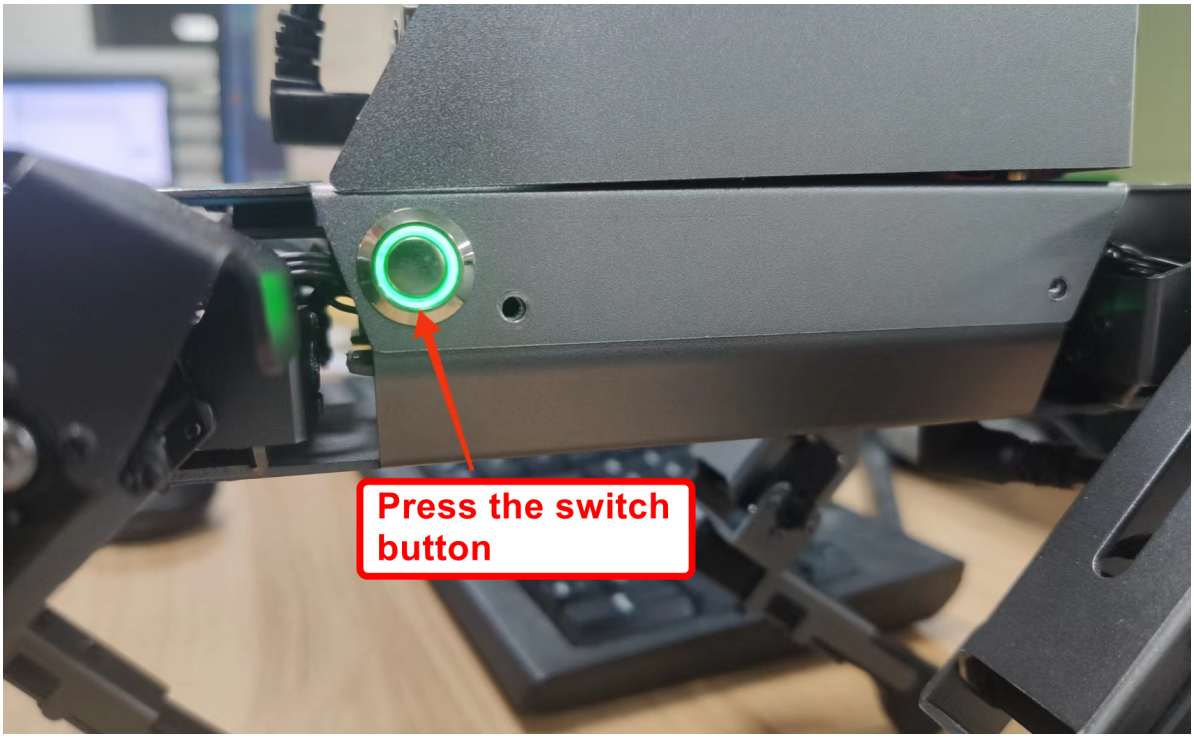
Color Tracking

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.

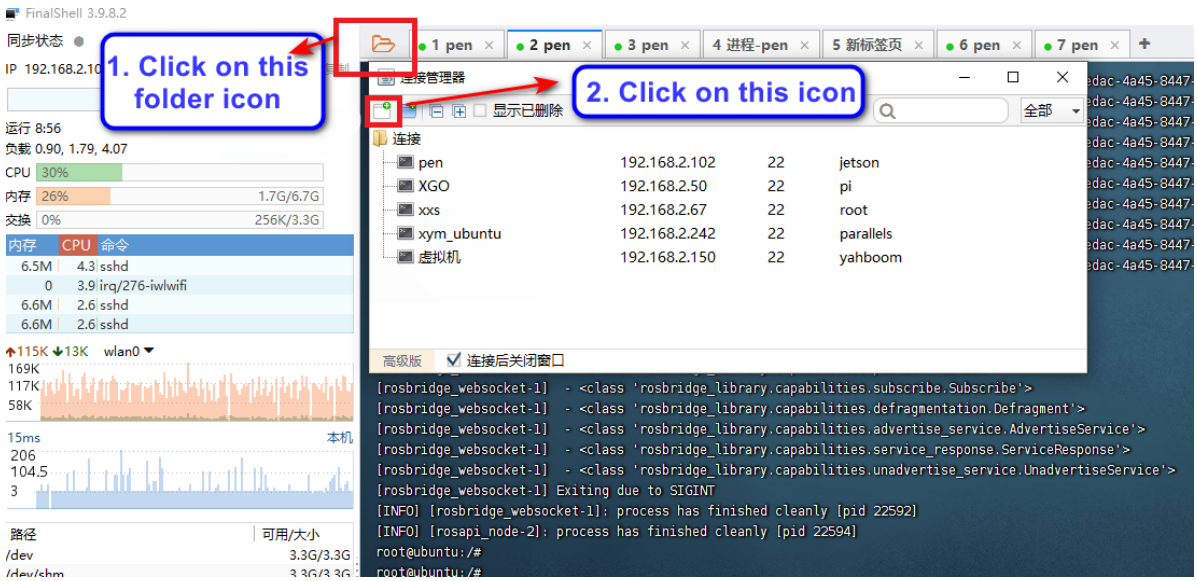
2. Start DOGZILLA chassis

PI4 version steps:

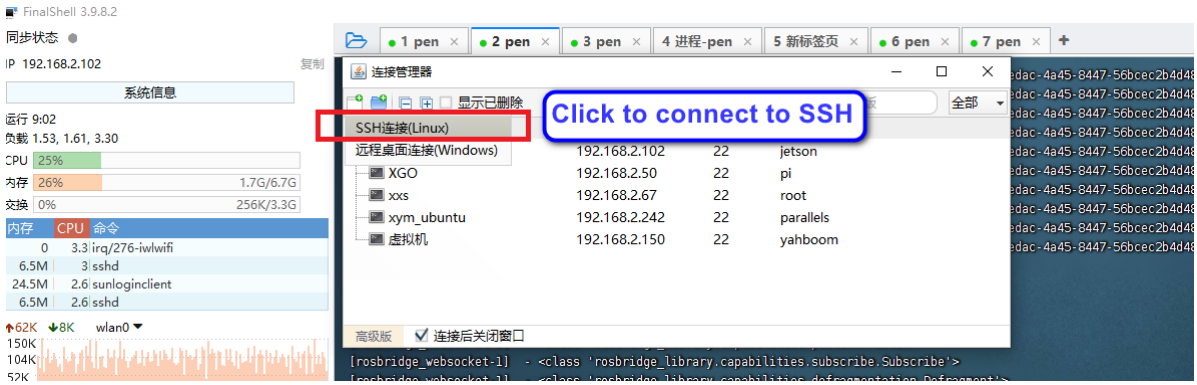
Then use the ssh terminal to connect to the robot dog.

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.

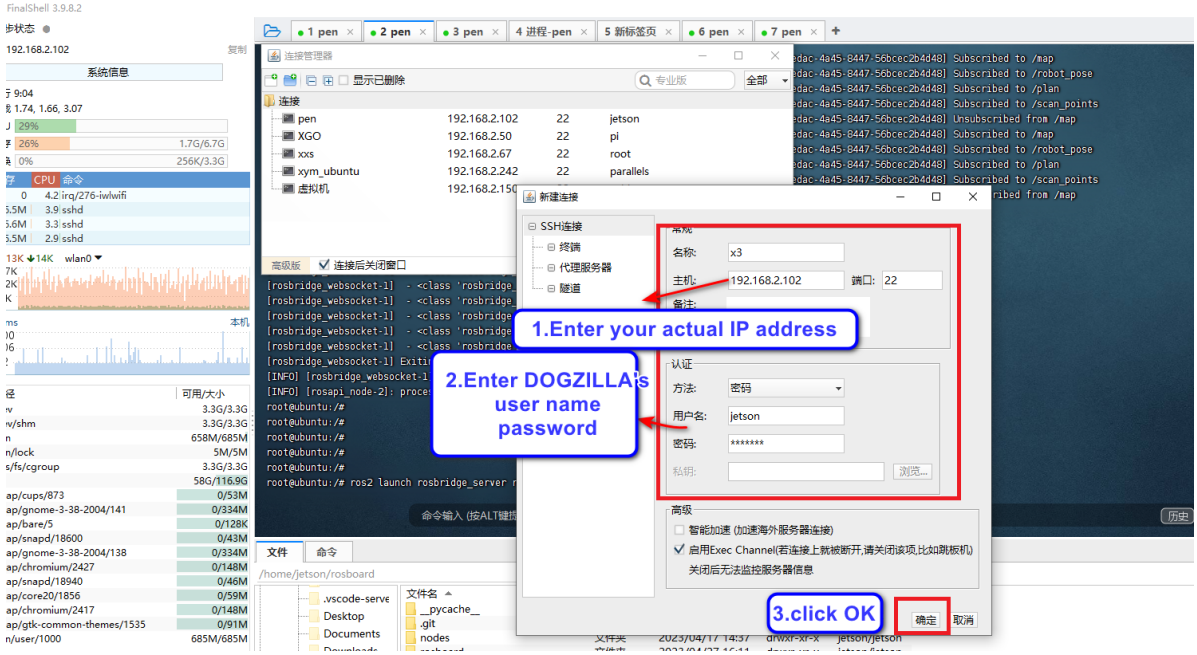
Open the shell tool. The shell tool I use here is FinalShell. Enter username, password, port, connection name and other information.



