tag identification

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.

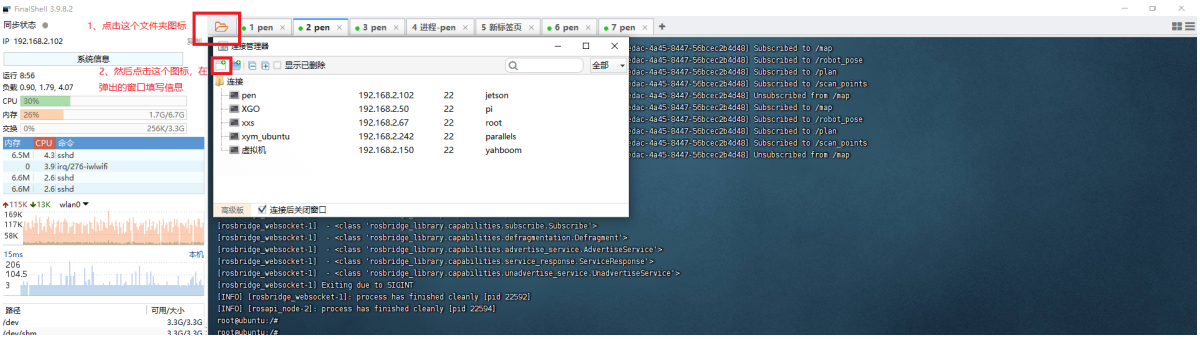
2. Start DOGZILLA chassis

PI4 version steps:

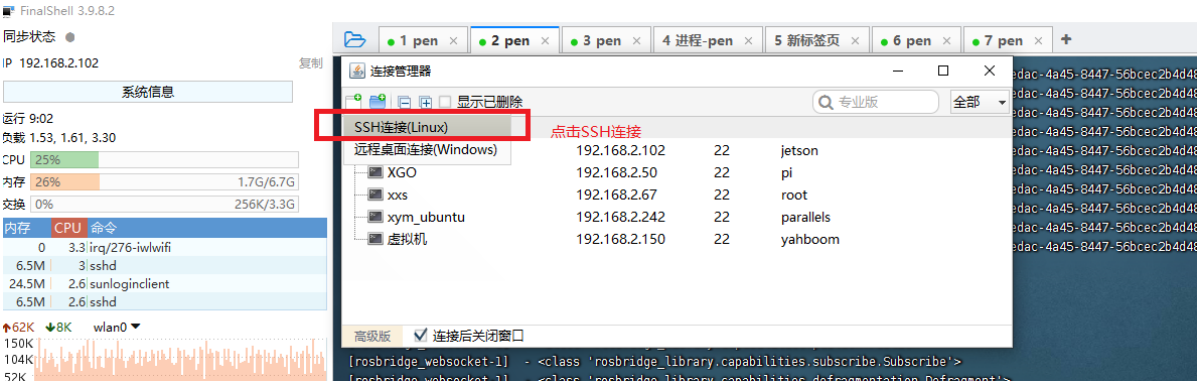
Then use the ssh terminal to connect to the robot dog.

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.

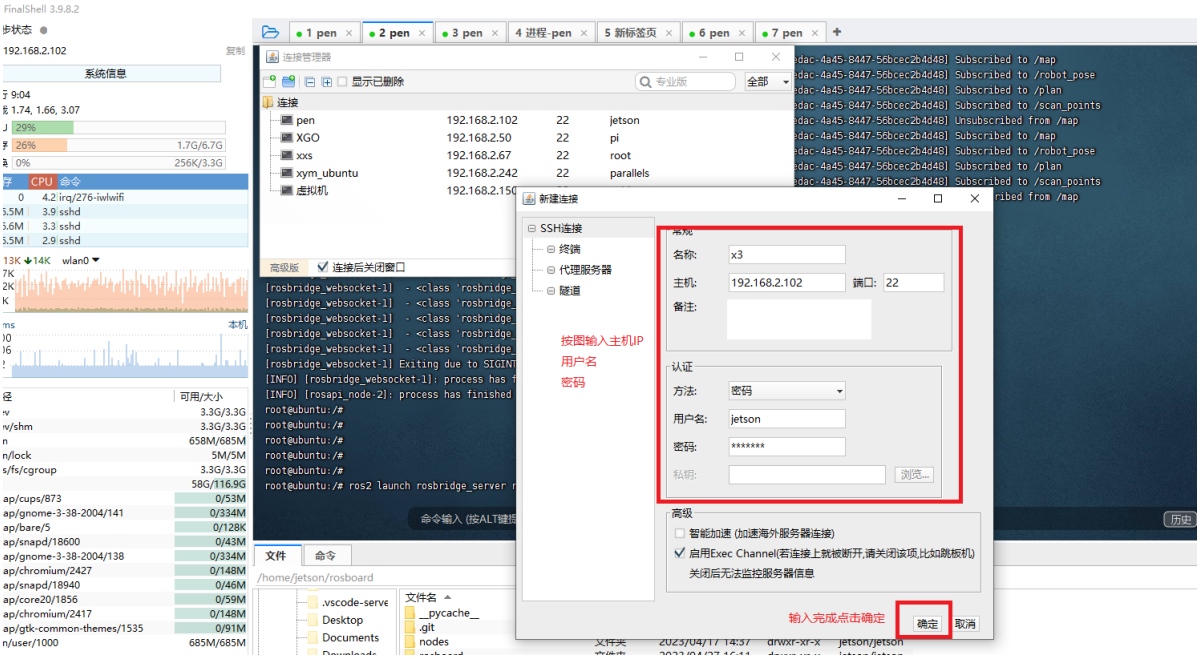
Open the shell tool. The shell tool I use here is FinalShell. Enter username, password, port, connection name and other information.



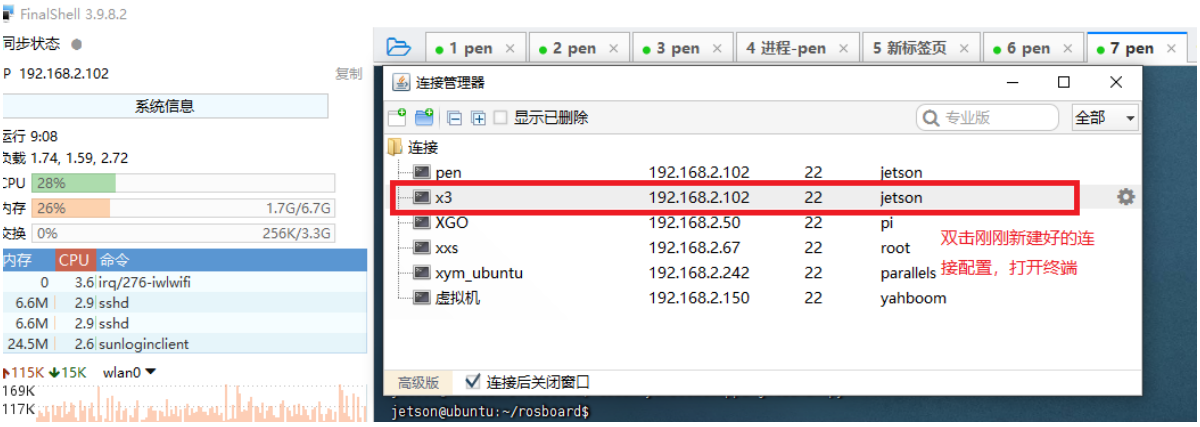
Select ssh connection to create a new ssh connection



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

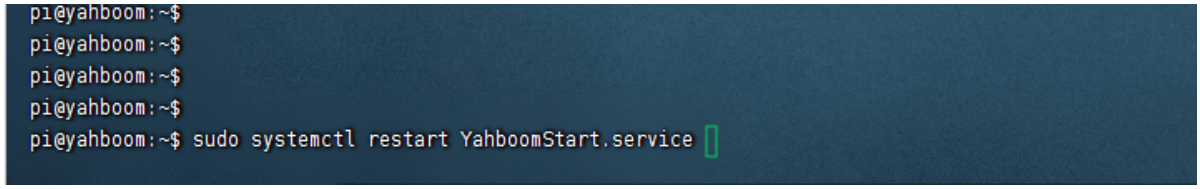


Select the ssh connection you just created here.



