap.yaml
```



A terminal window titled "yahboom@yahboom-virtual-machine: ~/yahboomcar_ws\$". It shows the source command for setup.bash and then the execution of the navigation2.launch.py script with use_sim_time set to False and the map loaded from mymap.yaml.

```
yahboom@yahboom-virtual-machine:~/yahboomcar_ws$ source install/setup.bash
yahboom@yahboom-virtual-machine:~/yahboomcar_ws$ ros2 launch yahboom_dog_navigation2 navigation2.launch.py use_sim_time:=False map:=~/home/yahboom/yahboomcar_ws/maps/mymap.yaml
```

Then press the Enter key to navigate. (**Because the mechanical dog itself does not provide odom, we need to position the mechanical dog at the coordinate origin when constructing the map**)



4. Multi-point navigation node startup

The steps are the same for PI4 and PI5 versions:

After the navigation module is started, we open a terminal and start the multi-point navigation node. Enter the command in the newly opened terminal:

```
cd ~/cartoros2
```

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

```
ros2 run yahboom_nav2_send_goal yahboom_nav2_send_goal
```

```
yahboom@ubuntu:~/cartoros2$ source install/setup.bash
yahboom@ubuntu:~/cartoros2$ ros2 run yahboom_nav2_send_goal yahboom_nav2_send_goal
```

Open another terminal and start the node for network communication. Enter the command in the newly opened terminal:

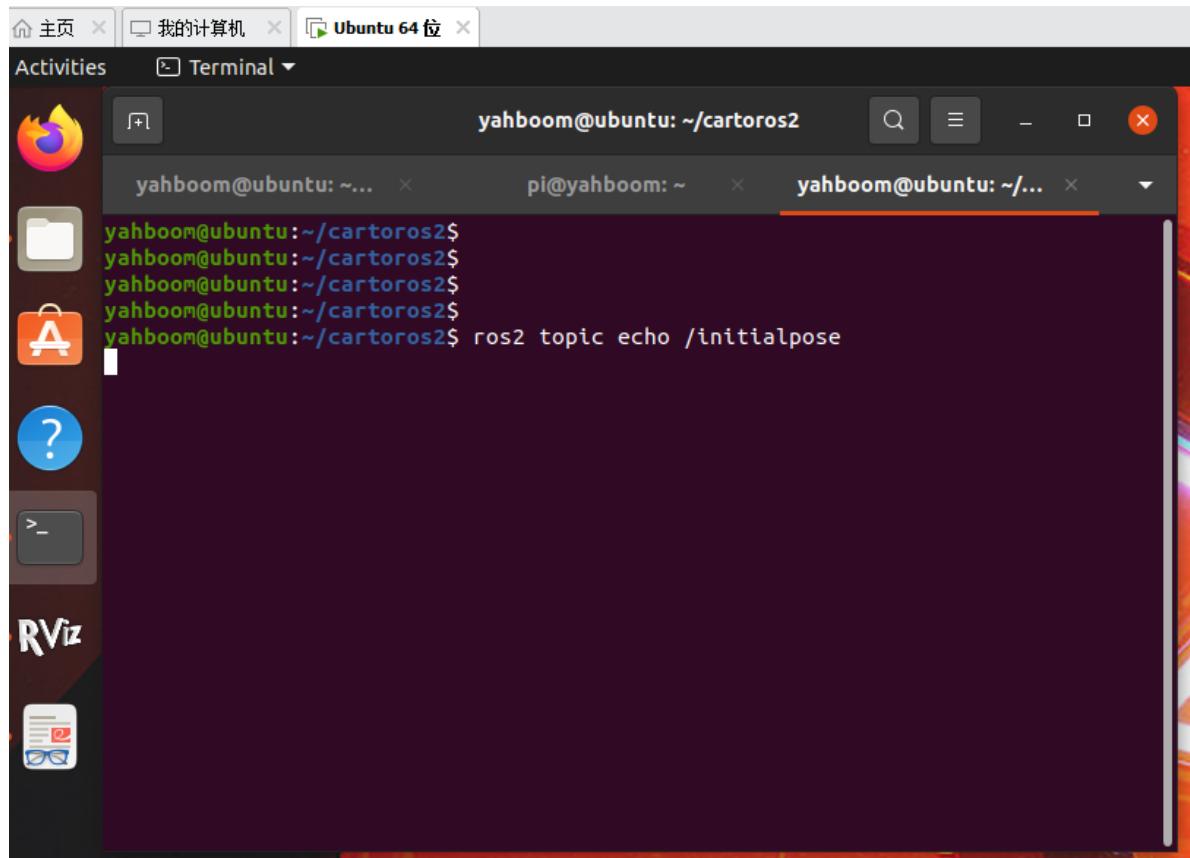
```
cd ~/cartoros2
source install/setup.bash
ros2 launch rosbridge_server rosbridge_websocket_launch.xml
```

5. Obtaining the navigation point position

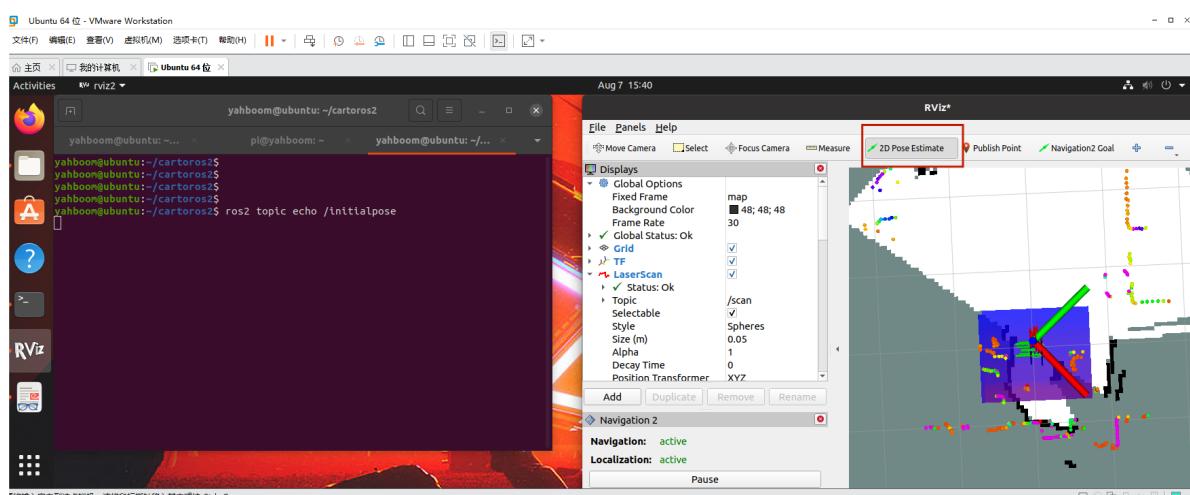
The steps are the same for PI4 and PI5 versions:

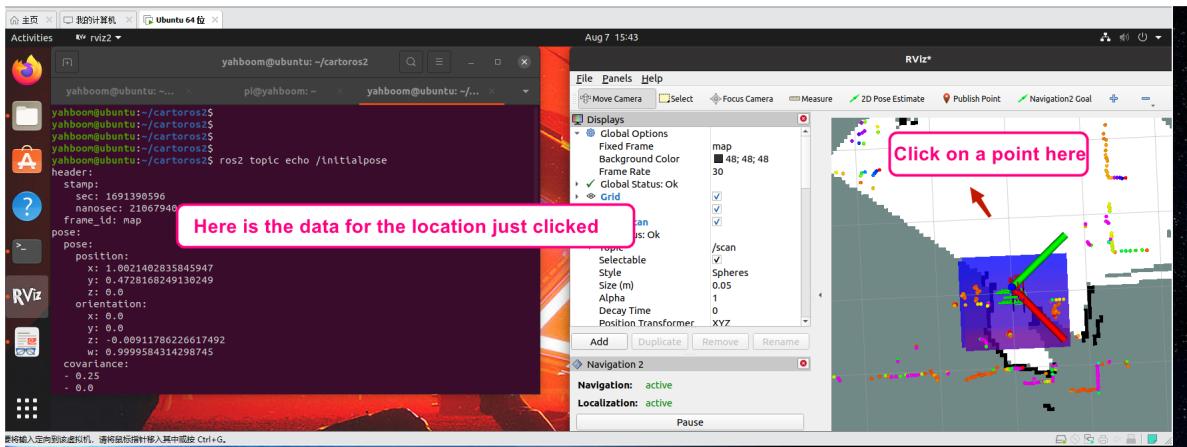
In the virtual machine terminal, press the shortcut key: ctrl + alt + T to reopen a terminal and subscribe to the /initialpose topic in the terminal to obtain the points we set in rviz. The command is as follows:

```
ros2 topic echo /initialpose
```



Click the button "2D Pose Estimate" in rviz and click a point on the map where you can walk. At this time, the location of the point just clicked will be printed in the terminal.



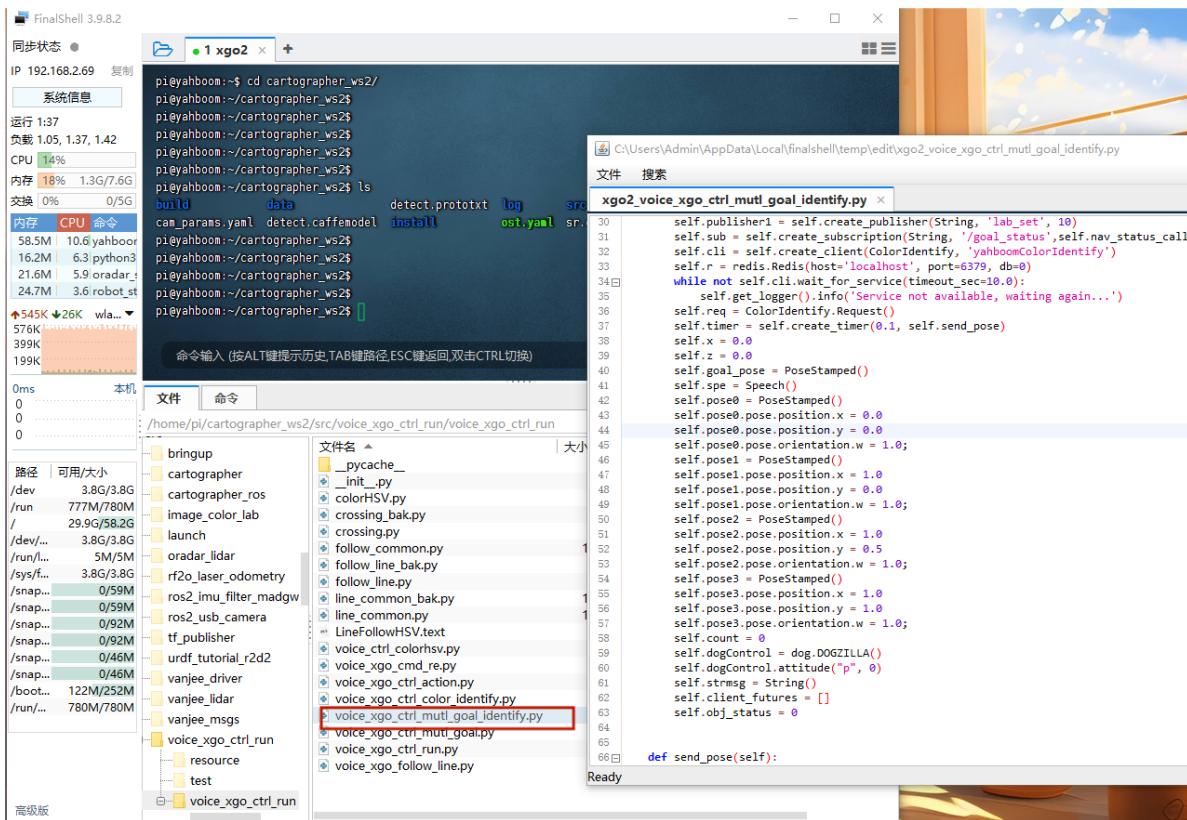


6. Voice navigation point modification

PI4 version steps:

According to the second step, open a new shell terminal, connect to the mechanical dog, and open the file in the path below.