Select ssh connection to create a new ssh connection



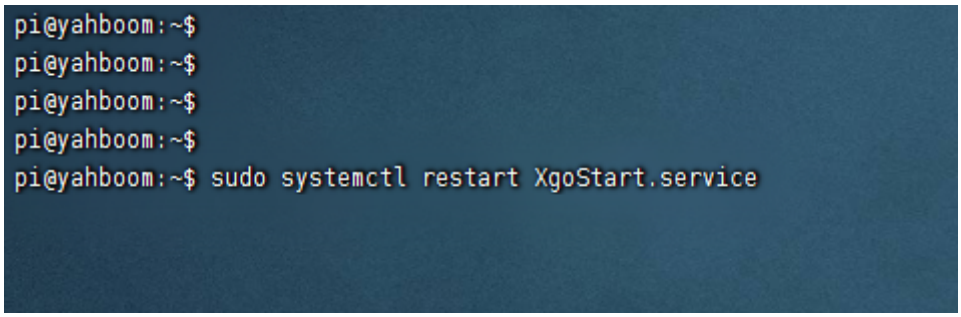
Here the username is pi, the password is yahboom, and the ip address is the IP address of the real robot dog.



Select the ssh connection you just created here.

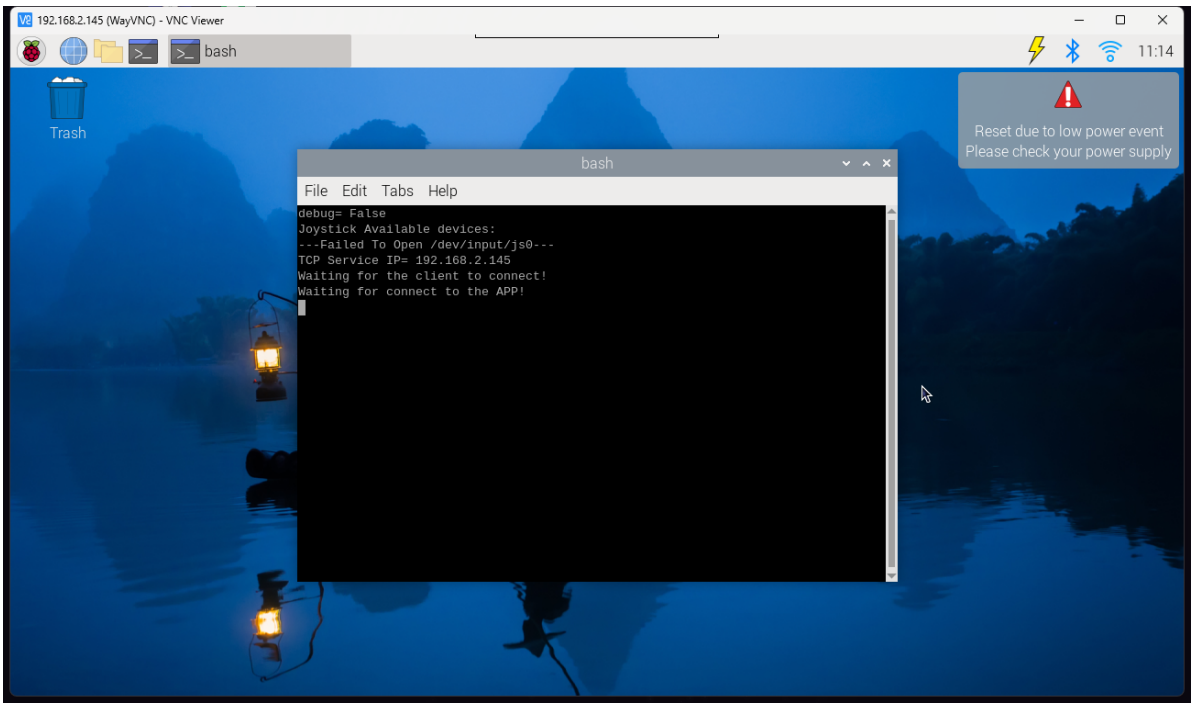
Enter the command in the terminal to start the chassis task.

```
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"**).



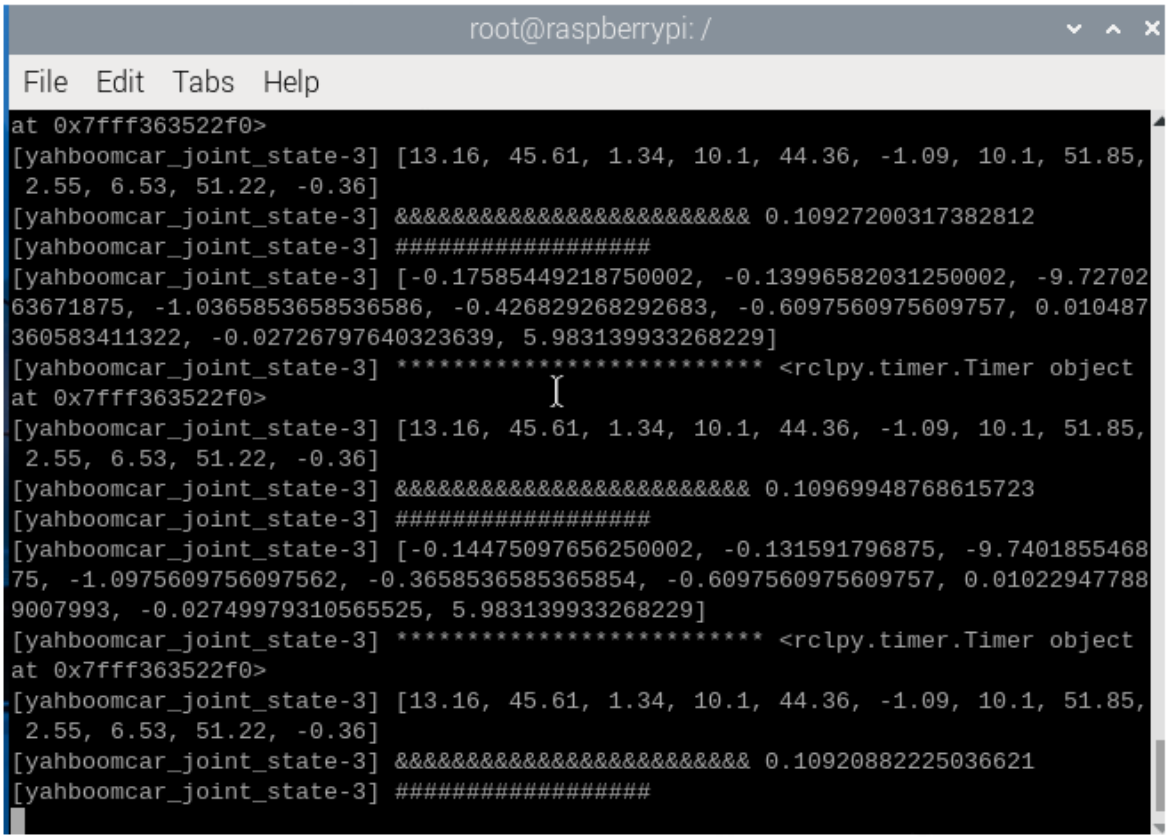
Then ctrl+c closes the large program and enter the following command to enter docker:

