

# 7 QR code instruction

---

## 7 QR code instruction

- 7.1 the experimental description
- 7.2 experimental target
- 7.3 the experimental operation
- 7.4 experimental results
- 7.5 the experiments are summarized

## 7.1 the experimental description

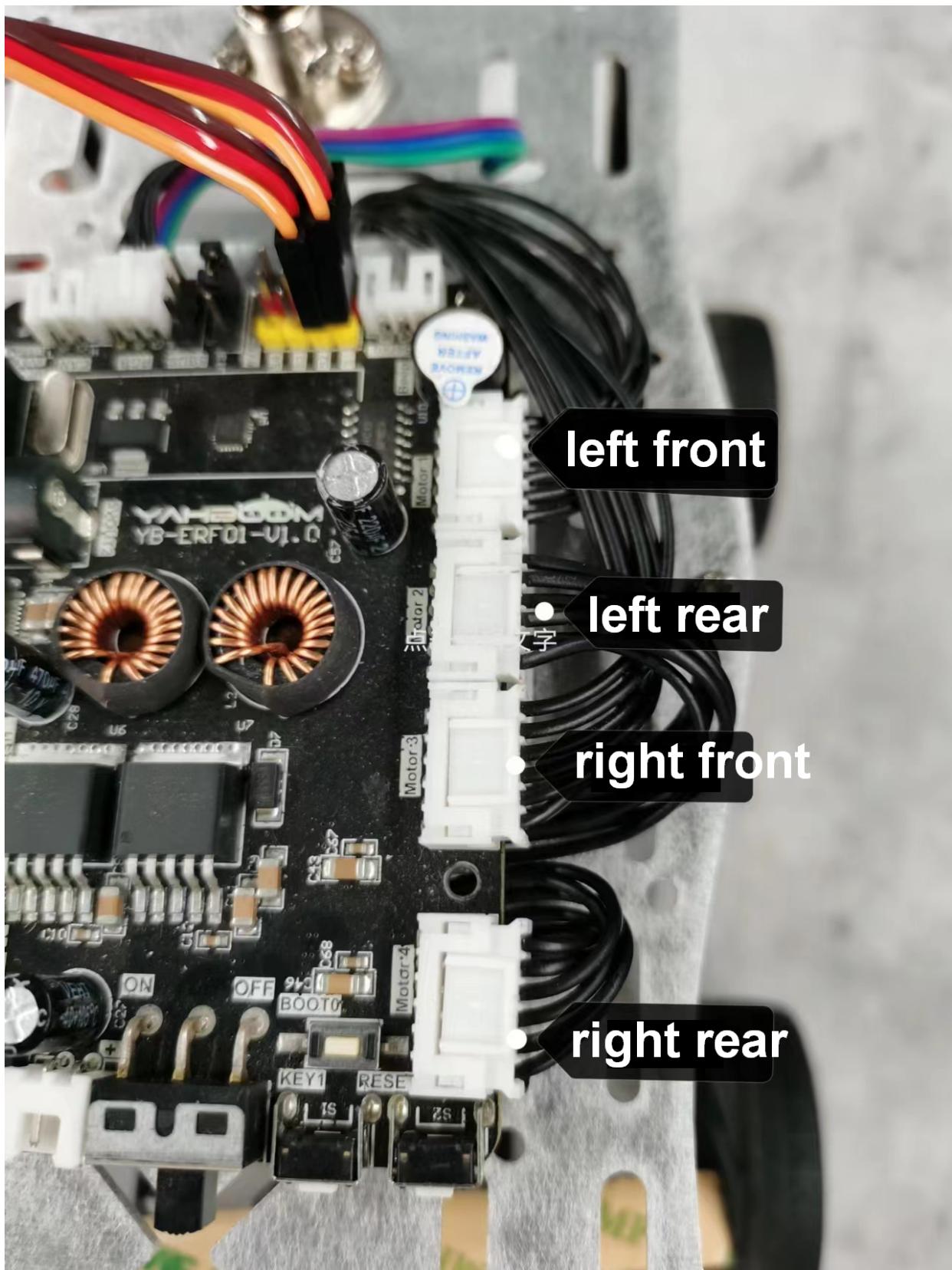
The present experiment was to belong to expand the class of experiments, the need to match other external devices to use, here to the car chassis and the ROS expansion Board is not part of K210 module kit contents, so the present experimental results are for reference only, if there is no corresponding device is not directly use the routine code.