```
#pi4
/home/pi/cartographer_ws2/install/voice_xgo_ctrl_run/lib/python3.8/site-
packages/voice_xgo_ctrl_run
```



Subscribe the corresponding position.x, position.y values to the data modification based on rviz.
As shown below:

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

```

pi@yahboom... x yahboom@u... yahbo...
C:\Users\Admin\AppData\Local\finalshell\temp\edit\xgo2_voice_xgo_ctrl_mutl_goal_identify.py
文件 搜索
xgo2_voice_xgo_ctrl_mutl_goal_identify.py x

stamp:
  sec: 1691569257
  nanosec: 493084935
frame_id: map
pose:
  position:
    x: 1.4283239841461182
    y: 0.4625771939754486
    z: 0.0
  orientation:
    x: 0.0
    y: 0.0
    z: -0.21076283150104438
    w: 0.9775372263283186
covariance:
- 0.25
- 0.0
- 0.0
- 0.0
- 0.0
- 0.0
- 0.0
- 0.25

```

```

30      self.publisher1 = self.create_publisher(String, 'lab_set', 10)
31      self.sub = self.create_subscription(String, '/goal_status',self.nav_stat
32      self.cli = self.create_client(ColorIdentify, 'yahboomColorIdentify')
33      self.r = redis.Redis(host='localhost', port=6379, db=0)
34      while not self.cli.wait_for_service(timeout_sec=10.0):
35          self.get_logger().info('Service not available, waiting again...')
36          self.req = ColorIdentify.Request()
37          self.timer = self.create_timer(0.1, self.send_pose)
38          self.x = 0.0
39          self.z = 0.0
40          self.goal_pose = PoseStamped()
41          self.spe = Speech()
42          self.pose0 = PoseStamped()
43          self.pose0.pose.position.x = 0.0
44          self.pose0.pose.position.y = 0.0
45          self.pose0.pose.orientation.w = 1.0;
46          self.pose1 = PoseStamped()
47          self.pose1.pose.position.x = 1.0
48          self.pose1.pose.position.y = 0.0
49          self.pose1.pose.orientation.w = 1.0;
50          self.pose2 = PoseStamped()
51          self.pose2.pose.position.x = 1.0
52          self.pose2.pose.position.y = 0.5
53          self.pose2.pose.orientation.w = 1.0;
54          self.pose3 = PoseStamped()
55          self.pose3.pose.position.x = 1.0
56          self.pose3.pose.position.y = 1.0
57          self.pose3.pose.orientation.w = 1.0;
58          self.count = 0
59          self.dogControl = dog.DOGZILLA()
60          self.dogControl.attitude("p", 0)
61          self.strmsg = String()
62          self.client_futures = []
63          self.ohi status = 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, there is no need to modify it.

PI5 version steps:

According to the method in the second step, enter the same docker terminal** (you can see the previous steps to enter the same docker step)**, enter the following command to open,

```
#path
cd /yahboomcar_ws/install/voice_xgo_ctrl_run/lib/python3.10/site-
packages/voice_xgo_ctrl_run
#Open the modified file
vi voice_xgo_ctrl_mutl_goal_identify.py
```

```
root@raspberrypi:~# cd /yahboomcar_ws/install/voice_xgo_ctrl_run/lib/python3.10/s
site-packages/voice_xgo_ctrl_run
root@raspberrypi:~/yahboomcar_ws/install/voice_xgo_ctrl_run/lib/python3.10/site-
packages/voice_xgo_ctrl_run# vi voice_xgo_ctrl_mutl_goal_identify.py
```

