

## 4. Robot keyboard control

According to different models, just set the purchased model in [.bashrc], X1 (normal four-wheel drive) X3 (Mailun) X3plus (Mailun robotic arm) R2 (Ackerman differential) etc. , this section takes X3 as an example

```
#Raspberry Pi 5 master needs to enter docker first, please perform this step  
#If running the script into docker fails, please refer to ROS/07, Docker tutorial  
~/run_docker.sh
```

Open the [.bashrc] file

```
sudo vim .bashrc
```

Find the [ROBOT\_TYPE] parameter and modify the corresponding model

```
export ROBOT_TYPE=X3 # ROBOT_TYPE: X1 X3 X3plus R2 X7
```

### 4.1. Button description

#### 4.1.1. Direction control

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

#### 4.1.2. Speed control

Button	Speed change	Button	Speed change
[q]	Linear speed and angular speed are both increased by 10%	[z]	Linear speed and angular speed are both reduced by 10%
[w]	Only the linear speed increases by 10%	[x]	Only the linear speed decreases by 10%
[e]	Only the angular velocity increases by 10%	[c]	Only the angular velocity decreases by 10%
[t]	Line speed X-axis/Y-axis direction switching	[s]	Stop keyboard control

## 4.2. Run the program

### 4.2.1. Keyboard control code `yahboom_keyboard.py` path:

```
~/yahboomcar_ws/src/yahboom_ctrl/scripts
```

### 4.2.2. Run

```
roslaunch yahboomcar Bringup bringup.launch #robotchassisstart
```

<P15 needs to open another terminal to enter the same docker container

1. In the above steps, a docker container has been opened. You can open another terminal on the host (car) to view:

```
docker ps -a
```

```
jetson@ubuntu:~$ docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
5b698ea10535 yahboomtechnology/ros-foxy:3.3.9 "/bin/bash" 3 days ago Up 9 hours
jetson@ubuntu:~$
```

2. Now enter the docker container in the newly opened terminal:

```
docker exec -it 5b698ea10535 /bin/bash
```

```
jetson@ubuntu:~$ docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
5b698ea10535 yahboomtechnology/ros-foxy:3.3.9 "/bin/bash" 3 days ago Up 9 hours
jetson@ubuntu:~$ docker exec -it 5b698ea10535 /bin/bash
-----
my_robot_type: x3 | my_lidar: al | my_camera: astrapro
-----
root@ubuntu:/#
```

After successfully entering the container, you can open countless terminals to enter the container.

```
roslaunch yahboomcar_ctrl yahboom_keyboard.launch #Keyboard control node
```

### 4.2.3. Analysis of `yahboom_keyboard.py`

- 1). Posted topic: cmd\_vel

```
pub = rospy.Publisher('cmd_vel', Twist, queue_size=1)
```

Therefore, you only need to package the speed and publish it through `pub.publish(twist)`. The speed subscriber of the chassis can receive the speed data and then drive the car.

- 2). Mainly used modules

- The select module is mainly used for socket communication, seeing changes in file descriptions, and completing work in a non-blocking manner.
- The termios module provides an interface for IO controlled POSIX calls for tty
- The tty module is mainly used to change the mode of the file descriptor fd

- 3). mobile dictionary and speed dictionary

- The mobile dictionary mainly stores characters related to direction control

```

moveBindings = {
    'i': (1, 0),
    'o': (1, -1),
    'j': (0, 1),
    'l': (0, -1),
    'u': (1, 1),
    ',': (-1, 0),
    '.': (-1, 1),
    'm': (-1, -1),
    'I': (1, 0),
    'O': (1, -1),
    'J': (0, 1),
    'L': (0, -1),
    'U': (1, 1),
    'M': (-1, -1),
}

```

- The speed dictionary mainly stores characters related to speed control

```

speedBindings = {
    'Q': (1.1, 1.1),
    'Z': (.9, .9),
    'W': (1.1, 1),
    'X': (.9, 1),
    'E': (1, 1.1),
    'C': (1, .9),
    'q': (1.1, 1.1),
    'z': (.9, .9),
    'w': (1.1, 1),
    'x': (.9, 1),
    'e': (1, 1.1),
    'c': (1, .9),
}

