ROS2 environment entity mechanical dog APP mapping

Quick use

1. Power on the mechanical dog

PI4 version steps:

Press the power switch on the side of the mechanical dog and wait for the mechanical dog to start up. And connect the robot dog's WiFi to the same LAN environment as the computer.

After the mechanical dog is started, the laser radar, imu, and mechanical dog joint status nodes have been automatically started.

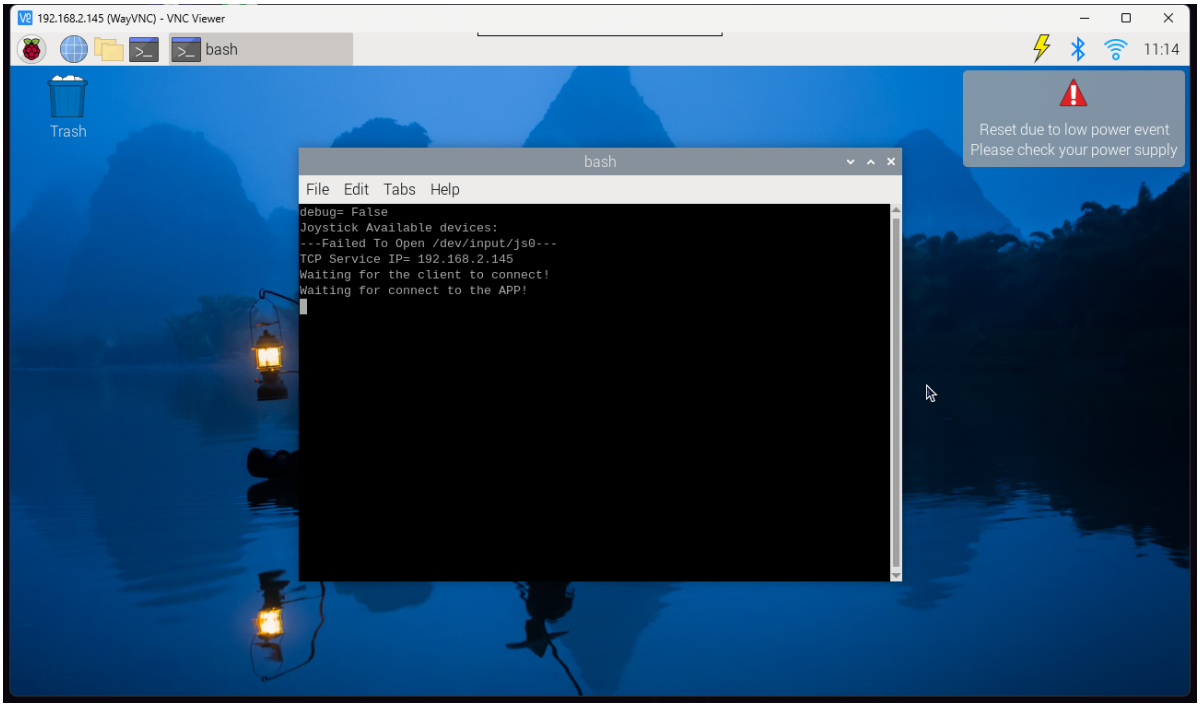
If you find that you cannot obtain lidar and other data, please close the mechanical dog program and restart the chassis program.

To close the large program, restart the chassis and modify the multi-level communication ID, please refer to the tutorial: 14. Radar mapping navigation\6. Obtaining the status of the physical mechanical dog in the ROS2 environment\Acquiring the real joint data of the mechanical dog in the ROS2 environment.pdf

PI5 version steps:

Press the power switch on the side of the mechanical dog and wait for the mechanical dog to start up. And connect the robot dog's WiFi to the same LAN environment as the computer.

After the mechanical dog is started, remotely connect to the mechanical dog through the IP address on the OLED.



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.

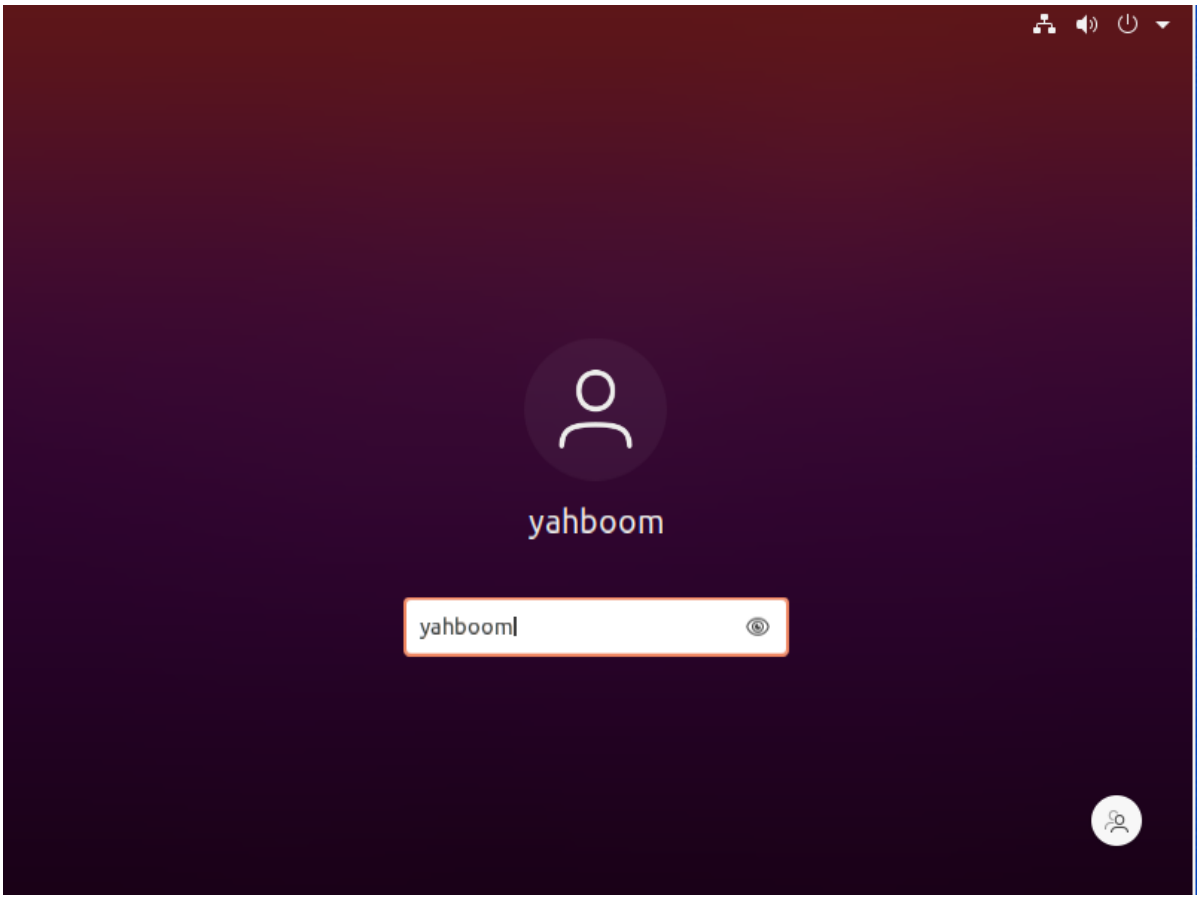
```
ros2 launch bringup Navigation_bringup.launch.py
```

```
root@raspberrypi: /
File  Edit  Tabs  Help
at 0x7fff363522f0>
[yahboomcar_joint_state-3] [8.06, 42.49, 0.85, 7.04, 43.12, 0.85, 12.14, 54.34,
0.85, 11.12, 54.96, 0.61]
[yahboomcar_joint_state-3] &&&&&&&&&&&&&&&&&&&&&&&&& 0.11081695556640625
[yahboomcar_joint_state-3] ##################
[yahboomcar_joint_state-3] [0.24045410156250002, -0.1160400390625, -9.7198486328
125, -1.0975609756097562, -0.48780487804878053, -0.5487804878048781, 0.008936438
739299774, 0.013750852101378972, -0.12733232821358575]
[yahboomcar_joint_state-3] ************************** <rclpy.timer.Timer object
at 0x7fff363522f0>
[yahboomcar_joint_state-3] [8.06, 42.49, 0.85, 7.04, 43.12, 0.85, 12.14, 54.34,
0.85, 11.12, 54.96, 0.61]
[yahboomcar_joint_state-3] &&&&&&&&&&&&&&&&&&&&&&&&& 0.11193227767944336
[yahboomcar_joint_state-3] ##################
[yahboomcar_joint_state-3] [0.23327636718750003, -0.14833984375, -9.7330078125,
-1.0365853658536586, -0.6097560975609757, -0.6707317073170732, 0.009365924020608
266, 0.013580941220124564, -0.12733232821358575]
[yahboomcar_joint_state-3] ************************** <rclpy.timer.Timer object
at 0x7fff363522f0>
[yahboomcar_joint_state-3] [8.06, 42.49, 0.85, 7.04, 43.12, 0.85, 12.14, 54.34,
0.85, 11.12, 54.96, 0.61]
[yahboomcar_joint_state-3] &&&&&&&&&&&&&&&&&&&&&&&&& 0.11203646659851074
[yahboomcar_joint_state-3] ##################
```

