Connection module proxy

1、Enable proxy

If using a factory produced virtual machine system, you can input it at the terminal:

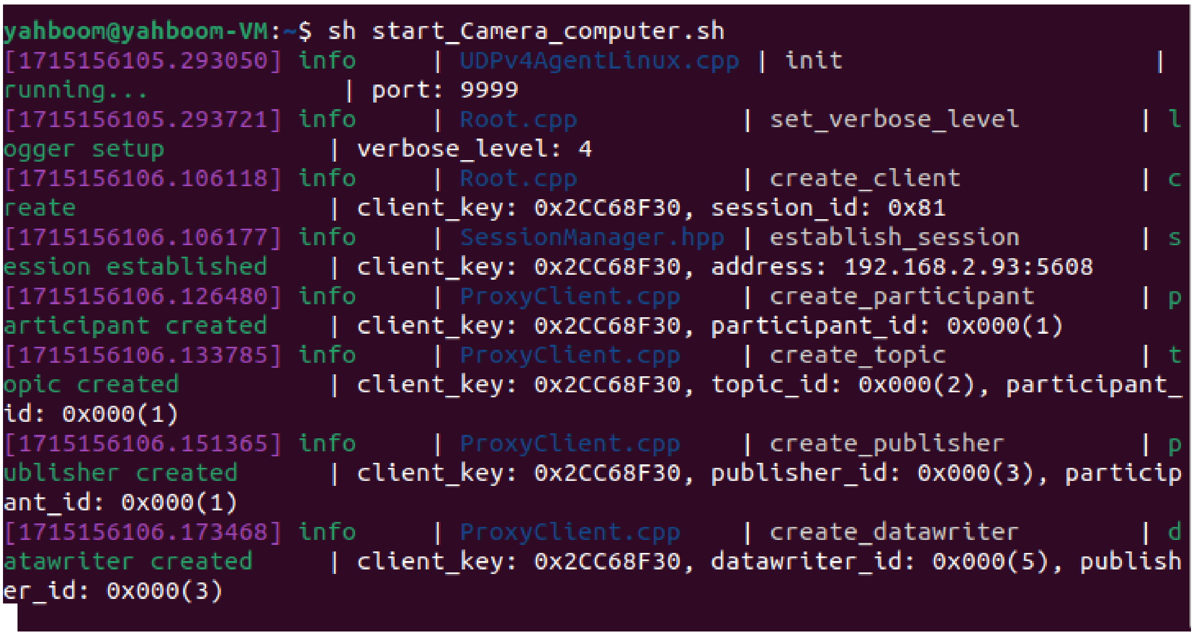
```
sh ~/start_Camera_computer.sh
```

If using a third-party virtual machine system, you need to first install the Docker development environment and open the terminal input:

```
docker run -it --rm -v /dev:/dev -v /dev/shm:/dev/shm --privileged --net=host
microros/micro-ros-agent:humble udp4 --port 9999 -v4
```

2、Connection proxy

After all the previous tutorials have been configured and confirmed to be correct,
Turn on the power switch of the car and automatically connect to the proxy. The connection is successful as shown in the following figure



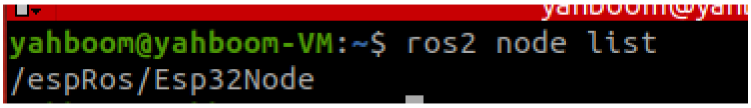
```
yahboom@yahboom-VM:~$ sh start_Camera_computer.sh
[1715156105.293050] info     | UDPv4AgentLinux.cpp | init                     |
running...             | port: 9999
[1715156105.293721] info     | Root.cpp            | set_verbose_level        | l
ogger setup            | verbose_level: 4
[1715156106.106118] info     | Root.cpp            | create_client            | c
reate                  | client_key: 0x2CC68F30, session_id: 0x81
[1715156106.106177] info     | SessionManager.hpp | establish_session         | s
ession established     | client_key: 0x2CC68F30, address: 192.168.2.93:5608
[1715156106.126480] info     | ProxyClient.cpp     | create_participant       | p
articipant created     | client_key: 0x2CC68F30, participant_id: 0x000(1)
[1715156106.133785] info     | ProxyClient.cpp     | create_topic             | t
opic created           | client_key: 0x2CC68F30, topic_id: 0x000(2), participant_
id: 0x000(1)
[1715156106.151365] info     | ProxyClient.cpp     | create_publisher         | p
ublisher created       | client_key: 0x2CC68F30, publisher_id: 0x000(3), particip
ant_id: 0x000(1)
[1715156106.173468] info     | ProxyClient.cpp     | create_datawriter        | d
atawriter created      | client_key: 0x2CC68F30, datawriter_id: 0x000(5), publish
er_id: 0x000(3)
```

