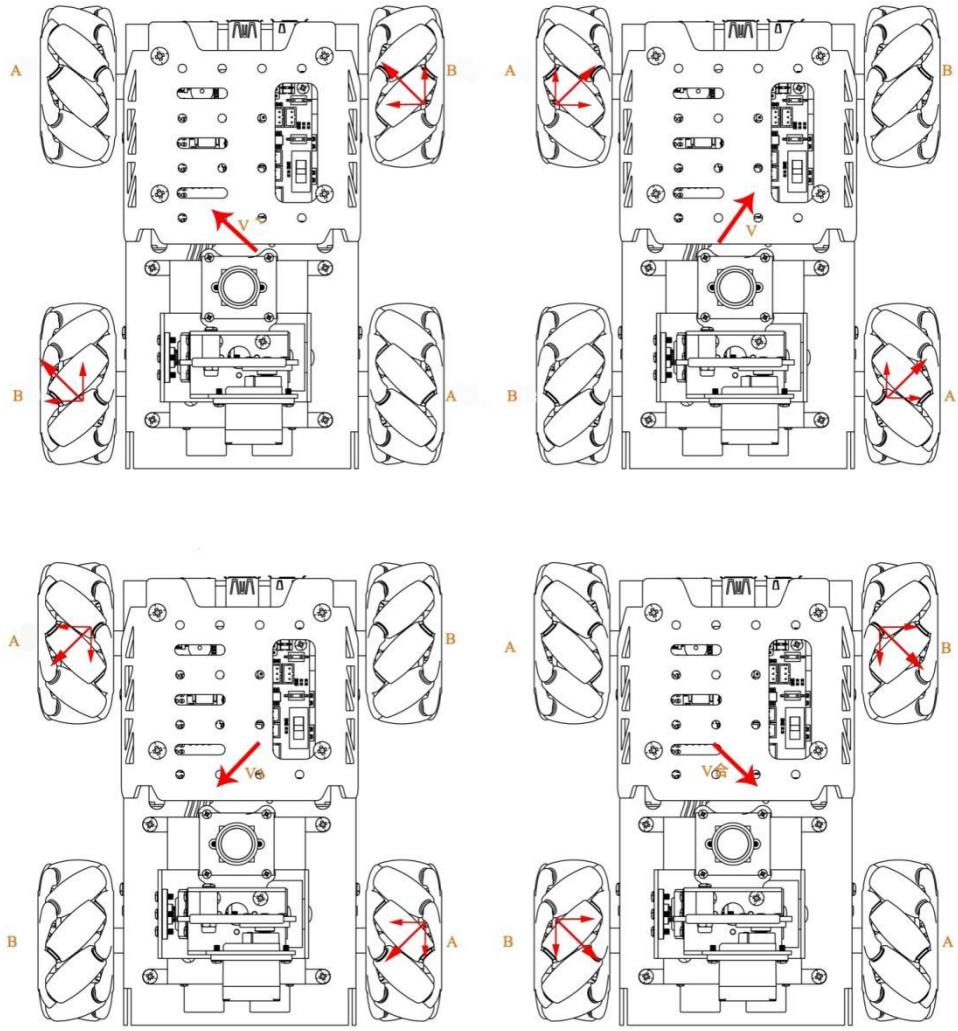
Lesson 5 Oblique Movement

1. Working Principle

According to the characteristics of mecanum wheel , when wheel A does not move and wheel B rotates clockwise, the car moves to the left front. When wheel B rotates counterclockwise, the car moves to the right rear. When the wheel B does not move and wheel A rotates clockwise, the car will move to the front right. When the wheel A rotates counterclockwise, the car move to the left rear. The force analysis for oblique movement:



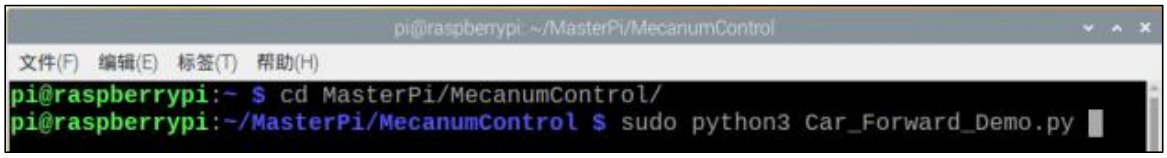
2. Operation Steps

The entered command should be case sensitive and “Tab” key can be used to fill in keyword.