2. Open the virtual machine

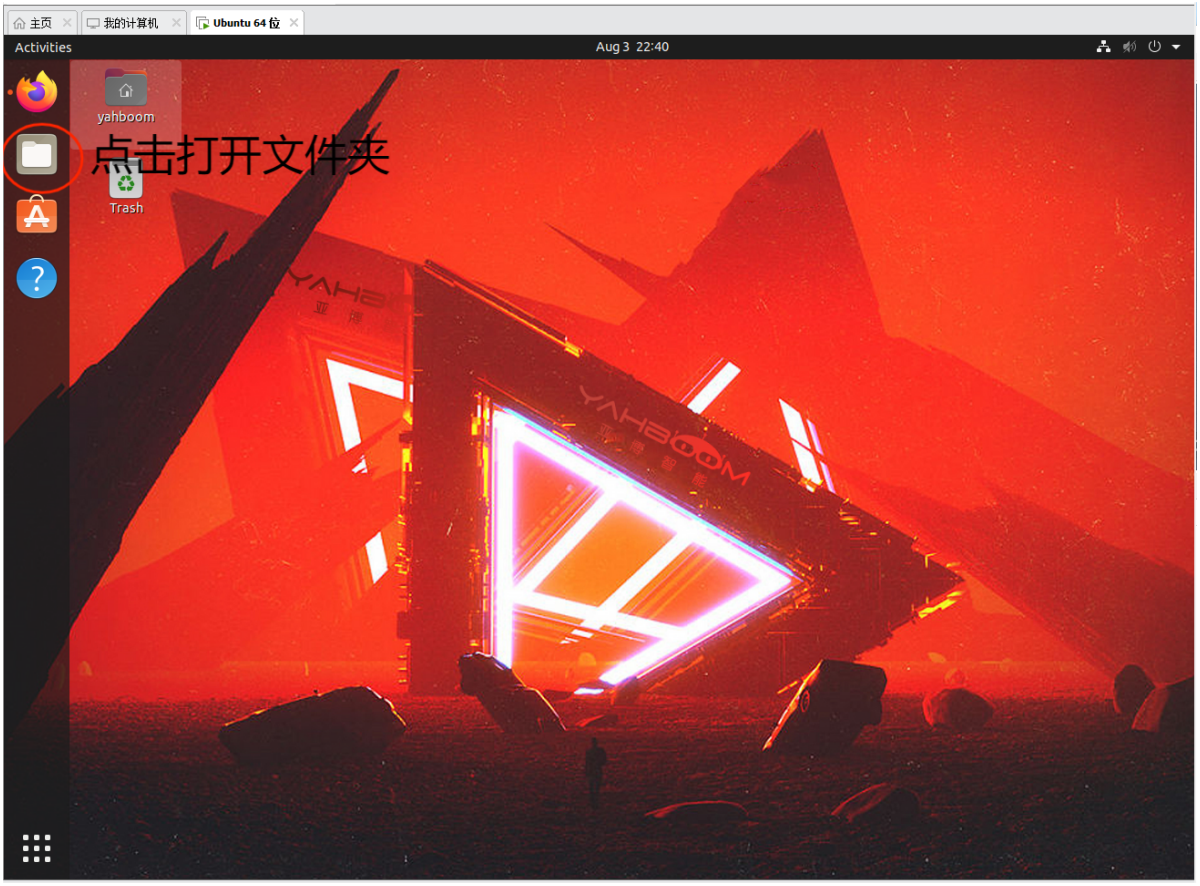
Note: The default virtual machine has been installed here.

Open the virtual machine, enter the password: yahboom and press the Enter key to enter the system desktop.

