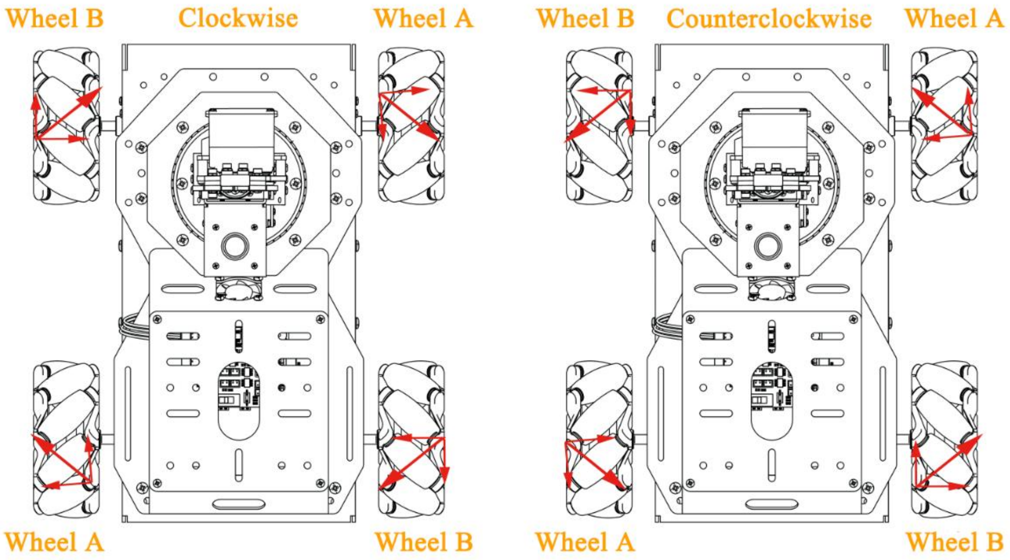
Lesson 4 Turning

1. Working Principle

According to the characteristics of mecanum wheel , running (all at the same speed) both wheels on one side in one direction while the other side in the opposite direction, will result in a stationary rotation of the chassis car.

Therefore, when the left mecanum wheels rotate clockwise and the right meacnum wheels rotate counterclockwise, the chassis car will be counterclockwise stationary rotation. When the left mecanum wheels rotate clockwise and the right wheels rotates counterclockwise, the chassis car will be clockwise stationary rotation.

The force analysis of two situation is as follow:



2. Operations Steps

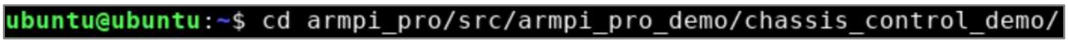
ℹ It should be case sensitive when entering command, and the “Tab” key can be used to complete the keywords.

1) Please refer to the content in “7. ArmPi Pro Basic Lesson/1.Mecanum

Wheel Chassis Lesson/ Lesson 2 Set Environment Development” to connect system desktop via No Machine.

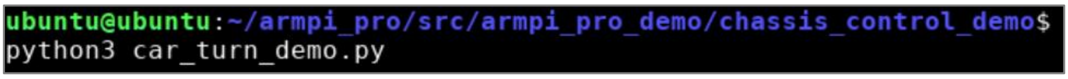
2) Click Applications in the lower left corner and select Terminal Emulator to enter the terminal.

3) Enter command “**cd armpi_pro/src/armpi_pro_demo/chassis_control_demo/**” and press “Enter” to enter the directory of game programmings.



```
ubuntu@ubuntu:~$ cd armpi_pro/src/armpi_pro_demo/chassis_control_demo/
```

4) Enter command “python3 car_forward_demo.py” and press “Enter” to start game.



```
ubuntu@ubuntu:~/armpi_pro/src/armpi_pro_demo/chassis_control_demo$
python3 car_turn_demo.py
```

5) If want to exit the game, press “Ctrl+C” in terminal. If fail to exit, please keep trying until the program is closed.

3. Project Outcome

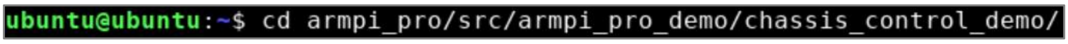
After starting the game, ArmPi Pro will be clockwise stationary rotation, and then counterclockwise stationary rotation.

