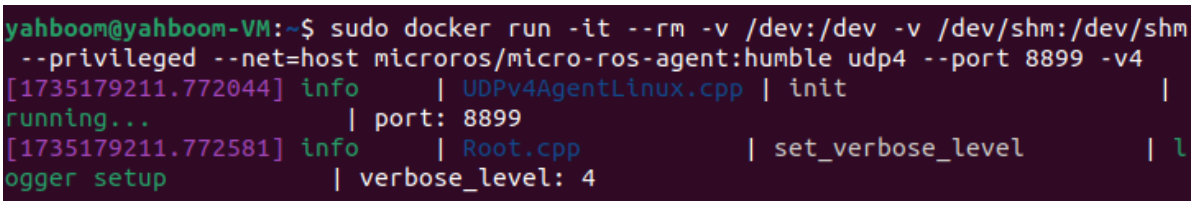
Keyboard remote control

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.

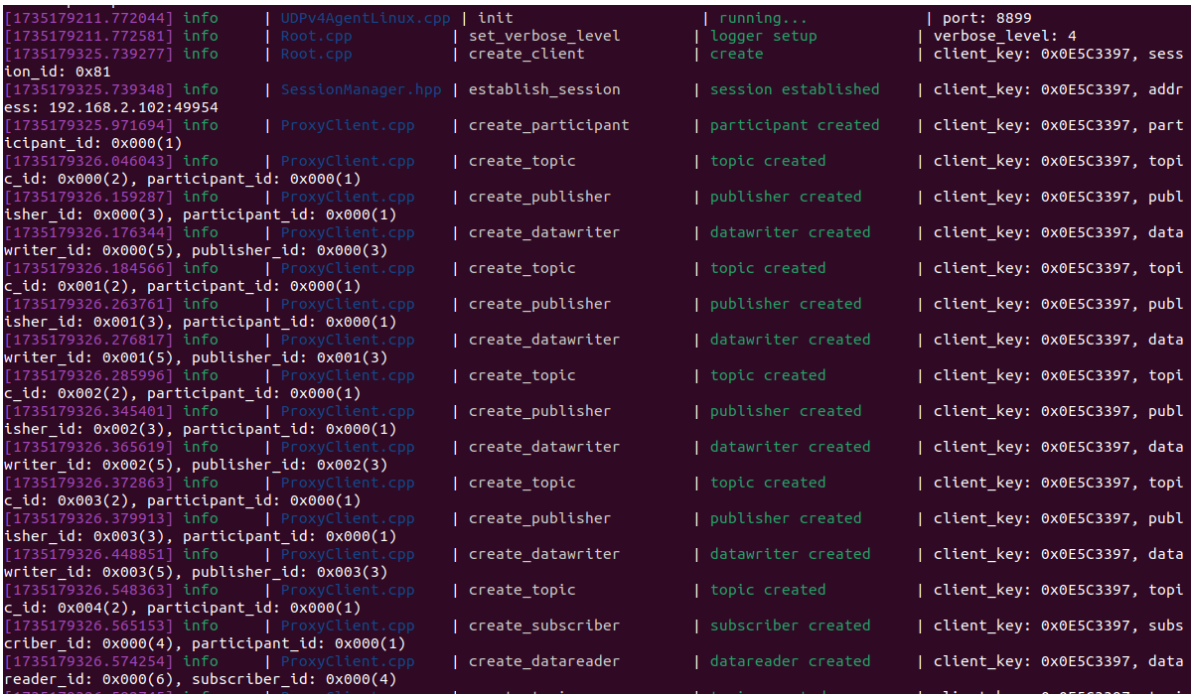
1. Start and connect the agent

Take the supporting 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
```



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,



2. Start the keyboard control program

2.1 Code path

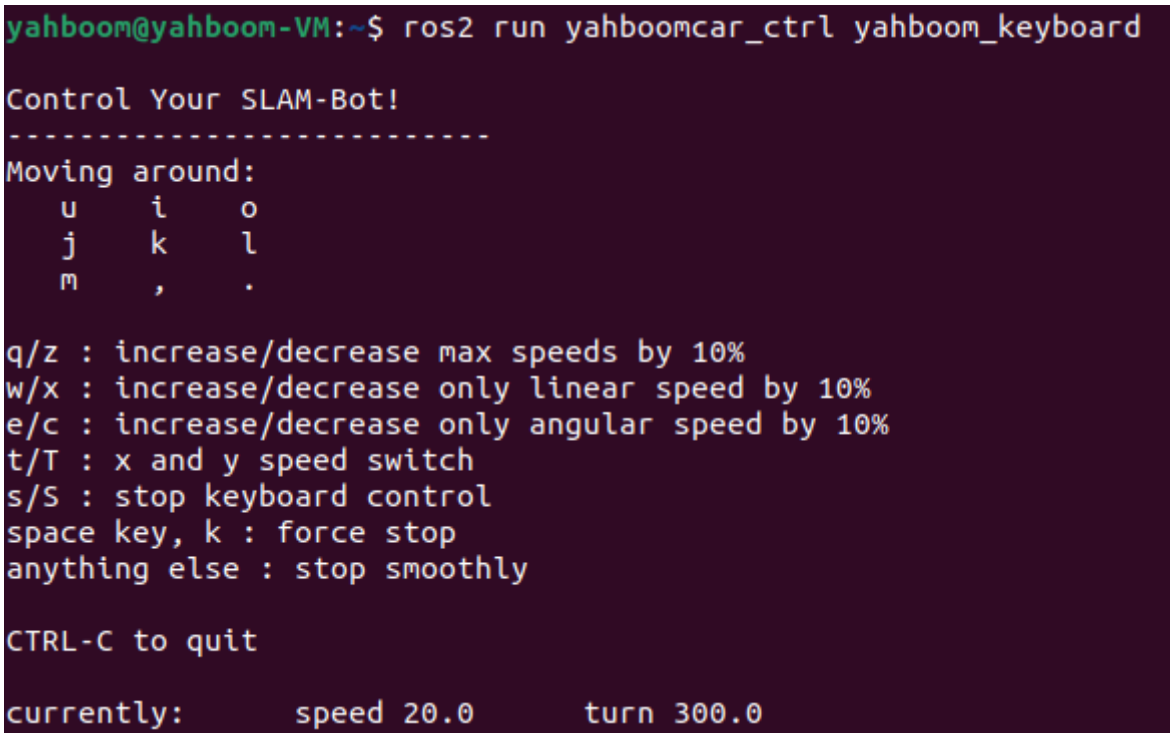
Virtual machine code path:

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

2.2 Run command

Take the matching virtual machine as an example, you can directly enter it in the terminal,

```
ros2 run yahboomcar_ctrl yahboom_keyboard
```



Then, according to the instructions below, press the corresponding button to control the movement of the car,

2.3 Direction control

Button	Linear angular velocity	Direction	Button	Linear angular velocity	Direction
【i】or 【I】	【linear，0】	Go forward	【u】or 【U】	【linear，angular】	Turn left
【,】	【-linear，0】	Go backward	【o】or 【O】	【linear，-angular】	Turn right
【j】or 【J】	【0，angular】	Rotate left	【m】or 【M】	【-linear，-angular】	Reverse left
【l】or 【L】	【0，-angular】	Rotate right	【.】	【-linear，angular】	Reverse right

2.4 Speed control

Button	Speed change	Button	Speed change
【q】	Increase both linear and angular velocities by 10%	【z】	Decrease both linear and angular velocities by 10%
【w】	Increase only linear velocity by 10%	【x】	Decrease only linear velocity by 10%
【e】	Increase only angular velocity by 10%	【c】	Decrease only angular velocity by 10%
【t】	Switch between linear velocity X-axis and Y-axis	【s】	Stop keyboard control

Note: Since the car is a two-wheeled balancing structure and cannot move sideways, the 【t】 key is meaningless.