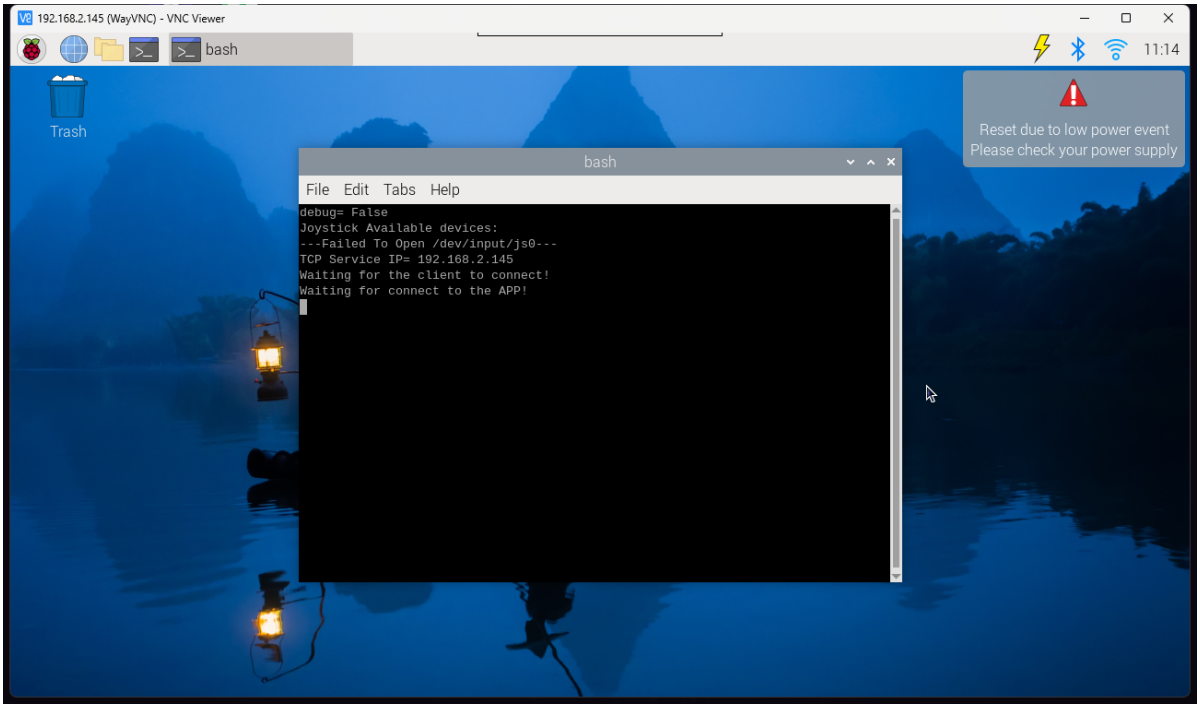
Enter the command in the terminal to start the chassis task.

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



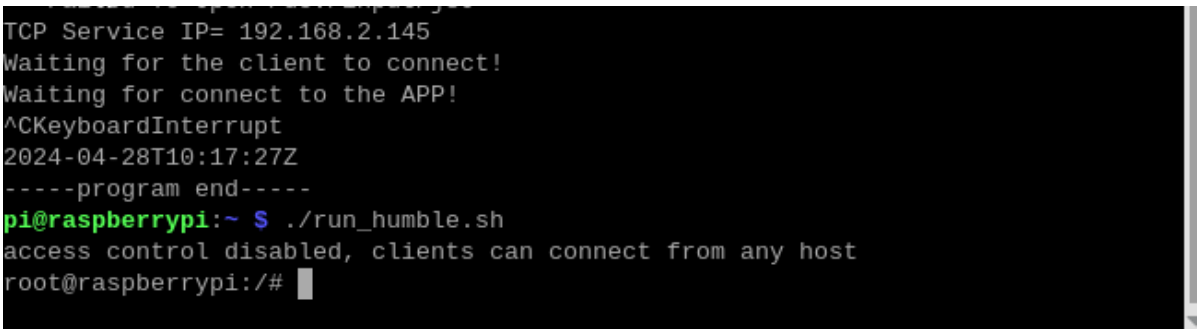
PI5 version steps:

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"**).



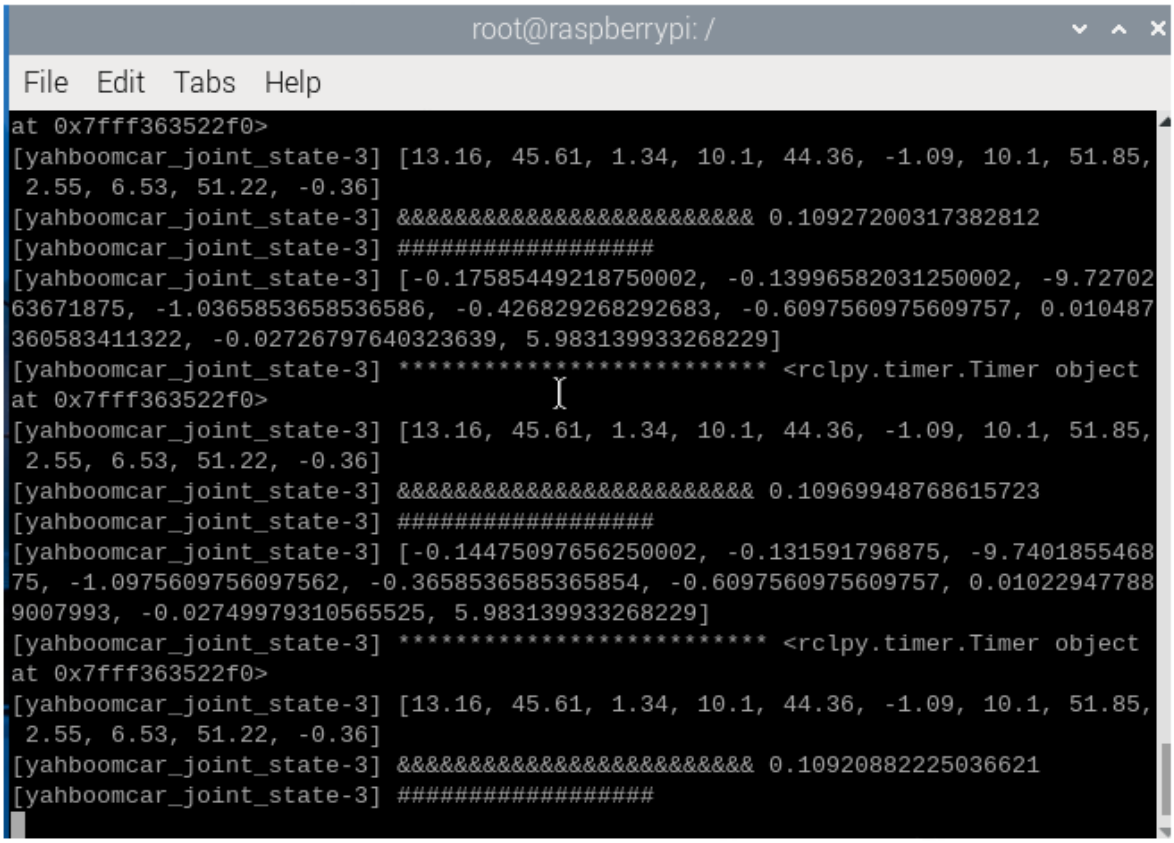
Then ctrl+c closes the large program and enter the following command to enter docker:

```
./run_humble.sh
```