Attention: If the connection is not successful, please check and confirm whether the ROS wifi image transmission module can be connected to the local area network normally, and whether the proxy IP address corresponds.

3、Test ROS nodes

Open the ROS2 terminal environment and enter the following command to view the /espRos/Esp32Node node name

```
ros2 node list
```



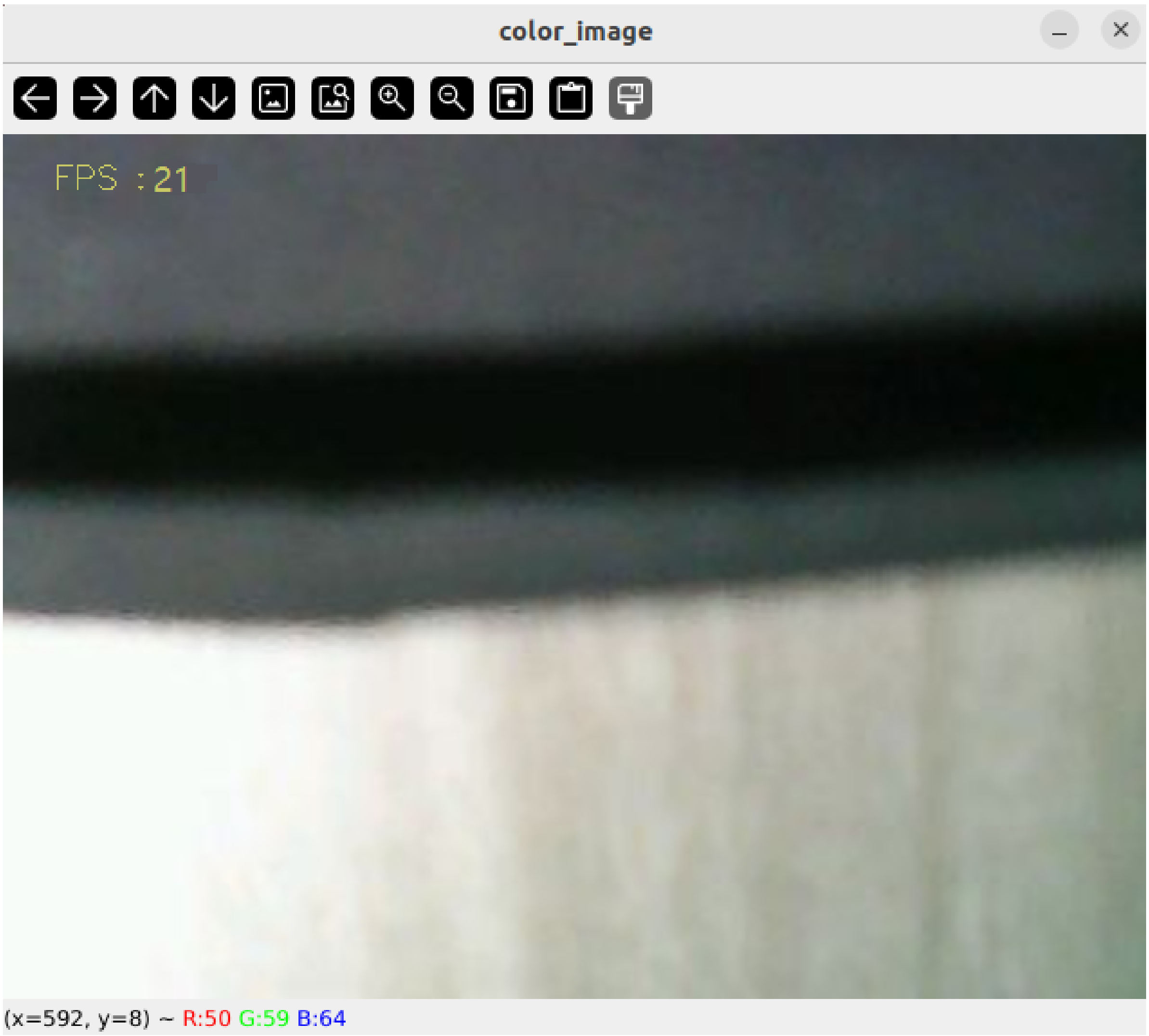
```
yahboom@yahboom-VM:~$ ros2 node list
/espRos/Esp32Node
```

