Handle ROS1 course

1、Operating environment Operating System: Ubuntu 18.04 ROS version: mellodic Devices: Jetson nano, Raspberry Pi, PC

2、Install the driver

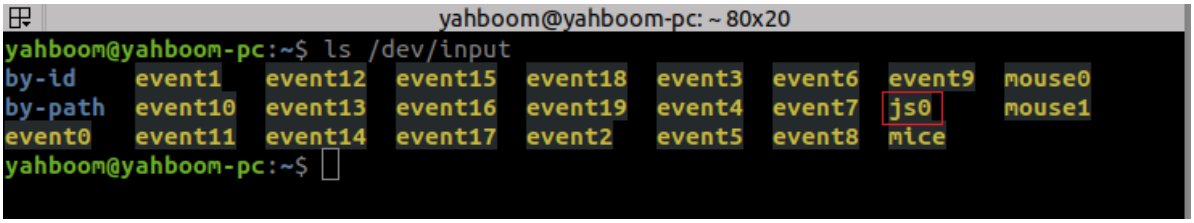
```
sudo apt install ros-melodic-joy ros-melodic-joystick-drivers
```

3、Usage steps

Connect the USB end of the wireless controller to the device

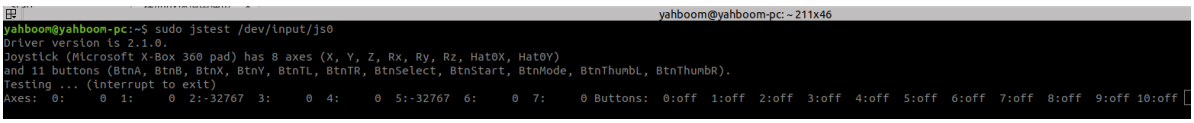
- View Device

```
ls /dev/input
```



- Test handle

```
sudo jstest /dev/input/js0
```

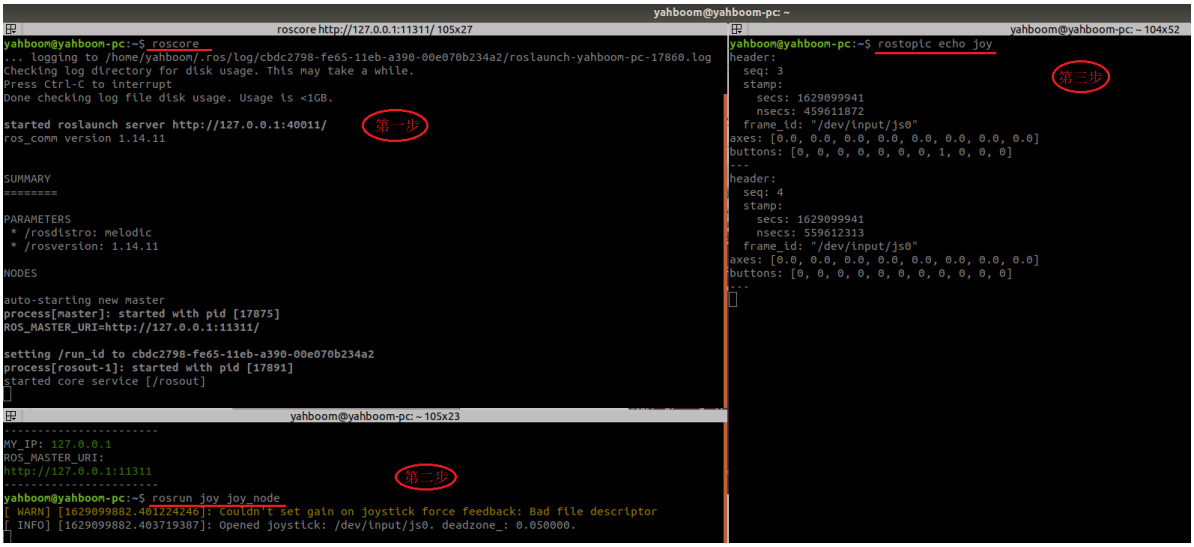


If jstest is not installed, run the following command:

```
sudo apt-get install joystick
```

- Operation handle node

```
roscore step1
rosrun joy joy_node Step3
rostopic echo joy Step3
```



4、Handle control Little Turtle

Copy the wireless handle control function package to the workspace, compile and update the environment