```
./run_humble.sh
```

```
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
```



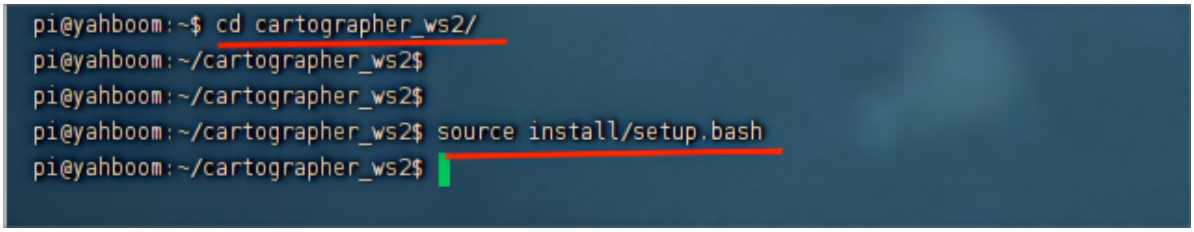
3. Start the image publishing node

PI4 version steps:

Enter the following command in the terminal

```
cd cartographer_ws2/
```

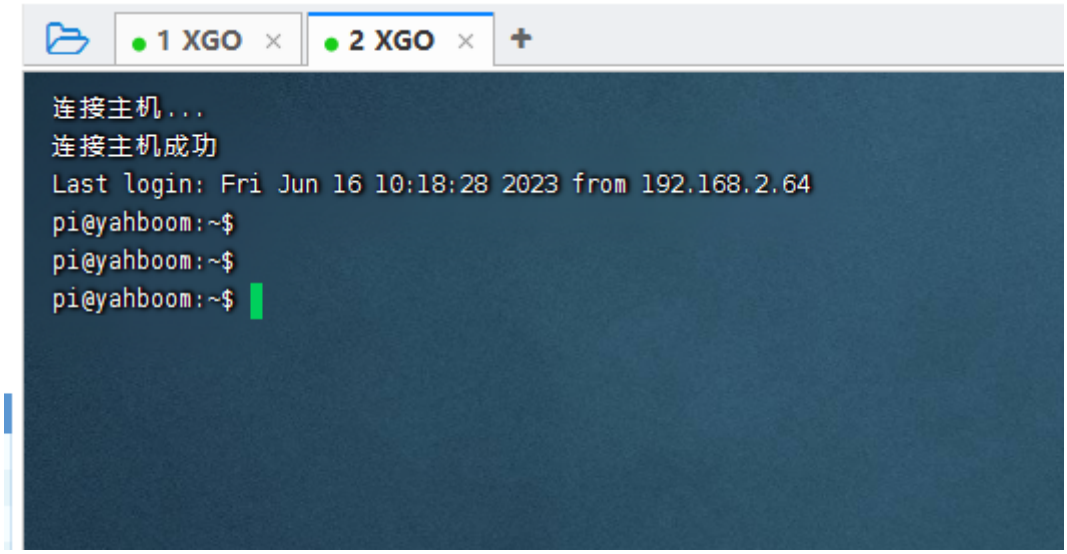
```
source install/setup.bash
```



Then enter the following command

```
ros2 run yahboom_image_publisher_c yahboom_image_publish_c
```

Restart a terminal in the same way as in item 2.

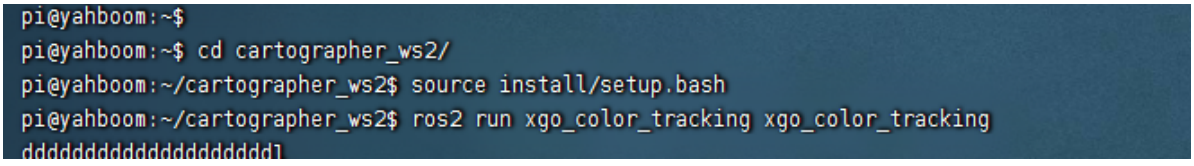


Enter the following command in a new terminal

```
cd cartographer_ws2/
```

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

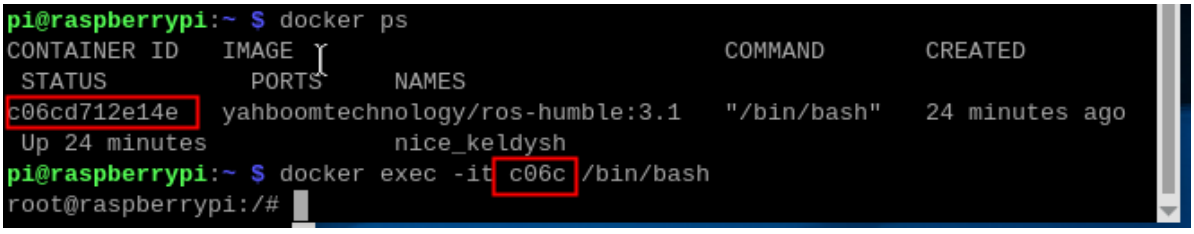
```
ros2 run yahboom_color_tracking yahboom_color_tracking
```



PI5 version steps:

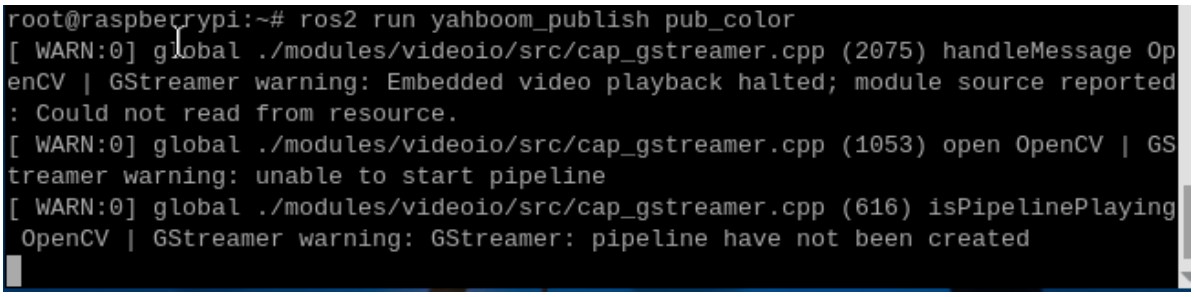
Enter in the root directory of the Raspberry Pi and enter the same terminal

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



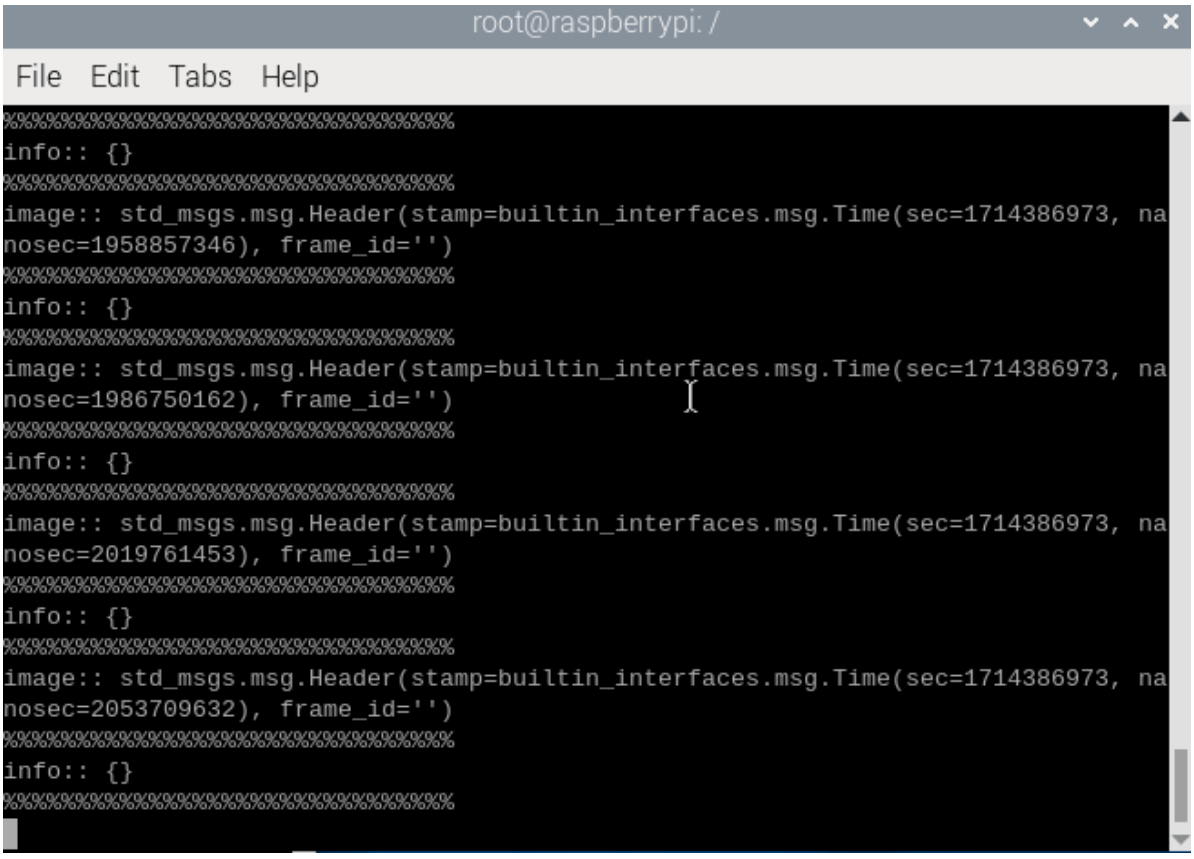
Then enter the following command

