

2. Robot keyboard control

2.1. Keyboard description

2.1.1. 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】

2.1.2. Speed control

Key	Speed change	Key	Speed change
【q】	Increase both linear and angular velocities by 10%	【z】	Decrease both linear and angular velocities by 10%
【w】	Increase only linear velocity by 10%	【x】	Decrease only linear velocity by 10%
【e】	Increase only angular velocity by 10%	【c】	Decrease only angular velocity by 10%
【t】	Switch between linear velocity X-axis and Y-axis	【s】	Stop keyboard control

2.2. Run the program

2.2.1. Keyboard control code yahboom_keyboard.py path:

```
/home/yahboom/YBAMR-COBOT-EDU-00001/src/yahboom_navrobo_ctrl/scripts/
```

2.2.2. Run

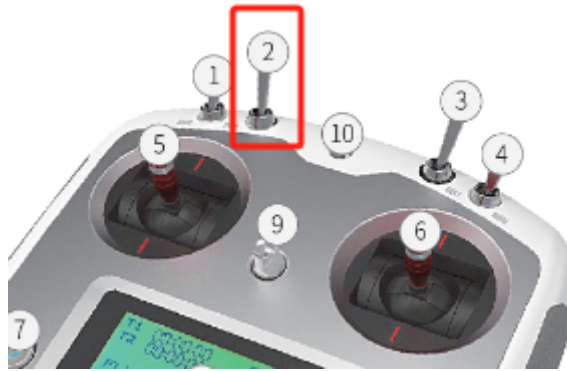
Stop the self-start chassis service

```
sudo supervisorctl stop ChassisServer
```

```
roslaunch scout_bringup scout_mini_robot_bringup.launch #Robot chassis startup
```

```
roslaunch yahboom_navrobo_ctrl yahboom_keyboard.launch #keyboard control node
```

Note: To start control, you need to first turn the SWB button to the upper gear position (control command mode) to release the remote control



2.2.3, Analysis of yahboom_keyboard.py

1), Published topic: cmd_vel

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

So, just package the speed and publish it through `pub.publish(twist)`, and the chassis speed subscriber will be able to receive the speed data and then drive the car.

2) Mainly used modules

- The select module is mainly used for socket communication, observes changes in file descriptions, and completes non-blocking work
- The termios module provides an IO-controlled POSIX call interface for tty
- The tty module is mainly used to change the mode of the file descriptor fd

3) Move dictionary and speed dictionary

- The move 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 the 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 operating system's IO interface; Monitor all file
    handles with the fileno() method
    rlist, _, _ = select.select([sys.stdin], [], [], 0.1)
    # Read a byte of 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

```
#Is the key string in the move dictionary
if key in moveBindings.keys():
    x = moveBindings[key][0]
    th = moveBindings[key][1]
    count = 0
#Is the key character 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 velocity and linear velocity have a limit value, and it is impossible to increase continuously. At startup, the program will first obtain the speed limit value, and when increasing the speed, the increased value will be judged.

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

7) Program flow chart