4. Function Extension

The default rotation speed of moving forwards is 0.3. This section will modify the clockwise rotation speed of car to 0.5 and the specific operation steps are as follow:

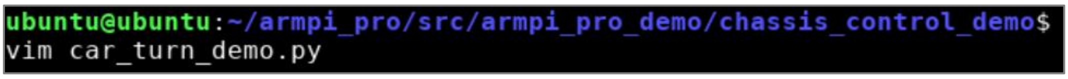
1) Click and select to enter the terminal.

2) Enter command "**cd armpi_pro/src/armpi_pro_demo/chassis_control_demo/**" and press "Enter" to come to the directory of game programmings.



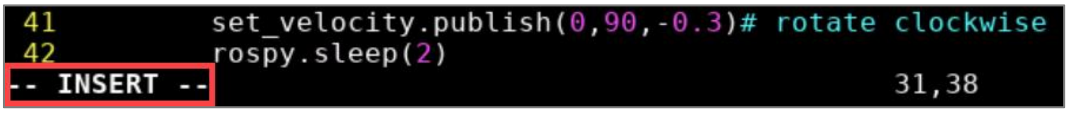
```
ubuntu@ubuntu:~$ cd armpi_pro/src/armpi_pro_demo/chassis_control_demo/
```

3) Enter command "**vim car_turn_demo.py**" and press "Enter" to open the program file.



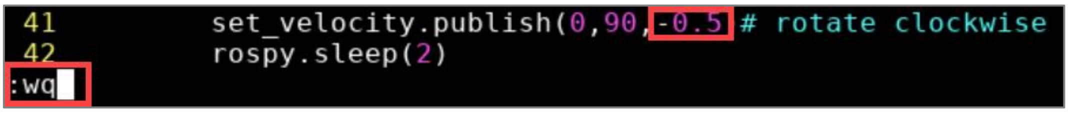
```
ubuntu@ubuntu:~/armpi_pro/src/armpi_pro_demo/chassis_control_demo$
vim car_turn_demo.py
```

4) Find the code to be modified and press "i". When the prompt "INSERT" appears in the lower left corner, it means the terminal has been switched to the editing mode.



```
        set_velocity.publish(0,90,-0.3)# rotate clockwise
        rospy.sleep(2)
-- INSERT --                                      31,38
```

5) In "**set_velocity.publish()**" function, the first parameter represents the rotation speed of the car and we modify it to -0.5. After modifying, press "Esc" and enter ":wq", and then press "Enter" to save and exit.

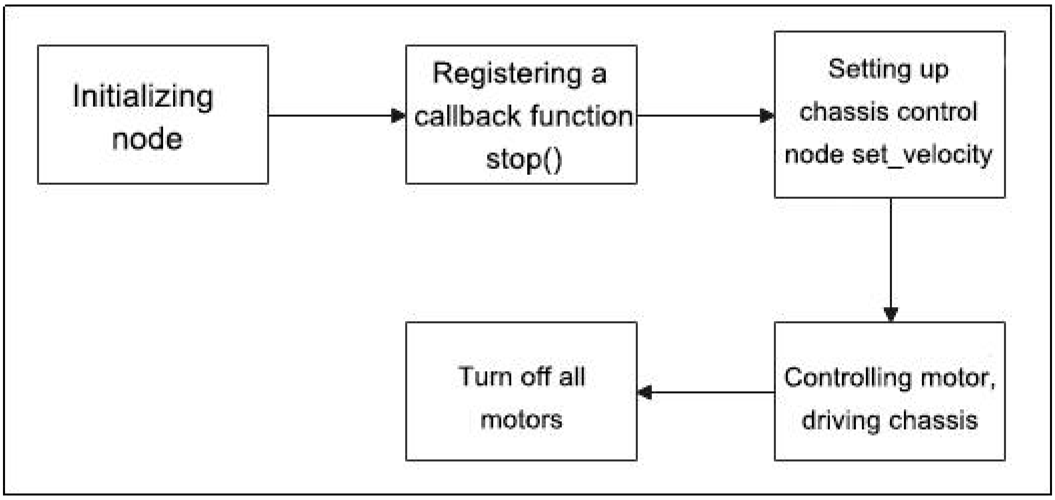


```
        set_velocity.publish(0,90,-0.5)# rotate clockwise
        rospy.sleep(2)
:wq
```

