

03. Car rotation

03. Car rotation

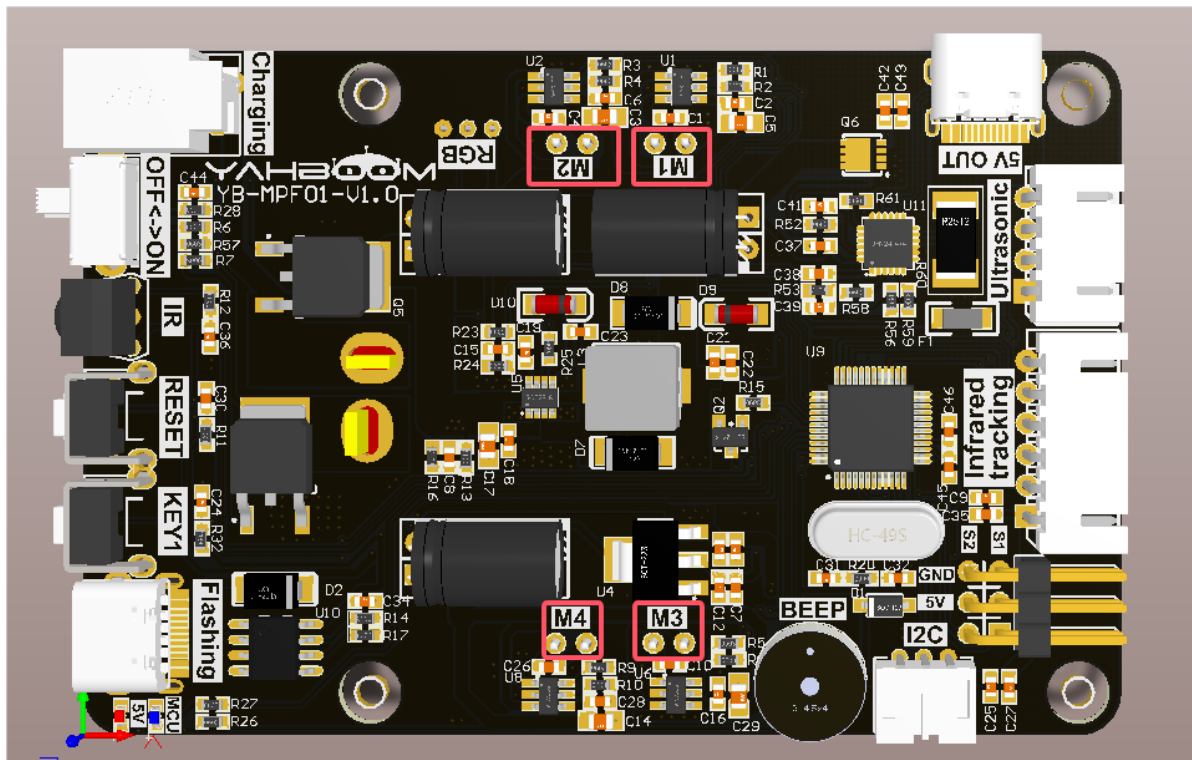
1. Learning objectives
2. Experimental preparation
3. Core code analysis
4. Experimental phenomenon

1. Learning objectives

Control the car to rotate left and right.

2. Experimental preparation

As shown in the figure below, the motor needs to be connected to the expansion board.



3. Core code analysis

Ordinary_wheel_motion library function needed to control the car's movement:

```
rotate_left(speed)
```

Parameter explanation: Control the car to rotate left

speed: [0,255], the larger the value, the faster the left rotation

Return value: None.

```
rotate_right(speed)
```

Parameter explanation: Control the car to rotate right

speed: [0,255], the larger the value, the faster the right rotation

Return value: None.

