

05. Motor control

05. Motor control

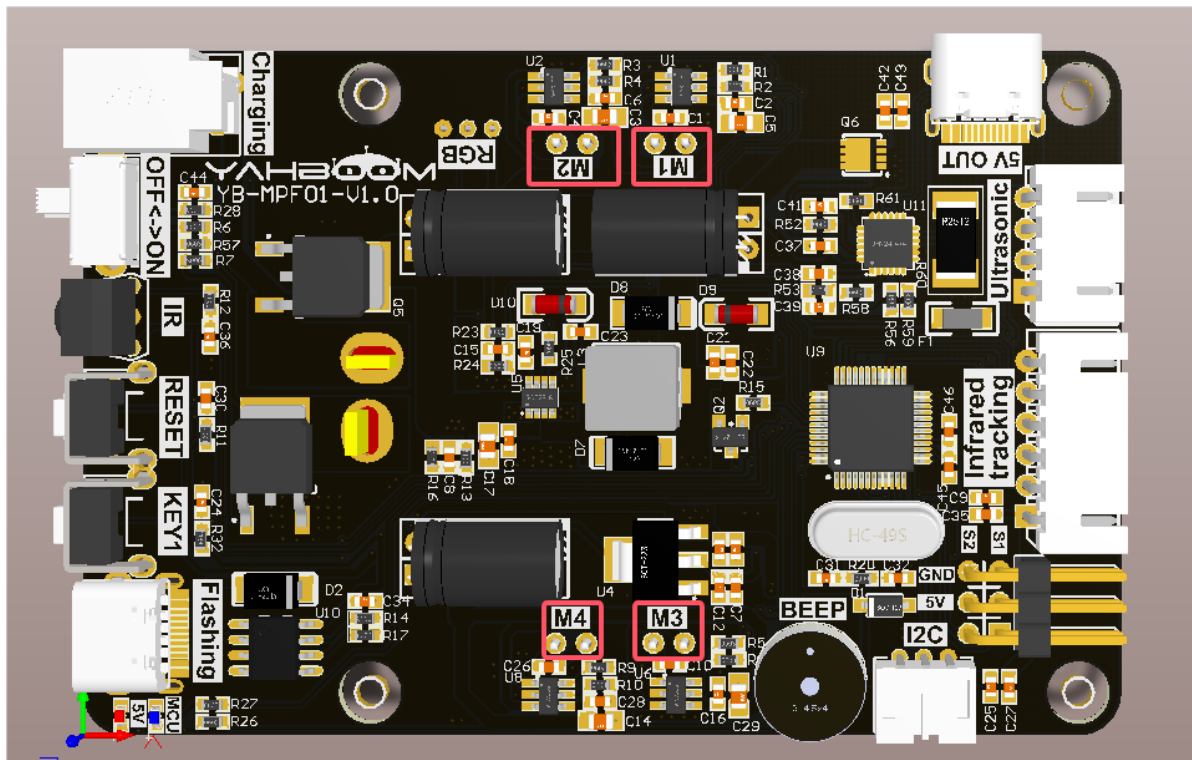
1. Learning objectives
2. Experimental preparation
3. Core code analysis
4. Experimental phenomenon

1. Learning objectives

Control the TT motor connected to the expansion board.

2. Experimental preparation

As shown in the figure below, connect the motor to the motor interface.



3. Core code analysis

Raspbot_Lib library functions needed to control the motor:

```
Ctrl_Muto(motor_id, motor_speed)
```

Parameter explanation: Control the forward and reverse rotation and speed of the motor

motor_id=[1,4]: 1: M1 motor, 2: M2 motor, 3: M3 motor, 4: M4 motor

motor_speed=[-255,255], positive number means forward rotation, negative number means backward rotation, 0 means stop

Return value: None.

```
Ctrl_Car(motor_id, motor_dir, motor_speed):
```

Parameter explanation: Control the forward and reverse rotation and speed of the motor. You need to use an additional judgment parameter to control the forward and reverse rotation

motor_id=[1,4]: 1: M1 motor, 2: M2 motor, 3: M3 motor, 4: M4 motor

motor_dir=[0,1]: 0: forward rotation, 1: reverse rotation

motor_speed=[0,255], the larger the value, the faster the motor speed, 0 means stop.

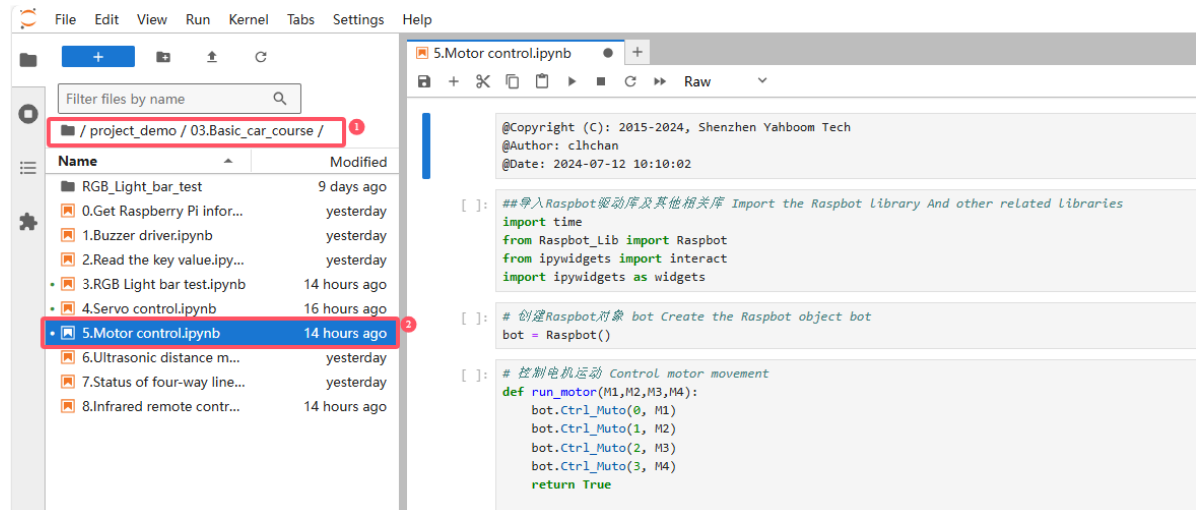
Return value: None.

