3、handwritten digital action

3.1、Experiment Description

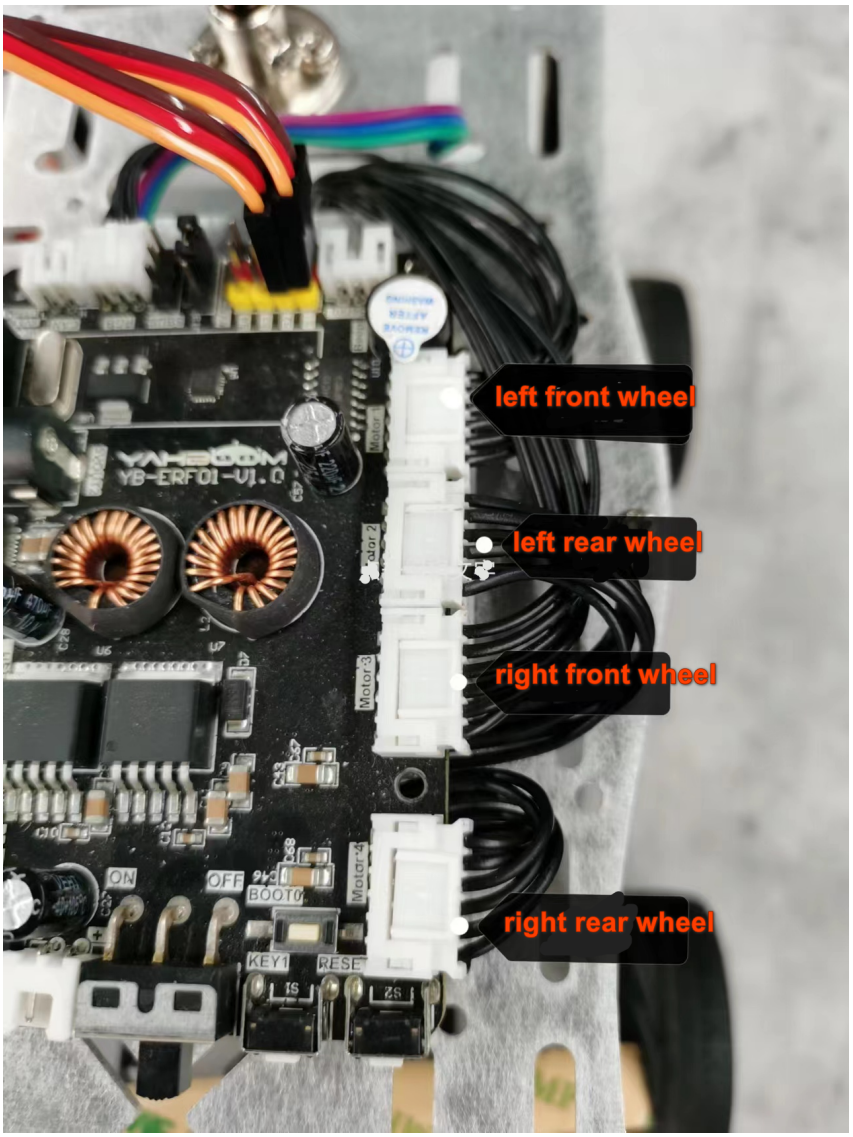
Note: This experiment is an expansion experiment and needs to be used with other external devices. The car chassis and ROS expansion board used here are not part of the K210 development board kit, so the effect of this experiment is for reference only. If there is no corresponding device, it cannot be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

