

3. Label coordinate positioning

Quick use

1. DOGZILLA POWER UP

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



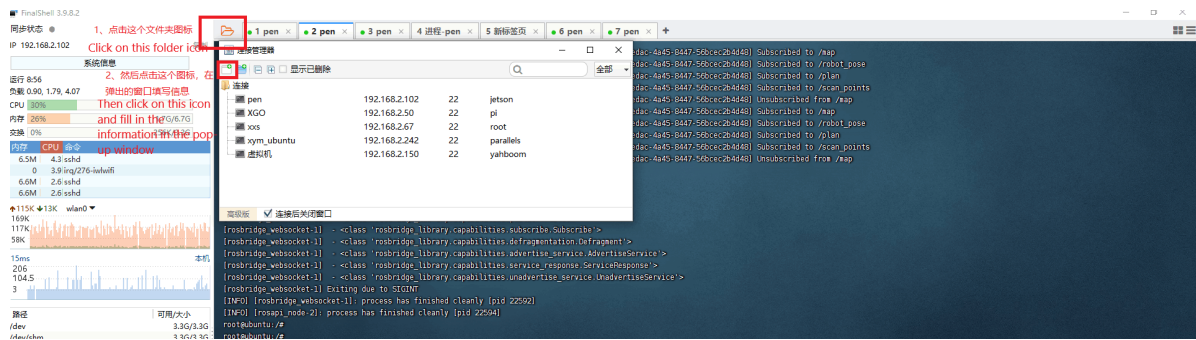
After startup, we can view the IP address on the robot dog's small screen.

2. Open shell to connect to DOGZILLA

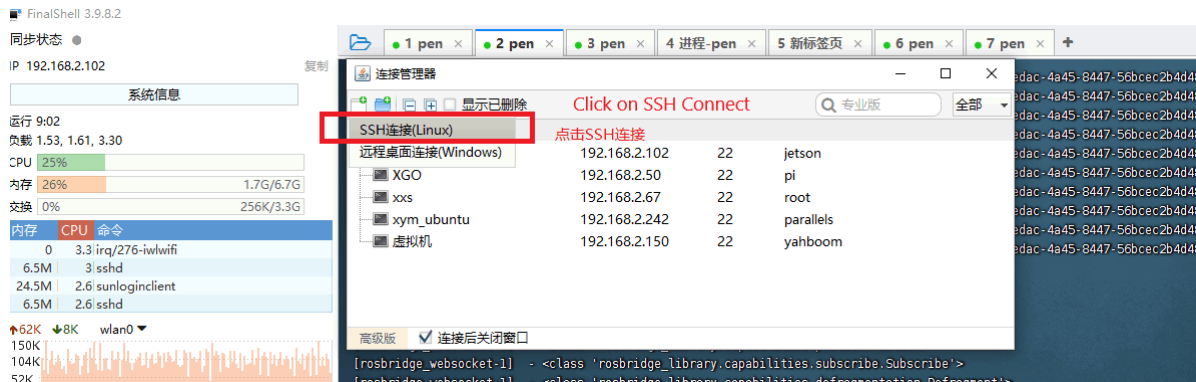
Then use the ssh terminal to connect to robot dog.

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.

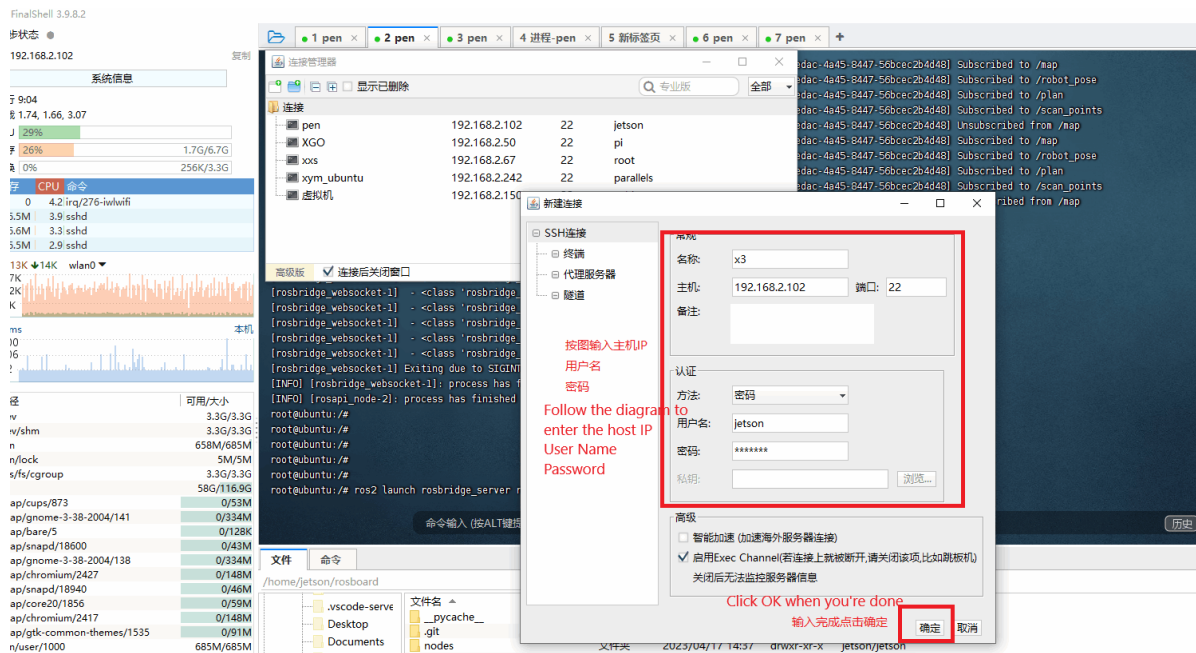
Open the shell utility, here I use FinalShell, enter the username, password, port, connection name and other information.