```
ros2 run yahboom_publish pub_color
```



Reopen the same docker terminal and enter the command:

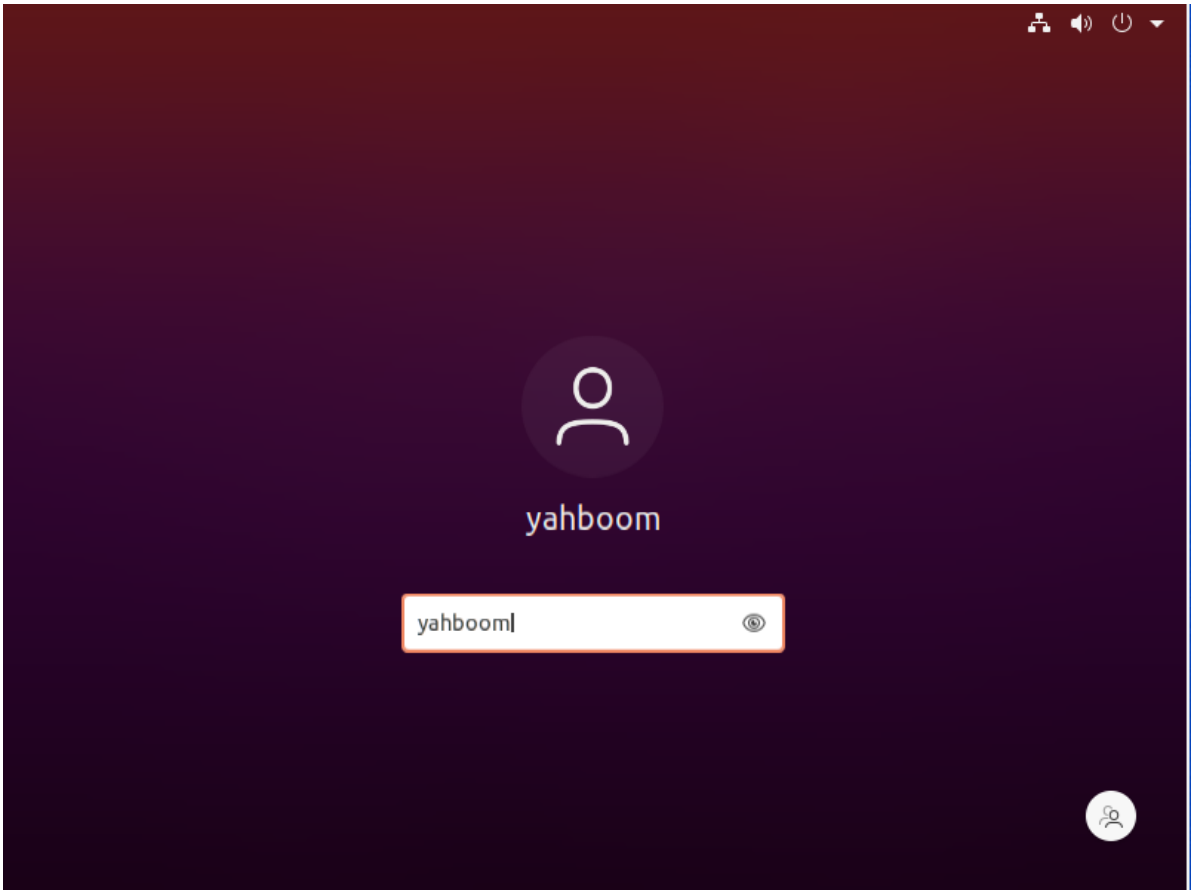
```
ros2 run yahboom_qrcode_tracking yahboom_qrcode_tracking
```



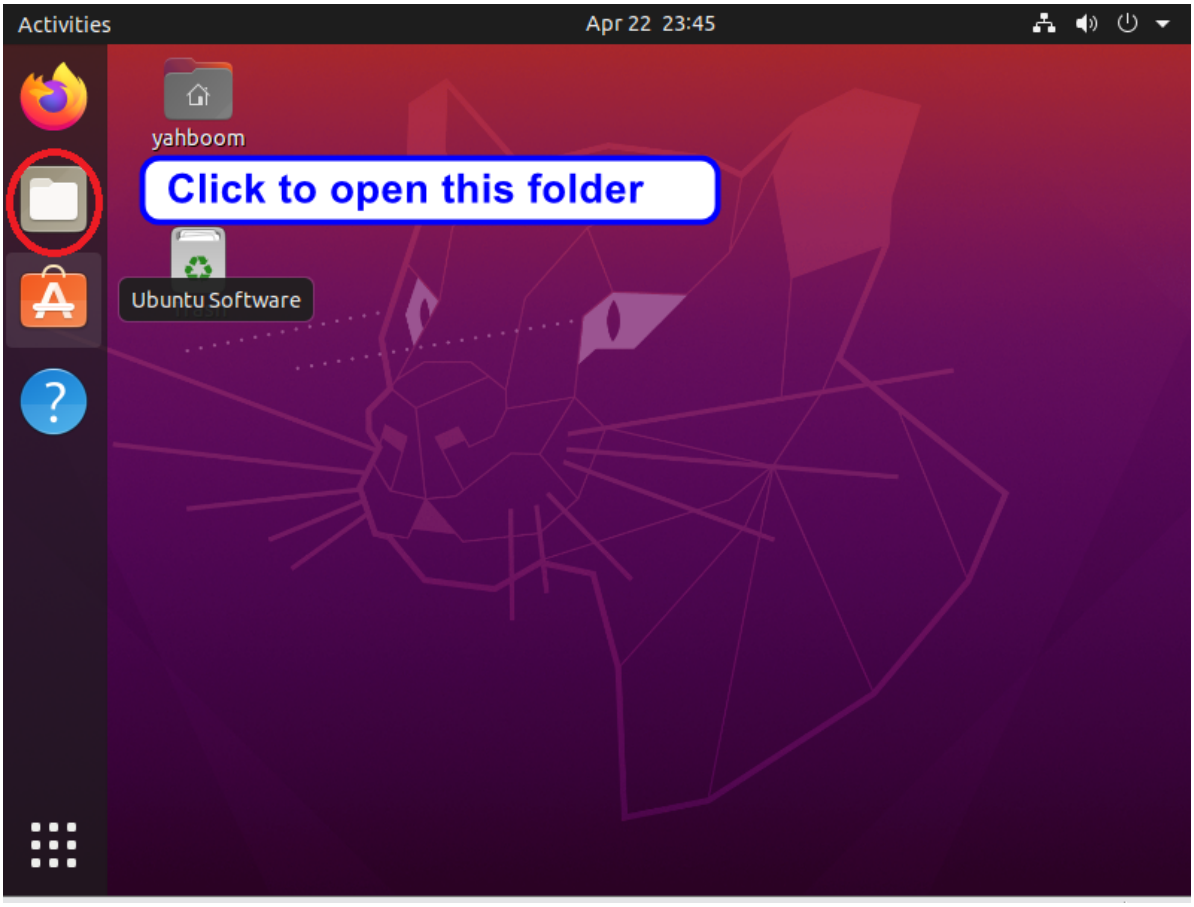
5. Set the recognition color through the web interface

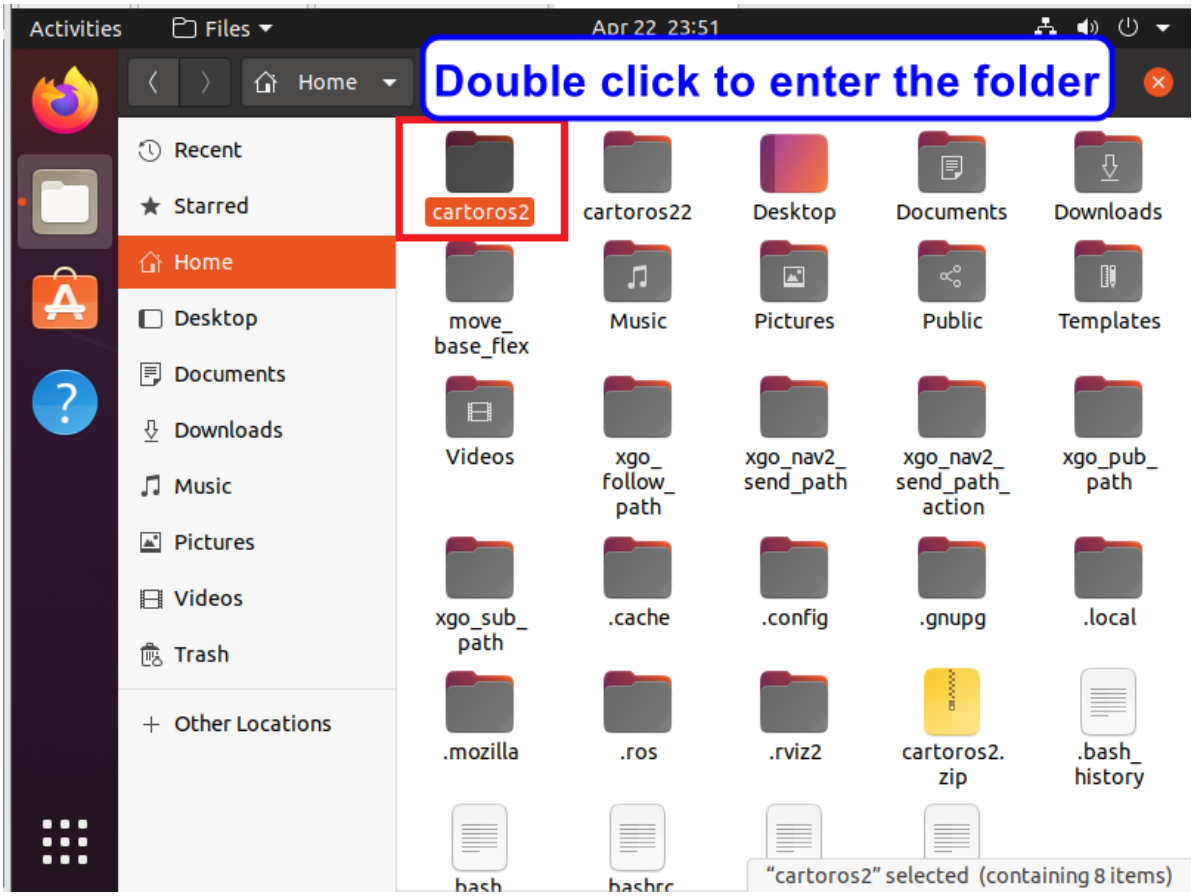
The steps are the same for PI4 and PI5 versions:

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

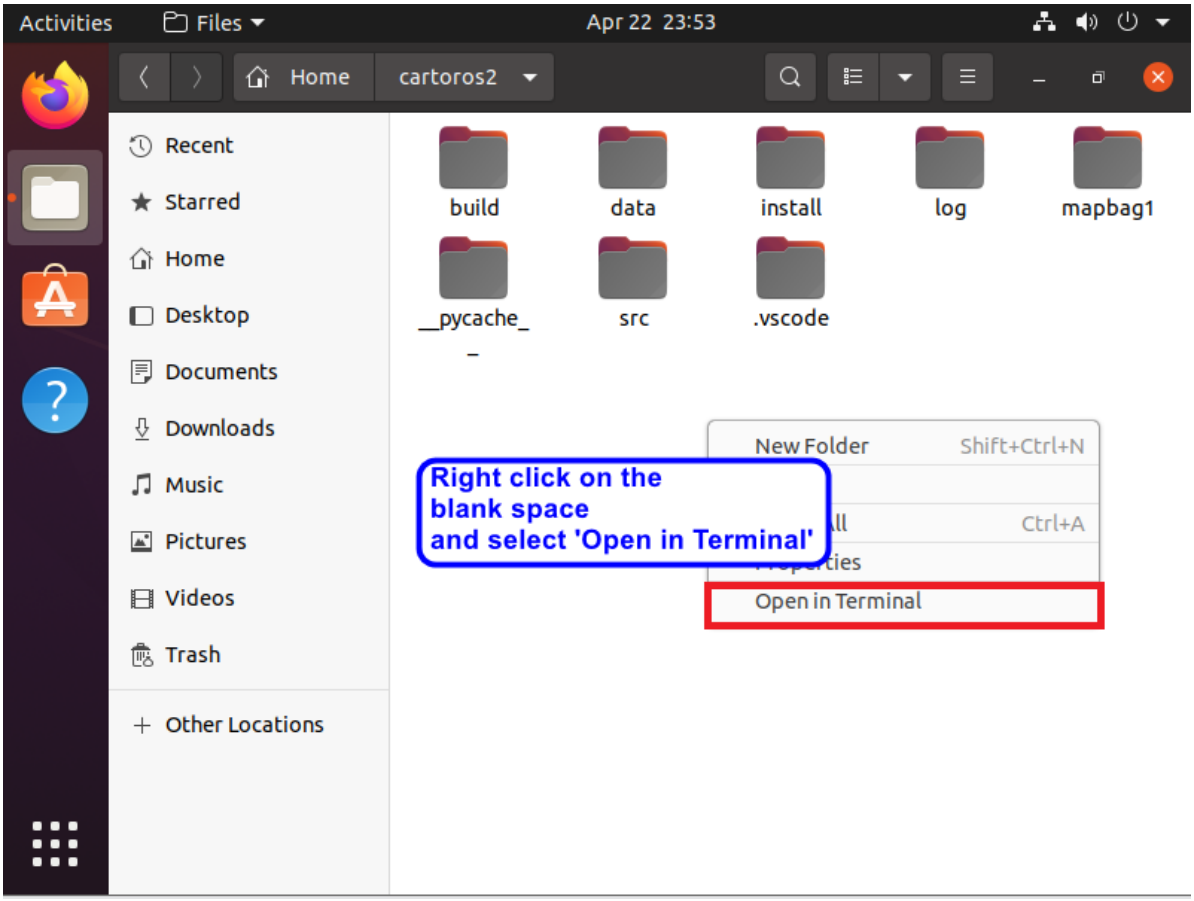


Click on the folder to open the cartoros2 folder.



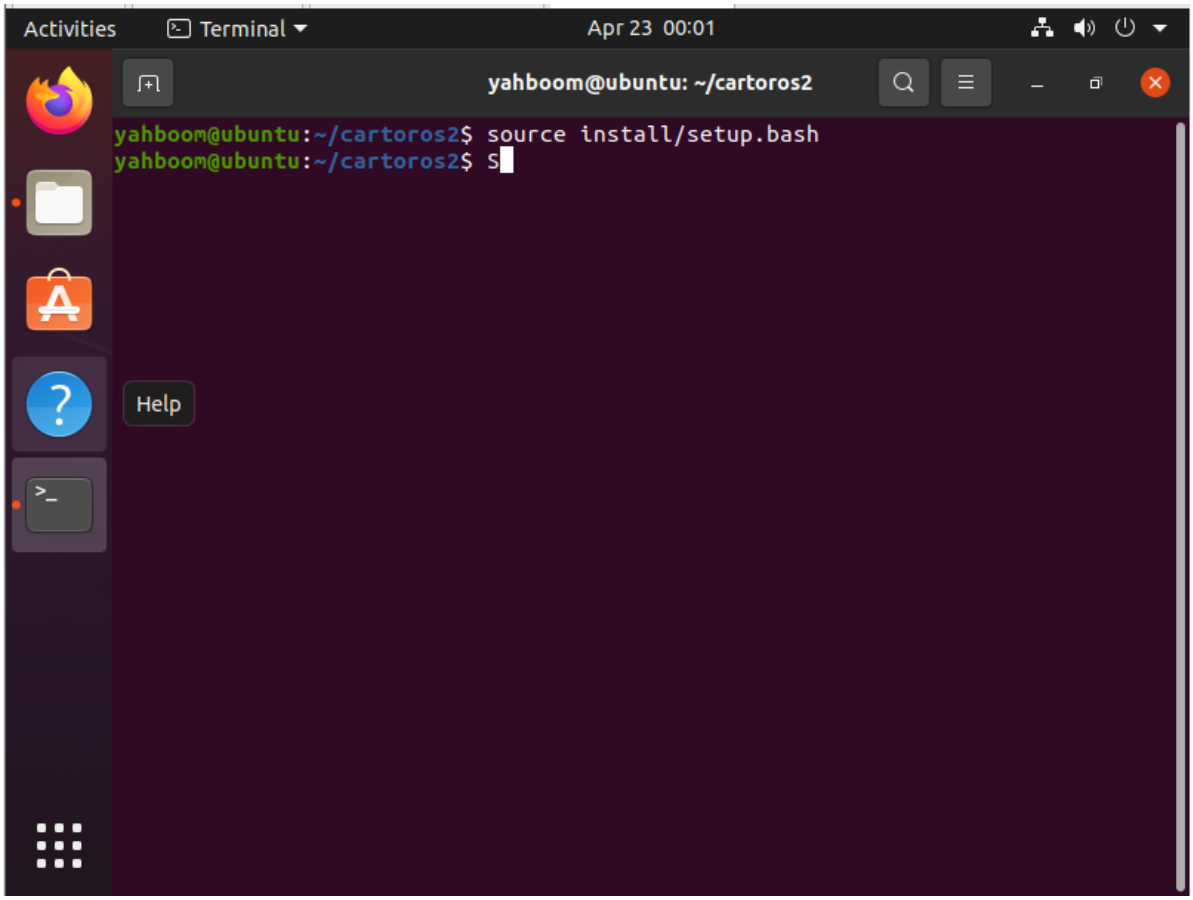


Open the terminal under the folder



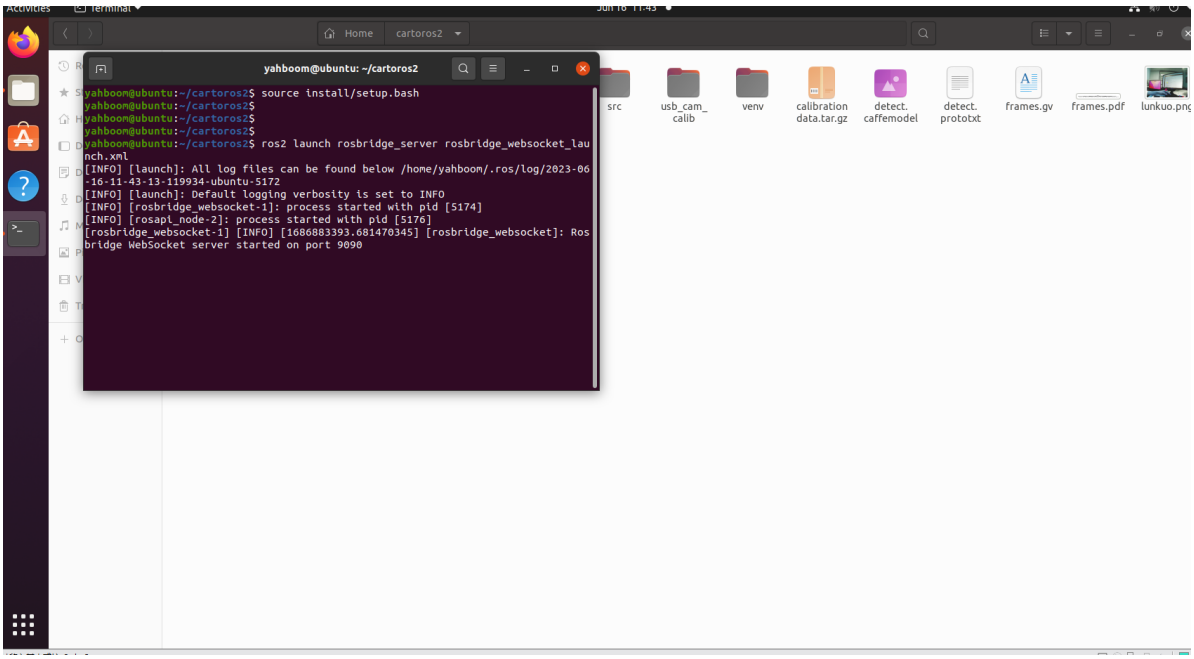
Then enter the following command

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



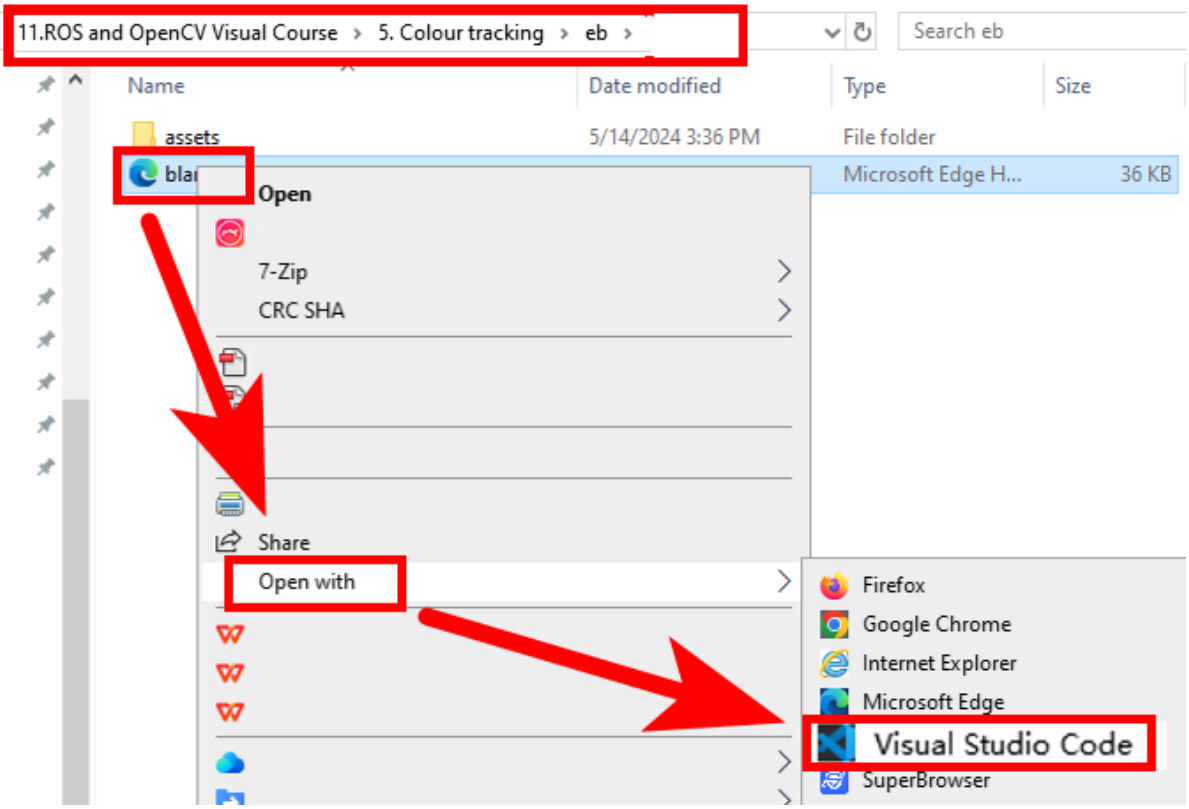