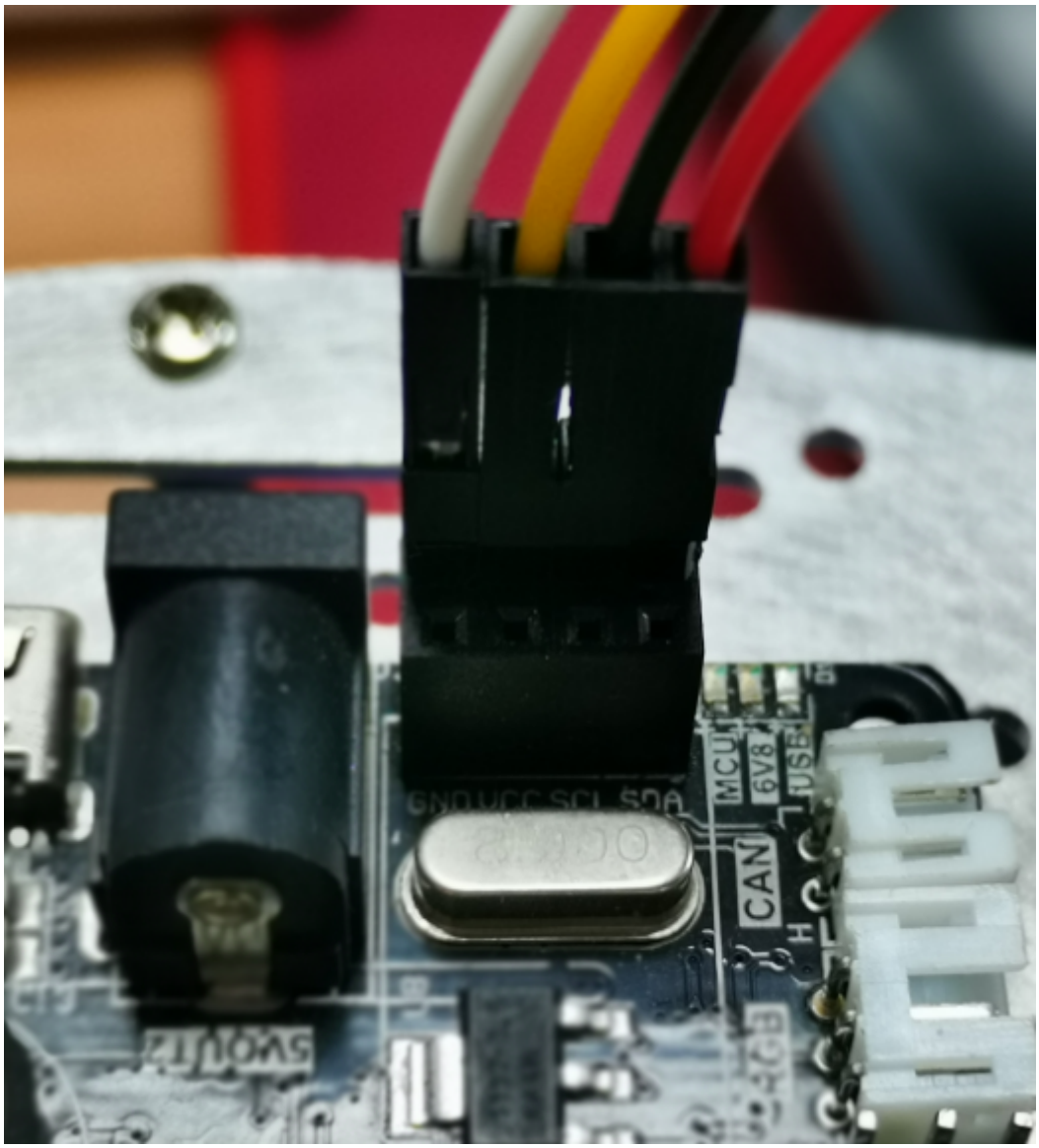
The motor Motor 1 is connected to the left front wheel, the motor Motor 2 is connected to the left rear wheel, the motor Motor 3 is connected to the right front wheel, and the motor Motor 4 is connected to the right rear wheel.



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.



