

## 2. Robot keyboard control

### 1. Program function description

After the program is started, the robot movement can be controlled by the keyboard.

### 2. Start the keyboard control program

#### 2.1 Code path

Code path:

```
/root/yahboomcar_ws/src/yahboomcar_ctrl/yahboomcar_ctrl
```

#### 2.2 Run the command

Raspberry Pi enters docker, enter in the terminal,

```
./docker_ros2.sh
```

The following interface appears, which means that you have successfully entered Docker.

```
pi@yahboom:~ $ ./docker_ros2.sh
access control disabled, clients can connect from any host
root@yahboom:/#
```

```
ros2 launch yahboomcar_bringup bringup.launch.py
```

Start the chassis drive, as shown below.

```
root@yahboom:~/yahboomcar_ws# ros2 launch yahboomcar_bringup bringup.launch.py
[INFO] [launch]: All log files can be found below /root/.ros/log/2024-08-14-10-18-50-939449-yahboom-52625
[INFO] [launch]: Default logging verbosity is set to INFO
[INFO] [Mcnamu_driver-1]: process started with pid [52626]
[Mcnamu_driver-1] [INFO] [1723630731.256365516] [driver_node]: Successfully started the chassis drive...
```

Open a new terminal and enter the same docker. Change the following da8c4f47020a to the ID displayed in the actual terminal. Open a new terminal and enter the same docker. Change the following da8c4f47020a to the ID displayed in the actual terminal.

```
docker ps
```

```
docker exec -it da8c4f47020a /bin/bash
```

```
pi@yahboom:~ $ docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS
da8c4f47020a   yahboomtechnology/ros-humble:0.0.4 "/ros_entrypoint.sh ..." 8 hours ago   Up 45 minutes
s             festive_payne
pi@yahboom:~ $ docker exec -it da8c4f47020a /bin/bash
root@yahboom:/#
```

Then enter in docker,

```
ros2 run yahboomcar_ctrl yahboom_keyboard
```

```
root@yahboom:~/yahboomcar_ws# ros2 run yahboomcar_ctrl yahboom_keyboard

Control Your Raspbot-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
█
```

The keyboard keys are described 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>

That is to say, install [i] to move forward, press [,] to move backward, press [l] to rotate right, press [j] to rotate left, and so on.

Speed control

<b>Key</b>	<b>Speed changes</b>	<b>Key</b>	<b>Speed changes</b>
<b>[q]</b>	10% increase in both linear and angular velocities	<b>[z]</b>	10% reduction in both linear and angular velocities
<b>[w]</b>	Only 10% increase in line speed	<b>[x]</b>	Only 10% reduction in line speed
<b>[e]</b>	Only 10% increase in angular velocity	<b>[c]</b>	Only 10% reduction in angular velocity
<b>[t]</b>	Linear speed X/Y direction switching	<b>[s]</b>	Stop keyboard control