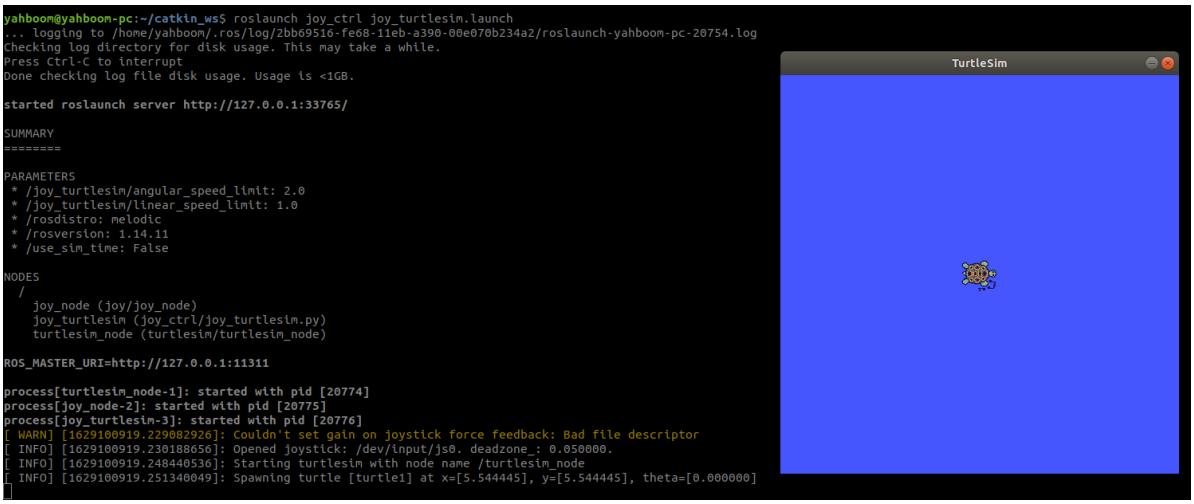
```
Catkin_ Make # Compile
Source dev/setup. bash # Update environment
Note: Any modifications to C++code require this step to take effect.
```

Start Python code command

```
roslaunch joy_ctrl joy_turtlesim.launch
```

Start C++code command

```
roslaunch joy_ctrl joy_turtle.launch
```

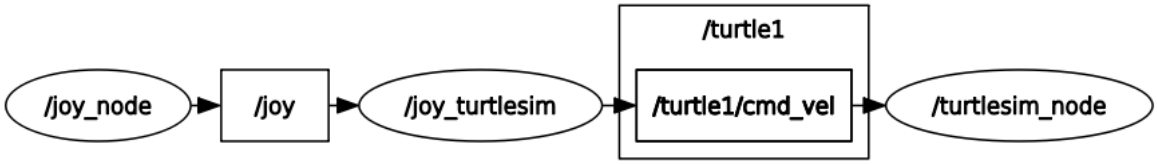


At this point, the handle can be used to control the operation of the little turtle.
Corresponding relationship between the handle and the operation of the little turtle

Handle	Little turtle
Left rocker up	forward
Left rocker down	back
Right rocker left	turn left
Right rocker right	turn right

- View node diagram

```
rqt_graph
```



5、Handle controlled turbobot

Due to the need to start gazebo, there is a high demand for device performance. It is recommended to use it on a PC, as Jetson nano and Raspberry pie may become particularly sluggish and may not run properly.

Install gazebo and turbobot simulations

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get install gazebo9 libgazebo9-dev ros-melodic-turtlebot3*
```

Copy the wireless handle control function package to the workspace, compile and update the environment

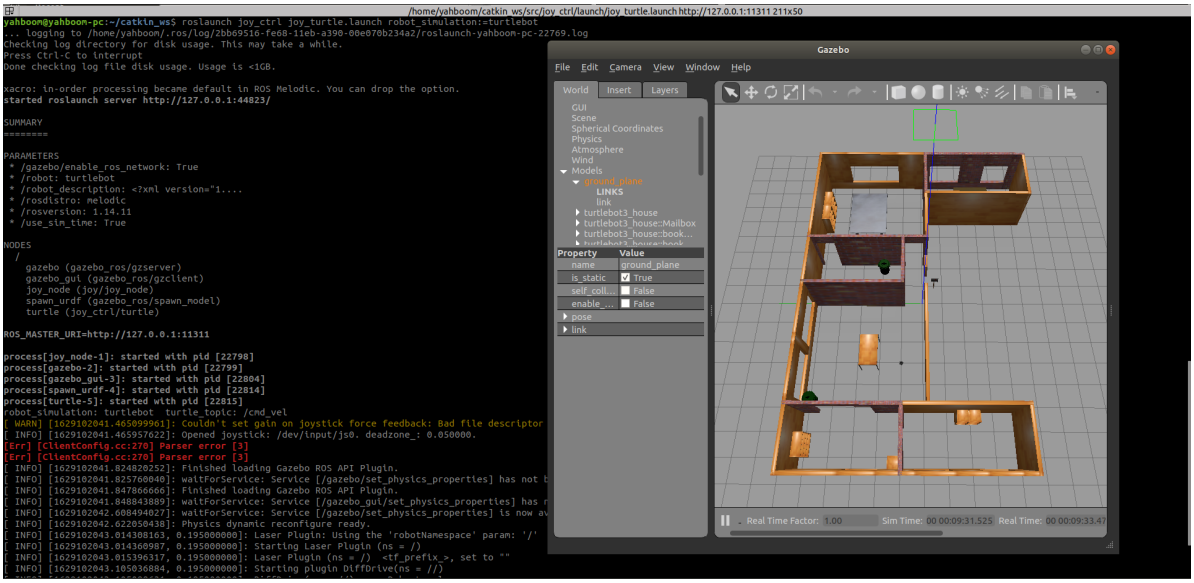
```
Catkin_ Make # Compile
Source dev/setup. bash # Update environment
Note: Any modifications to C++code require this step to take effect.
```

Start Python code command