1) Click  to enter the LX terminal.

2) Enter “**cd MasterPi/MecanumControl/**” command to come to the directory of game programmings.

3) Enter “**sudo python3 Car_Forward_Demo.py**” command and press “Enter”.



4) If want to exit the game, you can press “Ctrl+C”. If fail to exit, please try multiple times.

3. Project Outcome

After starting the game, MasterPi will move to the right front, to the right rear, to the left rear, and to the front left in sequence.

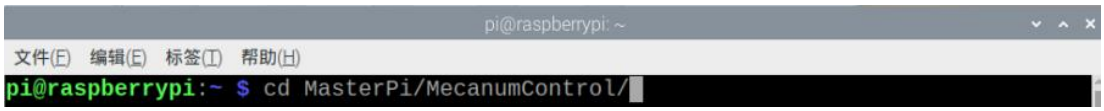
4. Function Extension

The default angle of oblique movement is 45° and the value can be modified to adjust the angle. This section will modify the value to 60 and the specific

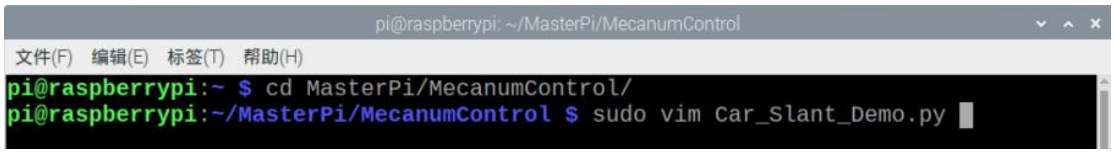
operation steps are as follow:

1) Click  to enter the LX terminal.

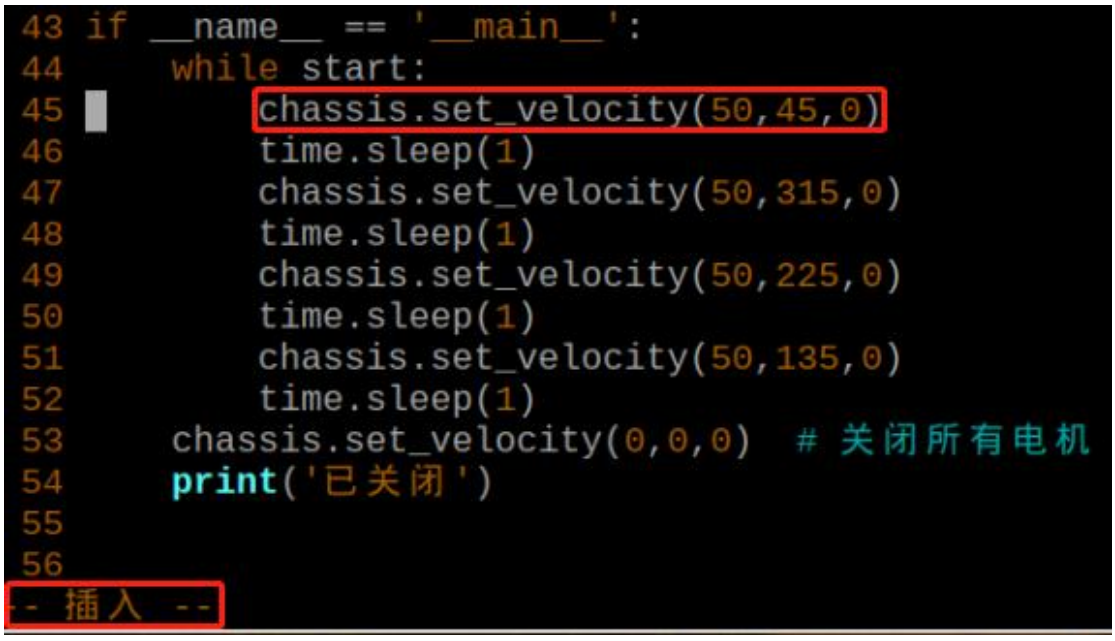
2) Enter "**cd MasterPi/MecanumControl/**" command and press "Enter" to come to the directory of game programmings.



3) Enter "sudo vim Car_Slant_Demo.py" command and press "Enter" to open the program file.