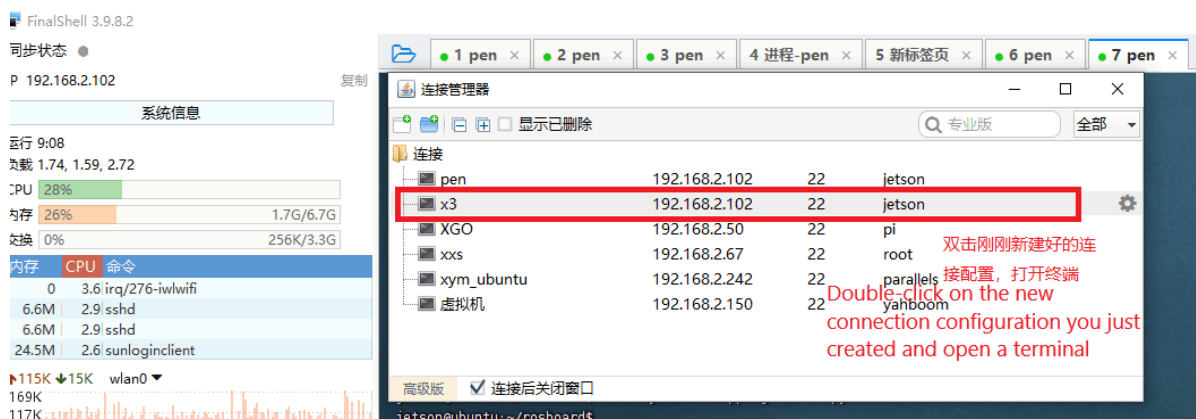
Select ssh connection to create a new ssh connection



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



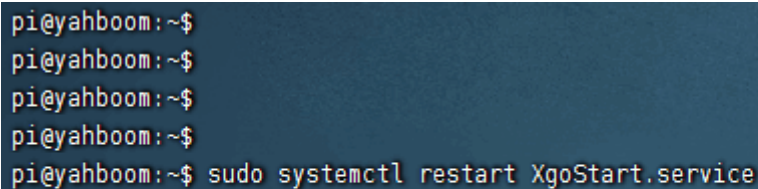
Here select the new ssh connection you just created.



3. Starting the DOGZILLA chassis

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

```
sudo systemctl restart XgoStart.service
```



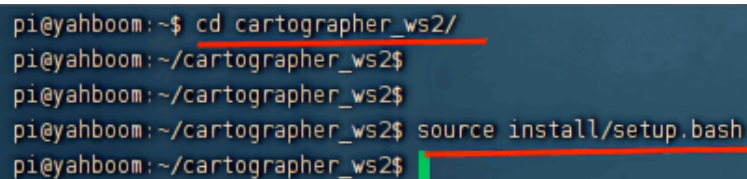
```
pi@yahboom:~$  
pi@yahboom:~$  
pi@yahboom:~$  
pi@yahboom:~$  
pi@yahboom:~$ sudo systemctl restart XgoStart.service
```

4. Start the image publishing node

Enter the following command in the terminal

```
cd cartographer_ws2/
```

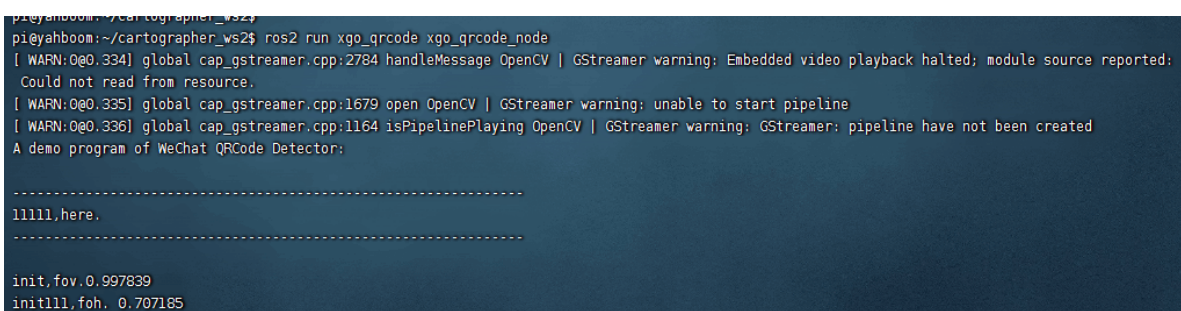
```
source install/setup.bash
```



```
pi@yahboom:~$ cd cartographer_ws2/  
pi@yahboom:~/cartographer_ws2$  
pi@yahboom:~/cartographer_ws2$  
pi@yahboom:~/cartographer_ws2$ source install/setup.bash  
pi@yahboom:~/cartographer_ws2$
```