Source code path: project_demo\03.Basic_car_course

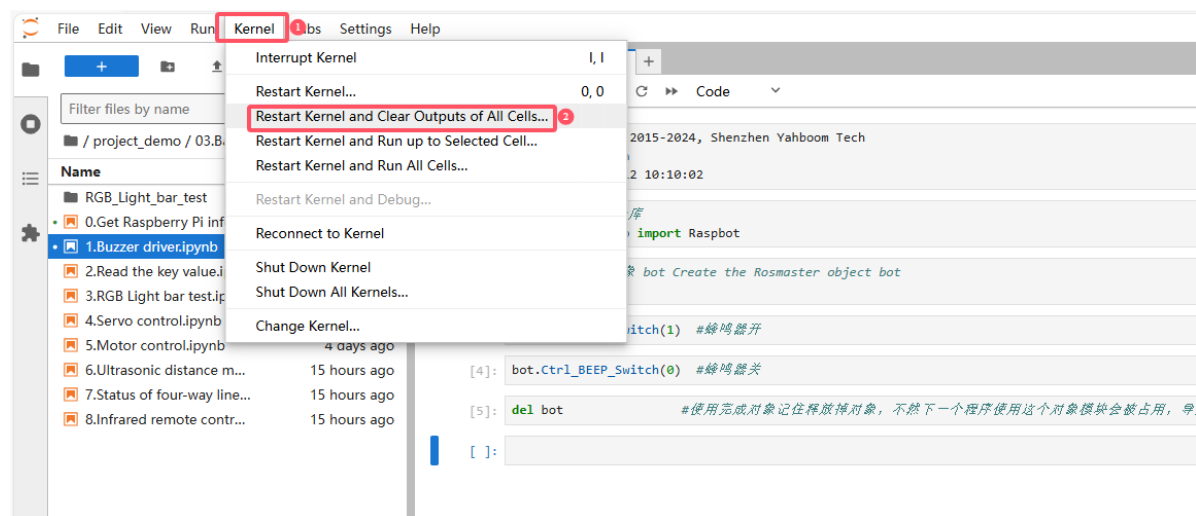
4. Experimental phenomenon

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

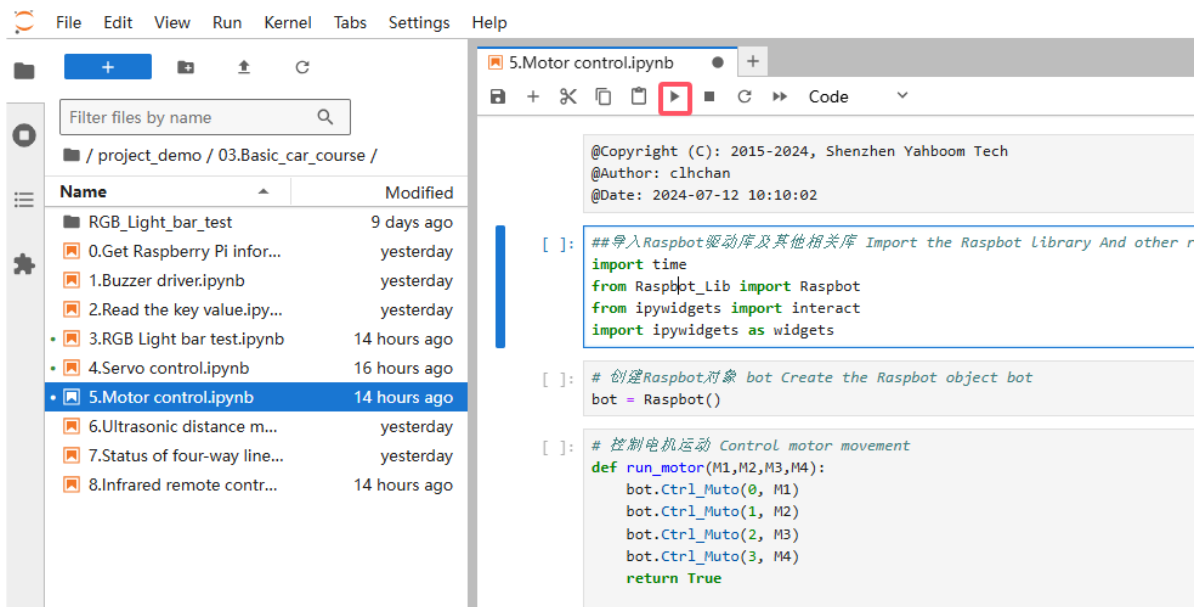
Enter the source code path and double-click the code to be run



Restart the kernel and clear all outputs



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 control the speed and forward and reverse rotation of the motor.