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
```



3. Start the image publishing node

PI4 version steps:

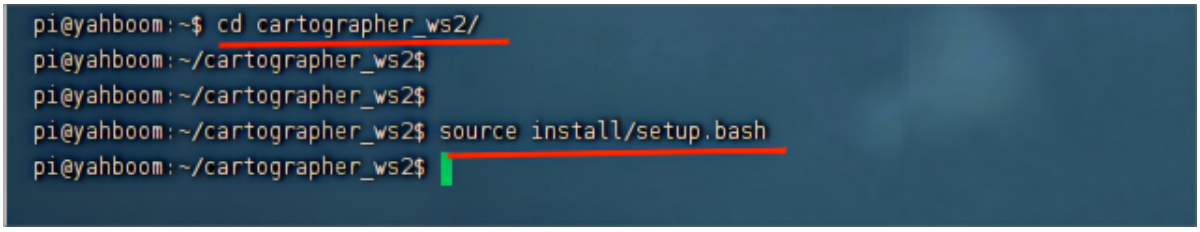
First enter two commands in the terminal

```
export PATH=/home/pi/opencv_install/bin:$PATH
export LD_LIBRARY_PATH=/home/pi/opencv_install/lib:$LD_LIBRARY_PATH
```

Then enter the following command in the terminal

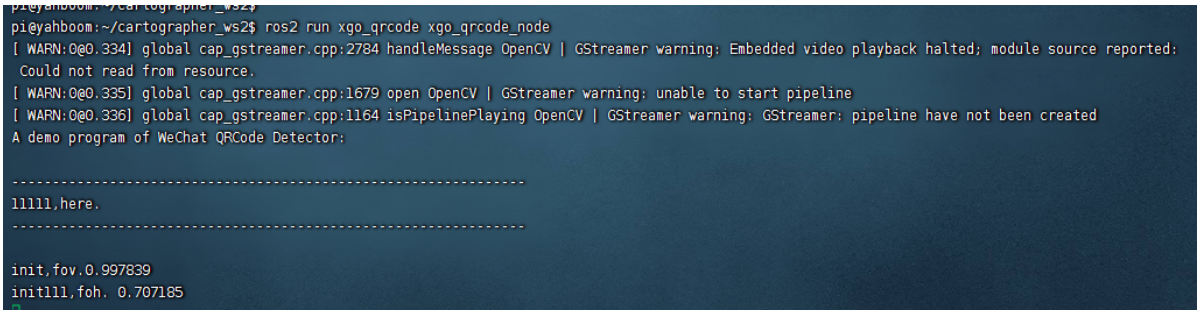
```
cd cartographer_ws2/
```

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



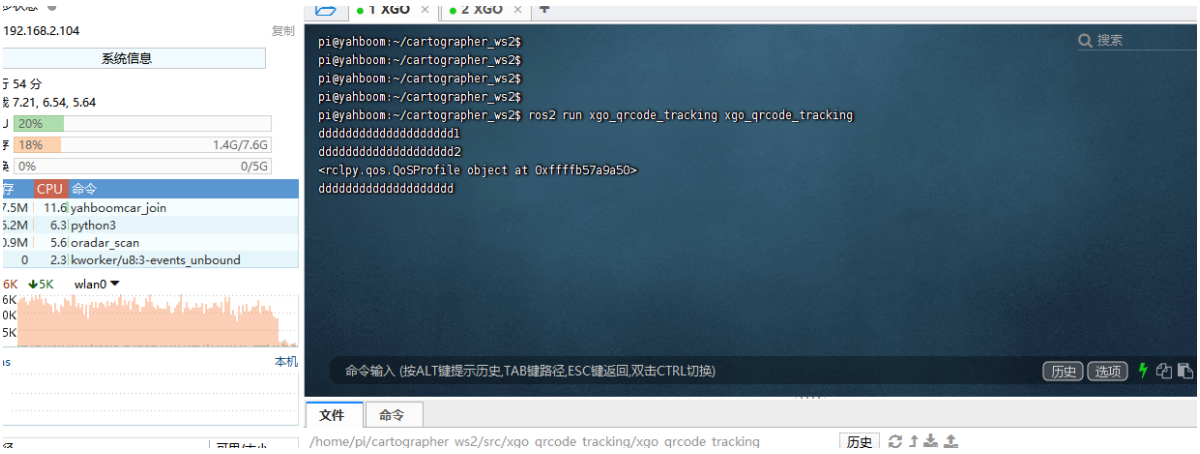
Then enter the following command

```
ros2 run yahboom_qrcode yahboom_qrcode_node
```



Reopen a terminal and enter the command:

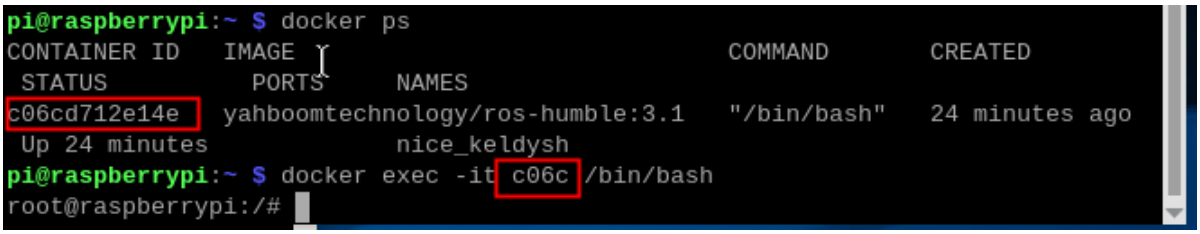
```
cd cartographer_ws2/
source install/setup.bash
ros2 run yahboom_qrcode_tracking yahboom_qrcode_tracking
```



PI5 version steps:

Enter in the root directory of the Raspberry Pi and enter the same terminal

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



Then enter the following command

