

# Control the motor to rotate forward and reverse

---

[Control the motor to rotate forward and reverse](#)

[7.1. Experimental Objectives](#)

[7.2. Experimental Preparation](#)

[7.3, Program source code](#)

[7.4, Experimental effect](#)

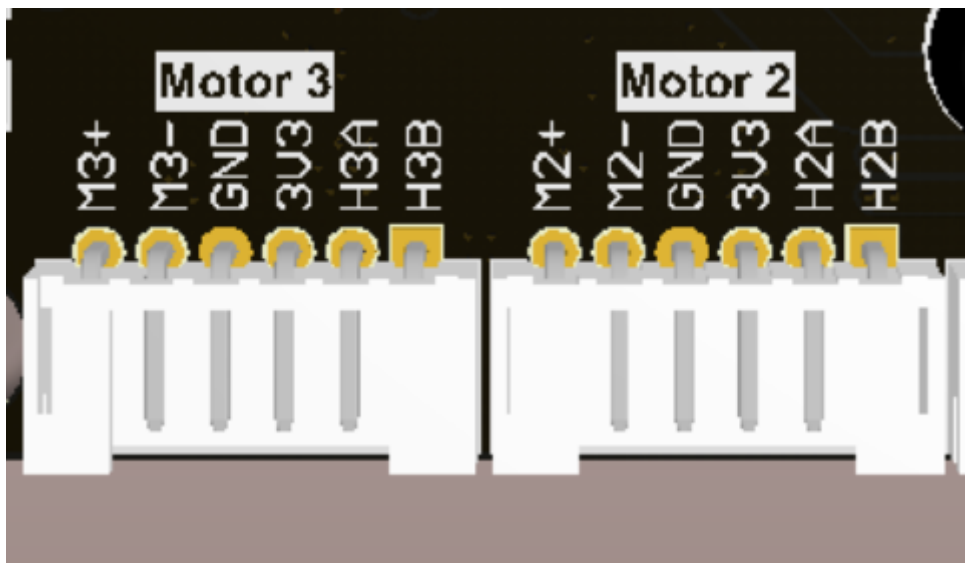
## 7.1. Experimental Objectives

Control the forward and reverse rotation of the robot's motor, and control the motor's movement speed by controlling the motor's PWM duty cycle.

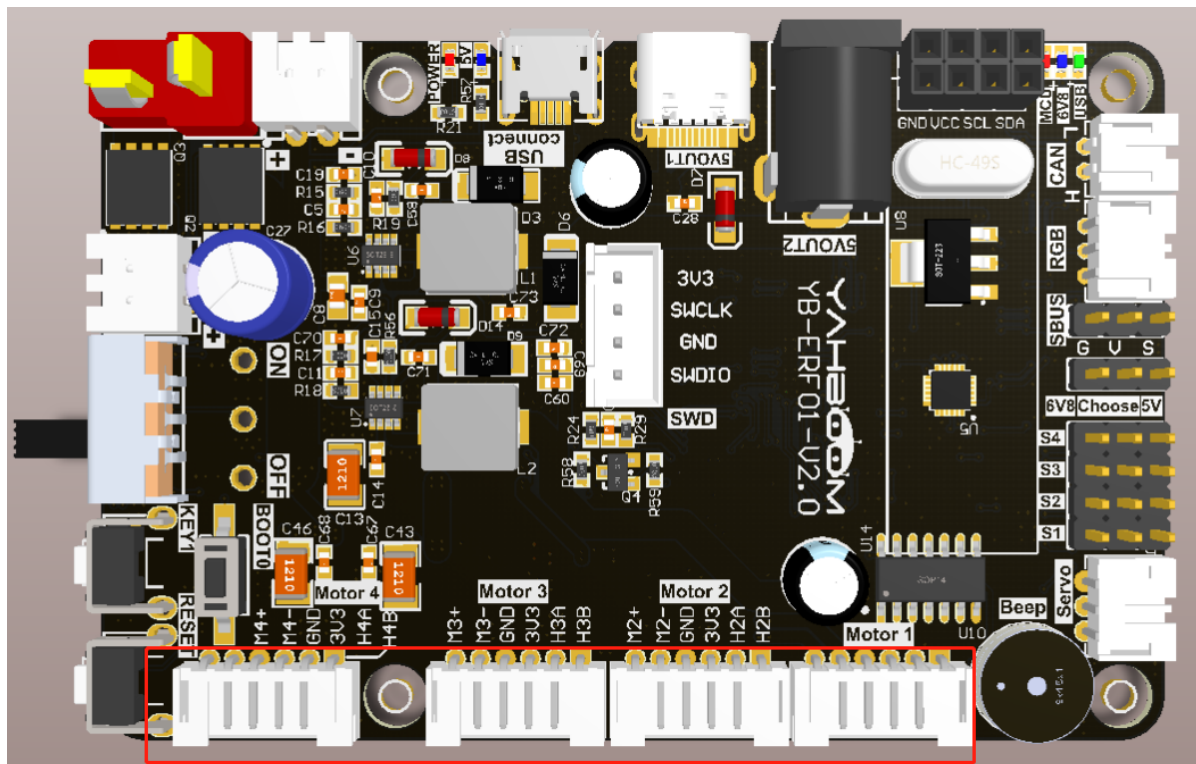
## 7.2. Experimental Preparation

The red box in the picture below is the motor MOTOR 1, MOTOR 2, MOTOR 3, MOTOR 4. The line sequence of the four motor interfaces is consistent. The motor interface has an anti-reverse connection function. Use the motor cable to connect to the motor.