If the/espRos/Esp32Node node cannot be searched, please check and confirm that the ROS DOMAIN ID of the. bashrc file on the virtual machine/computer must be 20. The ROS DOMAIN ID of the micro car must also be 20, otherwise the car cannot perform AI visual gameplay.

4、Viewing camera images

1. Open the terminal of the factory's virtual machine system and input

```
ros2 run yahboom_esp32_camera sub_img
```



2. If the camera is facing downwards and not in the middle, open a new terminal and enter (This step requires the virtual machine to connect to the agent of the car,How to connect can

be found in the **Quick Start Chapter**)

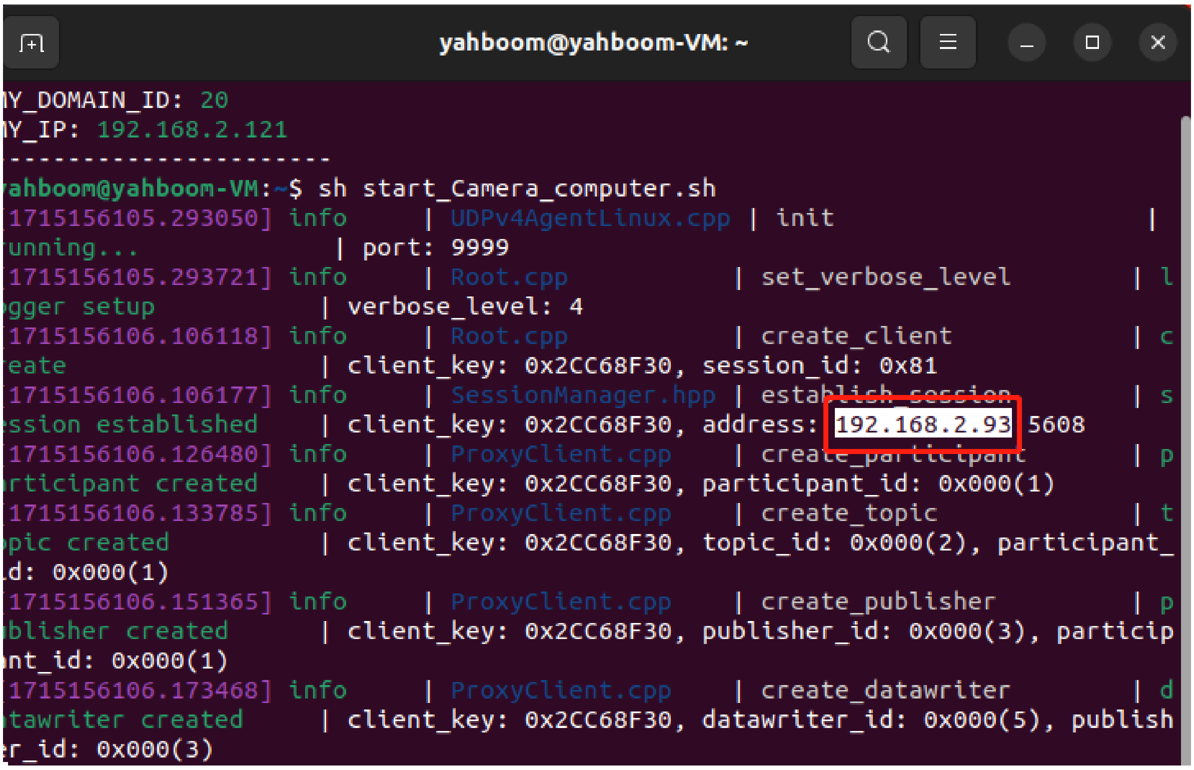
```
ros2 run yahboom_esp32_mediapipe control_servo
```

Wait for the camera to move in the middle, press "Ctrl+C" to terminate the program。
If the camera rudder maneuvers and does not return to the center, it is necessary to reinstall the camera while ensuring continuous power supply.