After opening, you can modify it according to the position.x, position.y values of the point pose1 just obtained. The values here are examples and should be subject to the values in your own environment. 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, there is no need to modify it.

```

root@raspberrypi: ~/yahboomcar_ws/install/voice_xgo_ctrl_run/lib/python3.... ▼ ^ >
File Edit Tabs Help Reset Please
while not self.cli.wait_for_service(timeout_sec=10.0):
    self.get_logger().info('Service not available, waiting again...')
self.req = ColorIdentify.Request()
self.timer = self.create_timer(0.1, self.send_pose)
self.x = 0.0
self.z = 0.0
self.goal_pose = PoseStamped()
self.spe = Speech()
self.pose0 = PoseStamped()
self.pose0.pose.position.x = 0.0
self.pose0.pose.position.y = 0.0
self.pose0.pose.orientation.w = 1.0;
self.pose1 = PoseStamped()
self.pose1.pose.position.x = 0.7165099382400513
self.pose1.pose.position.y = 0.05329164117574692
self.pose1.pose.orientation.w = 1.0;
self.pose2 = PoseStamped()
self.pose2.pose.position.x = 1.0
self.pose2.pose.position.y = 0.5
self.pose2.pose.orientation.w = 1.0;
self.pose3 = PoseStamped()
self.pose3.pose.position.x = 1.0
self.pose3.pose.position.y = 1.0
self.pose3.pose.orientation.w = 1.0;
self.count = 0
self.dogControl = dog.DOGZILLA()
self.dogControl.attitude("p", 0)
self.strmsg = String()
self.client_futures = []

```

48, 56 16%

After modification, save and exit.

vi editor operation: After opening the file, enter i to start editing. After editing is completed, press ESC, then enter: wq to save and exit the file!

8. Start the voice-controlled mechanical dog to find the ball

Open another shell or docker terminal and enter the following command in the terminal:

