

# Connect to ROS-WiFi cam module agent

## 1. Open the agent

If you are using the factory virtual machine system, you can enter in the terminal:

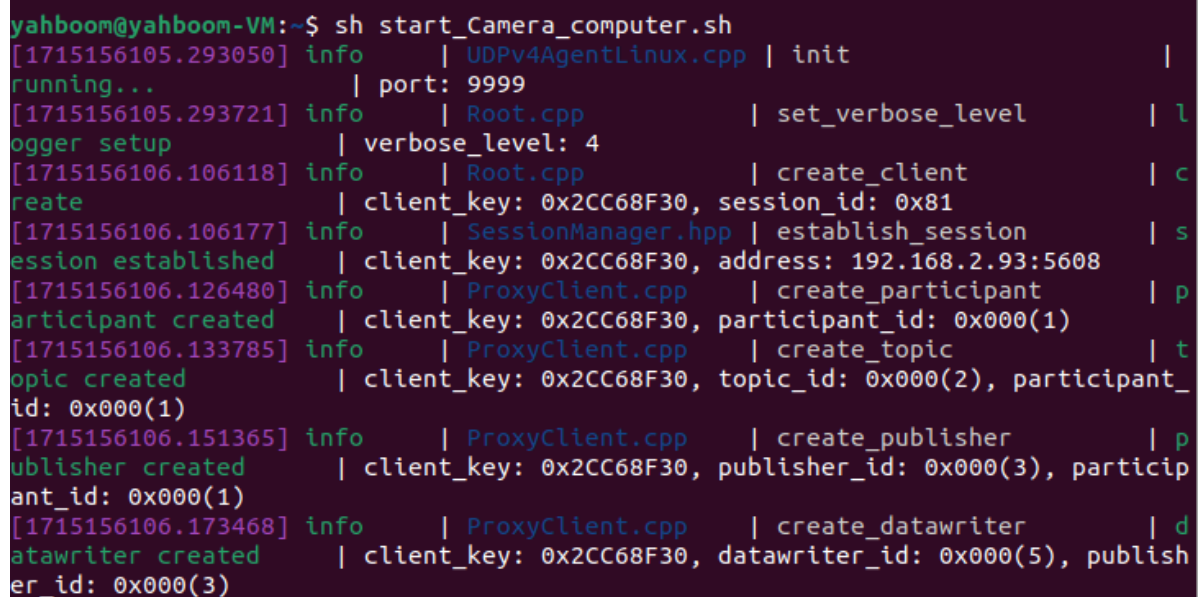
```
sh ~/start_Camera_computer.sh
```

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

```
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. Connect to the agent

After the previous tutorials are configured and confirmed, turn on the power switch of the car to automatically connect to the agent. The connection is successful as shown in the figure below



```
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  
logger setup        | verbose_level: 4  
[1715156106.106118] info      | Root.cpp            | create_client  
create              | client_key: 0x2CC68F30, session_id: 0x81  
[1715156106.106177] info      | SessionManager.hpp  | establish_session  
session established | client_key: 0x2CC68F30, address: 192.168.2.93:5608  
[1715156106.126480] info      | ProxyClient.cpp     | create_participant  
participant created | client_key: 0x2CC68F30, participant_id: 0x000(1)  
[1715156106.133785] info      | ProxyClient.cpp     | create_topic  
topic created       | client_key: 0x2CC68F30, topic_id: 0x000(2), participant_id: 0x000(1)  
[1715156106.151365] info      | ProxyClient.cpp     | create_publisher  
publisher created   | client_key: 0x2CC68F30, publisher_id: 0x000(3), participant_id: 0x000(1)  
[1715156106.173468] info      | ProxyClient.cpp     | create_datawriter  
datawriter created  | client_key: 0x2CC68F30, datawriter_id: 0x000(5), publisher_id: 0x000(3)
```

Note: 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 node

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