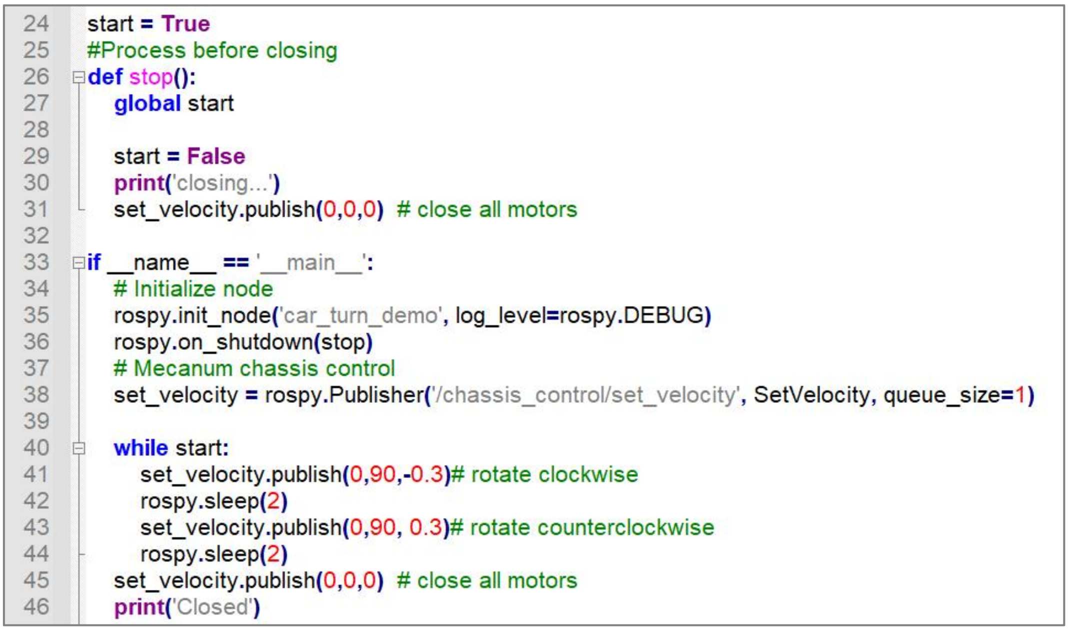
Note: The adjustable range of speed is from -2 to 2. When the value is negative, the motor will rotate clockwise. Please modify according to the actual situation.

6) After modifying, you can refer to the content of "2.Operation Steps" to check the effect.

5. Program Analysis


Initializing node
Registering a callback function stop()
Setting up chassis control node set_velocity
Controlling motor, driving chassis
Turn off all motors

The source code of the program is located in: **home/ubuntu/armpi_pro/src/armpi_pro_demo/chassis_control_demo/car_turn_demo.py**



```
start = True
#Process before closing
def stop():
    global start

    start = False
    print('closing...')
    set_velocity.publish(0,0,0)  # close all motors

if __name__ == '__main__':
    # Initialize node
    rospy.init_node('car_turn_demo', log_level=rospy.DEBUG)
    rospy.on_shutdown(stop)
    # Mecanum chassis control
    set_velocity = rospy.Publisher('/chassis_control/set_velocity', SetVelocity, queue_size=1)

    while start:
        set_velocity.publish(0,90,-0.3)# rotate clockwise
        rospy.sleep(2)
        set_velocity.publish(0,90, 0.3)# rotate counterclockwise
        rospy.sleep(2)
    set_velocity.publish(0,0,0)  # close all motors
    print('Closed')
```

Control motor through **set_velocity.publish()** function. There are three parameters in function. Take the code "**set_velocity.publish(0,90,-0.3)**" as an example:

1) The first parameter "0" represents the motor speed, its unit is mm/s and it ranges from -100 to 100. When the value is negative, the motor rotates counterclockwise.

2) The second parameter "90" represents the movement direction of car, its

unit is degree and it ranges from 0 to 360. The value of 90° refer to move forward. 270° refers to move backward. 0° refers to move to the right. 180° refers to move the left. Other movement directions are obtained according to the same reference method.

3) The third parameter “-0.3” represents the rotation speed of the car, its unit is 5°/s and it ranges from -2 to 2. When the parameter value is positive, the car will rotate clockwise. When the parameter value is negative, the car will rotate counterclockwise.