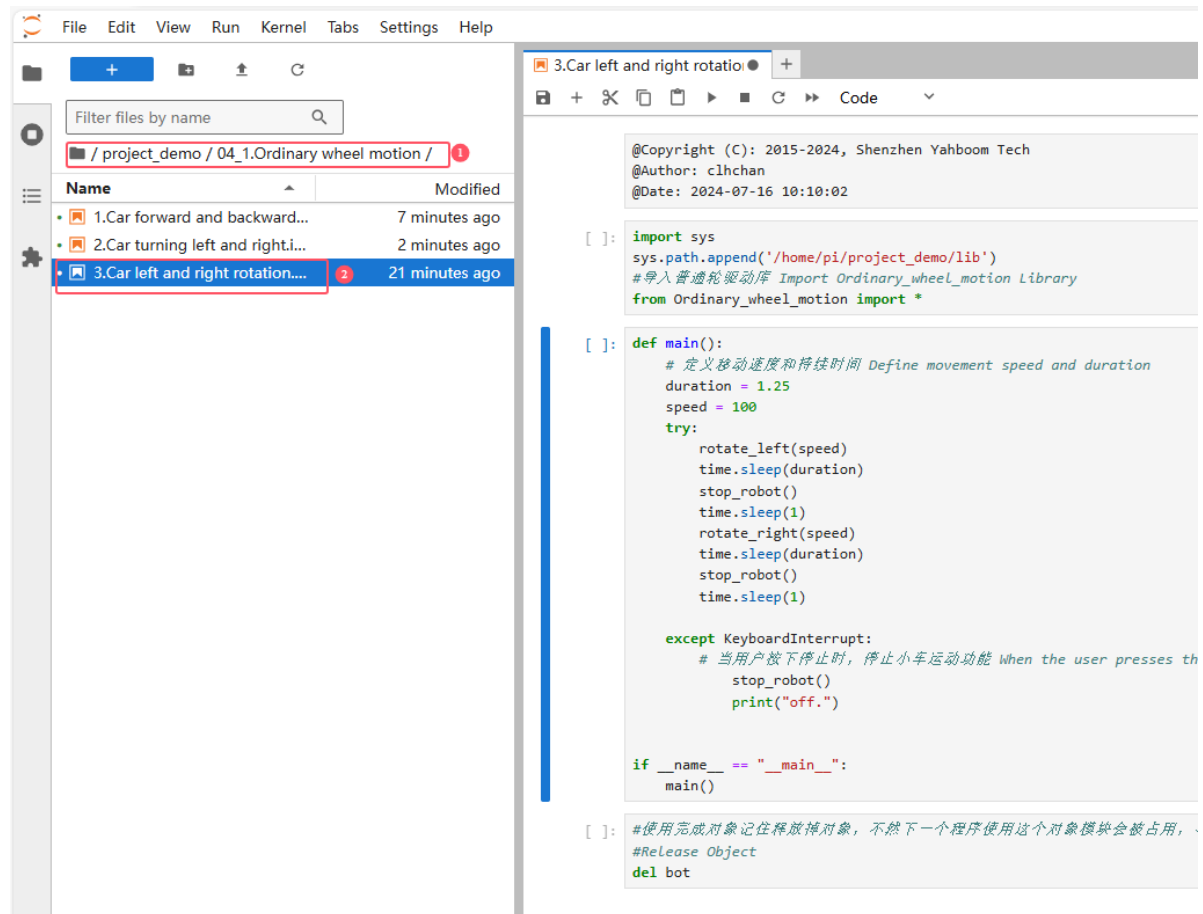
Source code path: project_demo/04_1.Ordinary wheel motion