Then start rosbridge and enter the following command

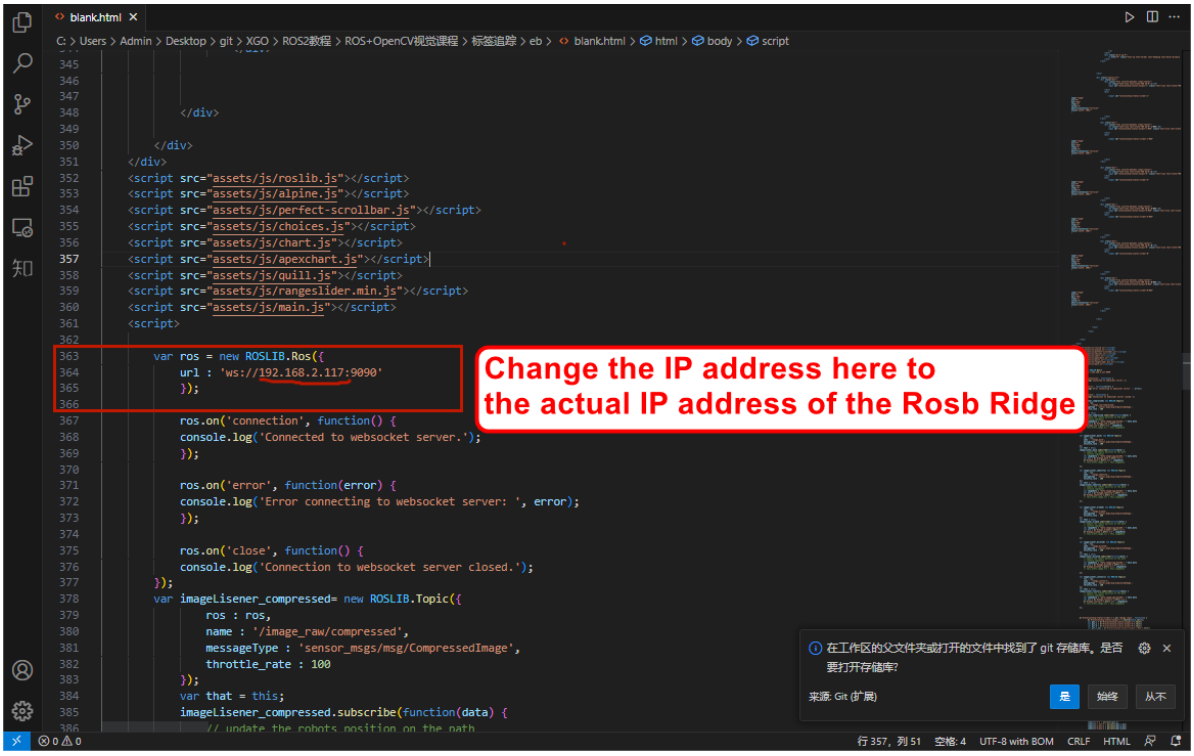
```
ros2 launch rosbridge_server rosbridge_websocket_launch.xml
```

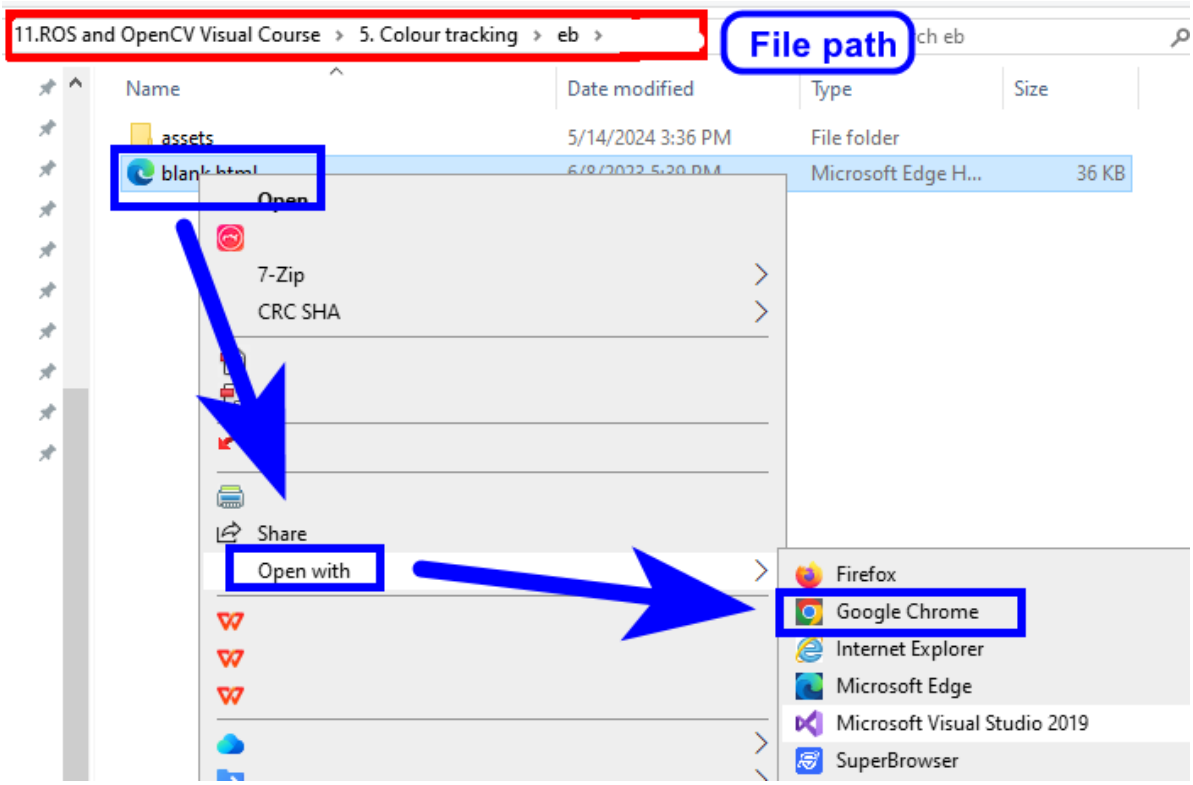


Find the blank.html file in the eb folder in the updated directory of this tutorial and open it with Google Chrome.

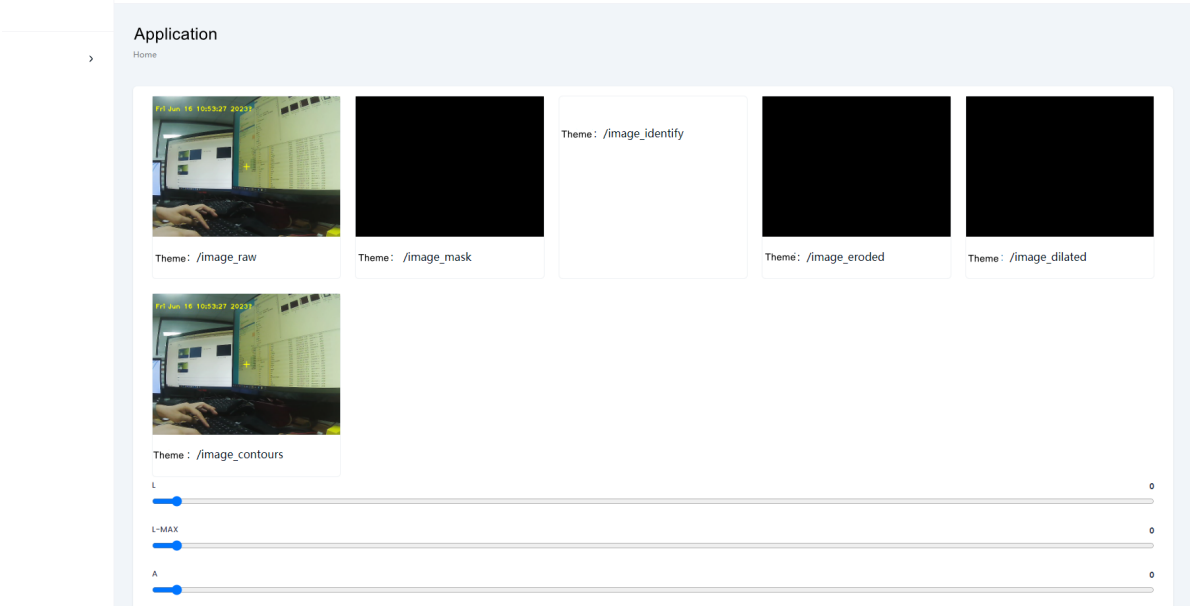
Note: The IP address of rosbridge needs to be set here. Obtain the IP address of the virtual machine, then open the blank.html file, modify the IP address of line 363 and save it, as shown in the figure below.


11.ROS and OpenCV Visual Course › 5. Colour tracking › eb ›
Search eb
Name
Date modified
Type
Size
assets
5/14/2024 3:36 PM
File folder
Microsoft Edge H...
36 KB
Open
7-Zip
CRC SHA
Share
Open with
Firefox
Google Chrome