3.2、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\mnist_car.py

3.3、Experimental operation

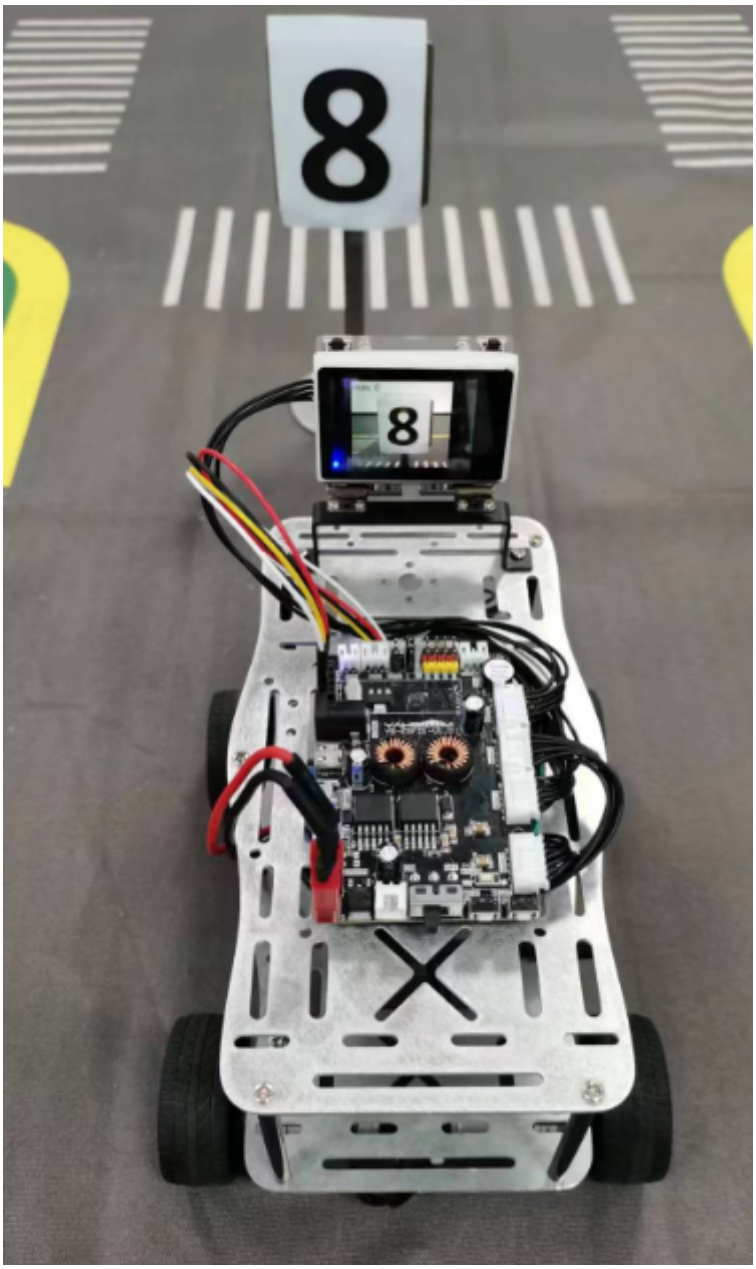
1. ROS expansion board flash firmware: ROS-CAR.hex
2. Connect the baseboard motor to the ROS expansion board, connect the left front motor according to M1, connect the left rear motor with M2, connect the right front motor with M3, and connect the right rear motor with M4.
3. Please download the trolley driver library and PID control library in the 06-export\library directory to the root directory of the memory card in advance.