```
roslaunch joy_ctrl joy_turtlesim.launch
```

Start C++code command

```
roslaunch joy_ctrl joy_turtle.launch robot_simulation:=turtlebot
```



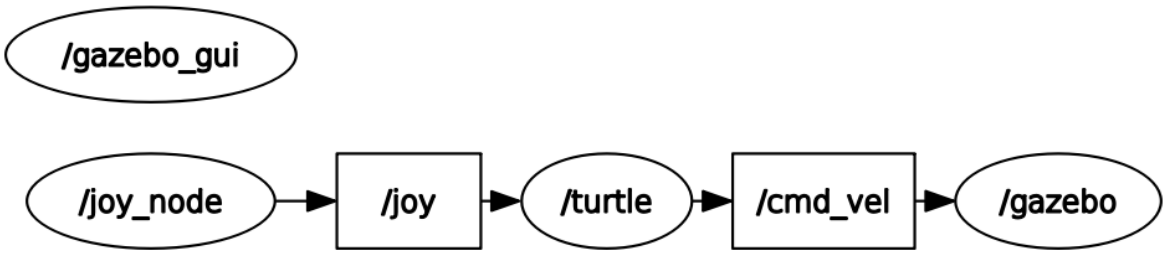
At this point, the handle can be used to control the operation of the turbobot.

Corresponding relationship between the handle and the operation of the little turtlebot

Handle	turtlebot
Left rocker up	forward
Left rocker down	back
Right rocker left	turn left
Right rocker right	turn right

- View node diagram

```
rqt_graph
```



- gazebo close command

```
killall gzserver gzclient
```

Appendix

jetson nano

```
joy_data.buttons: header:
  seq: 335
  stamp:
    secs: 1628324636
    nsecs: 962988952
  frame_id: "/dev/input/js0"
axes: [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
buttons: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

axes（8）

Code Parsing	Handle buttons
axes[0]	Left rocker (left positive and right negative)
axes[1]	Left rocker (up positive and down negative)
axes[2]	Right rocker (left positive and right negative)
axes[3]	Right rocker (up positive and down negative)
axes[4]	
axes[5]	
axes[6]	Left button (left positive and right negative)
axes[7]	Left button (up positive and down negative)

buttons（15）

Code Parsing	Handle buttons
buttons[0]	A
buttons[1]	B
buttons[2]	
buttons[3]	X
buttons[4]	Y
buttons[5]	
buttons[6]	L1
buttons[7]	R1
buttons[8]	L2
buttons[9]	R2
buttons[10]	SELECT
buttons[11]	START
buttons[12]	
buttons[13]	Press left rocker
buttons[14]	Press right rocker

Raspberry Pi

```
joy_data.buttons: header:
  seq: 264
  stamp:
    secs: 1628326479
    nsecs: 848359307
  frame_id: "/dev/input/js0"
axes: [-0.0, -0.0, 0.0, -0.0, -0.0, 0.0, 0.0, 0.0]
buttons: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

axes (8)

Code Parsing	Handle buttons
axes[0]	Left rocker (left positive and right negative)
axes[1]	Left rocker (up positive and down negative)
axes[2]	L2(Press:-1，release:1)
axes[3]	Right rocker (left positive and right negative)
axes[4]	Right rocker (up positive and down negative)
axes[5]	R2(Press:-1，release:1)
axes[6]	Left button (left positive and right negative)
axes[7]	Left button (up positive and down negative)

buttons（11）

Code Parsing	Handle buttons
buttons[0]	A
buttons[1]	B
buttons[2]	X
buttons[3]	Y
buttons[4]	L1
buttons[5]	R1
buttons[6]	SELECT
buttons[7]	START
buttons[8]	MODE
buttons[9]	Press left rocker
buttons[10]	Press right rocker