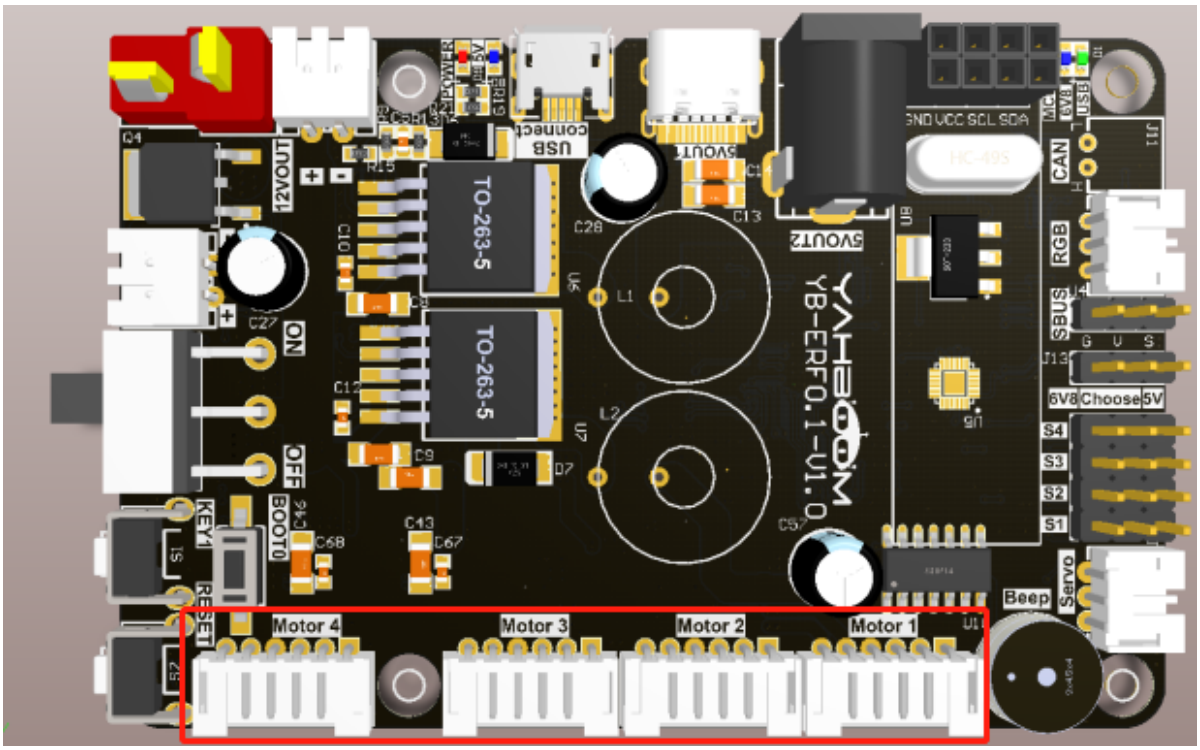
7. Control motor

7.1. Experimental goal

Control the forward and reverse rotation of the motor on the Rosmaster, and control the speed of the motor by controlling the PWM duty cycle of the motor.

7.2. Experiment preparation

The red squares in the picture below are the motor MOTOR 1, MOTOR 2, MOTOR 3, and MOTOR 4. The motor interface has an anti-reverse connection function, which can be connected to the motor using Rosmaster's motor cable. Here you need to pay attention to connecting the motors according to different models. Here is an example of a Mecanum wheel car. MOTOR 1 is connected to the left front motor of the car, MOTOR 2 is connected to the left rear motor of the car, and MOTOR 3 is connected to the right front motor of the car. MOTOR 4 is connected to the right rear motor of the dolly.



Rosmaster_Lib library functions required to control the Rosmaster motor:

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

Parameter explanation: Control the motor PWM pulse, thereby controlling the motor speed. This function does not use the encoder speed function.

speed_X=[-100, 100], a positive number means forward rotation, and a negative number means backward rotation.

Return value: None.

7.3. Experimental effect

Refer to 4.3.1 and 4.3.2 Checking ROS expansion board and entering Docker container, then run the program. double-click on the jupyter lab interface to enter /root/yahboomcar_ros2_ws/Rosmaster/Sample.

Double-click and select 7.motor.ipynb, and click the button in 4.3.3 to operate the program step by step.

7.4. Program source code

Go to docker, Reference code path:

/root/yahboomcar_ros2_ws/Rosmaster/Sample/7.motor.ipynb