Then enter the following command

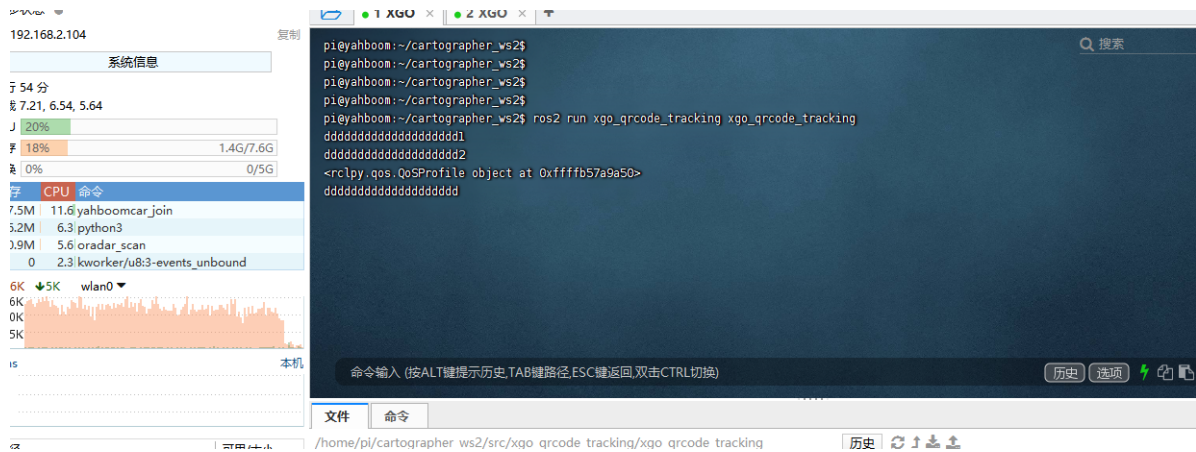
```
ros2 run yahboom_qrcode yahboom_qrcode_node
```



```
pi@yahboom:~/cartographer_ws2$  
pi@yahboom:~/cartographer_ws2$ ros2 run xgo_qrcode xgo_qrcode_node  
[ WARN:0@0.334] global cap_gstreamer.cpp:2784 handleMessage OpenCV | GStreamer warning: Embedded video playback halted; module source reported:  
Could not read from resource.  
[ WARN:0@0.335] global cap_gstreamer.cpp:1679 open OpenCV | GStreamer warning: unable to start pipeline  
[ WARN:0@0.336] global cap_gstreamer.cpp:1164 isPipelinePlaying OpenCV | GStreamer warning: GStreamer: pipeline have not been created  
A demo program of WeChat QRCode Detector:  
  
-----  
11111,here.  
-----  
  
init,fov.0.997839  
init1111,fov. 0.707185
```

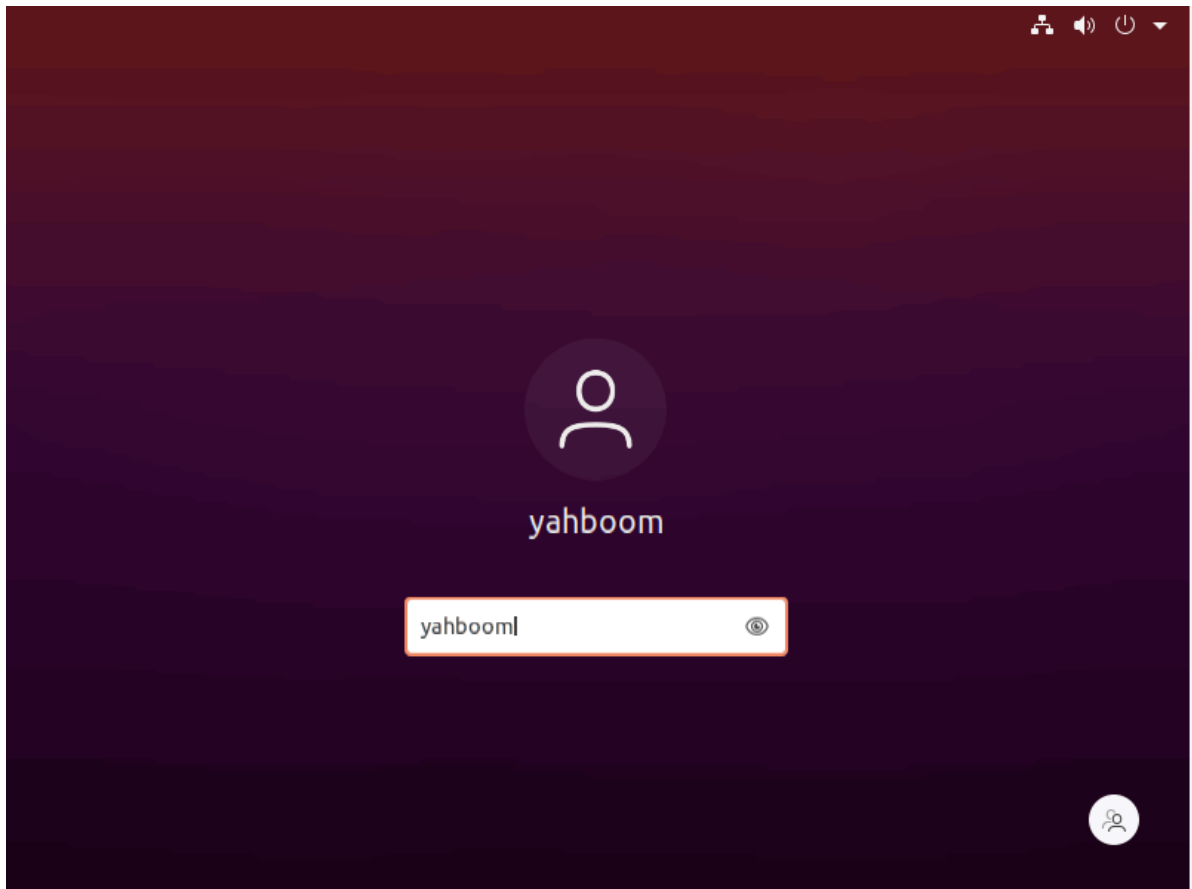
Reopen a terminal and enter the command:

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

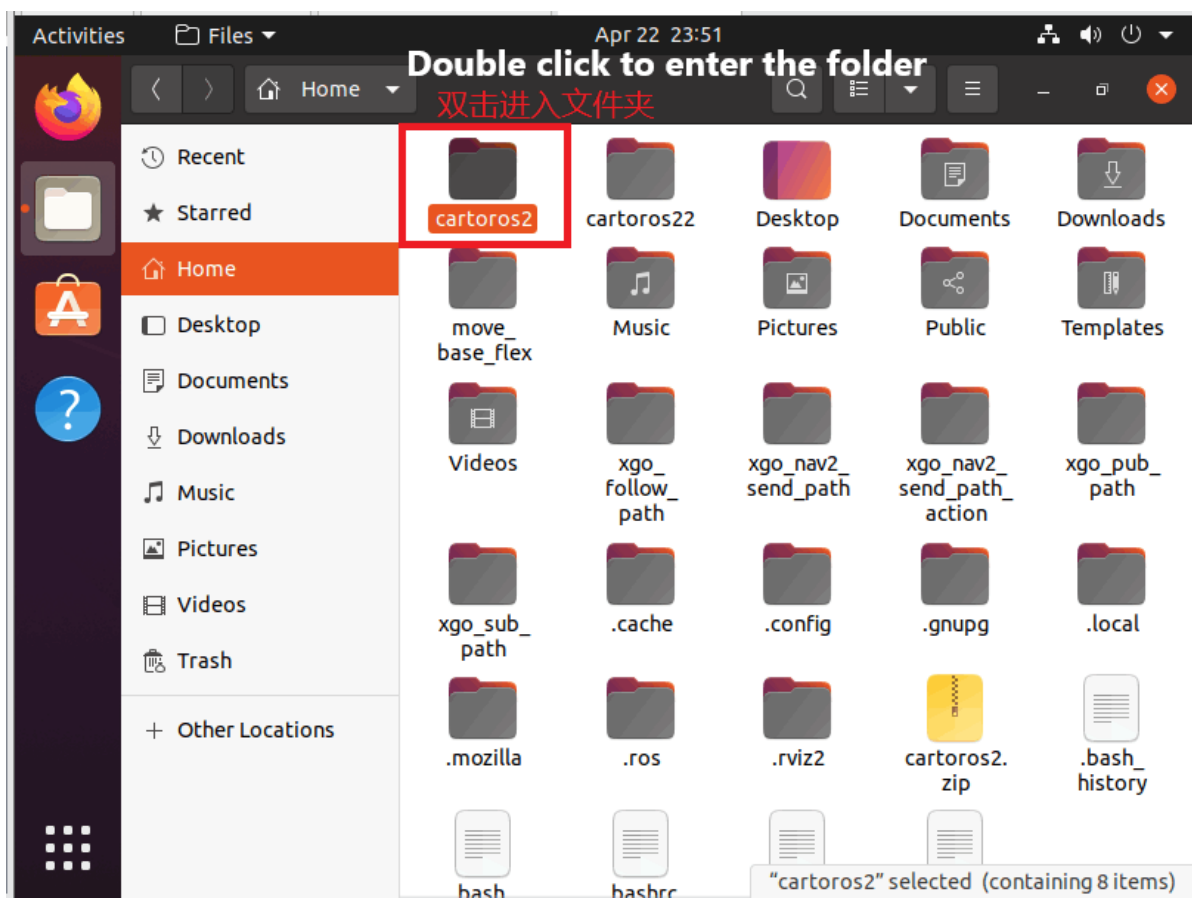
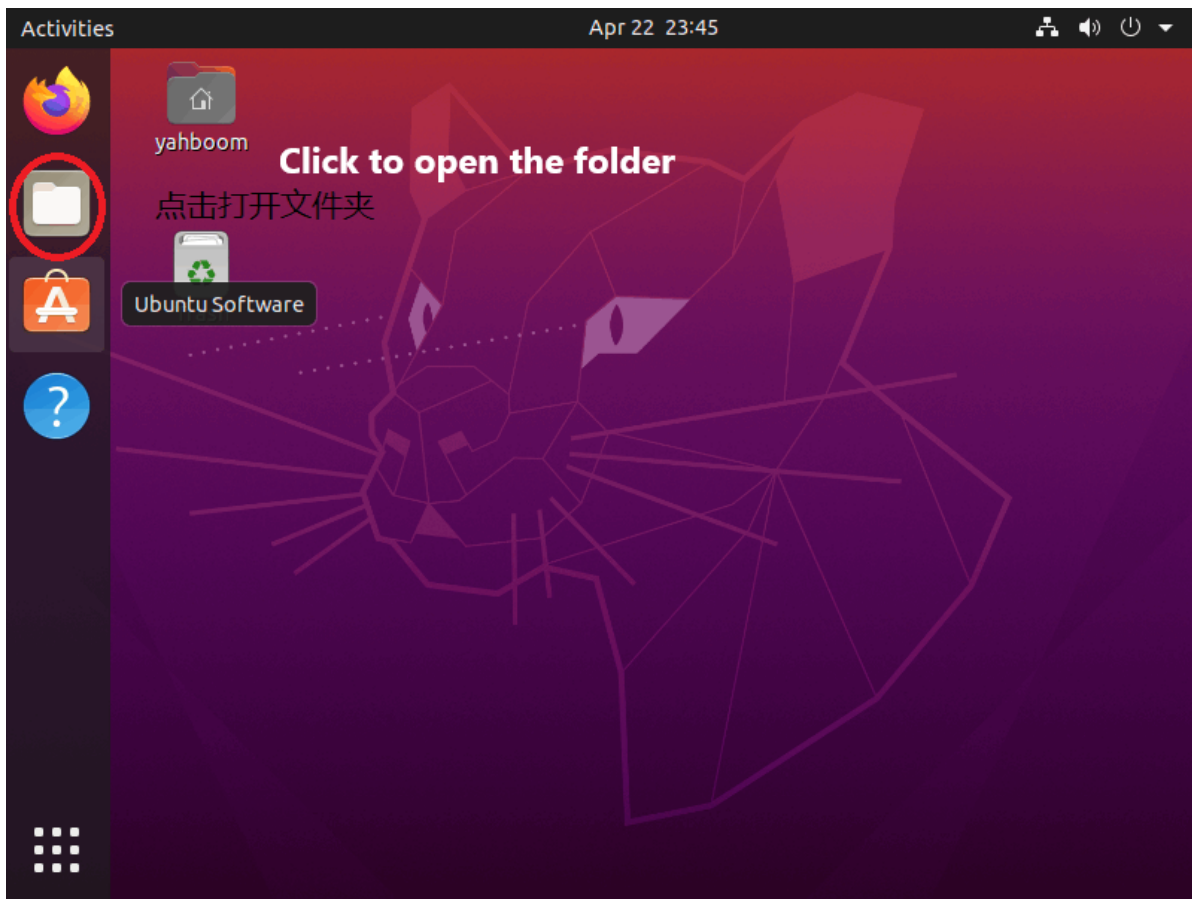