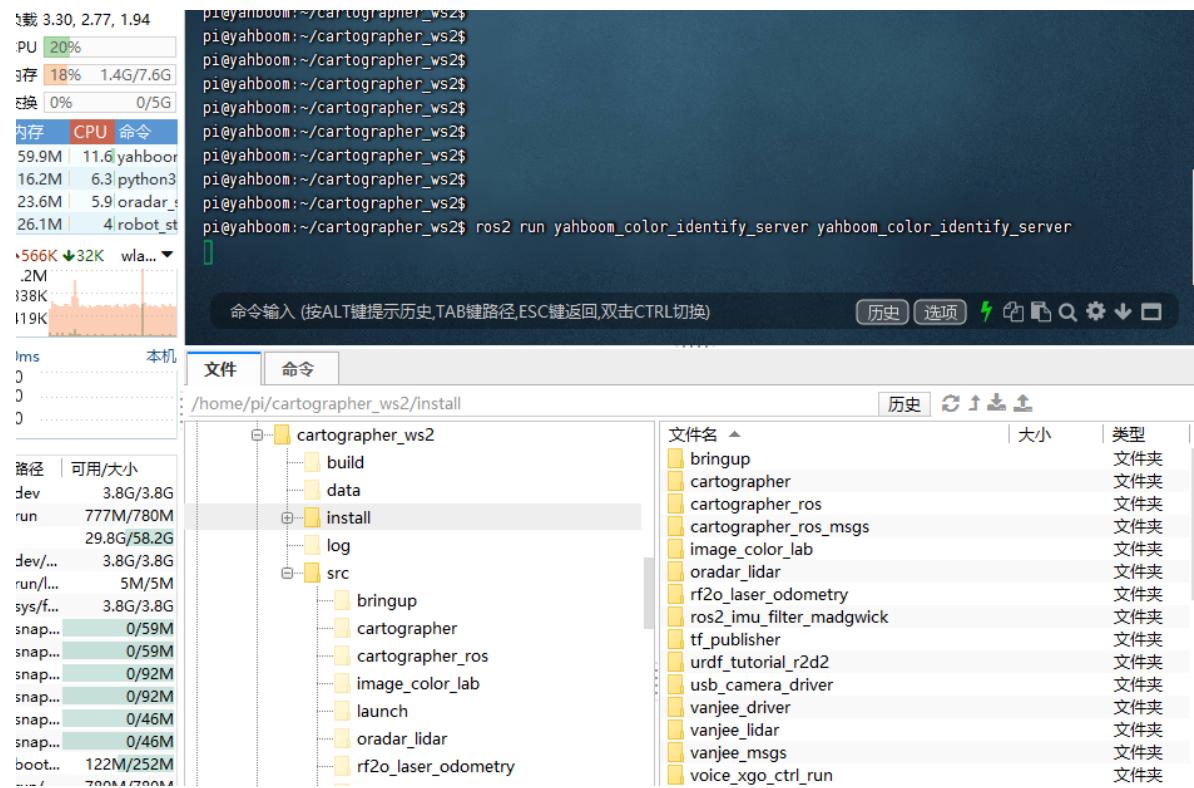
Note: This terminal is used to open the remote connection to the mechanical dog.

```

#pi4
cd ~/cartographer_ws2/
source install/setup.bash
ros2 run yahboom_color_identify_server yahboom_color_identify_server
#pi5 (need to enter docker terminal)
ros2 run yahboom_color_identify_server yahboom_color_identify_server

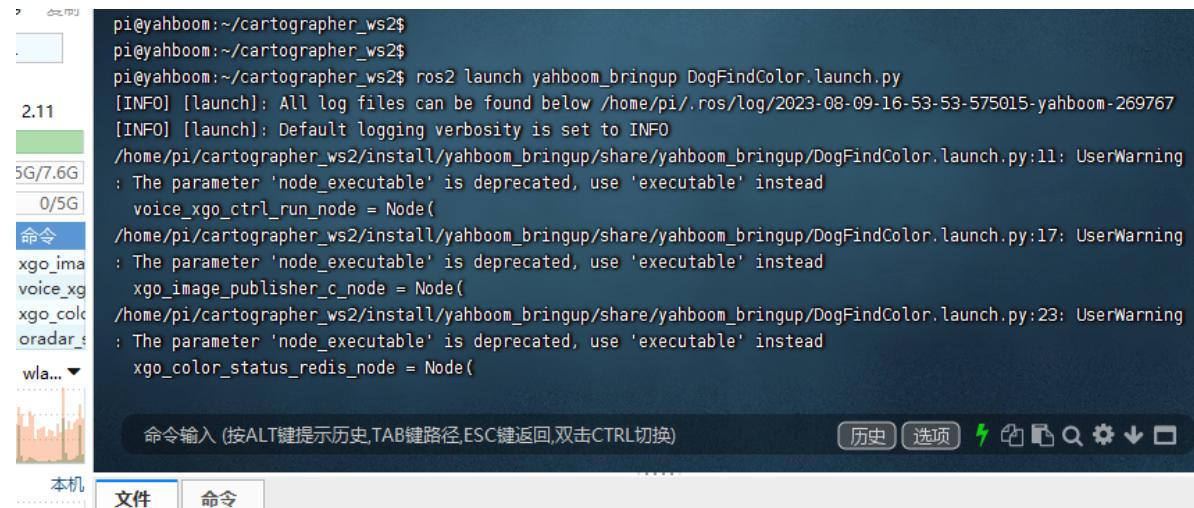
```

The following is the PI4 version as an example,



Open a new shell terminal and enter the command:

```
#pi4
cd ~/cartographer_ws2/
source install/setup.bash
ros2 launch yahboom Bringup DogFindColor.launch.py
#pi5 (need to enter the same docker terminal)
ros2 launch yahboom Bringup DogFindColor.launch.py
```



Then he said to the mechanical dog: "Hello, Xiaoya"

The mechanical dog replied: "Yes"

Then say to the mechanical dog: "" Navigate to position 1

As shown in the picture below, the mechanical dog will automatically navigate to position 1.

