# 7. Control the motor forward and reverse

7.1. Experimental objectives

By controlling the PWM duty cycle of the motor, the motion speed of the motor is controlled.

7.2. Experiment Preparation

The position of the red box in the picture below is MOTOR 1, MOTOR 2, MOTOR 3, MOTOR 4. The line sequence of the four motor interfaces is consistent. The motor interface has the function of anti-reverse connection, and it can be connected to the motor by using the motor line.

The sequence of the motor interface is shown as follows:
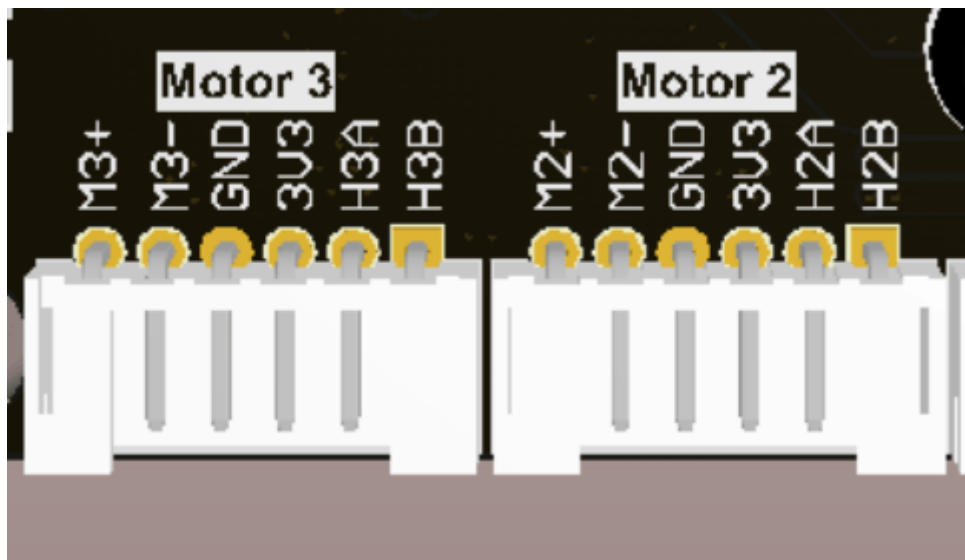
MOTOR 1 is connected to the left front MOTOR of the car, MOTOR 2 is connected to the left rear MOTOR of the car, motor 3 is connected to the right front motor of the car, and MOTOR 4 is connected to the right rear motor of the car.
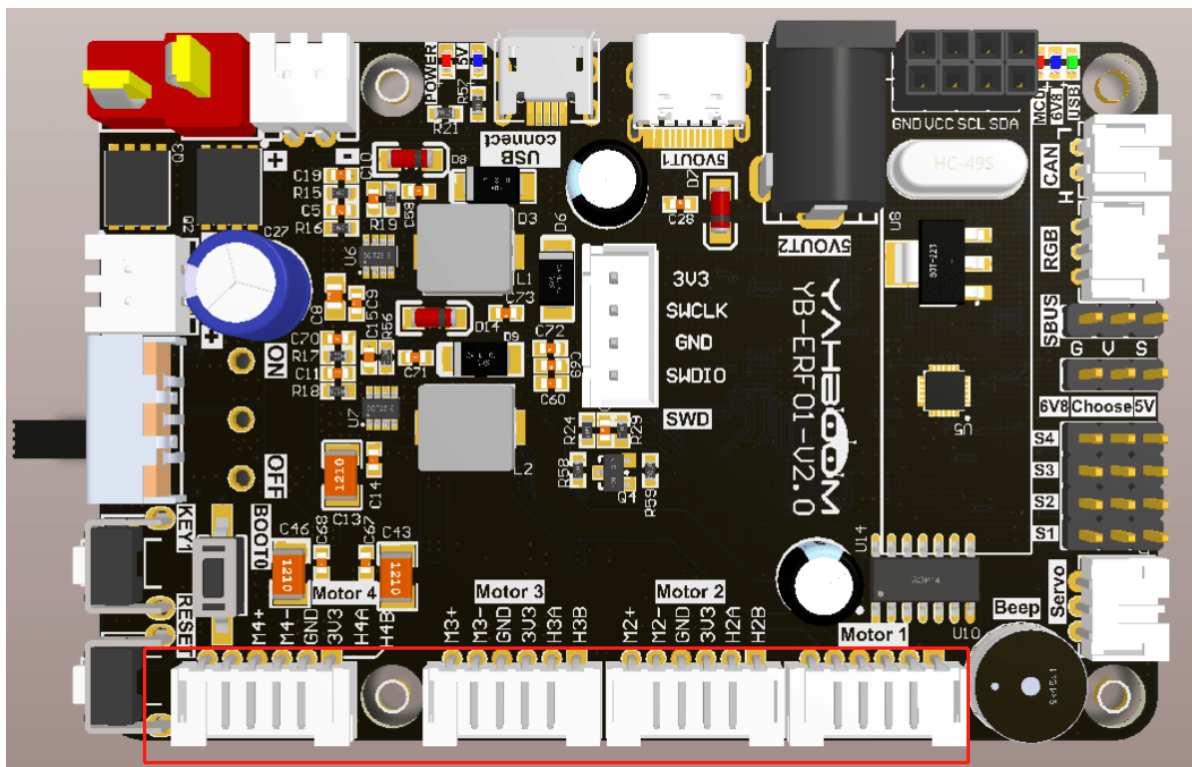
One end of the robot motor cabling interface is a white interface, and the other end is a black interface. The white interface is connected to the expansion board, and the black interface is connected to the motor.
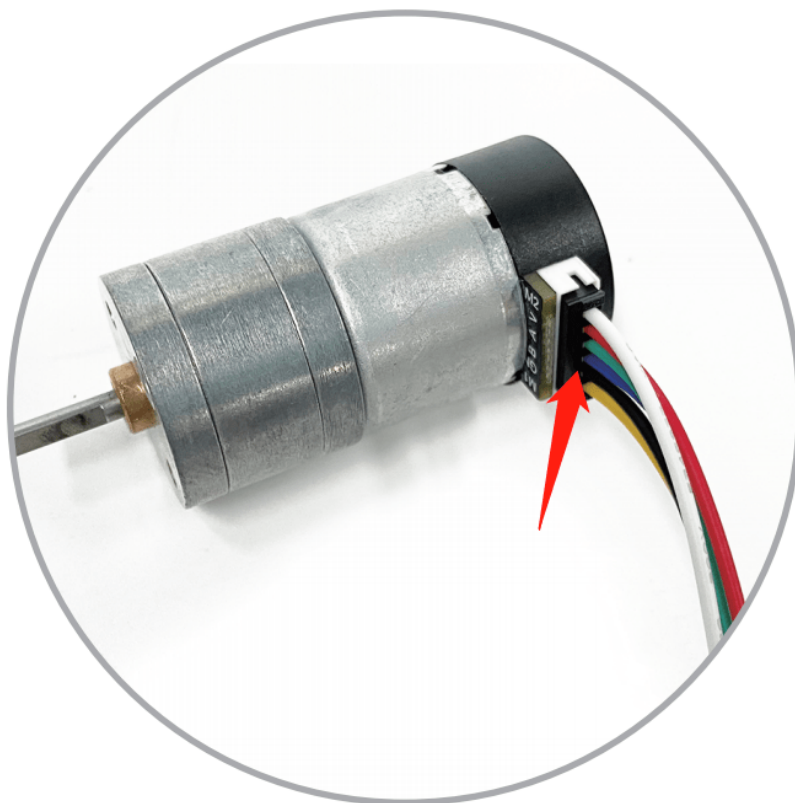
Motor cable black shell connector connected to the motor

Control motor need SunriseRobotLib library functions:

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

Parameter interpretation: Control the motor PWM pulse, thus controlling the motor speed. This function does not use encoder speed measurement.

speed_X=[-100, 100], positive numbers for forward rotation, negative numbers for backward rotation.

Returned 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: / root/sunriseRobot _basic/Samples / 1/4 _motor. lpynb

7.4. Experimental results

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

Click the Run button in sequence. When the cell controlling the motor is run, four sliding bars will appear, indicating the control speed of the motor M1/M2/M3/M4 respectively. Drag the sliding bar to change the speed of the motor.

```python
# 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));
```

M1   0
M2   -31
M3   21
M4   26

Since this function is to control the rotation of the motor by modifying the PWM duty ratio, it can only be used to test whether the motor is working normally, and this function is less used in the actual use process.