

# Robot keyboard 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 [Read me] to set the IP and ROS\_DOMAIN\_ID on the board.

## 1. Program function description

After the program is started, the car movement can be controlled through the keyboard.

## 2. Start and connect to the agent

Taking 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 8090 -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 8090 -v4
[1704167422.995513] info | UDPv4AgentLinux.cpp | init |
running... | port: 8090
[1704167422.995832] info | Root.cpp | set_verbose_level | 1
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.

```
[1702630014.015846] info | ProxyClient.cpp | create_participant | participant created | client_key: 0x0B62A009, part
icipant_id: 0x000(1)
[1702630014.135363] info | ProxyClient.cpp | create_topic | topic created | client_key: 0x0B62A009, topl
c_id: 0x000(2), participant_id: 0x000(1)
[1702630014.223689] info | ProxyClient.cpp | create_publisher | publisher created | client_key: 0x0B62A009, publ
isher_id: 0x000(3), participant_id: 0x000(1)
[1702630014.415510] info | ProxyClient.cpp | create_datawriter | datawriter created | client_key: 0x0B62A009, data
writer_id: 0x000(5), publisher_id: 0x000(3)
[1702630014.428530] info | ProxyClient.cpp | create_topic | topic created | client_key: 0x0B62A009, topl
c_id: 0x001(2), participant_id: 0x000(1)
[1702630014.527190] info | ProxyClient.cpp | create_publisher | publisher created | client_key: 0x0B62A009, publ
isher_id: 0x001(3), participant_id: 0x000(1)
[1702630014.543889] info | ProxyClient.cpp | create_datawriter | datawriter created | client_key: 0x0B62A009, data
writer_id: 0x001(5), publisher_id: 0x001(3)
[1702630014.554490] info | ProxyClient.cpp | create_topic | topic created | client_key: 0x0B62A009, topl
c_id: 0x002(2), participant_id: 0x000(1)
[1702630014.737059] info | ProxyClient.cpp | create_publisher | publisher created | client_key: 0x0B62A009, publ
isher_id: 0x002(3), participant_id: 0x000(1)
[1702630014.755072] info | ProxyClient.cpp | create_datawriter | datawriter created | client_key: 0x0B62A009, data
writer_id: 0x002(5), publisher_id: 0x002(3)
[1702630014.818985] info | ProxyClient.cpp | create_topic | topic created | client_key: 0x0B62A009, topl
c_id: 0x003(2), participant_id: 0x000(1)
[1702630014.840001] info | ProxyClient.cpp | create_subscriber | subscriber created | client_key: 0x0B62A009, subs
criber_id: 0x000(4), participant_id: 0x000(1)
[1702630014.864010] info | ProxyClient.cpp | create_datareader | datareader created | client_key: 0x0B62A009, data
reader_id: 0x000(6), subscriber_id: 0x000(4)
[1702630014.959908] info | ProxyClient.cpp | create_topic | topic created | client_key: 0x0B62A009, topl
c_id: 0x004(2), participant_id: 0x000(1)
[1702630015.033537] info | ProxyClient.cpp | create_subscriber | subscriber created | client_key: 0x0B62A009, subs
criber_id: 0x001(4), participant_id: 0x000(1)
[1702630015.140350] info | ProxyClient.cpp | create_datareader | datareader created | client_key: 0x0B62A009, data
reader_id: 0x001(6), subscriber_id: 0x001(4)
[1702630015.158510] info | ProxyClient.cpp | create_topic | topic created | client_key: 0x0B62A009, topl
c_id: 0x005(2), participant_id: 0x000(1)
[1702630015.241039] info | ProxyClient.cpp | create_subscriber | subscriber created | client_key: 0x0B62A009, subs
criber_id: 0x002(4), participant_id: 0x000(1)
[1702630015.347393] info | ProxyClient.cpp | create_datareader | datareader created | client_key: 0x0B62A009, data
reader_id: 0x002(6), subscriber_id: 0x002(4)
```

### 3、 Start the keyboard control program

Enter the following command in the terminal to start the keyboard control program.

```
ros2 run yahboomcar_ctrl yahboom_keyboard
```

```
yahboom@yahboom-VM:~$ ros2 run yahboomcar_ctrl yahboom_keyboard
```

```
Control Your SLAM-Bot!
-----
Moving around:
  u   i   o
  j   k   l
  m   ,   .

q/z : increase/decrease max speeds by 10%
w/x : increase/decrease only linear speed by 10%
e/c : increase/decrease only angular speed by 10%
t/T : x and y speed switch
s/S : stop keyboard control
space key, k : force stop
anything else : stop smoothly

CTRL-C to quit

currently:      speed 0.2      turn 1.0
█
```

Keyboard key descriptions are as follows

Directional control.

<b>[i] or [I]</b>	<b>[linear, 0]</b>	<b>[u] or [U]</b>	<b>[linear, angular]</b>
<b>[,]</b>	<b>[-linear, 0]</b>	<b>[o] or [O]</b>	<b>[linear, -angular]</b>
<b>[j] or [J]</b>	<b>[0, angular]</b>	<b>[m] or [M]</b>	<b>[-linear, -angular]</b>
<b>[l] or [L]</b>	<b>[0, -angular]</b>	<b>[.]</b>	<b>[-linear, angular]</b>

According to the control table description, you can go forward by pressing the **[i]** key. Press **[,]** to go back, Press **[l]** to rotate to the right, press **[j]** to rotate to the left, and so on.

speed control table

<b>speed change</b>	<b>speed change</b>	<b>keyboard keys</b>	<b>speed change</b>
<b>[q]</b>	Linear speed and angular speed increased by 10%	<b>[z]</b>	Linear speed and angular speed are reduced by 10%
<b>[w]</b>	Only line speed increases by 10%	<b>[x]</b>	Only line speed is reduced by 10%
<b>[e]</b>	Only the angular velocity is increases by 10%	<b>[c]</b>	Only the angular velocity is reduced by 10%
<b>[t]</b>	Linear speed X-axis/Y-axis direction switching	<b>[s]</b>	Stop keyboard control

Note: Since the car has a four-wheel drive structure with ordinary tires and cannot move sideways, the [t] button has no meaning. Before each use of keyboard control, you need to click on the terminal that starts the program, otherwise the key event cannot be detected.