

7、QR code command action

7、QR code command action

7.1、Experiment Description

7.2、Experimental goal

7.3、Experimental steps

7.4、Experimental effect

7.5、Experiment summary

7.1、Experiment Description

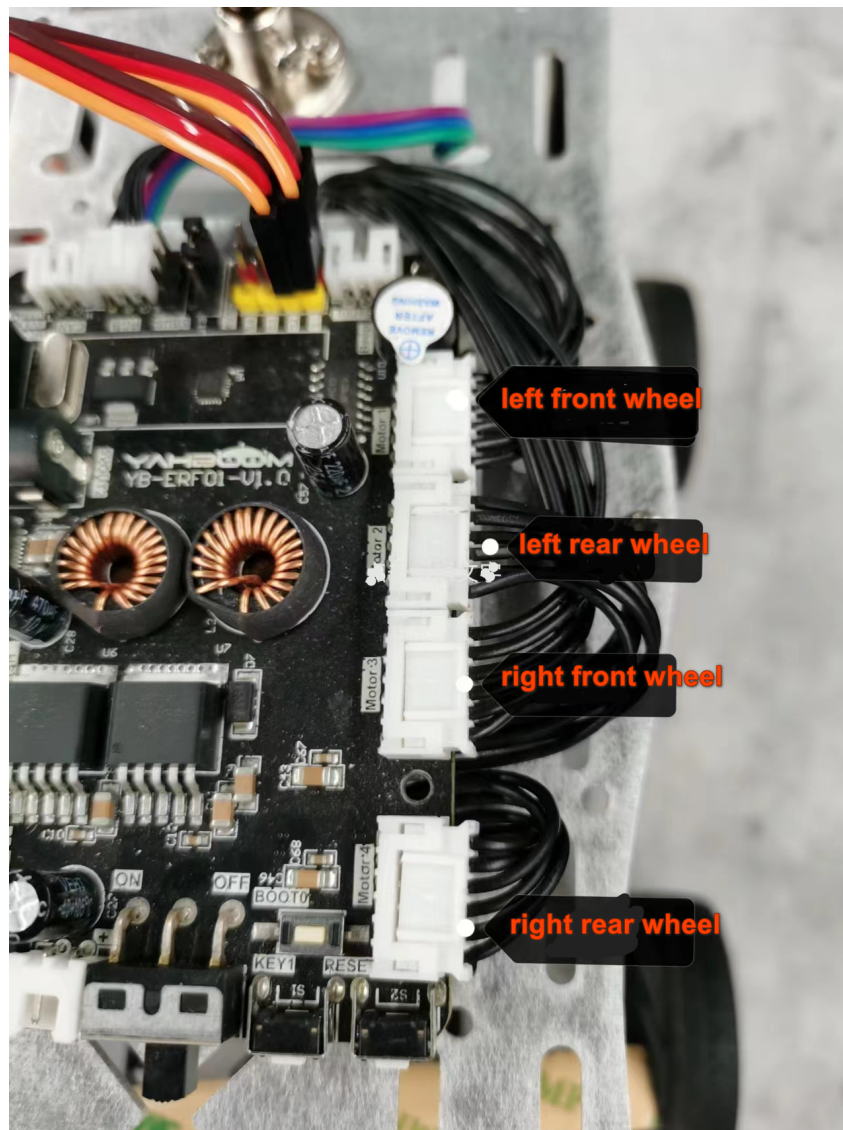
Note: This experiment is an expansion experiment and needs to be used with other external devices. The car chassis and ROS expansion board used here are not part of the K210 development board kit, so the effect of this experiment is for reference only. If there is no corresponding device, it cannot be used. Use this example code directly.

The ROS expansion board needs to flash the firmware in advance: ROS-CAR.hex

Since the voltage of the motor used this time is 8.4V, the battery of the ROS expansion board cannot be inserted into a 12.6V battery, and an 8.4V battery must be inserted.

The connecting wire of the trolley motor is shown in the figure below:

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



The line sequence of the connection between the K210 development board and the ROS expansion board is shown in the figure below:

The white wire is connected to GND, the yellow wire is connected to VCC, the black wire is connected to SCL, and the red wire is connected to SDA.

It should be noted here that the logo in the diagram is the I2C line sequence logo, but the K210 uses serial port communication. Since the burned ROS-CAR.hex file has changed this interface to a serial port signal, the actual ROS expansion board The corresponding relationship of the interface is: SCL is actually TX, and SDA is actually RX.



7.2、 Experimental goal

This lesson mainly learns the function of K210 development board and car chassis for visual line inspection.

The reference code path for this experiment is: 06-export\qrcode_motion.py

7.3、 Experimental steps

1. ROS expansion board flash firmware: ROS-CAR.hex
2. Insert the RGB light bar into the RGB light interface of the ROS expansion board.
3. Please download the trolley driver library and PID control library in the 06-export\library directory to the root directory of the memory card in advance.
4. Open CanMV IDE, open the qrcode_motion.py code and download it to the K210 development board.

5. Connect the K210 development board to the ROS expansion board through a 4PIN cable.
6. Put the car into the white background, move the K210 development board bracket to an appropriate angle, and turn on the switch of the car.
7. Put the QR code with specific information in front of the camera of the car. After the K210 development board recognizes the corresponding information, it will perform the corresponding operation.

7.4、 Experimental effect

After the system initialization is completed, the car will not move when it does not recognize the QR code. When it recognizes the QR code with the "turn left" character, it will turn left. When it recognizes the "turn right" character When using the QR code, it will turn right, the "spin left" character is rotate left, the "spin right" character is rotate right, the "go ahead" character is forward, and the "fall back" character is backward. After the corresponding character is recognized, the function will run, and it will stop automatically after a certain period of time, and the QR code can be recognized again after stopping.

```
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、 Experiment summary

The trolley recognizes the two-dimensional code and executes the instruction function based on the two-dimensional code recognition function. After the two-dimensional code is recognized, the corresponding action is executed immediately. If the recognized two-dimensional code character is not the preset content, no movement is performed.