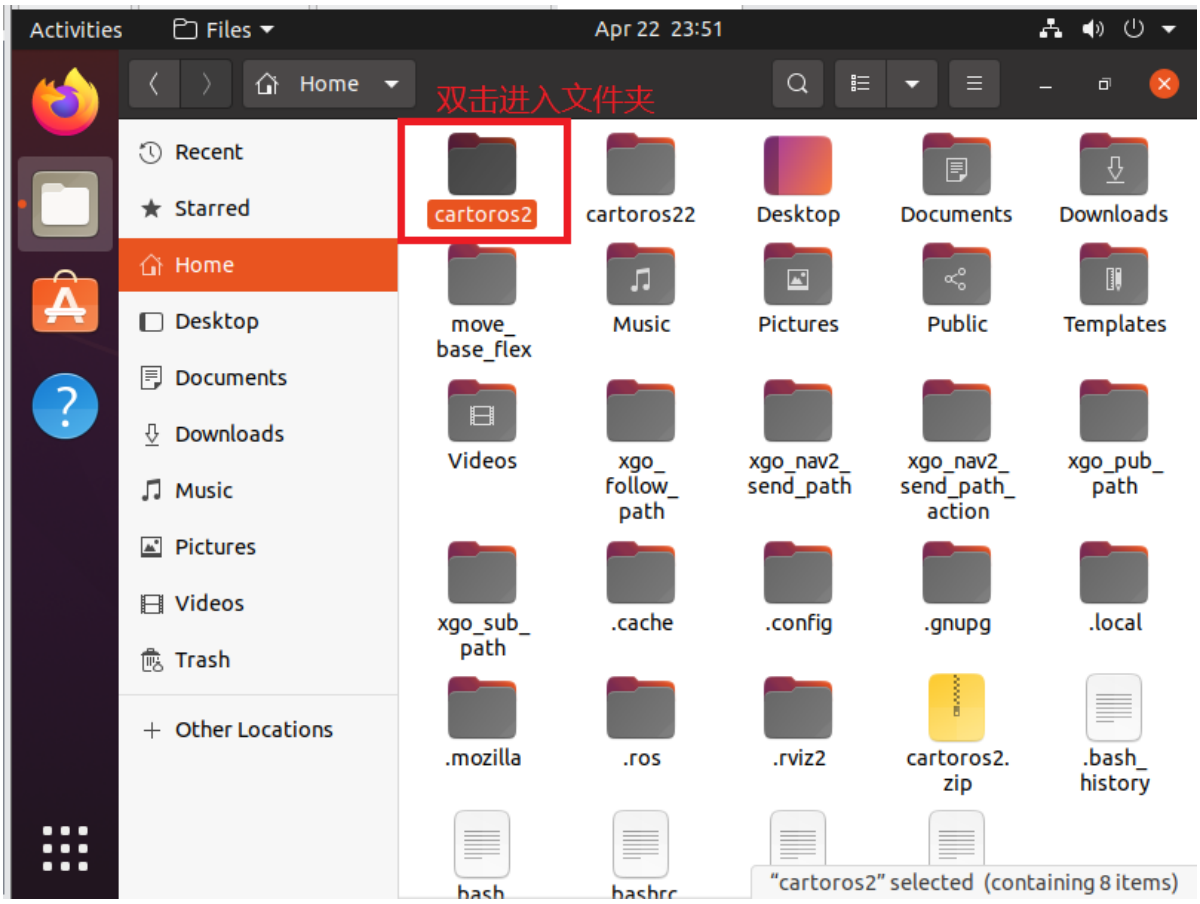


3. Start ROSBridge

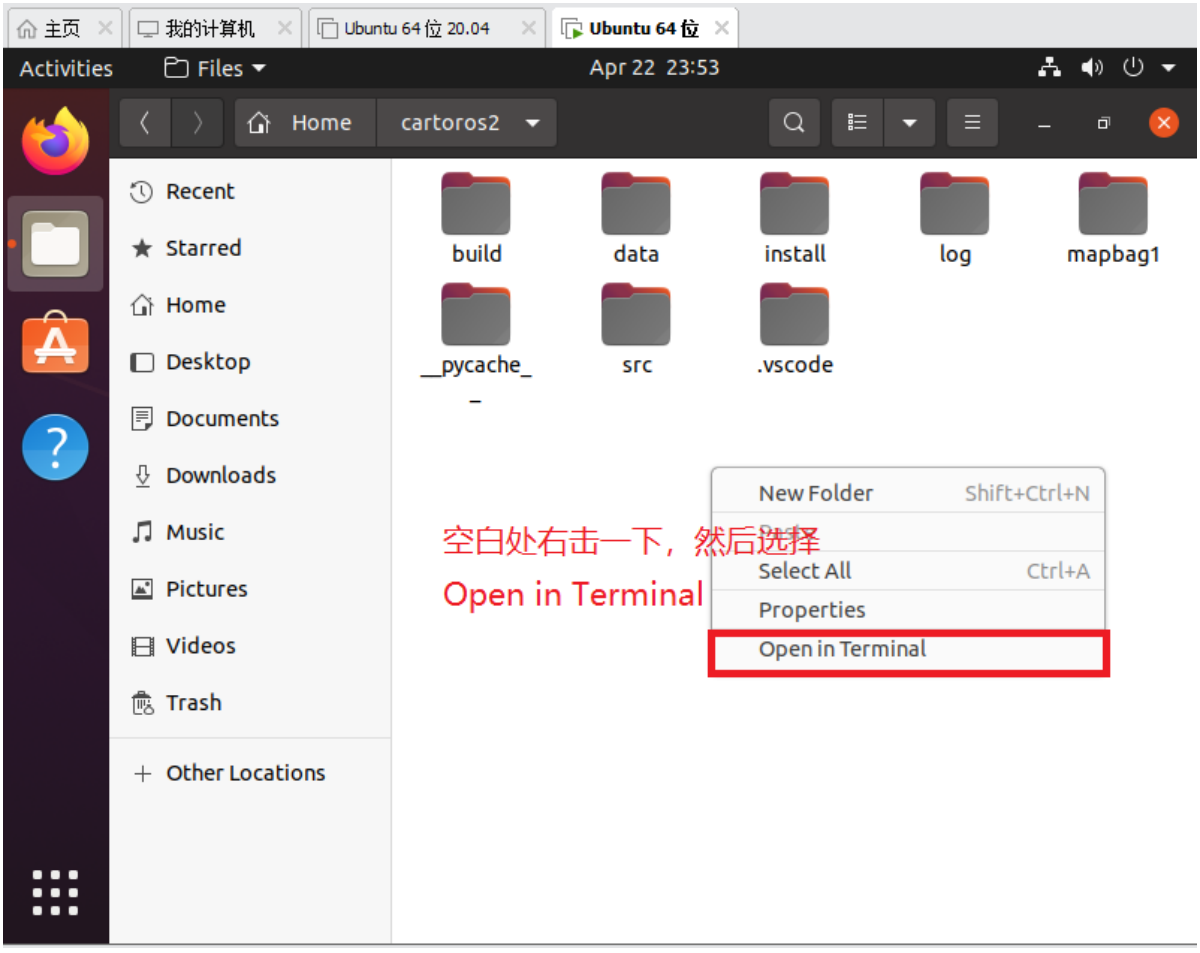
Note: This step is the same for virtual machines of PI4 and PI5 versions