Library path: project_demo/lib

4. Experimental phenomenon

Turn on the robot, open the computer browser to enter the Jupyter lab editor

Enter the source code path and double-click the code to be run



The screenshot shows the Jupyter Lab interface. On the left, the file explorer displays the directory structure: `/ project_demo / 04_1.Ordinary wheel motion /`. Below this, a list of files is shown with their modification times: `1.Car forward and backward...` (7 minutes ago), `2.Car turning left and right...` (2 minutes ago), and `3.Car left and right rotation....` (21 minutes ago). The file `3.Car left and right rotation....` is selected. On the right, the code editor shows the source code for `3.Car left and right rotation`. The code includes a copyright notice, imports, and a `main` function that defines movement speed and duration, and calls `rotate_left` and `rotate_right` functions. The code is as follows:

```
@Copyright (C): 2015-2024, Shenzhen Yahboom Tech
@Author: clhchan
@Date: 2024-07-16 10:10:02

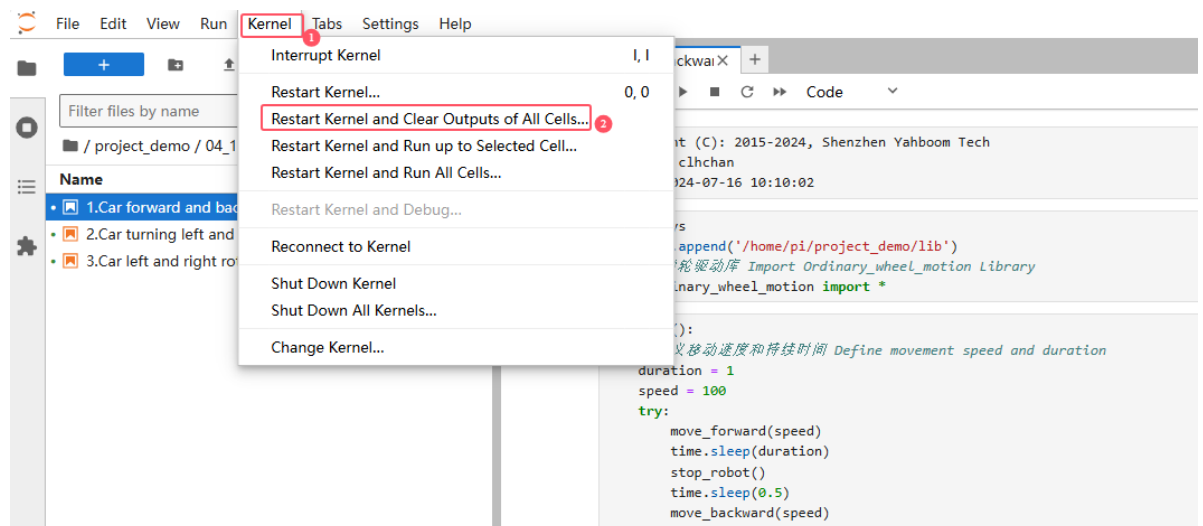
[ ]: import sys
sys.path.append('/home/pi/project_demo/lib')
#导入普通轮驱动库 Import Ordinary_wheel_motion Library
from Ordinary_wheel_motion import *

[ ]: def main():
    # 定义移动速度和持续时间 Define movement speed and duration
    duration = 1.25
    speed = 100
    try:
        rotate_left(speed)
        time.sleep(duration)
        stop_robot()
        time.sleep(1)
        rotate_right(speed)
        time.sleep(duration)
        stop_robot()
        time.sleep(1)
    except KeyboardInterrupt:
        # 当用户按下停止时, 停止小车运动功能 When the user presses th
        stop_robot()
        print("off.")

if __name__ == "__main__":
    main()

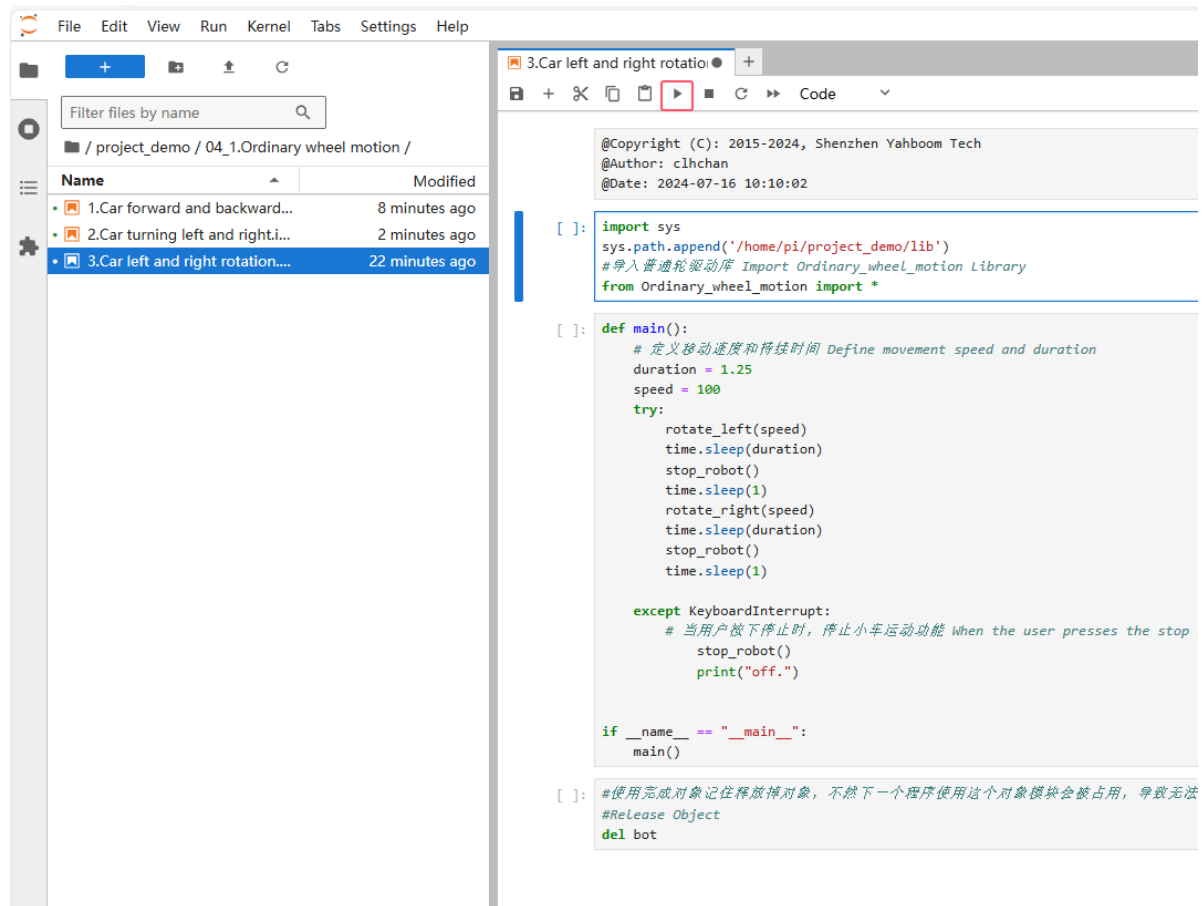
[ ]: #使用完成对象记得释放掉对象, 不然下一个程序使用这个对象模块会被占用,
#Release Object
del bot
```

Restart the kernel and clear all outputs



The screenshot shows the Jupyter Lab interface with the `Kernel` menu open. The menu options are: `Interrupt Kernel`, `Restart Kernel...`, `Restart Kernel and Clear Outputs of All Cells...`, `Restart Kernel and Run up to Selected Cell...`, `Restart Kernel and Run All Cells...`, `Restart Kernel and Debug...`, `Reconnect to Kernel`, `Shut Down Kernel`, `Shut Down All Kernels...`, and `Change Kernel...`. The option `Restart Kernel and Clear Outputs of All Cells...` is highlighted. The background shows the same file explorer and code editor as the previous screenshot.

Click the first code block, then click the run button to start running one by one



After the program runs, as the code blocks run, we can see that the car rotates left for 1.25 seconds, right for 1.25 seconds, and finally stops.