5. Setting the recognition colours through the web interface

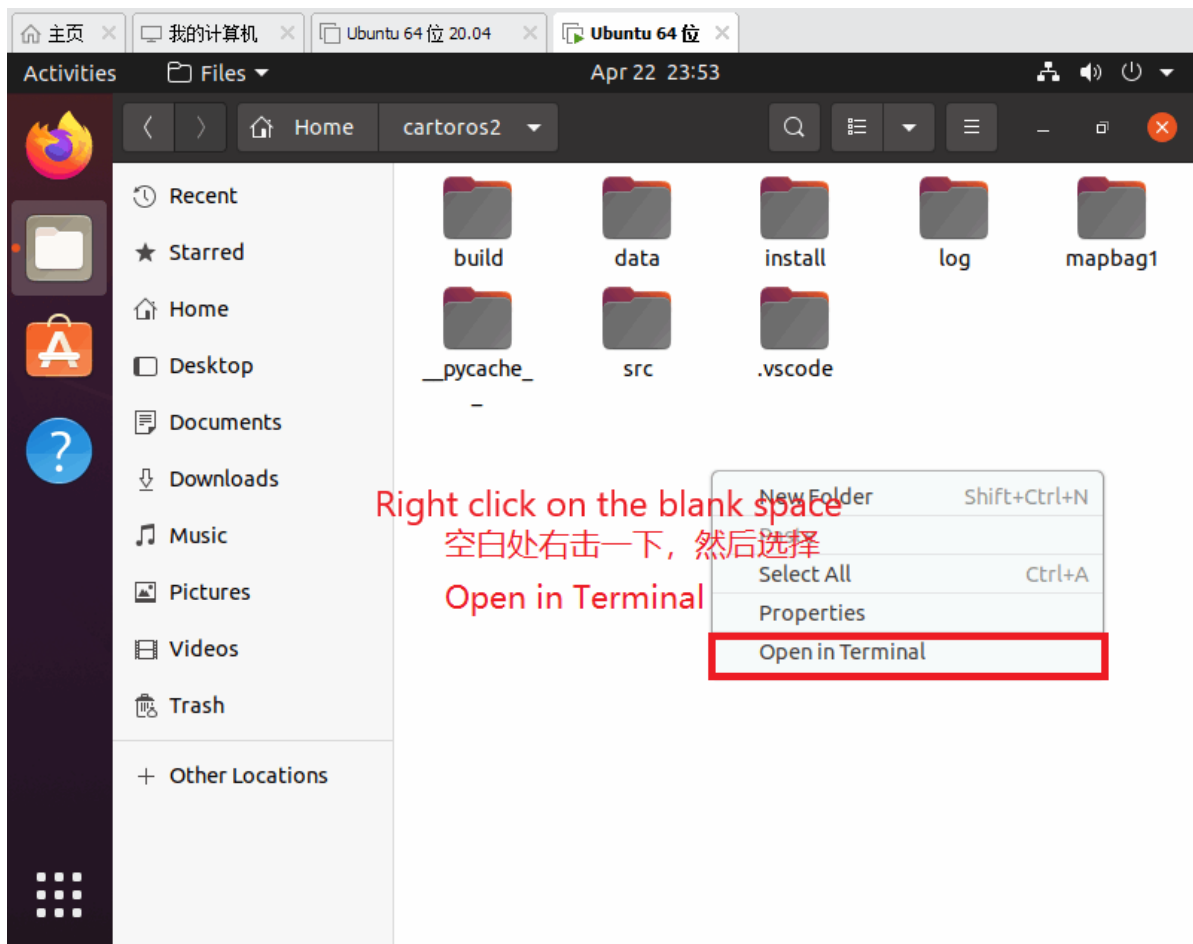
Open the virtual machine and enter the username yahboom, password yahboom.



Click on the folder to open the cartoros2 folder.