Then double click on the cartoros2 folder



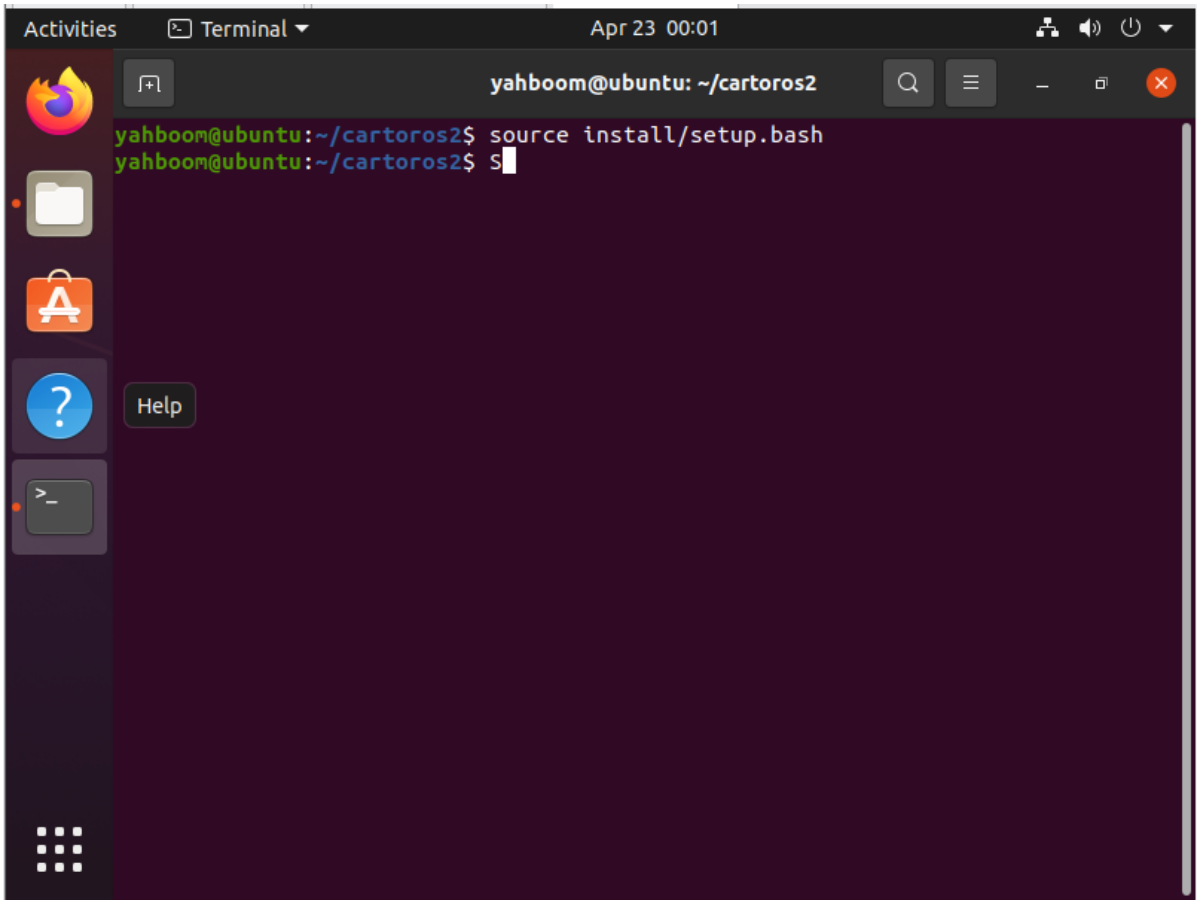
Then right-click in an empty space of the folder and select Open in Terminal



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

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

After completing the input, press the Enter key.



Then enter the following command and press Enter to start rosbridge

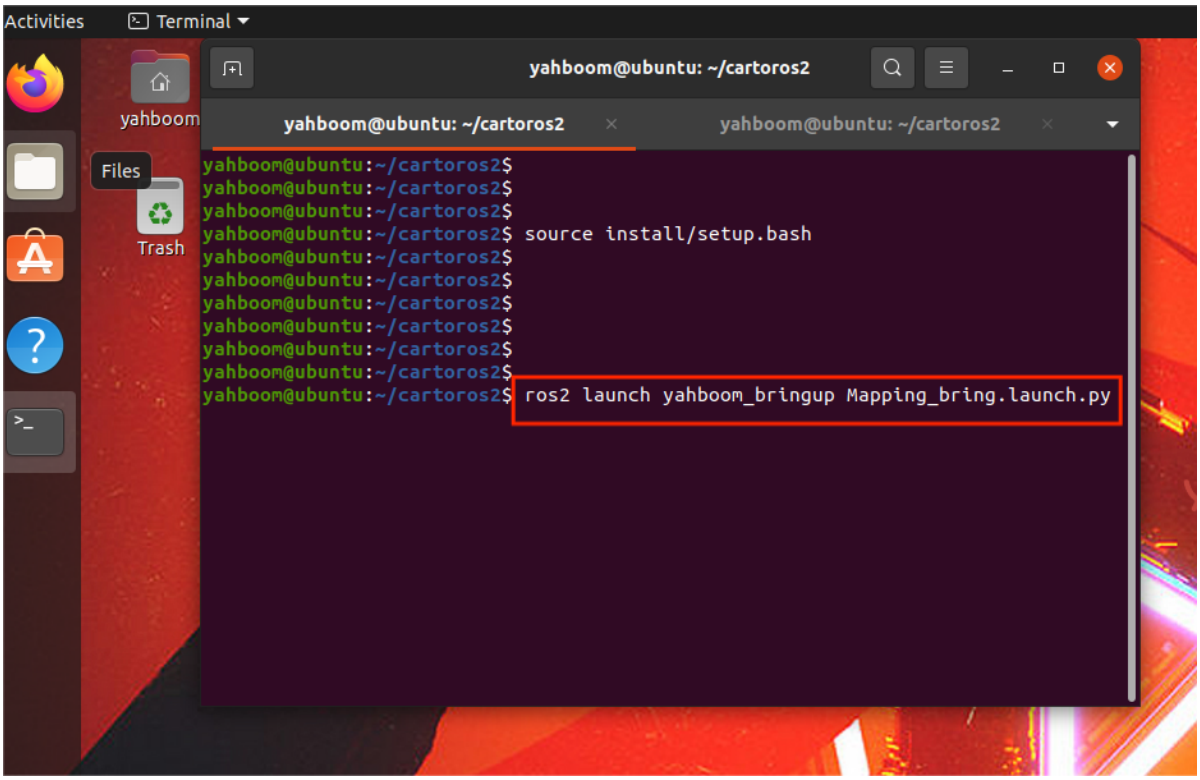
```
ros2 launch rosbridge_server rosbridge_websocket_launch.xml
```

Then press the shortcut key: ctrl + shift + t

Enter the command in the newly opened terminal

```
source install/setup.bash
ros2 launch yahboom_bringup Mapping_bring.launch.py
```

Then press the Enter key to start drawing.