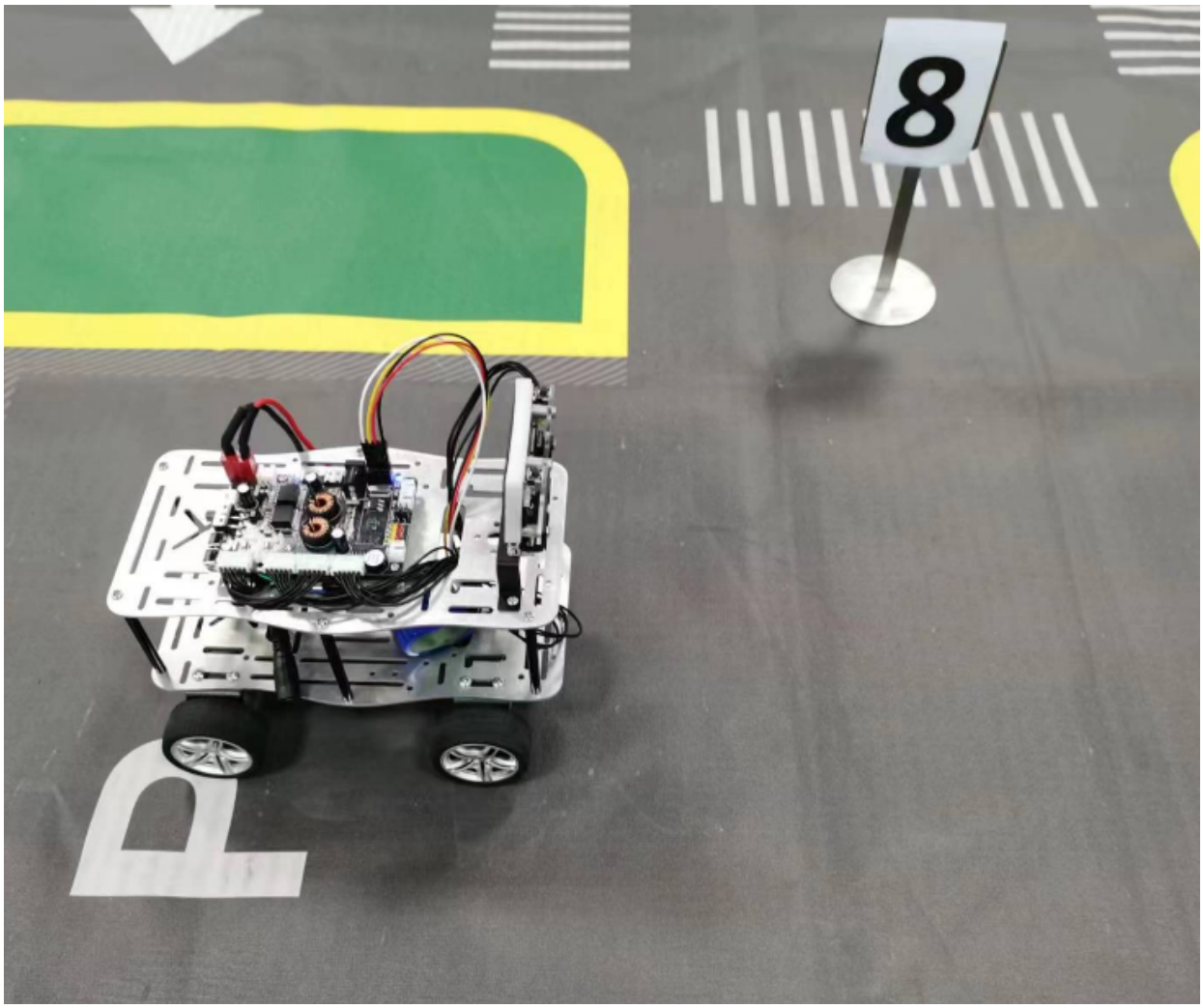
4. Open CanMV IDE and download the follow_color.py code into the K210 development board.
5. Put the car into the white background, move the K210 development board bracket to an appropriate angle, and turn on the switch of the car.
6. Put the car into the white background, move the K210 development board bracket to an appropriate angle, and turn on the switch of the car.

3.4、Experimental effect

After the system initialization is completed, the car will move forward for a certain distance, then stop, and analyze whether there is a number 8 in the current camera screen. If other numbers are detected, please change `ACTION_NUM = 8` to other numbers.

If the number 8 is detected three times in a row, the function of backing into the garage will start to run.





The function of reversing into storage is divided into two parts, which can be modified according to different maps.

```
car_count = 0
car_state = 0
motion_index = 0
speed_line = -0.2
speed_angular = -1.6
back_count = 22
turn_count = 22
run_end = 0
ACTION_NUM = 8
```

3.5、Experiment summary

The handwritten digital action function of the trolley mainly realizes the function of car recognition and digital reversing storage. Since the reversing movement of the trolley is a fixed action, it is necessary to re-adjust the speed and running time after the map changes to achieve the optimal effect.