

## 5、road sign indicating action

---

### 5、road sign indicating action

#### 5.1、Experiment Description

#### 5.2、Experimental goal

#### 5.3、practical operation steps

#### 5.4、Experimental effect

#### 5.5、Experiment summary

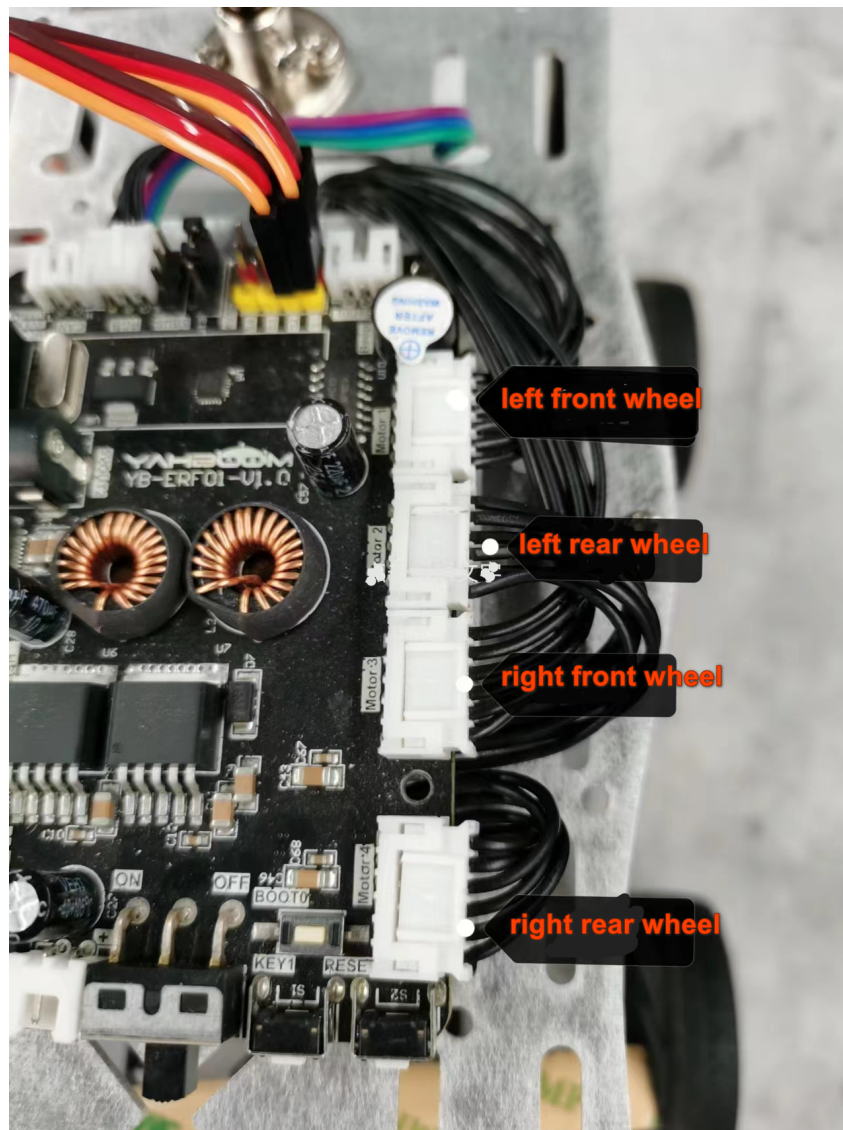
## 5.1、Experiment Description

**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 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.



## 5.2、 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\sign\_motion.py

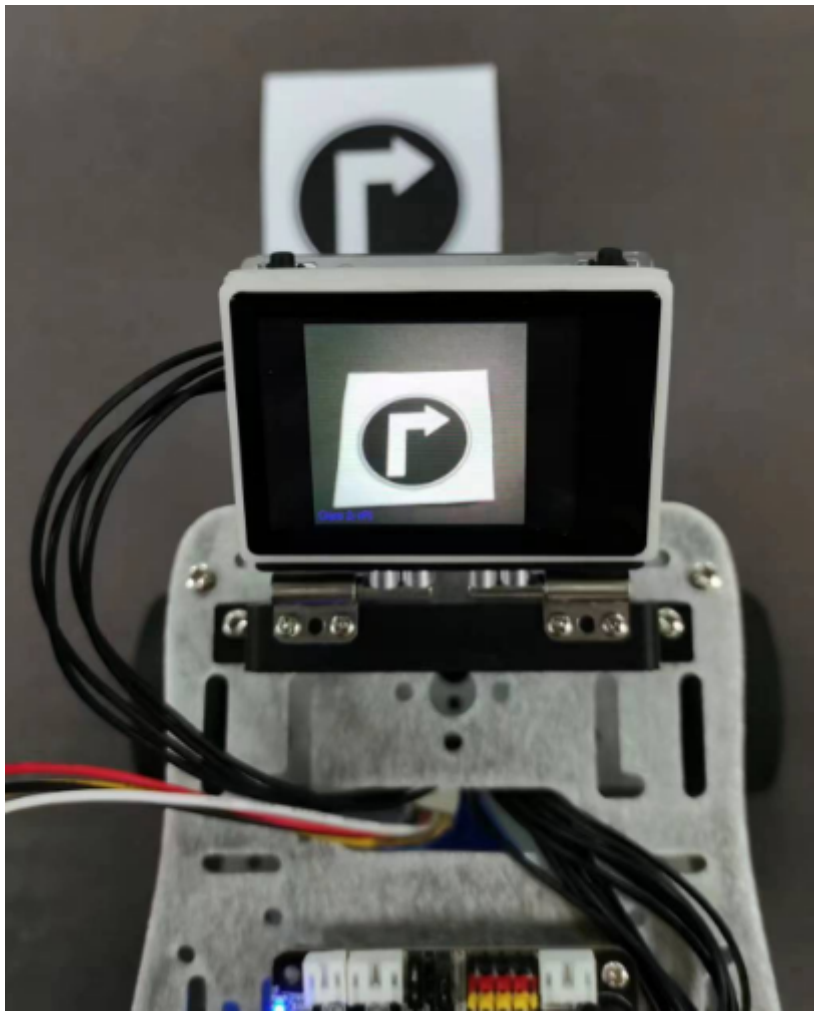
## 5.3、 practical operation steps

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 the CanMV IDE, open the sign\_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 a white or black background, move the K210 development board bracket to an appropriate angle, and turn on the switch of the car.
7. First of all, you need to learn the left-turn icon. The operation steps are the same as the self-learning method. Take five pictures of the left-turn icon according to the screen prompts.



8. Then learn the right-turn icon and follow the screen prompts to take five pictures of the right-turn icon.



9. Then learn the stop icon and follow the on-screen prompts to take five pictures of the stop icon.
10. After the learning is completed, the car starts to move forward, and when it detects that there is a corresponding icon in the image, it executes the corresponding action.

