3、Robot keyboard control

1、Program Running Example

After successfully installing the Rosmaster library and compiling the workspace, we can enter the following command to start the car keyboard control. Take our company's Rosmaster-X3 McNamm wheel as an example, input at the terminal,

```
#Drive trolley chassis
ros2 run yahboomcar_bringup Mcnamu_driver_X3
#Enable keyboard control
ros2 run yahboomcar_ctrl yahboom_keyboard
```

2、Key Description

Direction control instruction table

【i】或【I】	【linear，0】	【u】或【U】	【linear，angular】
【,】	【-linear，0】	【o】或【O】	【linear，-angular】
【j】或【J】	【0，angular】	【m】或【M】	【-linear，-angular】
【l】或【L】	【0，-angular】	【.】	【-linear，angular】

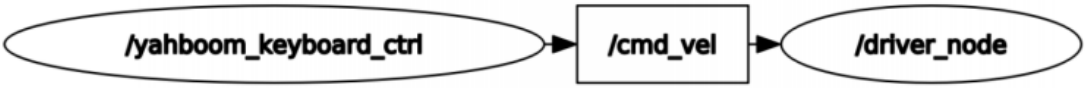
Speed control command table

button	speed change	button	speed change
[q]	Linear and angular velocity are both increased by 10%	[z]	Linear and angular velocities are both reduced by 10%
[w]	10% increase in line speed only	[x]	10% reduction in line speed only
[e]	10% increase in angular velocity only	[c]	Only the angular velocity is reduced by 10%
[t]	Line speed X-axis/Y-axis direction switching	[s]	stop keyboard control

3、Node Communication Diagram

Enter the following command to view the node communication diagram，

```
ros2 run rqt_graph rqt_graph
```



4、Source code parsing

Keyboard control can only control the motion of the car, therefore, there is only one/cmd_ Vel speed publisher,

```
self.pub = self.create_publisher(Twist,'cmd_vel',1)
```

The program also defines two dictionaries to detect changes in keyboard letters when pressed,

```
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),
}

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),
}
```

Entering the while loop, the program will read the values pressed by the keyboard and make layer by layer judgments,

```
key = yahboom_keyboard.getKey()
if key=="t" or key == "T": xspeed_switch = not xspeed_switch
elif key == "s" or key == "S":
...
if key in moveBindings.keys():
...
elif key in speedBindings.keys():
..
```

Finally, based on multi-layer judgment, assign values to twist. linear. x, twist. linear. y, twist. regular. z and publish them.

```
if xspeed_switch: twist.linear.x = speed * x
else: twist.linear.y = speed * x
twist.angular.z = turn * th
if not stop: yahboom_keyboard.pub.publish(twist)
if stop:yahboom_keyboard.pub.publish(Twist())
```

Please refer to the following path for detailed code:
~/driver_ws/src/yahboomcar_ctrl/yahboomcar_ctrl/yahboom_keyboard.py。