

# Quick start keyboard control

Note: The ROS\_DOMAIN\_ID of the Raspberry Pi and the microROS control board need to be consistent. You can check [MicroROS Control Board Parameter Configuration] to set the microROS control board ROS\_DOMAIN\_ID. Check the tutorial [Connect MicroROS Agent] to determine whether the IDs are consistent.

## 1. Program function description

After the car is connected to the agent, it will publish sensor data such as radar and imu. You can run commands in the supporting virtual machine/Raspberry Pi 5 to query this information. You can also publish control data of sensors such as speed and buzzer.

## 2. Query car information

### 2.1. Start and connect to the agent

After the Raspberry Pi is successfully powered on, open the terminal and enter the following command to open the agent.

```
sh ~/start_agent_rpi5.sh
```

```
pi@raspberrypi:~$ sh ~/start_agent_rpi5.sh
[1705911763.838436] info | TermiosAgentLinux.cpp | init | running... | fd: 3
[1705911763.839055] info | Root.cpp | set_verbose_level | logger setup | verbose_level: 4
```

Press the reset button on the microROS control board and wait for the car to connect to the agent. The connection is successful as shown in the figure below.

```
[1705911851.265754] info | ProxyClient.cpp | create_participant | participant created | client
key: 0x6BB64C97, participant_id: 0x000(1)
[1705911851.273538] info | ProxyClient.cpp | create_topic | topic created | client
key: 0x6BB64C97, topic_id: 0x000(2), participant_id: 0x000(1)
[1705911851.279639] info | ProxyClient.cpp | create_publisher | publisher created | client
key: 0x6BB64C97, publisher_id: 0x000(3), participant_id: 0x000(1)
[1705911851.283998] info | ProxyClient.cpp | create_datawriter | datawriter created | client
key: 0x6BB64C97, datawriter_id: 0x000(5), publisher_id: 0x000(3)
[1705911851.289506] info | ProxyClient.cpp | create_topic | topic created | client
key: 0x6BB64C97, topic_id: 0x001(2), participant_id: 0x000(1)
[1705911851.294457] info | ProxyClient.cpp | create_publisher | publisher created | client
key: 0x6BB64C97, publisher_id: 0x001(3), participant_id: 0x000(1)
[1705911851.299026] info | ProxyClient.cpp | create_datawriter | datawriter created | client
key: 0x6BB64C97, datawriter_id: 0x001(5), publisher_id: 0x001(3)
[1705911851.305475] info | ProxyClient.cpp | create_topic | topic created | client
key: 0x6BB64C97, topic_id: 0x002(2), participant_id: 0x000(1)
[1705911851.309535] info | ProxyClient.cpp | create_publisher | publisher created | client
key: 0x6BB64C97, publisher_id: 0x002(3), participant_id: 0x000(1)
[1705911851.313202] info | ProxyClient.cpp | create_datawriter | datawriter created | client
key: 0x6BB64C97, datawriter_id: 0x002(5), publisher_id: 0x002(3)
[1705911851.319437] info | ProxyClient.cpp | create_topic | topic created | client
key: 0x6BB64C97, topic_id: 0x003(2), participant_id: 0x000(1)
[1705911851.323740] info | ProxyClient.cpp | create_subscriber | subscriber created | client
key: 0x6BB64C97, subscriber_id: 0x000(4), participant_id: 0x000(1)
[1705911851.329366] info | ProxyClient.cpp | create_datareader | datareader created | client
```

## 2.2. Enter the car docker

Open another terminal and enter the following command to enter docker:

```
sh ros2_humble.sh
```

When the following interface appears, you have successfully entered docker. Now you can control the car through commands.

```
pi@raspberrypi:~$ ./ros2_humble.sh
access control disabled, clients can connect from any host
MY_DOMAIN_ID: 20
root@raspberrypi:/#
```

Enter the following command in the terminal to query the agent node:

```
ros2 node list
```

```
root@raspberrypi:~# ros2 node list
/YB_Car_Node
```

## 3. Start the keyboard control program

Enter the following command in the docker terminal to control the car using the keyboard.

```
ros2 run yahboomcar_ctrl yahboom_keyboard
```

```
root@raspberrypi:/# 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
```

Enter the following command in the docker terminal to control the car using the keyboard.

keyboard keys	Key Function
[i]	Go forward

keyboard keys	Key Function
【.】	move back
【l】	Right rotation
【j】	left rotation
【u】	Turn left
【o】	Turn right
【m】	Reverse left
【.】	Reverse to the right