```
ros2 run yahboom_qrcode yahboom_qrcode_node
```

```
root@raspberrypi:/# ros2 run yahboom_qrcode yahboom_qrcode_node
[ WARN:0] global ./modules/videoio/src/cap_gstreamer.cpp (2075) handleMessage Op
enCV | GStreamer warning: Embedded video playback halted; module source reported
: Could not read from resource.
[ WARN:0] global ./modules/videoio/src/cap_gstreamer.cpp (1053) open OpenCV | GS
treamer warning: unable to start pipeline
[ WARN:0] global ./modules/videoio/src/cap_gstreamer.cpp (616) isPipelinePlaying
 OpenCV | GStreamer warning: GStreamer: pipeline have not been created
A demo program of WeChat QRCode Detector:

------------------------------------------------------------
11111,here.
------------------------------------------------------------

init,fov.0.997839
init111,foh. 0.707185
```

Reopen the same docker terminal and enter the command:

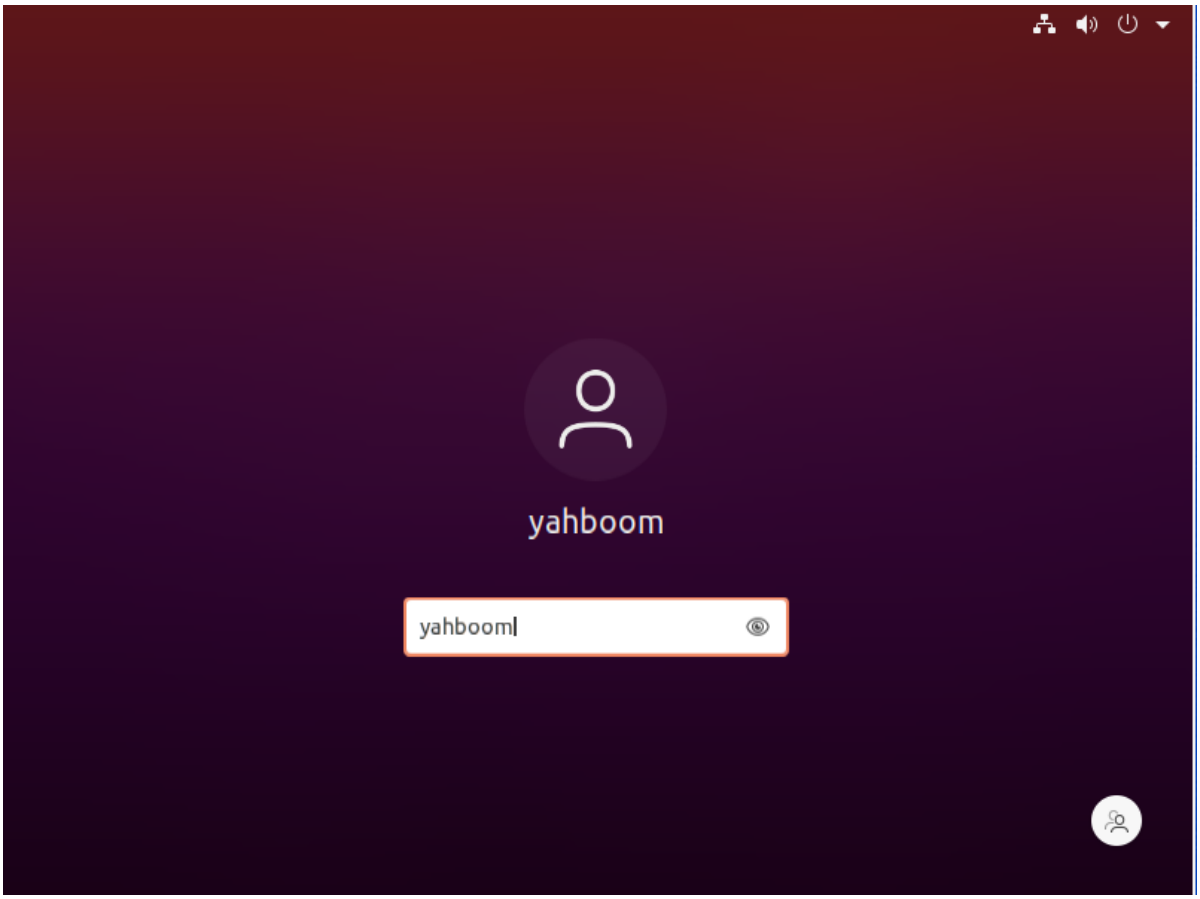
```
ros2 run yahboom_qrcode_tracking yahboom_qrcode_tracking
```

```
root@raspberrypi: /
File  Edit  Tabs  Help
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
info:: {}
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
image:: std_msgs.msg.Header(stamp=builtin_interfaces.msg.Time(sec=1714386973, na
nosec=1958857346), frame_id='')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
info:: {}
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
image:: std_msgs.msg.Header(stamp=builtin_interfaces.msg.Time(sec=1714386973, na
nosec=1986750162), frame_id='')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
info:: {}
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
image:: std_msgs.msg.Header(stamp=builtin_interfaces.msg.Time(sec=1714386973, na
nosec=2019761453), frame_id='')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
info:: {}
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
image:: std_msgs.msg.Header(stamp=builtin_interfaces.msg.Time(sec=1714386973, na
nosec=2053709632), frame_id='')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
info:: {}
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

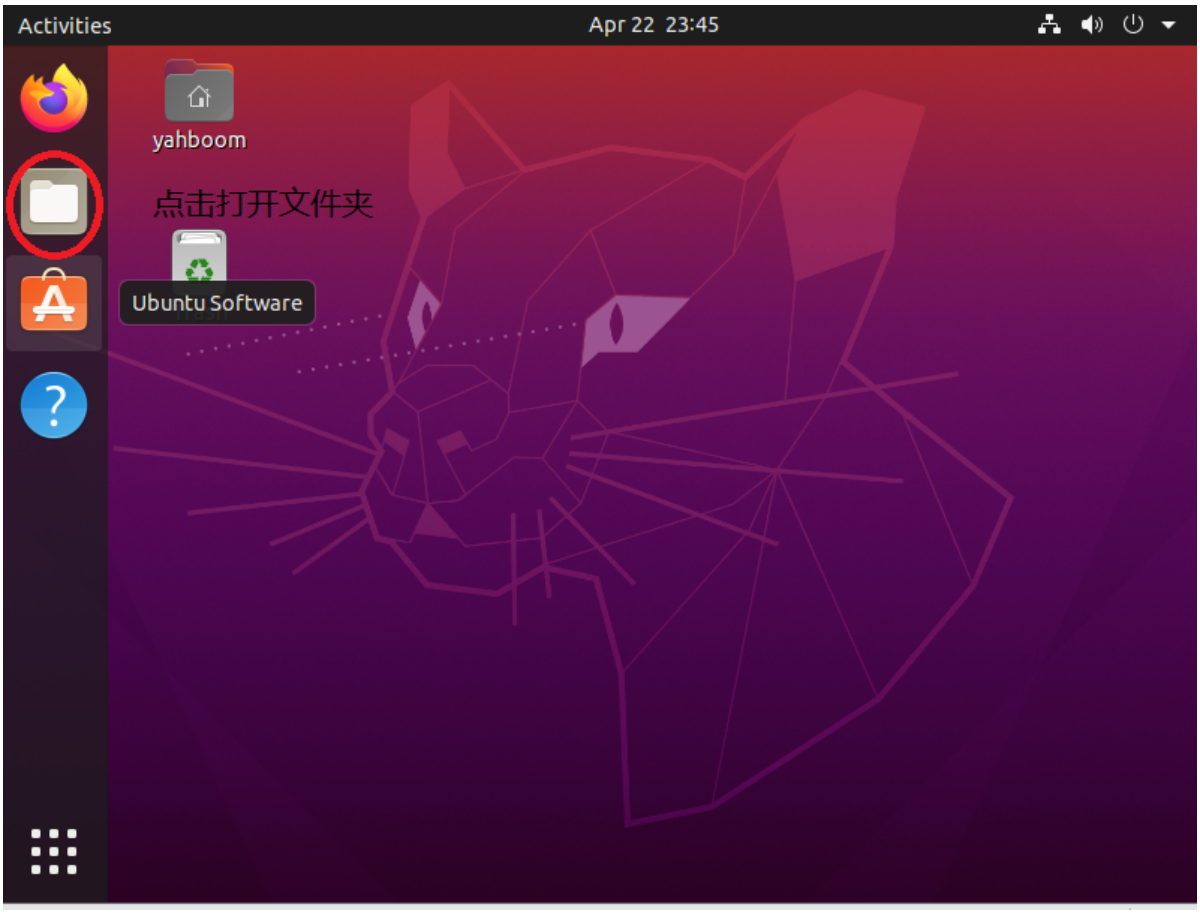
5. Set the recognition color through the web interface