ROS expansion Board needs advance programming firmware: ROS-CAR. hex

Due to the use of the motor voltage is 8. 4V, so the ROS expansion Board battery may not be inserted 12. 6V battery, you need to insert 8. 4V battery.

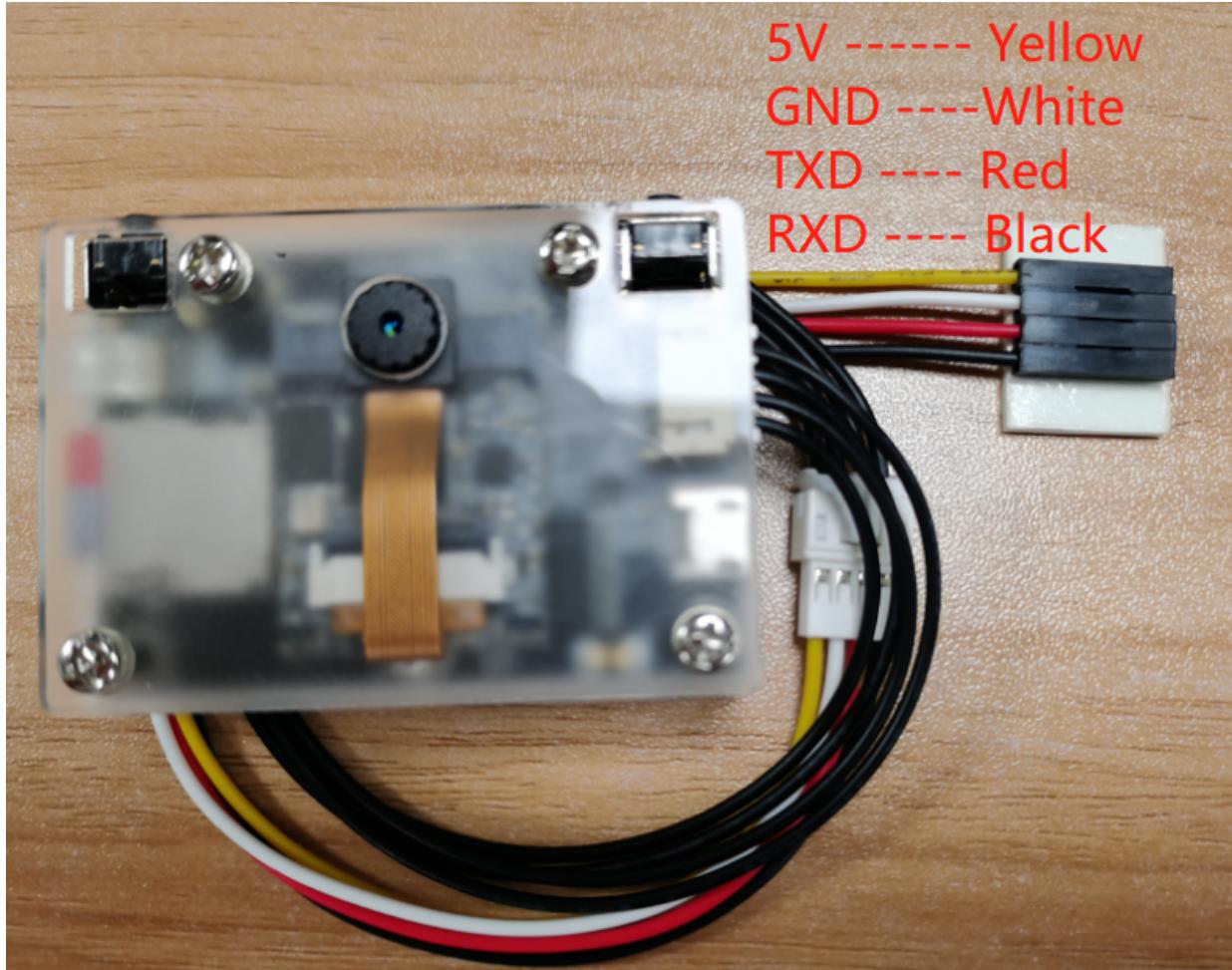
Trolley motor connected to the line as shown below:

Motor Motor 1 is connected to the Left Front wheel, the Motor Motor 2 is connected to the left rear wheel, motor Motor 3 is connected to the right front wheel, the Motor Motor 4 is connected to the right rear wheel.

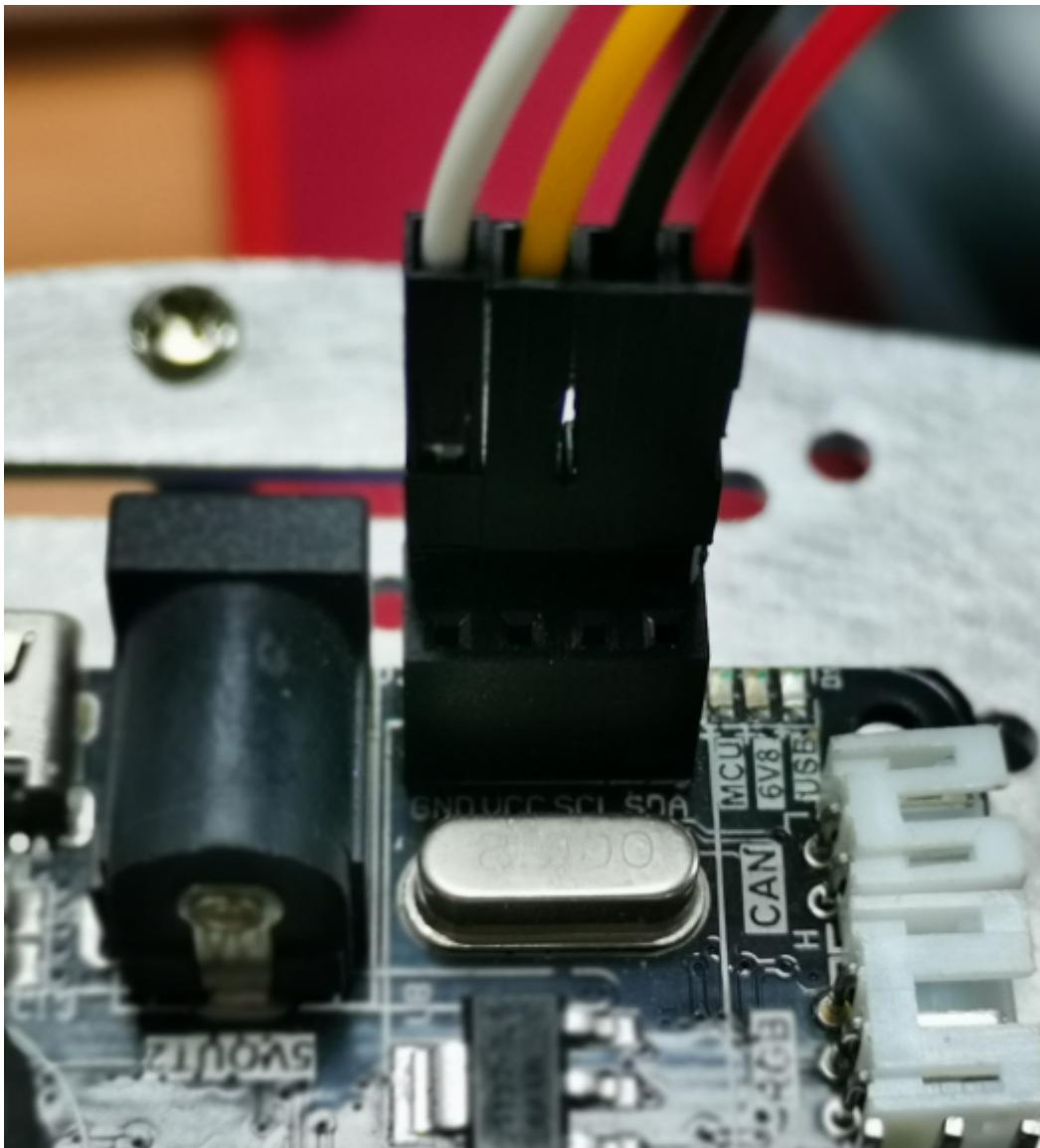


K210 module with the ROS expansion Board connected to the line sequence as shown below:

White(GND) connected to GND, yellow(5V) is connected to VCC, the black(RXD) connect the SCL, the red(TXD) connected to SDA.



Here you note that the illustration of the logo for the I2C line sequence identity, but K210 using serial communication, due to the burning of the ROS-CAR. the hex file has been put on this interface modification for the serial signal, so in fact the ROS expansion Board on the interface corresponding relationship is: the SCL is actually TX, SDA is actually RX.



## 7.2 experimental target

This lesson is mainly learning K210 module with the car chassis to do a visual inspection of the line features.

The present experiments the reference code path is: CanMV\06-export\qrcode\_motion.py

## 7.3 the experimental operation

1. ROS expansion Board to burn the firmware: ROS-CAR. hex
2. Insert the RGB light strip into the RGB light interface of the ROS expansion board.
3. Please CanMV\06-export\library directory under the trolley driver library and PID control Library in advance to download to the memory card root directory.
4. Open CanMV IDE open qrcode\_motion. py code and downloaded to the K210 module.
5. The K210 module via the 4PIN cable is connected to the ROS expansion Board.

6. The trolley into the white background, breaking move K210 module bracket to a suitable angle, turn the car switch.
7. With the specific information of the two-dimensional code to the trolley front of the camera, the K210 module to identify the corresponding information, perform the corresponding operation.