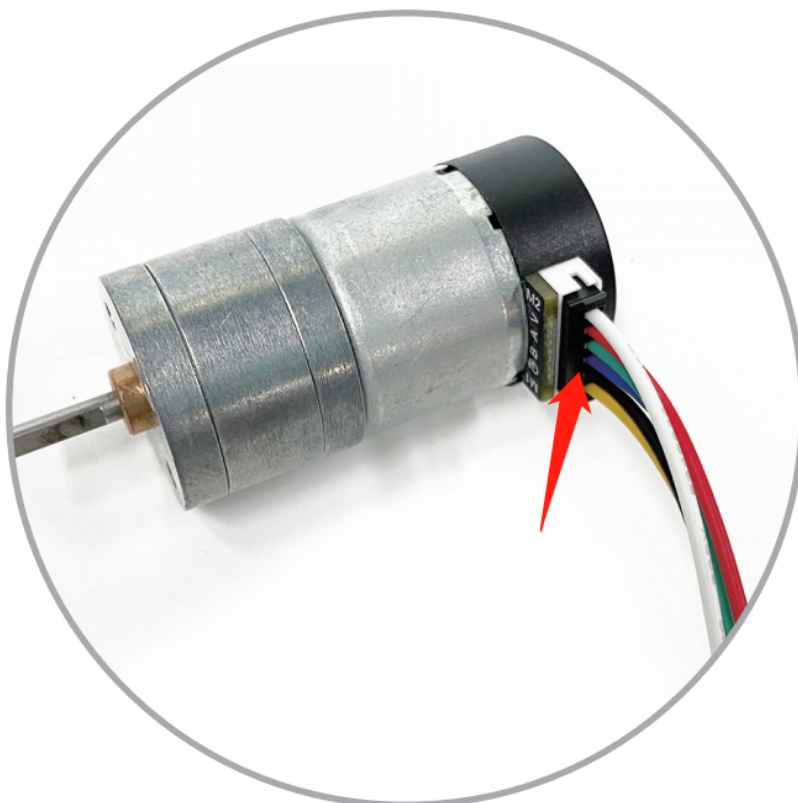
The line sequence of the motor interface is shown in the figure below:



MOTOR 1 is connected to the left front motor of the car, MOTOR 2 is connected to the left rear motor of the car, MOTOR 3 is connected to the right front motor of the car, and MOTOR 4 is connected to the right rear motor of the car.



The robot motor cable interface has a white interface at one end and a black interface at the other end. The white interface is connected to the expansion board and the black interface is connected to the motor.



SunriseRobotLib library functions needed to control the motor:

```
set_motor(speed_1, speed_2, speed_3, speed_4)
```

Parameter explanation: Control the motor PWM pulse to control the motor speed. This function does not use the encoder speed measurement function.

speed\_X=[-100, 100], positive numbers are forward rotation, negative numbers are backward rotation.

Return value: None.

### 7.3, Program source code

Turn on the power switch of the robot, and open the browser of the robot system or remote computer to enter the Jupyter lab editor.

Reference code path: /home/sunrise/sunriseRobot/Samples/1\_Basic/4\_motor.ipynb

### 7.4, Experimental effect

Open jupyterlab, find the corresponding program source code, please close other running source code files first.

Click the run button in sequence. When running to the cell that controls the motor, four sliders will appear, indicating the control speed of motor M1/M2/M3/M4 respectively. Drag the slider to change the speed of the motor.

```
# 控制电机运动 Control motor movement
def run_motor(M1, M2, M3, M4):
    bot.set_motor(M1, M2, M3, M4)
    return M1, M2, M3, M4

# 创建四个滑块来控制电机 Create four sliders to control the motor
interact(run_motor, \
    M1=widgets.IntSlider(min=-100,max=100,step=1,value=0), \
    M2=widgets.IntSlider(min=-100,max=100,step=1,value=0), \
    M3=widgets.IntSlider(min=-100,max=100,step=1,value=0), \
    M4=widgets.IntSlider(min=-100,max=100,step=1,value=0));
```



Motor	Value
M1	0
M2	-31
M3	21
M4	26

Since this function controls the rotation of the motor by modifying the PWM duty cycle, it can only be used to test whether the motor is working properly. This function is rarely used in actual use.