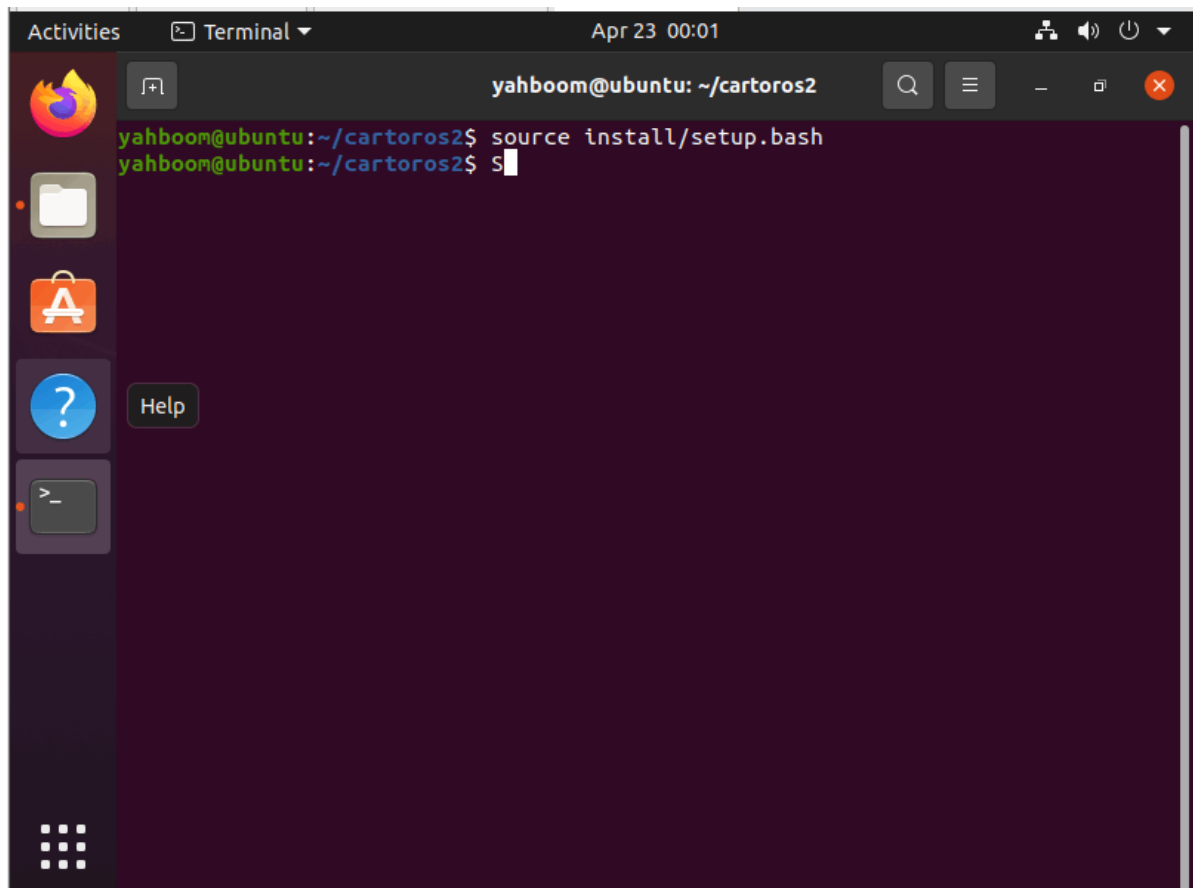


Open a terminal under the folder



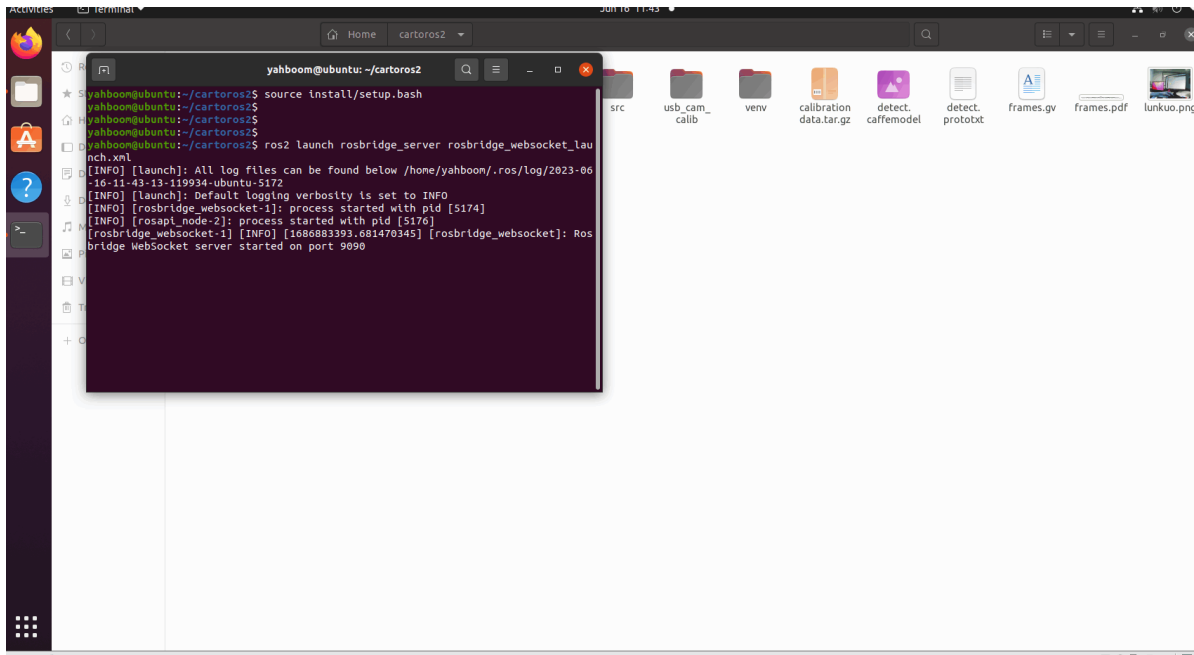
Then enter the following command