## 7.4 experimental results

Wait for the system initialization is complete, the car does not recognize the QR code and when not in motion, when the identification with the "turn left" the character of the two-dimensional code, will perform a left turn, when the identification with the "turn right" character of the code, will perform a turn to the right, and "spin left" character is left-handed, "spin right" character is right-handed, "go ahead", the character is moving forward, "fall back" character is backwards. Recognition to the corresponding character after running the function, after a certain time will automatically stop, stop can again identify two-dimensional code.

```
def parse_payload(payload):
    global car_count
    if payload == "turn left":
        print("car turn left")
        car_count = 20
        bot.set_car_motion(0, 0, 3)
    elif payload == "turn right":
        print("car turn right")
        car_count = 20
        bot.set_car_motion(0, 0, -3)
    if payload == "spin left":
        print("car spin left")
        car_count = 40
        bot.set_car_motion(0, 0, 3)
    elif payload == "spin right":
        print("car spin right")
        car_count = 40
        bot.set_car_motion(0, 0, -3)
    elif payload == "go ahead":
        print("car go ahead")
        car_count = 20
        bot.set_car_motion(0.5, 0, 0)
    elif payload == "fall back":
        print("car fall back")
        car_count = 20
        bot.set_car_motion(-0.5, 0, 0)
```

## **7.5 the experiments are summarized**

Trolley identify the QR code to execute the instruction functionality is based on the two-dimensional code recognition function to recognize the QR code immediately after the execution of the corresponding action, if you recognize the two-dimensional character is not the default content, you do not do any exercise.