Handle remote control

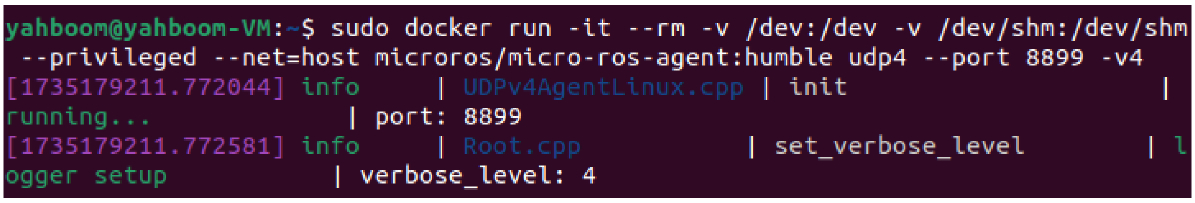
Note: The virtual machine needs to be in the same LAN as the car, and the ROS_DOMAIN_ID needs to be consistent. You can check [Must-read before use] to set the IP and ROS_DOMAIN_ID on the board.

Here, we take the [PS2-2.4G wireless controller] sold by Yabo Intelligent Technology Co., Ltd. as an example to explain how to quickly start the program and control the car. The code here is only applicable to the above-mentioned controllers, not other controllers.

1. Start and connect the agent

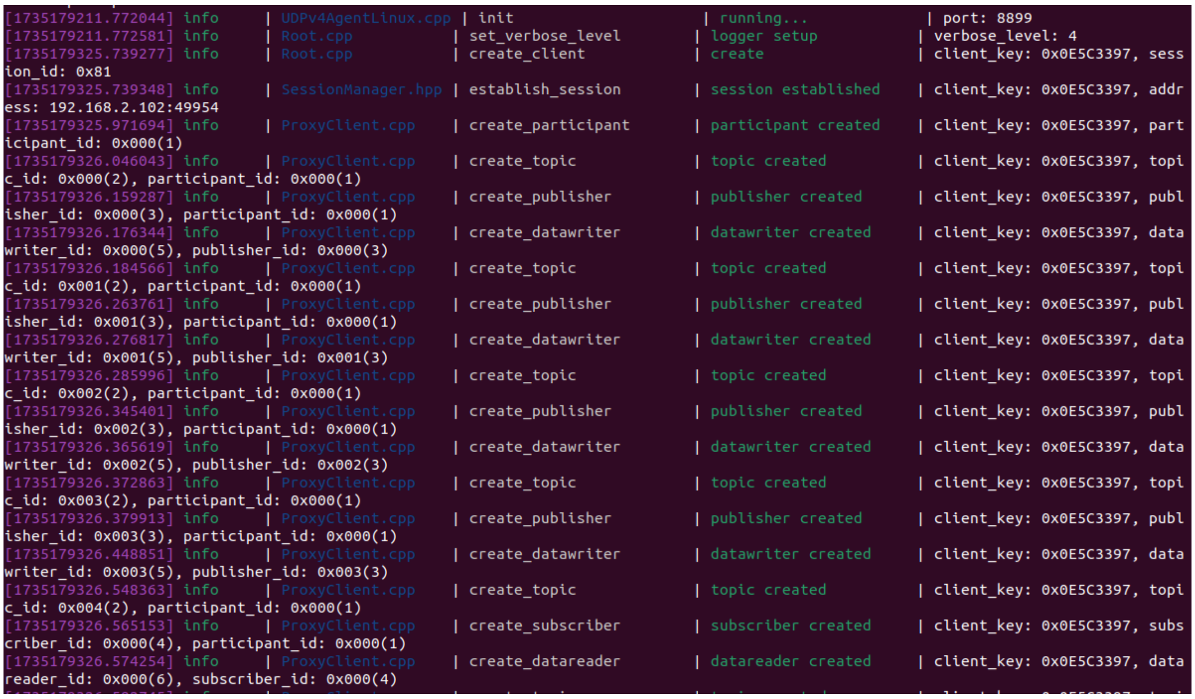
Take the matching virtual machine as an example, enter the following command to start the agent,

```
sudo docker run -it --rm -v /dev:/dev -v /dev/shm:/dev/shm --privileged --
net=host microros/micro-ros-agent:humble udp4 --port 8899 -v4
```



```
yahboom@yahboom-VM:~$ sudo docker run -it --rm -v /dev:/dev -v /dev/shm:/dev/shm
 --privileged --net=host microros/micro-ros-agent:humble udp4 --port 8899 -v4
[1735179211.772044] info     | UDPv4AgentLinux.cpp | init                     |
running...              | port: 8899
[1735179211.772581] info     | Root.cpp            | set_verbose_level        | l
ogger setup             | verbose_level: 4
```