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



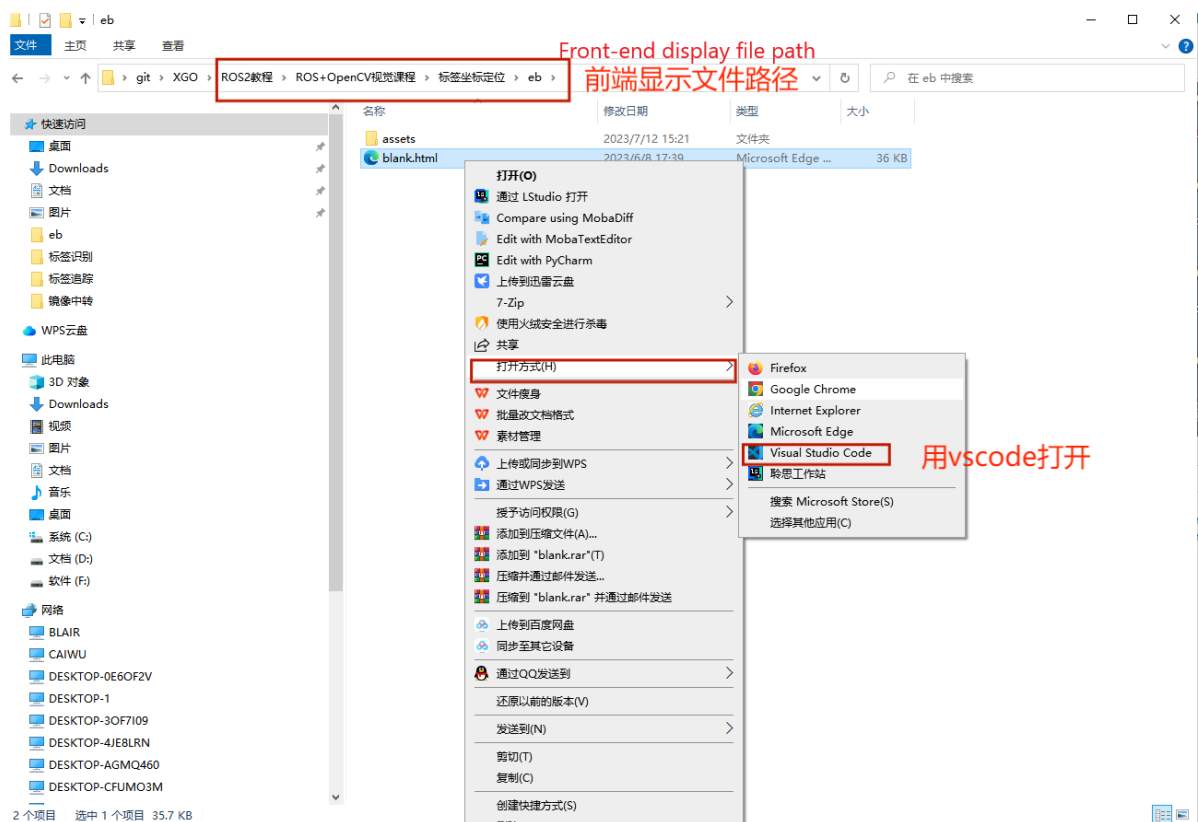
Then start rosbriidge and enter the following command