3. If the camera is inverted, open a new terminal input

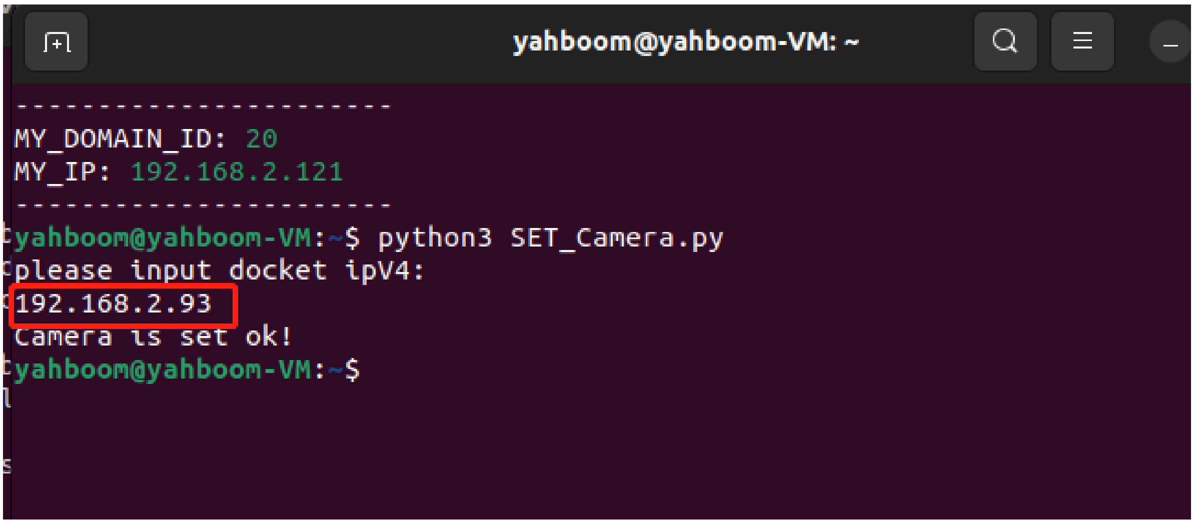
```
python3 ~/SET_Camera.py
```

Then enter the IP address to connect to the ROS wifi image transmission module, which can be viewed at the terminal that connects to the ROS wifi image transmission module proxy