Internet Explorer
Microsoft Edge
Visual Studio Code
SuperBrowser


blank.html
C: › Users › Admin › Desktop › git › XGO › ROS2教程 › ROS+OpenCV视觉课程 › 标签追踪 › eb › blank.html › html › body › script
<script src="assets/js/roslib.js"></script>
<script src="assets/js/alpine.js"></script>
<script src="assets/js/perfect-scrollbar.js"></script>
<script src="assets/js/choices.js"></script>
<script src="assets/js/chart.js"></script>
<script src="assets/js/apexchart.js"></script>
<script src="assets/js/quill.js"></script>
<script src="assets/js/rangeslider.min.js"></script>
<script src="assets/js/main.js"></script>
var ros = new ROSLIB.Ros({
url : 'ws://192.168.2.117:9090'
});
Change the IP address here to
the actual IP address of the Rosb Ridge
ros.on('connection', function() {
console.log('Connected to websocket server.');
});
ros.on('error', function(error) {
console.log('Error connecting to websocket server: ', error);
});
ros.on('close', function() {
console.log('Connection to websocket server closed.');
});
var imageLisener_compressed= new ROSLIB.Topic({
ros : ros,
name : '/image_raw/compressed',
messageType : 'sensor_msgs/msg/CompressedImage',
throttle_rate : 100
});
var that = this;
imageLisener_compressed.subscribe(function(data) {
要打开存储库?
来源: Git (扩展)
是
始终
从不
行 357，列 51
空格: 4
UTF-8 with BOM
CRLF
HTML



As shown in the picture below, you can see the pictures transmitted by the camera.



Then we set the LAB value of the color through the slider.

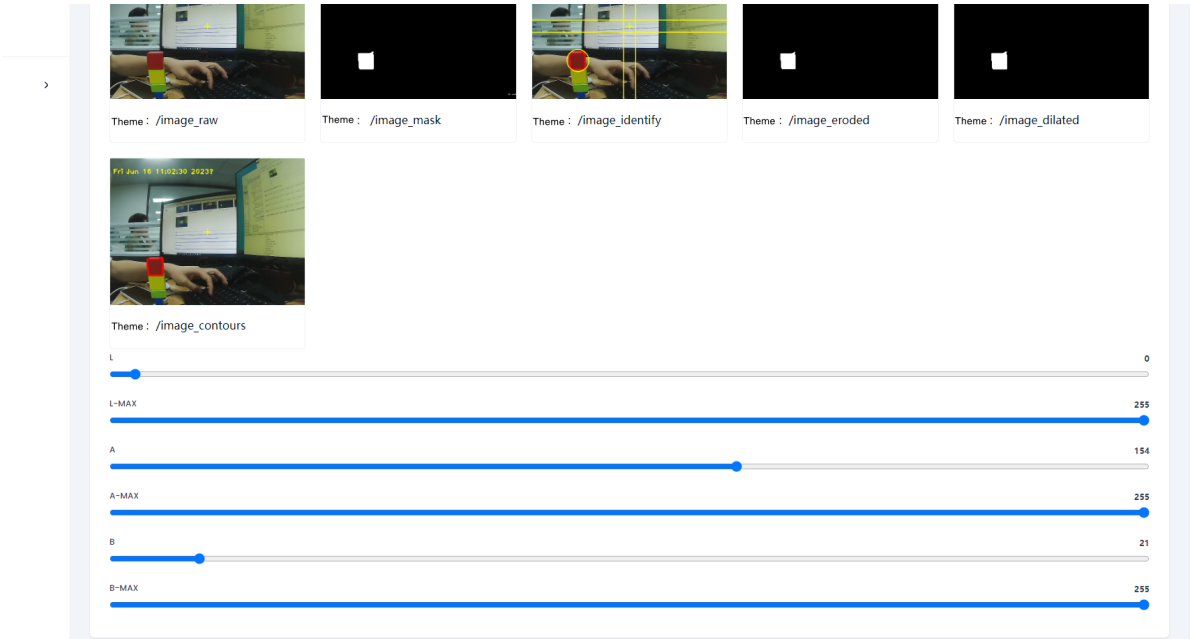
```
Yellow: {"l":96, "a": 55, "b":188, "l_max": 252 , "a_max": 141, "b_max": 255}
```