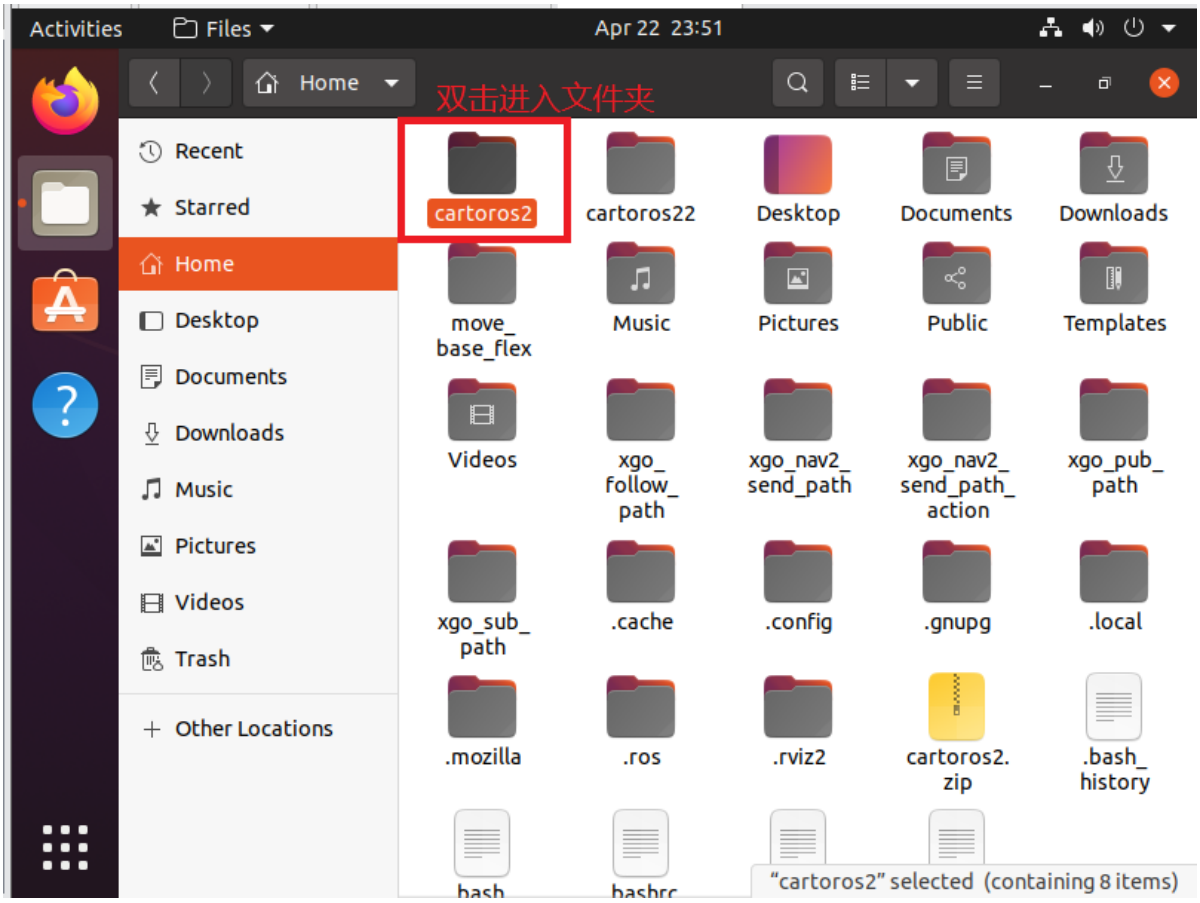
The steps are the same for PI4 and PI5 versions:

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

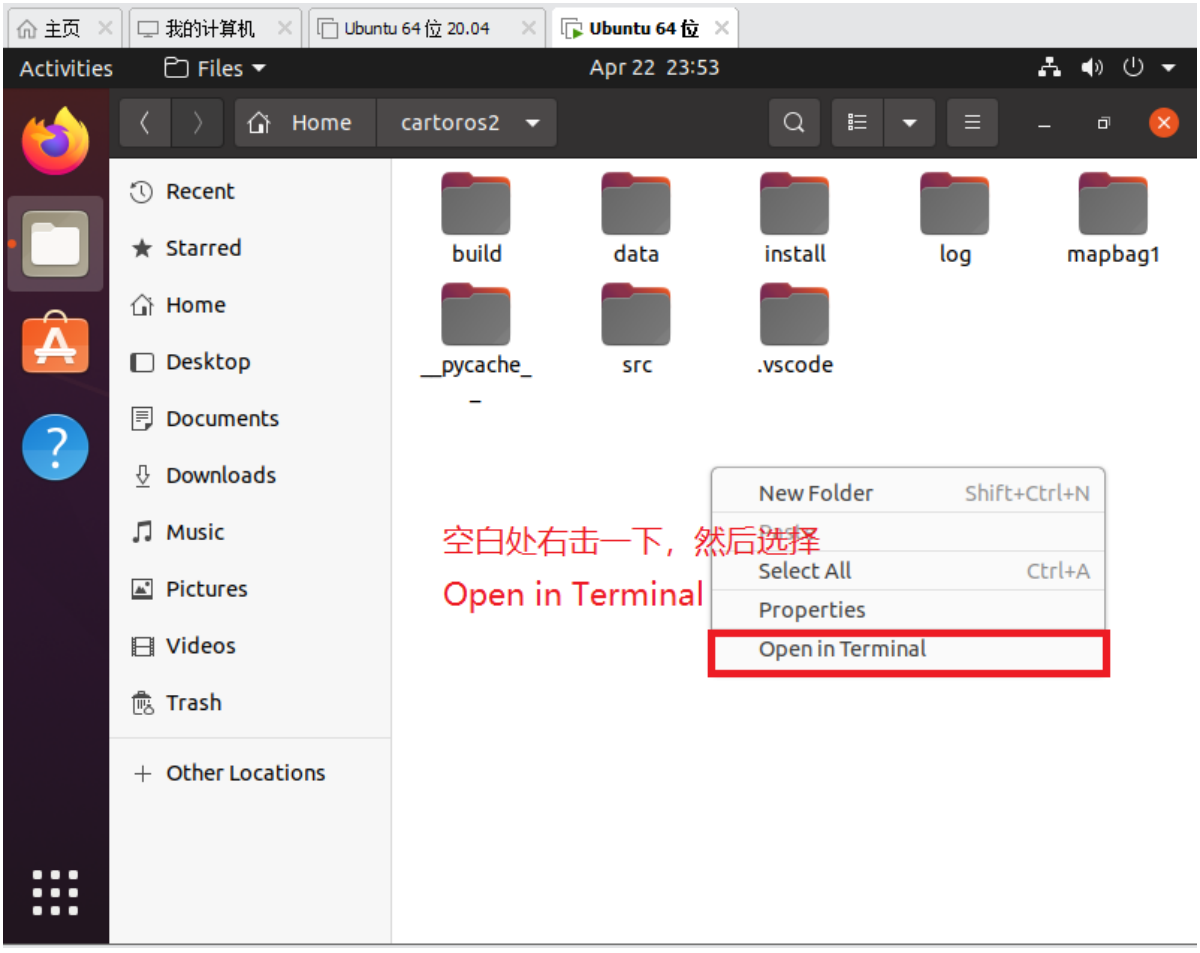


Click on the folder to open the cartoros2 folder.