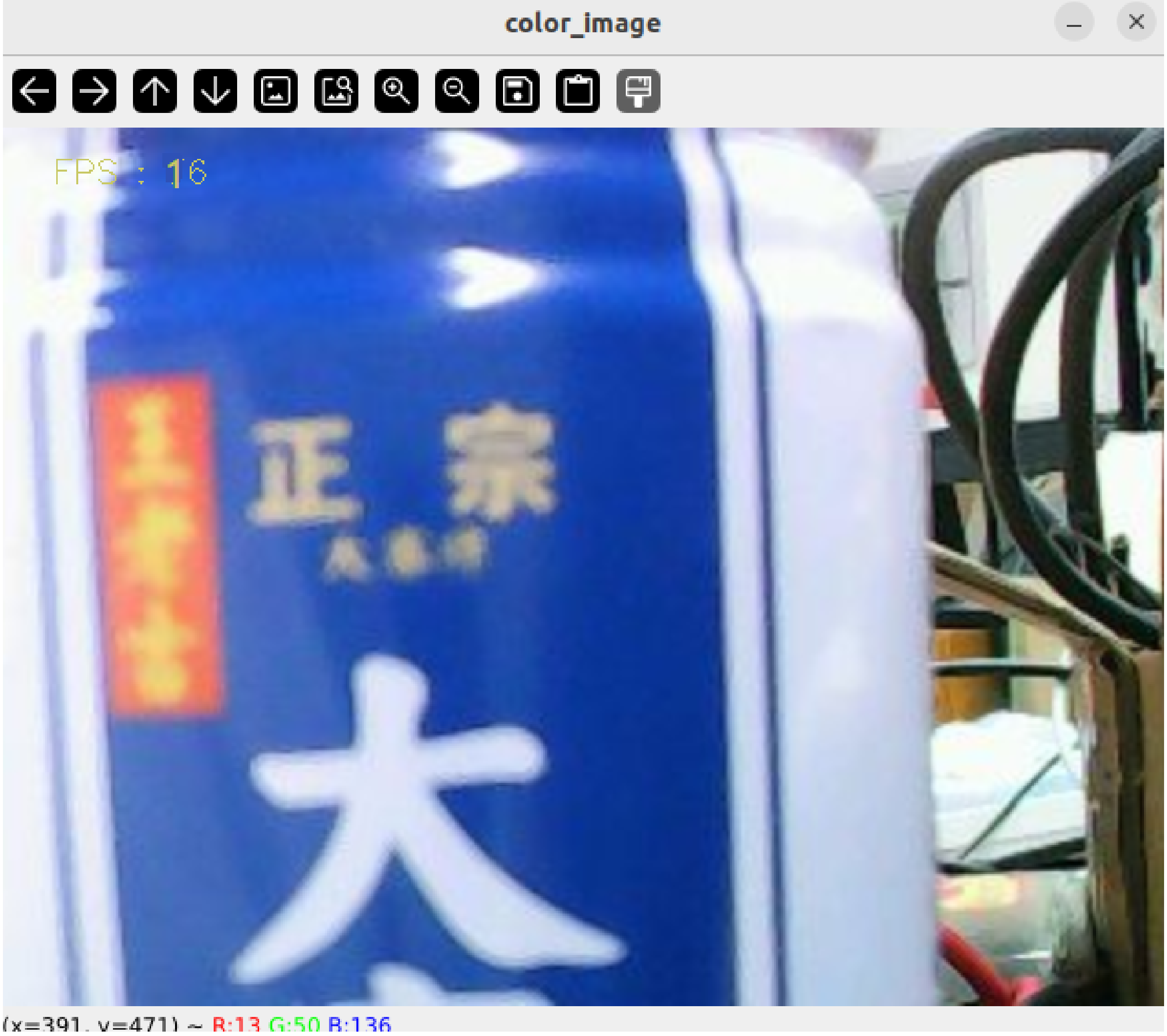
```
Y_DOMAIN_ID: 20
Y_IP: 192.168.2.121
----------------------
ahboom@yahboom-VM:~$ sh start_Camera_computer.sh
[1715156105.293050] info     | UDPv4AgentLinux.cpp | init                     |
unning...                 | port: 9999
[1715156105.293721] info     | Root.cpp            | set_verbose_level        | l
gger setup               | verbose_level: 4
[1715156106.106118] info     | Root.cpp            | create_client            | c
eate                     | client_key: 0x2CC68F30, session_id: 0x81
[1715156106.106177] info     | SessionManager.hpp  | establish_session        | s
ssion established        | client_key: 0x2CC68F30, address: 192.168.2.93 5608
[1715156106.126480] info     | ProxyClient.cpp     | create_participant       | p
rticipant created        | client_key: 0x2CC68F30, participant_id: 0x000(1)
[1715156106.133785] info     | ProxyClient.cpp     | create_topic             | t
pic created              | client_key: 0x2CC68F30, topic_id: 0x000(2), participant_
d: 0x000(1)
[1715156106.151365] info     | ProxyClient.cpp     | create_publisher         | p
blisher created          | client_key: 0x2CC68F30, publisher_id: 0x000(3), particip
nt_id: 0x000(1)
[1715156106.173468] info     | ProxyClient.cpp     | create_datawriter        | d
tawriter created         | client_key: 0x2CC68F30, datawriter_id: 0x000(5), publish
r_id: 0x000(3)
```

Enter the IP address at the terminal based on the detected IP address, and then press enter



```
-------------------------
MY_DOMAIN_ID: 20
MY_IP: 192.168.2.121
-------------------------
yahboom@yahboom-VM:~$ python3 SET_Camera.py
please input docket ipV4:
192.168.2.93
Camera is set ok!
yahboom@yahboom-VM:~$
```

The image has been restored



If it is a third-party system, you need to port it yourself according to the feature package of yahboomcar_ws.