Then, turn on the car switch and wait for the car to connect to the agent. The connection is successful as shown in the figure below,



```
[1735179211.772044] info     | UDPv4AgentLinux.cpp | init                     | running...              | port: 8899
[1735179211.772581] info     | Root.cpp           | set_verbose_level        | logger setup            | verbose_level: 4
[1735179325.739277] info     | Root.cpp           | create_client            | create                  | client_key: 0x0E5C3397, sess
ion_id: 0x81
[1735179325.739348] info     | SessionManager.hpp | establish_session        | session established     | client_key: 0x0E5C3397, addr
ess: 192.168.2.102:49954
[1735179325.971694] info     | ProxyClient.cpp    | create_participant       | participant created     | client_key: 0x0E5C3397, part
icipant_id: 0x000(1)
[1735179326.046043] info     | ProxyClient.cpp    | create_topic             | topic created           | client_key: 0x0E5C3397, topi
c_id: 0x000(2), participant_id: 0x000(1)
[1735179326.159287] info     | ProxyClient.cpp    | create_publisher         | publisher created       | client_key: 0x0E5C3397, publ
isher_id: 0x000(3), participant_id: 0x000(1)
[1735179326.176344] info     | ProxyClient.cpp    | create_datawriter        | datawriter created      | client_key: 0x0E5C3397, data
writer_id: 0x000(5), publisher_id: 0x000(3)
[1735179326.184566] info     | ProxyClient.cpp    | create_topic             | topic created           | client_key: 0x0E5C3397, topi
c_id: 0x001(2), participant_id: 0x000(1)
[1735179326.263761] info     | ProxyClient.cpp    | create_publisher         | publisher created       | client_key: 0x0E5C3397, publ
isher_id: 0x001(3), participant_id: 0x000(1)
[1735179326.276817] info     | ProxyClient.cpp    | create_datawriter        | datawriter created      | client_key: 0x0E5C3397, data
writer_id: 0x001(5), publisher_id: 0x001(3)
[1735179326.285996] info     | ProxyClient.cpp    | create_topic             | topic created           | client_key: 0x0E5C3397, topi
c_id: 0x002(2), participant_id: 0x000(1)
[1735179326.345401] info     | ProxyClient.cpp    | create_publisher         | publisher created       | client_key: 0x0E5C3397, publ
isher_id: 0x002(3), participant_id: 0x000(1)
[1735179326.365619] info     | ProxyClient.cpp    | create_datawriter        | datawriter created      | client_key: 0x0E5C3397, data
writer_id: 0x002(5), publisher_id: 0x002(3)
[1735179326.372863] info     | ProxyClient.cpp    | create_topic             | topic created           | client_key: 0x0E5C3397, topi
c_id: 0x003(2), participant_id: 0x000(1)
[1735179326.379913] info     | ProxyClient.cpp    | create_publisher         | publisher created       | client_key: 0x0E5C3397, publ
isher_id: 0x003(3), participant_id: 0x000(1)
[1735179326.448851] info     | ProxyClient.cpp    | create_datawriter        | datawriter created      | client_key: 0x0E5C3397, data
writer_id: 0x003(5), publisher_id: 0x003(3)
[1735179326.548363] info     | ProxyClient.cpp    | create_topic             | topic created           | client_key: 0x0E5C3397, topi
c_id: 0x004(2), participant_id: 0x000(1)
[1735179326.565153] info     | ProxyClient.cpp    | create_subscriber        | subscriber created      | client_key: 0x0E5C3397, subs
criber_id: 0x000(4), participant_id: 0x000(1)
[1735179326.574254] info     | ProxyClient.cpp    | create_datareader        | datareader created      | client_key: 0x0E5C3397, data
reader_id: 0x000(6), subscriber_id: 0x000(4)
```

Then, connect the receiver of the handle to the host of the virtual machine and choose to connect to the virtual machine.

2. Start the handle control program

2.1 Code path

Virtual machine code path:

```
/home/yahboom/yahboomcar_ws/src/yahboomcar_ctrl/yahboomcar_ctrl
```

2.2 Run command

Take the matching virtual machine as an example, input in the terminal,

Start the chassis driver,

```
ros2 launch yahboomcar_bringup yahboomcar_bringup_launch.py
```

Open another terminal and input,

```
ros2 launch yahboomcar_ctrl yahboomcar_joy_launch.py
```

Observe the handle indicator light, if it is always on, it means the connection is successful. After the program starts, press [START] and the buzzer of the car will sound. Press the R1 key, and the terminal prints the following information, turning on the handle control. You can use the left joystick to control the car forward and backward; you can use the right joystick to control the car to turn left and right;

```
[yahboom_joy-2] [INFO] [1735197789.731509428] [joy_ctrl]: Play:true
```