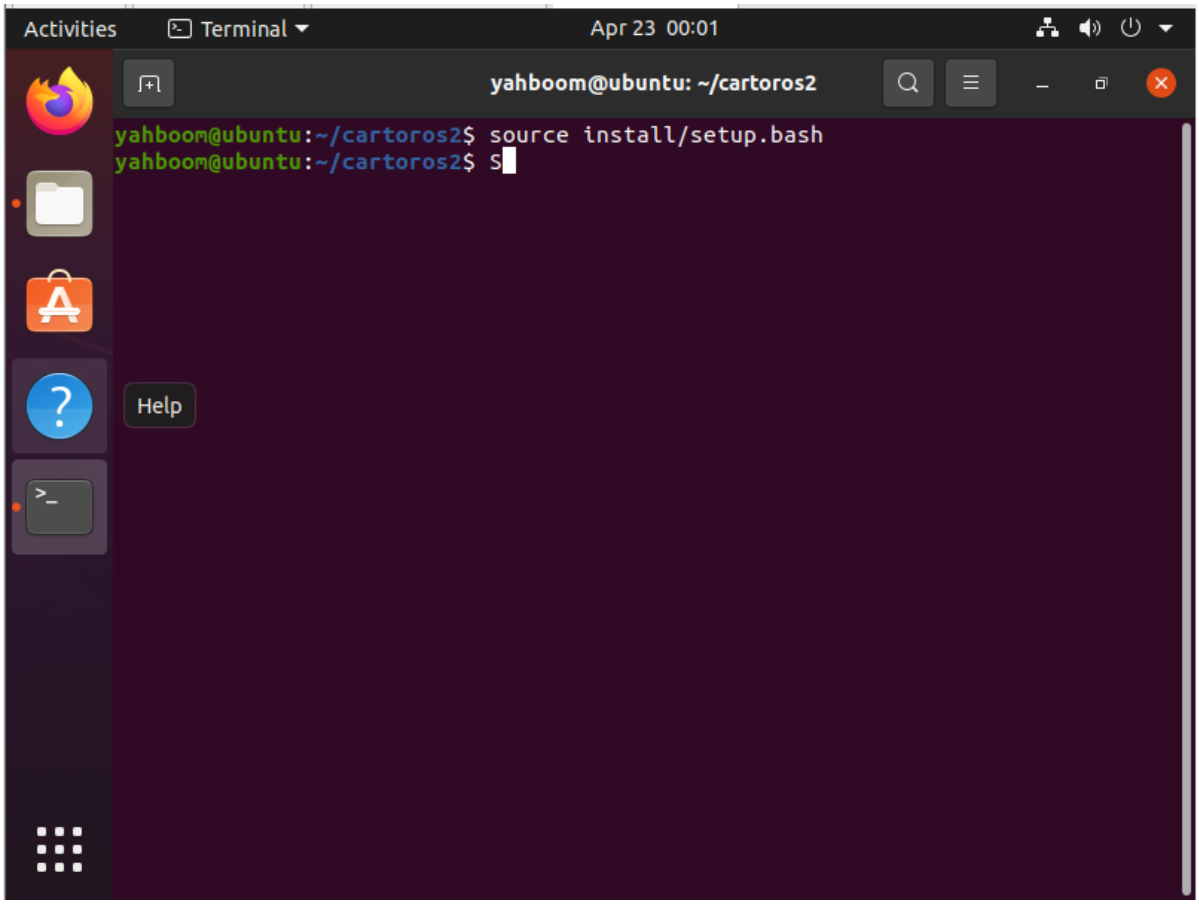


Open the terminal under the folder



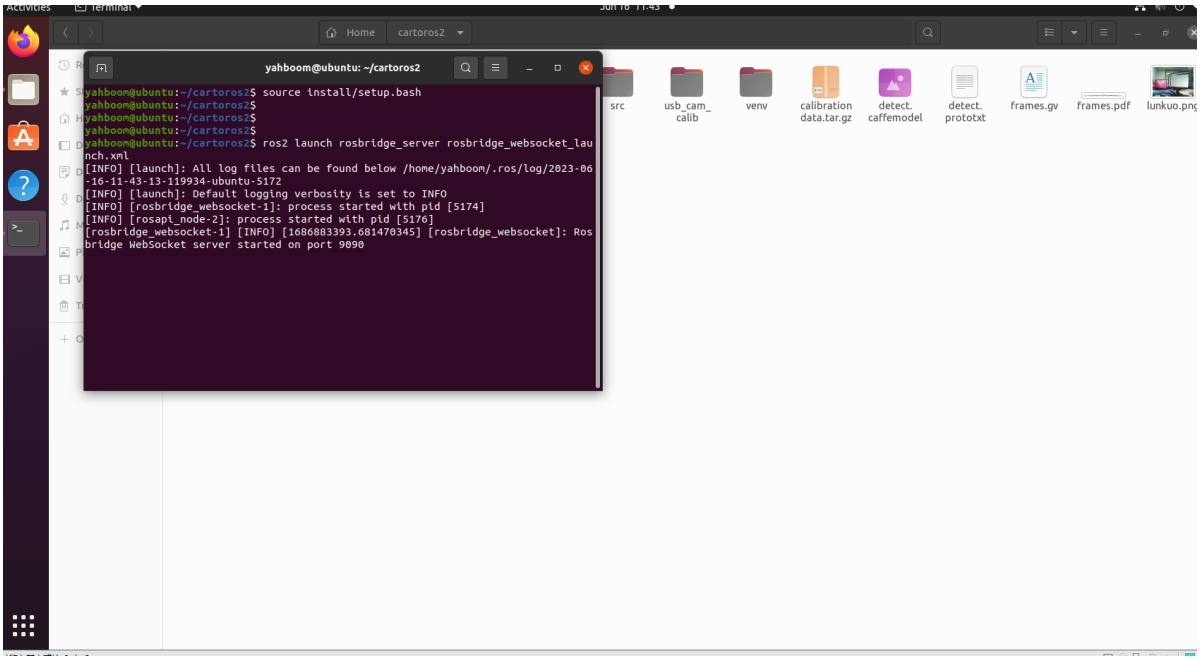
Then enter the following command

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



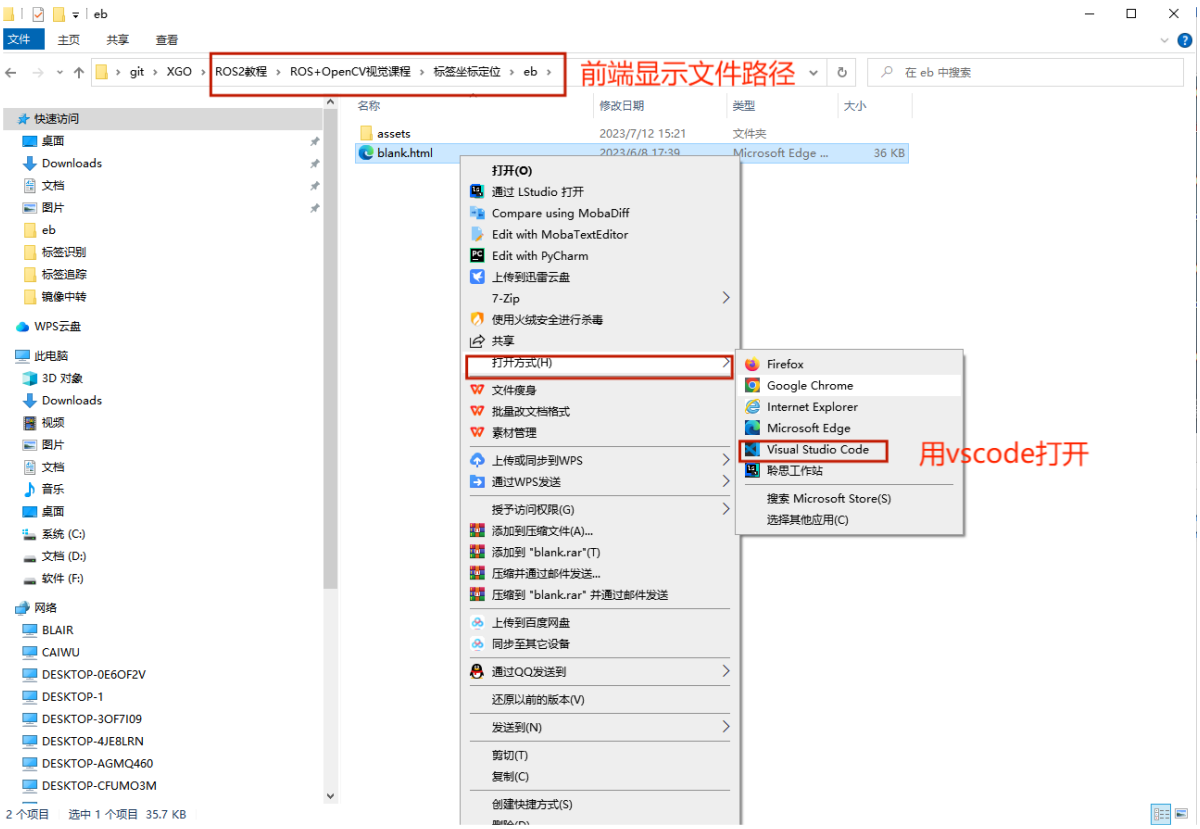
Then start rosbridge and enter the following command

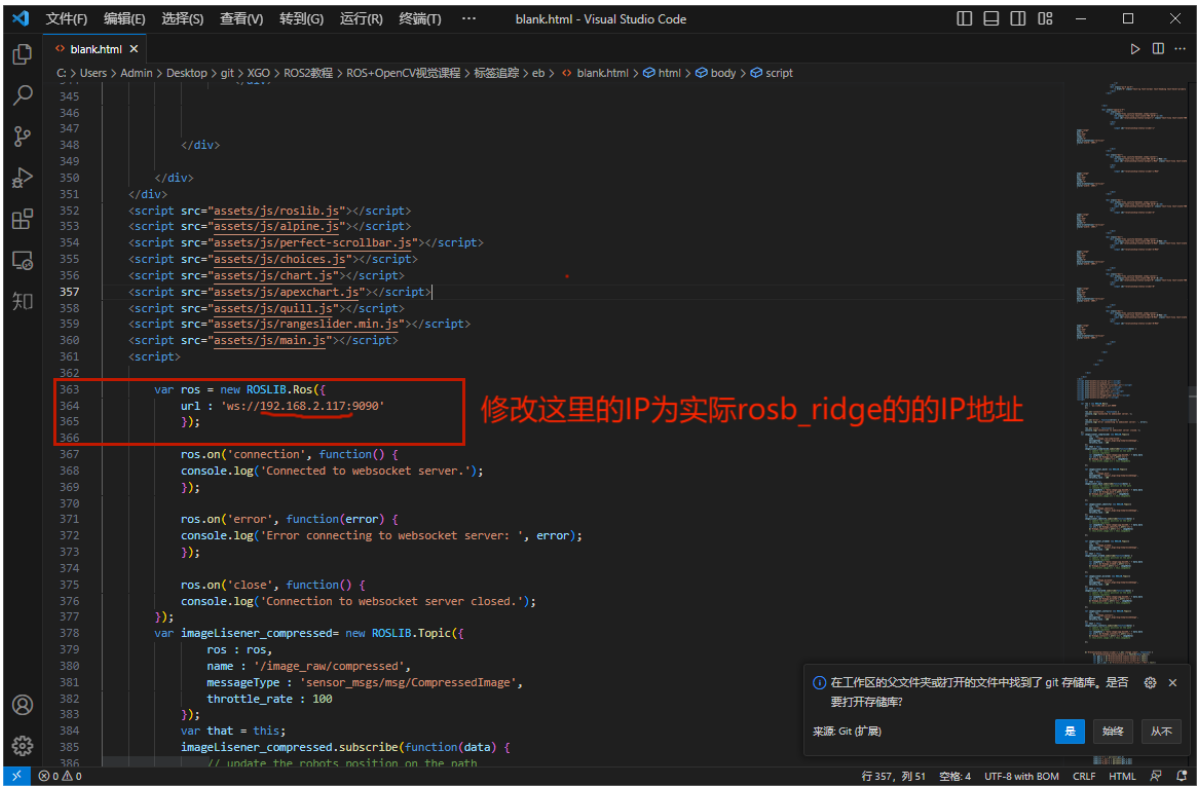
```
ros2 launch rosbridge_server rosbridge_websocket_launch.xml
```