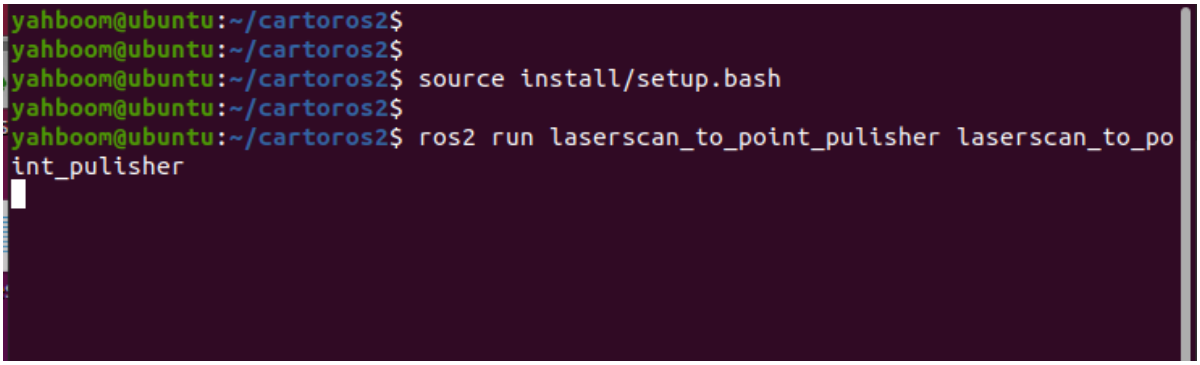


4. Control the mechanical dog to walk and build a map

Click on the terminal and press the shortcut key: ctrl + shift + t

Enter the command in the terminal to start the point cloud publishing node:

```
source install/setup.bash
ros2 run laserscan_to_point_pulisher laserscan_to_point_pulisher
```



Click on the terminal and press the shortcut key: ctrl + shift + t

Enter the command in the terminal to start the positioning publishing node:

```
source install/setup.bash
ros2 launch robot_pose_publisher_ros2 robot_pose_publisher_launch.py
```

5. Camera display

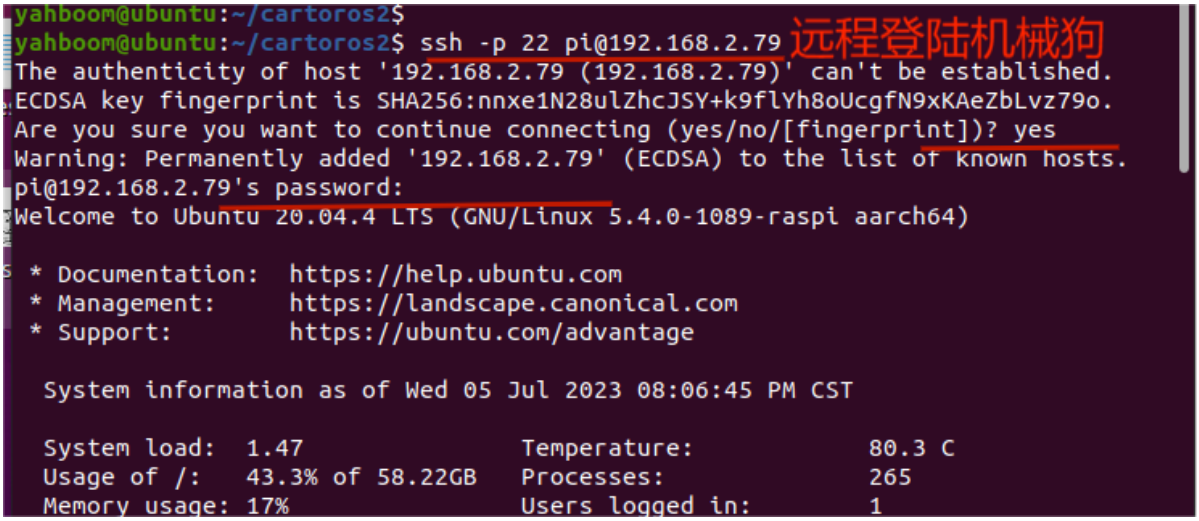
PI4 version steps:

Click on the terminal and press the shortcut key: ctrl + shift + t

Enter the command in the terminal to remotely connect to the robot dog. Currently, the robot dog and the virtual machine are in the same LAN. The IP of the robot dog is: 192.168.2.79. In actual operation, it depends on the specific IP address of your own robot dog.

```
ssh -p 22pi@192.168.2.79
```

Press Enter and enter yes, then enter the login password of the mechanical dog: yahboom



Then enter the command to start the camera in the terminal:

```
ros2 launch usb_cam demo_launch.py
```

```
pi@yahboom:~$
pi@yahboom:~$ ros2 launch usb_cam demo_launch.py
[INFO] [launch]: All log files can be found below /home/pi/.ros/log/2023-07-05-2
0-13-03-911670-yahboom-89445
[INFO] [launch]: Default logging verbosity is set to INFO
'opt/ros/foxy/share/usb_cam/config/params.yaml
 INFO] [usb_cam_node_exe-1]: process started with pid [89529]
[usb_cam_node_exe-1] [INFO] [1688559184.442941567] [usb_cam]: camera_name value:
 test_camera
[usb_cam_node_exe-1] [WARN] [1688559184.443275107] [usb_cam]: framerate: 30.0000
00
[usb_cam_node_exe-1] [INFO] [1688559184.461739301] [usb_cam]: camera calibration
 URL: package://usb_cam/config/camera_info.yaml
[usb_cam_node_exe-1] [INFO] [1688559184.464124605] [usb_cam]: Starting 'test_cam
era' (/dev/video0) at 640x480 via mmap (mjpeg2rgb) at 30 FPS
[usb_cam_node_exe-1] [swscaler @ 0xaaaaf30867f0] No accelerated colorspace conve
rsion found from yuv422p to rgb24.
[usb_cam_node_exe-1] [INFO] [1688559184.545634240] [usb_cam]: This devices suppr
```

启动相机节点

PI5 version steps:

Open a terminal in the virtual machine, enter the IP on the mechanical dog, and log in remotely. The example IP here is 192.168.2.145.

```
ssh -p 22pi@192.168.2.145
```

Press Enter and enter yes, then enter the login password of the mechanical dog: yahboom