```

#### 4. Get the current key information

```

def getKey():
    #tty.setraw(): Change the file descriptor fd mode to raw; fileno():
    #Return an integer file descriptor (fd)
    tty.setraw(sys.stdin.fileno())
    # select(): Directly call the IO interface of the operating system;
    #monitor all file handles with the fileno() method
    rlist, _, _ = select.select([sys.stdin], [], [], 0.1)
    # Read a byte input stream
    if rlist: key = sys.stdin.read(1)
    else: key = ''
    #tcsetattr sets the tty attribute of the file descriptor fd from the
    #attribute
    termios.tcsetattr(sys.stdin, termios.TCSADRAIN, settings)
    return key

```

#### 5. Determine whether t/T or s/S is pressed

```

if key=="t" or key == "T": xspeed_switch = not xspeed_switch
elif key == "s" or key == "S":
    print ("stop keyboard control: {}".format(not stop))
    stop = not stop

```

6. Determine whether the string is in the dictionary

```

#whether the key string is in the mobile dictionary
if key in moveBindings.keys():
    x = moveBindings[key][0]
    th = moveBindings[key][1]
    count = 0
#whether the key character is in the speed dictionary
elif key in speedBindings.keys():
    speed = speed * speedBindings[key][0]
    turn = turn * speedBindings[key][1]
    count = 0

```

6), speed limit

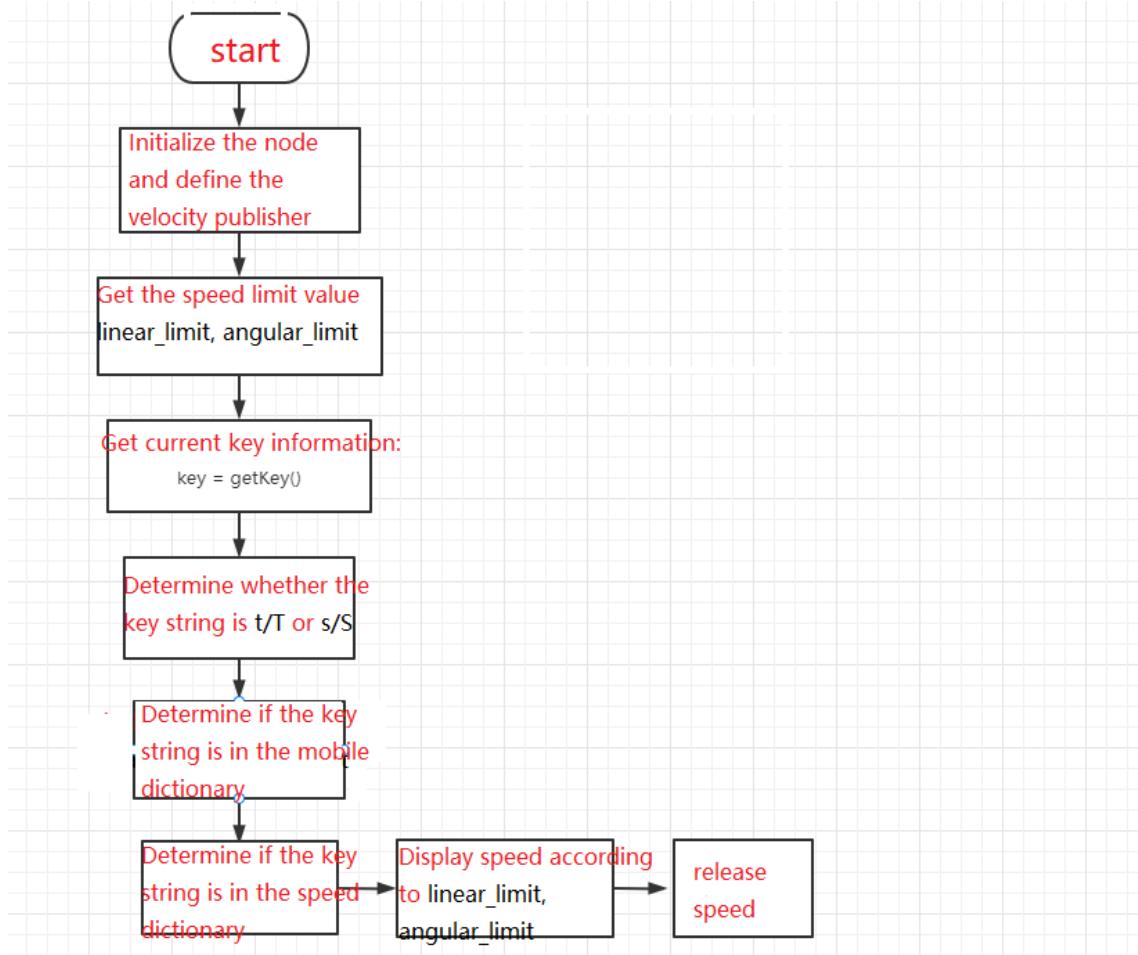
Both angular speed and linear speed have a limit value, and it is impossible to increase forever. When starting, the program will first obtain the speed limit value, and when increasing the speed, the increased value will be judged and processed.

```

linear_limit = rospy.get_param('~linear_speed_limit', 1.0)
angular_limit = rospy.get_param('~angular_speed_limit', 5.0)