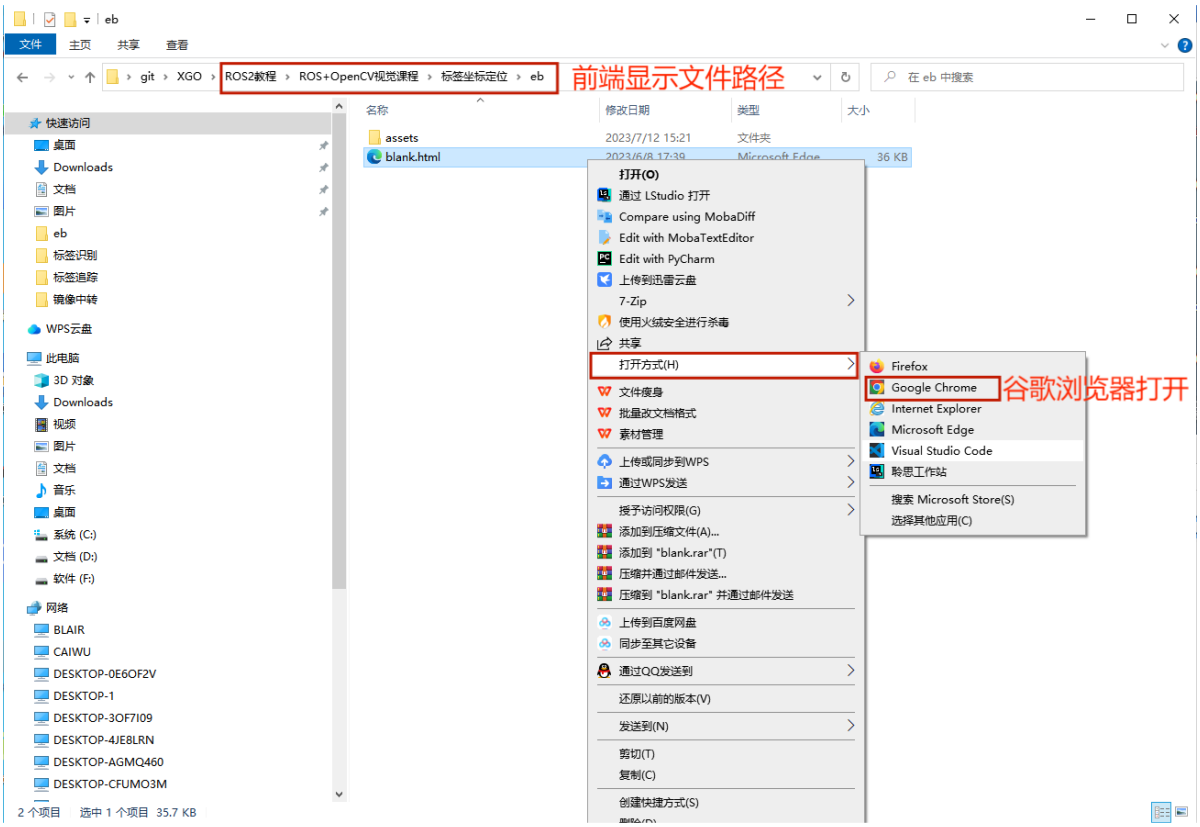


Find the blank.html file in the eb folder in the updated directory of this tutorial and open it with Google Chrome.

Note: The IP address of rosbridge needs to be set here. Obtain the IP address of the virtual machine, then open the blank.html file, modify the IP address of line 363 and save it, as shown in the figure below.

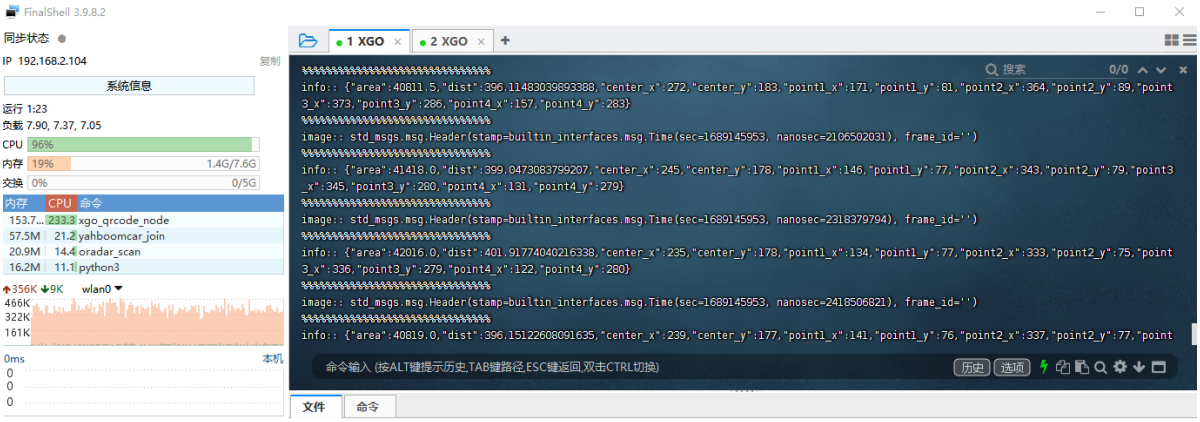

前端显示文件路径
用vscode打开


blank.html - Visual Studio Code
修改这里的IP为实际rosb_ridge的的IP地址



As shown in the picture below, use your mobile phone to display a QR code. Here is the WeChat QR code. You can see the picture transmitted by the camera and the recognized QR code data.

You can see the printed QR code data in the finalshell terminal, where (center_x, center_y) is the center position of the QR code, (point1_x, point1_y), (point2_x, point2_y), (point3_x, point3_y), (point4_x ,point4_y) are the four fixed point coordinates of the QR code.


FinalShell 3.9.8.2
同步状态
IP 192.168.2.104
复制
系统信息
运行 1:23
负载 7.90, 7.37, 7.05
CPU 96%
内存 19% 1.4G/7.6G
交换 0% 0/5G
内存 CPU 命令
153.7... 233.3 xgo_qrcode_node
57.5M 21.2 yahboomcar_join
20.9M 14.4 oradar_scan
16.2M 11.1 python3
356K 9K wlan0
466K
322K
161K
0ms 本机
0
0
0
1 XGO
2 XGO
搜索
0/0
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
info:: {"area":40811.5,"dist":396.11483039893388,"center_x":272,"center_y":183,"point1_x":171,"point1_y":81,"point2_x":364,"point2_y":89,"point3_x":373,"point3_y":286,"point4_x":157,"point4_y":283}
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
image:: std_msgs.msg.Header(stamp=builtin_interfaces.msg.Time(sec=1689145953, nanosec=2106502031), frame_id='')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
info:: {"area":41418.0,"dist":399.0473083799207,"center_x":245,"center_y":178,"point1_x":146,"point1_y":77,"point2_x":343,"point2_y":79,"point3_x":345,"point3_y":280,"point4_x":131,"point4_y":279}
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
image:: std_msgs.msg.Header(stamp=builtin_interfaces.msg.Time(sec=1689145953, nanosec=2318379794), frame_id='')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
info:: {"area":42016.0,"dist":401.91774040216338,"center_x":235,"center_y":178,"point1_x":134,"point1_y":77,"point2_x":333,"point2_y":75,"point3_x":336,"point3_y":279,"point4_x":122,"point4_y":280}
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
image:: std_msgs.msg.Header(stamp=builtin_interfaces.msg.Time(sec=1689145953, nanosec=2418506821), frame_id='')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
info:: {"area":40819.0,"dist":396.15122608091635,"center_x":239,"center_y":177,"point1_x":141,"point1_y":76,"point2_x":337,"point2_y":77,"point
命令输入 (按ALT键提示历史,TAB键路径,ESC键返回,双击CTRL切换)
历史
选项
文件
命令