```
ros2 node list
```

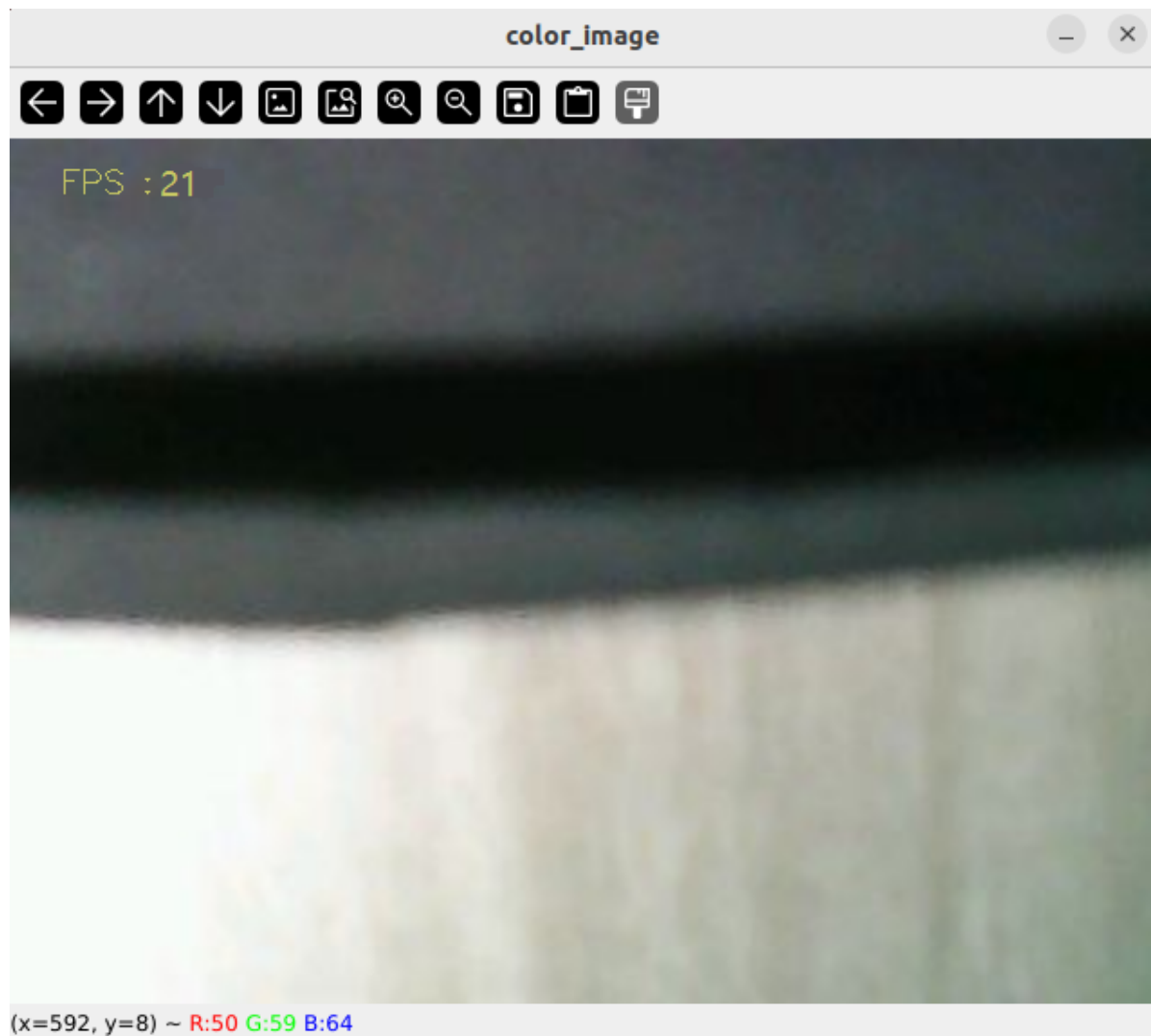
```
yahboom@yahboom-VM:~$ ros2 node list  
/espRos/Esp32Node
```

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 vision gameplay.

## 4. View the camera screen

1. Open the terminal of the factory virtual machine system and enter

```
ros2 run yahboom_esp32_camera sub_img
```



2. If the camera is inverted, open a new terminal and enter

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

Then enter the IP address connected to the ROS-wifi image transmission module, and you can view it in the terminal connected to the ROS-wifi image transmission module agent

```
yahboom@yahboom-VM: ~  
MY_DOMAIN_ID: 20  
MY_IP: 192.168.2.121  
-----  
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  
logger setup        | verbose_level: 4     |  
1715156106.106118] info      | Root.cpp             | create_client    | c  
create              | client_key: 0x2CC68F30, session_id: 0x81  
1715156106.106177] info      | SessionManager.hpp   | establish_session | s  
session established | client_key: 0x2CC68F30, address: 192.168.2.93 5608  
1715156106.126480] info      | ProxyClient.cpp      | create_participant | p  
participant created | client_key: 0x2CC68F30, participant_id: 0x000(1)  
1715156106.133785] info      | ProxyClient.cpp      | create_topic      | t  
topic created       | client_key: 0x2CC68F30, topic_id: 0x000(2), participant_id: 0x000(1)  
1715156106.151365] info      | ProxyClient.cpp      | create_publisher   | p  
publisher created   | client_key: 0x2CC68F30, publisher_id: 0x000(3), participant_id: 0x000(1)  
1715156106.173468] info      | ProxyClient.cpp      | create_datawriter  | d  
datawriter created  | client_key: 0x2CC68F30, datawriter_id: 0x000(5), publisher_id: 0x000(3)
```

According to the ip found Enter the IP address in the terminal and press Enter

```
yahboom@yahboom-VM: ~  
-----  
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 will be restored



If it is a third-party system, you need to transplant it yourself according to the function package of yahboomcar\_ws.