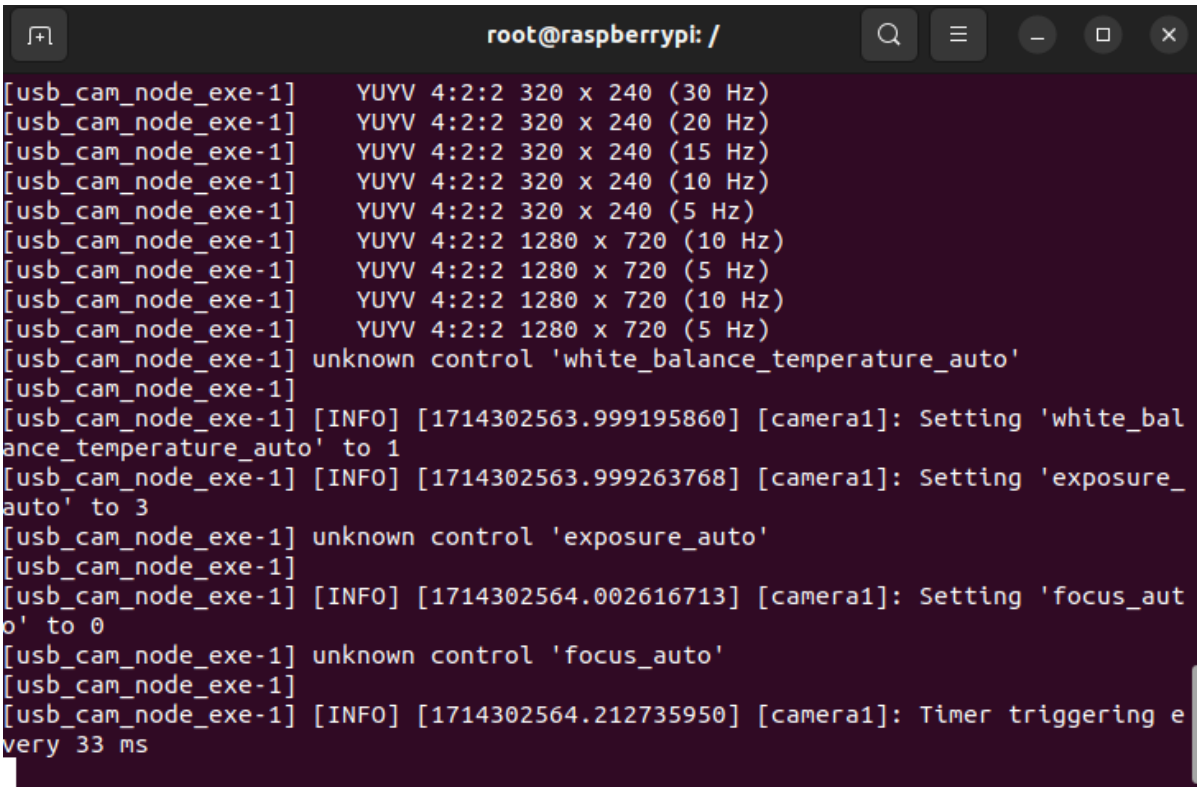
Then enter docker ps to view the current ID, and then enter the command to enter the same docker

```
docker ps
docker exec -it (id)/bin/bash
```

```
pi@raspberrypi:~ $ docker ps
CONTAINER ID   IMAGE                                COMMAND       CREATED
 STATUS          PORTS     NAMES
5029cf9a6f99   yahboomtechnology/ros-humble:3.0   "/bin/bash"   50 minutes ago
 Up 50 minutes             loving_black
pi@raspberrypi:~ $ docker exec -it 5029 /bin/bash
root@raspberrypi:/#
```

Then enter the command to start the camera in the terminal:

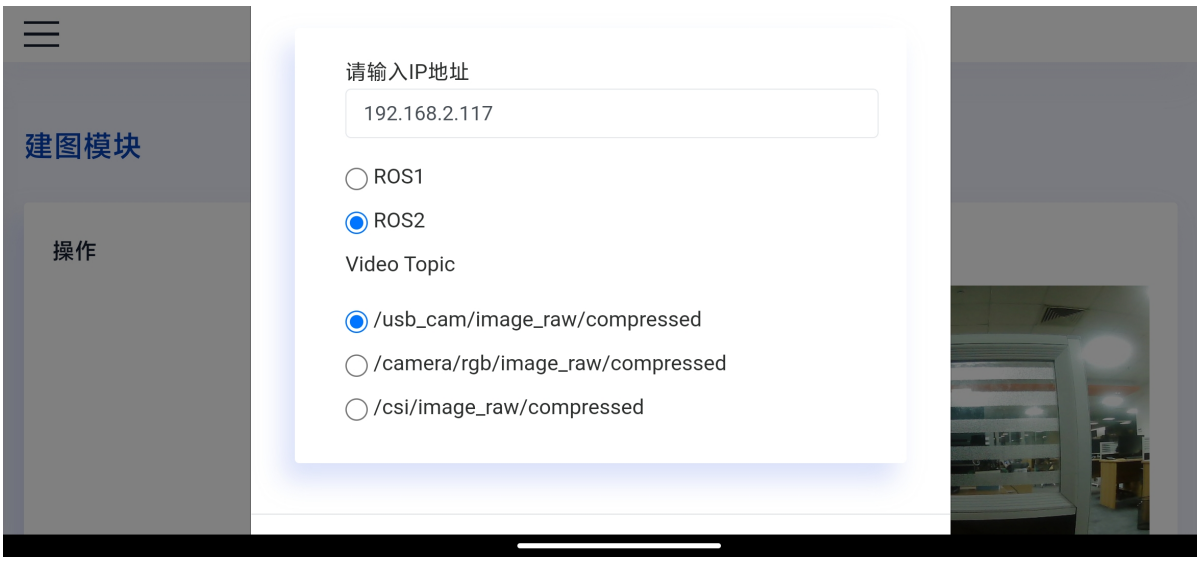
```
ros2 launch usb_cam camera.launch.py
```