```
Red: {"l":0, "a": 155, "b":21, "l_max": 255 , "a_max": 255, "b_max": 255}
```

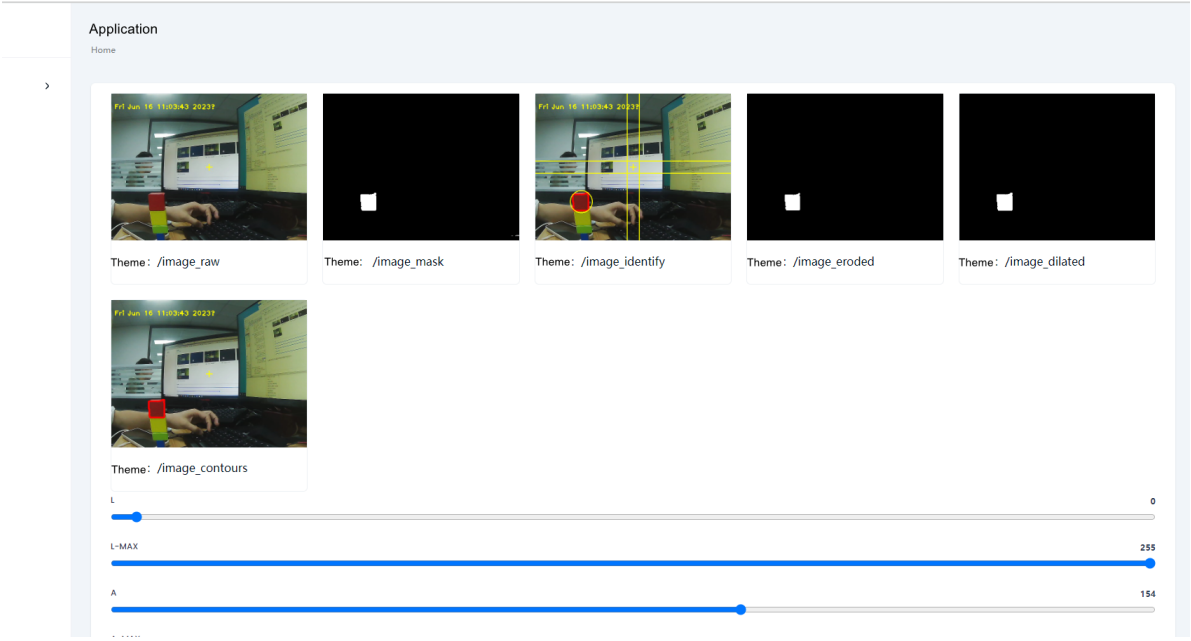
```
Green: {"l":26, "a": 7, "b":170, "l_max": 143 , "a_max": 110, "b_max": 255}
```

```
Blue: {"l":0, "a": 0, "b":0, "l_max": 255 , "a_max": 255, "b_max": 102}
```

Above are the LAB values of several colors. We can choose one to set. For example, we set red, as shown in the figure below, and move the slider.



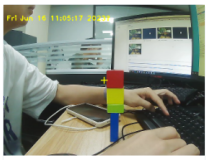
In the picture we can see that the red square has been identified.



Then the mechanical dog will adjust its attitude so that the red block is near the center of the screen.

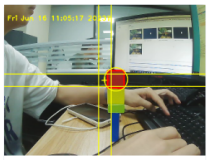
Application

Home



Theme: /image_raw

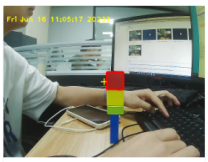
Theme: /image_mask



Theme: /image_identify

Theme: /image_eroded

Theme: /image_dilated



Theme: /image_contours

L 0

L-MAX 255

A 154

A-MAX 255