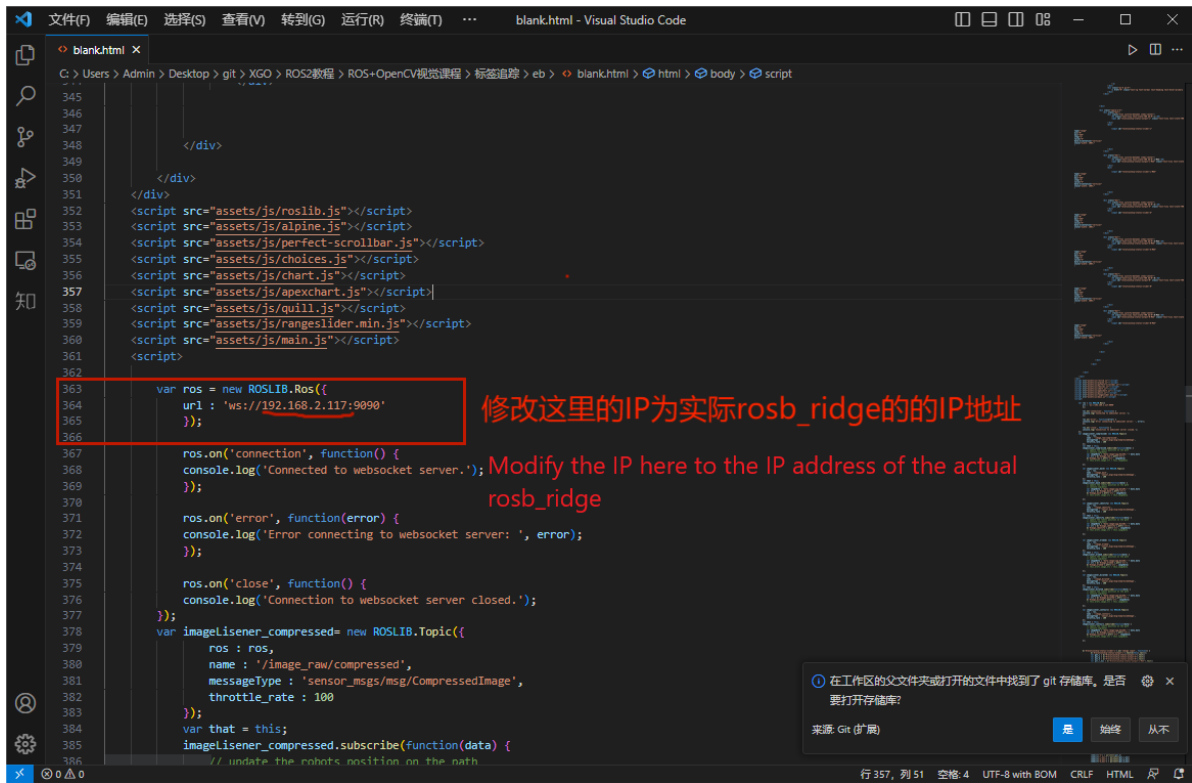
```
ros2 launch rosbridge_server rosbridge_websocket_launch.xml
```



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

Note: Here you need to set the IP address of rosbridge. Get the IP address of the virtual machine, then open the blank.html file, change the IP address in line 363 of the code and save it as shown below.





As shown in the figure below, show a QR code with your mobile phone, in my case a WeChat QR code, and you can see the image transmitted by the camera, as well as the recognised QR code data.



In finalshell terminal, you can see the data of the printed QR code, where (centre_x, centre_y) is the centre position of the QR code, and (point1_x, point1_y), (point2_x, point2_y), (point3_x, point3_y), (point4_x, point4_y) are the coordinates of the four fixed points of the QR code.