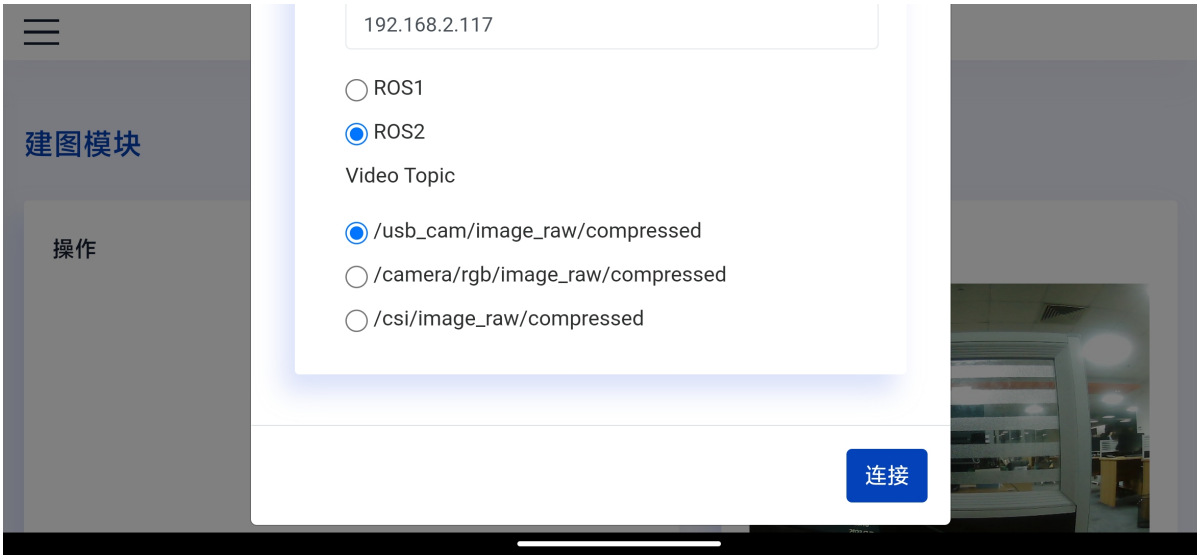
6. APP display

Open the mobile APP ROS Robot

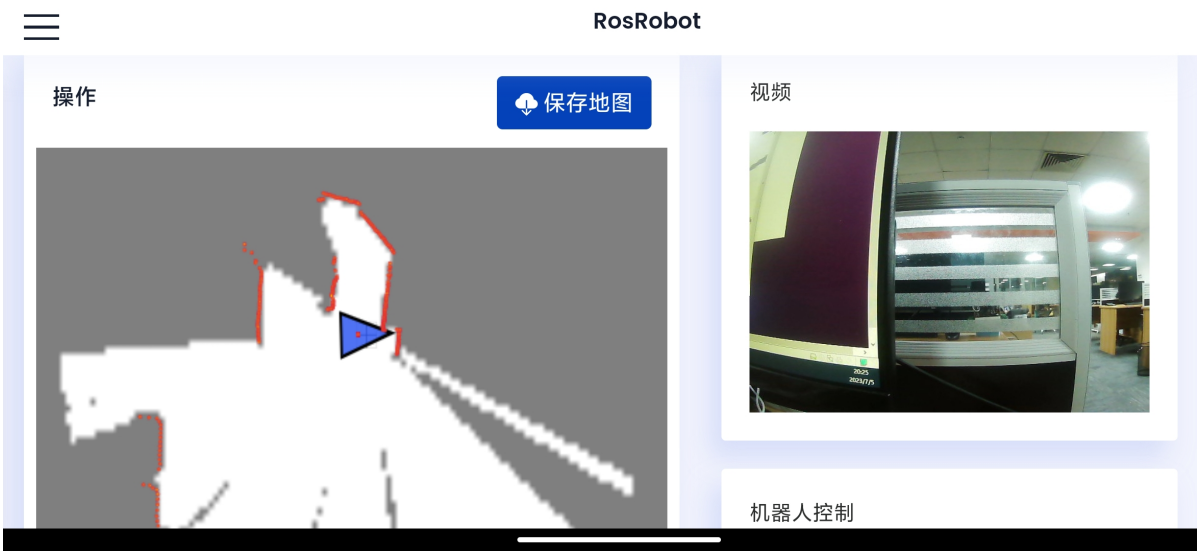
Enter the IP address of the virtual machine. Take the current virtual machine IP as: 192.168.2.117 as an example. At the same time, select the camera topic of the mechanical dog as /usb_cam/imge_raw/compress and select ros2.



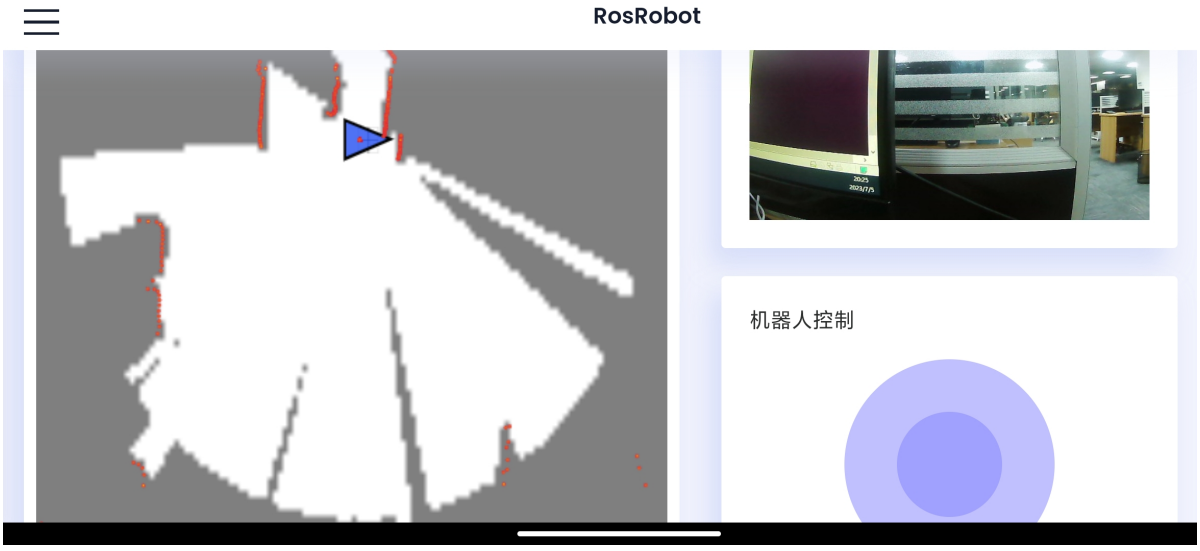
Then swipe down and click Connect.



After entering the map interface, you can see the map scanned by the radar and the camera image.



Swipe down to see the remote sensing control in the lower right corner. You can use remote sensing to control the mechanical dog for mapping.