4) Find the code to be modified and press "i" to enter the editing mode.



5) In "set_velocity" function, the second parameter represents the directional angle of moving forwards and we modify it to 60. After modifying, press "Esc" and enter ":wq", and then press "Enter" to save and exit.

```
43 if __name__ == '__main__':
44     while start:
45         chassis.set_velocity(50,60,0)
46         time.sleep(1)
47         chassis.set_velocity(50,315,0)
48         time.sleep(1)
49         chassis.set_velocity(50,225,0)
50         time.sleep(1)
51         chassis.set_velocity(50,135,0)
52         time.sleep(1)
53     chassis.set_velocity(0,0,0)  # 关闭所有电机
54     print('已关闭')
55
56
:wq
```

Note: The adjustable range of the directional angle is from 0 to 360. 0 refers to move to the right; 90 refers to move forwards; 180 refers to move to the left; 270 refers to move backwards.

5. Program Analysis

The source code of program is located in: **/home/pi/MasterPi/MecanumControl/Car_Slant_Demo.py**

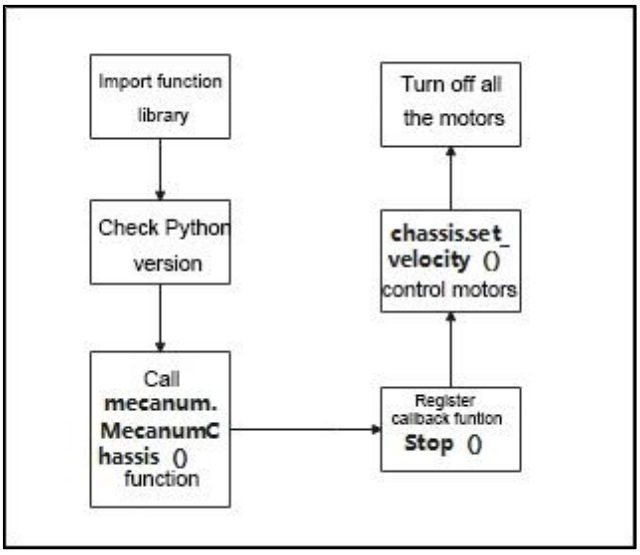
5.1 Import Parameter Module

Import module	Function
import sys	Importing the Python sys module is used for getting access to the relevant function and variables

import time	Importing the Python time module is used for time-related functionalities, such as delay operations.
import signal	The receiving and processing of the signal
HiwonderSDK.mecanum	The control module related to mecanum wheel chassis

5.2 Program Logic and Corresponding Code Analysis

The diagram of program logic refers to the below figure.



From the above figure, the program's logical flow is mainly divided into importing the function library, calling relevant functions, and controlling the motors. The following documentation will be written based on the program's logical flowchart mentioned above.

◆ **Import Function Library**

During initialization, the function library needs to be first imported for subsequent calling. For the detailed content of importing, please refer to "**3.1**

Import Parameter Module".

```
3  import sys
4  sys.path.append('/home/pi/MasterPi/')
5  import time
6  import signal
7  import HiwonderSDK.mecanum as mecanum
```

◆ **Check Python Version**

```
9  if sys.version_info.major == 2:
10     print('Please run this program with python3!')
11     sys.exit(0)
```

sys.version_info.major is used to check the major version of Python. If the version is equal to 2, the program will print a message and exit the program.

◆ **Call the mecanum.MecanumChassis（）Function**

```
26 chassis = mecanum.MecanumChassis()
```

The call to **mecanum.MecanumChassis()** constructor creates an object of the **MecanumChassis** class and assigns it to the chassis variable.

◆ **Motor Control**

```
40 if __name__ == '__main__':
41     while start:
42         chassis.set_velocity(50,45,0)
43         time.sleep(1)
44         chassis.set_velocity(50,315,0)
45         time.sleep(1)
46         chassis.set_velocity(50,225,0)
47         time.sleep(1)
48         chassis.set_velocity(50,135,0)
49         time.sleep(1)
50     chassis.set_velocity(0,0,0)  # 关闭所有电机
51     print('已关闭')
52
```

Control motor through **set_velocity** function. There are three parameters in function. Here takes an example of the code "**chassis.set_velocity(50,45,0)**":

1) The first parameter "**50**" 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 "**45**" 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**" 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.