```

7), program flow chart



## Precautions for using the handle

- When connecting the USB handle receiver, it is recommended to connect it to the outermost USB-HUB expansion board instead of directly connecting to the motherboard or the middle USB-HUB expansion board (X3plus). If it is directly connected to the motherboard or the middle USB-HUB expansion board (X3plus), due to the aluminum alloy blocks on the top and bottom, it will seriously interfere with the handle signal reception.
- After plugging and unplugging the handle receiver, you need to restart the handle program, otherwise you will not be able to control the car.
- After starting the handle control program, if the handle cannot control the car, it may be caused by the wrong handle control mode. You can press and hold the handle mode button for about 15 seconds to switch modes. After the green indicator light is always on, press the start button again, such as If a buzzer sounds, the switch is successful. If there is no response, you can press and hold the mode button on the handle again for 15 seconds.

**Jetson series support mode:** PC/PCS mode, the POWER MODE indicator light is red by default in PC mode. You can connect the handle receiver to the USB port of the computer to connect the wireless handle. Enter the URL in the browser: <https://gamepad-tester.com/>. Pressing the button URL will display the change of the button value, as shown in the figure below:



**Raspberry Pi series supported modes:** X-BOX mode. In X-BOX mode, the POWER MODE indicator light is green by default. You can connect the handle receiver to the USB port of the computer to connect the wireless handle. Enter the URL in the browser: <https://gamepad-tester.com/>. Pressing the button URL will display the change of the button value, as shown in the figure below:

The screenshot shows the Gamepad Tester interface. At the top, there's a navigation bar with links for Home, For Developers, Controllers, Codes, Browser Support, and Dataset. Below the navigation, it displays three player sections: PLAYER 1 (Xbox Controller connected), PLAYER 2 (None detected), and PLAYER 3 (None detected). A large central image of an Xbox 360 controller is shown. A red box highlights the 'Xbox Controller' section, which includes the text 'Xbox 360 Controller (XInput STANDARD GAMEPAD)'. Below this, a table provides detailed information about the connection: INDEX (0), CONNECTED (Yes), MAPPING (standard), and TIMESTAMP (649701.00000). Further details like Pose, HapticActuators, Hand, DisplayId, and Vibratio are listed. A green LED indicator on the left side of the controller is also highlighted with a red box.



无线手柄

- After re-plugging or unplugging the handle receiver or restarting the motherboard, the handle will be reset to factory mode. If it cannot be controlled, the mode will need to be switched again every time it is plugged in, unplugged or restarted.
- If matching is not successful, the POWER MODE indicator light will flash in red and green for a few seconds and then go into sleep mode.