7. Save the map

When the map scan is completed, we press the shortcut key Ctrl shift t in the terminal to reopen a terminal window and enter the following command

First enter the following command to activate the environment

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

Then enter the following command to stop mapping,

```
ros2 service call /finish_trajectory cartographer_ros_msgs/srv/FinishTrajectory "
{trajectory_id: 0}"
```

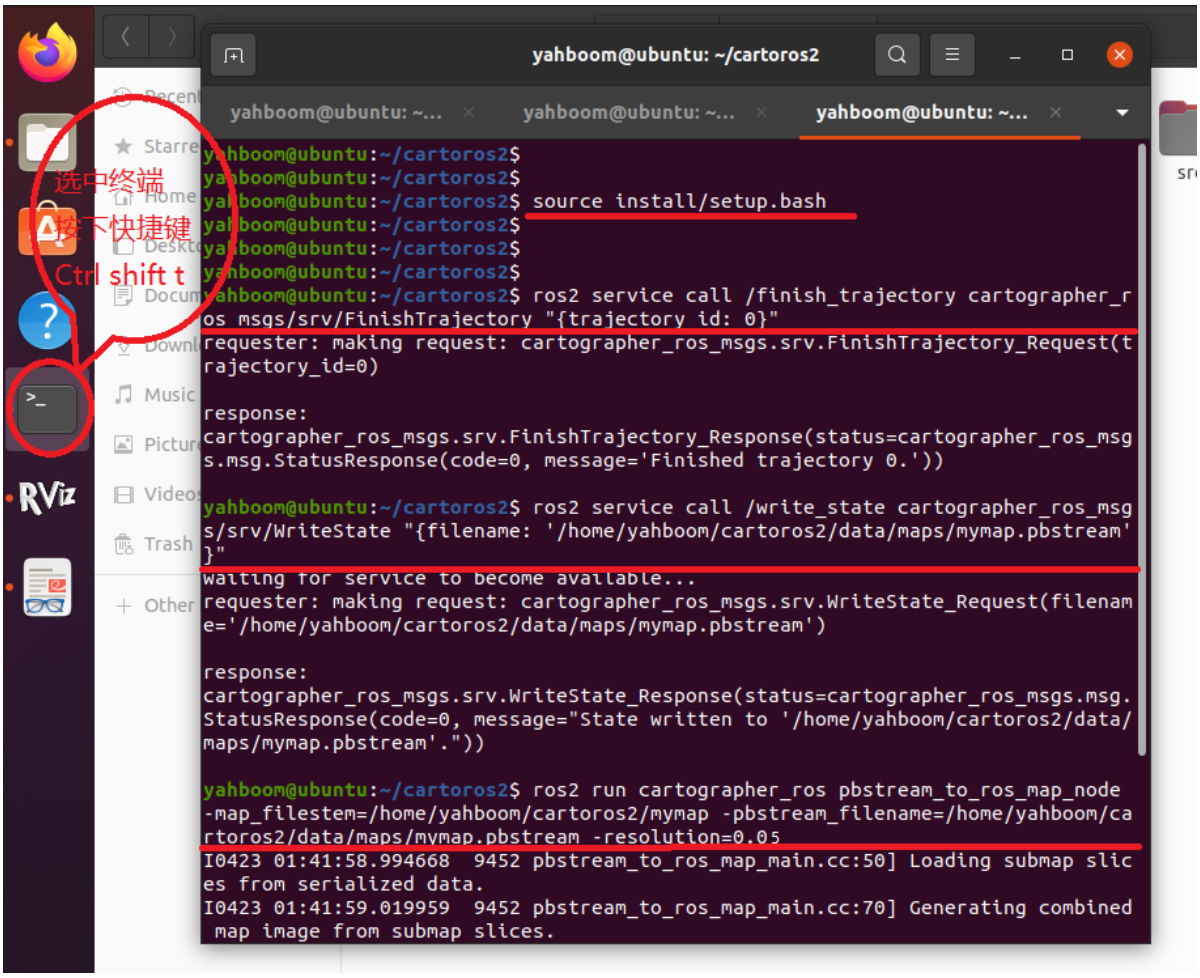
Then enter the following command to save the pbstream file

```
#pi4
ros2 service call /write_state cartographer_ros_msgs/srv/WriteState "{filename:
'/home/yahboom/cartoros2/data/maps/mymap.pbstream'}"

#pi5
ros2 service call /write_state cartographer_ros_msgs/srv/WriteState "{filename:
'/home/yahboom/yahboomcar_ws/maps/mymap.pbstream'}"
```

The path of the filename parameter is the path where the pbstream file of the map is saved.

Finally enter the following command to convert the pbstream file to a pgm file.

```
#pi4
ros2 run cartographer_ros pbstream_to_ros_map_node -
map_filestem=/home/yahboom/cartoros2/data/maps/mymap -
pbstream_filename=/home/yahboom/cartoros2/data/maps/mymap.pbstream -
resolution=0.05

#pi5
ros2 run cartographer_ros cartographer_pbstream_to_ros_map -
map_filestem=/home/yahboom/yahboomcar_ws/maps/mymap -
pbstream_filename=/home/yahboom/yahboomcar_ws/maps/mymap.pbstream -
resolution=0.05
```


选中终端
按下快捷键
Ctrl shift t
yahboom@ubuntu: ~/cartoros2
yahboom@ubuntu: ~...
yahboom@ubuntu: ~...
yahboom@ubuntu: ~...
yahboom@ubuntu:~/cartoros2$
yahboom@ubuntu:~/cartoros2$
yahboom@ubuntu:~/cartoros2$ source install/setup.bash
yahboom@ubuntu:~/cartoros2$
yahboom@ubuntu:~/cartoros2$
yahboom@ubuntu:~/cartoros2$
yahboom@ubuntu:~/cartoros2$ ros2 service call /finish_trajectory cartographer_ros msgs/srv/FinishTrajectory "{trajectory_id: 0}"
requester: making request: cartographer_ros_msgs.srv.FinishTrajectory_Request(trajectory_id=0)
response:
cartographer_ros_msgs.srv.FinishTrajectory_Response(status=cartographer_ros_msgs.msg.StatusResponse(code=0, message='Finished trajectory 0.'))
yahboom@ubuntu:~/cartoros2$ ros2 service call /write_state cartographer_ros_msgs/srv/WriteState "{filename: '/home/yahboom/cartoros2/data/maps/mymap.pbstream'}"
waiting for service to become available...
requester: making request: cartographer_ros_msgs.srv.WriteState_Request(filename='/home/yahboom/cartoros2/data/maps/mymap.pbstream')
response:
cartographer_ros_msgs.srv.WriteState_Response(status=cartographer_ros_msgs.msg.StatusResponse(code=0, message="State written to '/home/yahboom/cartoros2/data/maps/mymap.pbstream'."))
yahboom@ubuntu:~/cartoros2$ ros2 run cartographer_ros pbstream_to_ros_map_node -map_filestem=/home/yahboom/cartoros2/mymap -pbstream_filename=/home/yahboom/cartoros2/data/maps/mymap.pbstream -resolution=0.05
I0423 01:41:58.994668 9452 pbstream_to_ros_map_main.cc:50] Loading submap slices from serialized data.
I0423 01:41:59.019959 9452 pbstream_to_ros_map_main.cc:70] Generating combined map image from submap slices.