## 5.4、 Experimental effect

After waiting for the system initialization to complete, after learning the road signs according to the above operation steps, the car starts to move forward. When it detects a left-turn road sign in the camera image, it executes the left-turn function and then goes straight; when it detects a right-turn road sign, Perform a right turn function before going straight; when a stop sign is detected, the car stops.

The turning range and function of the car can be modified in the `car_control` function, `car_count` indicates the turning time, and `set_car_motion` sets the turning speed.

```
def car_control(class_id):  
    global car_count, car_state  
    car_state = 1  
    if class_id == 1:# 向左转 turn left  
        car_state = 1  
        car_count = 10  
        bot.set_car_motion(0, 0, 3)  
    elif class_id == 2:# 向右转 turn right  
        car_state = 1  
        car_count = 10  
        bot.set_car_motion(0, 0, -3)  
    elif class_id == 3: # 停止 stop  
        car_state = 0  
        car_count = 0  
        bot.set_car_motion(0, 0, 0)
```

## 5.5、 Experiment summary

The car road sign recognition function is modified based on the self-learning function of the K210 development board. The learning steps are similar to the self-learning steps. First learn the left-turn icon, then learn the right-turn icon, and finally learn the parking icon. Due to the movement of different cars and different maps The effect is different, you can modify the content in the `car_control` function of the car to optimize the running effect.