

3 handwriting digital operation

[3 handwriting digital operation](#)

- [3.1 experiment description](#)
- [3.2 experimental target](#)
- [3.3 experimental operation](#)
- [3.4 experimental results](#)
- [3.5 the experiments are summarized](#)

3.1 experiment description

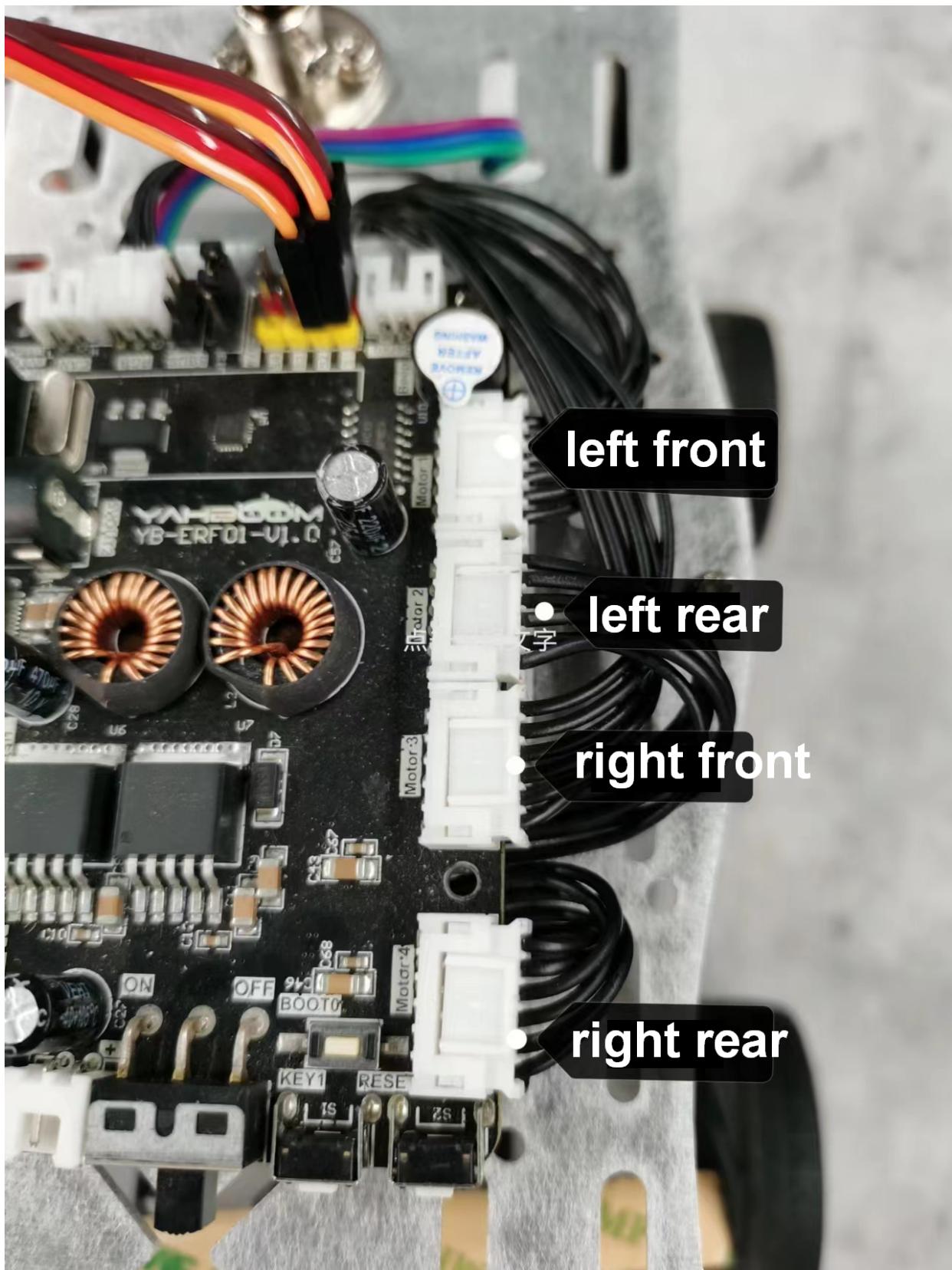
The present experiment was to belong to expand the class of experiments, the need to match other external devices to use, here to the car chassis and the ROS expansion Board is not part of K210 module kit contents, so the present experimental results are for reference only, if there is no corresponding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

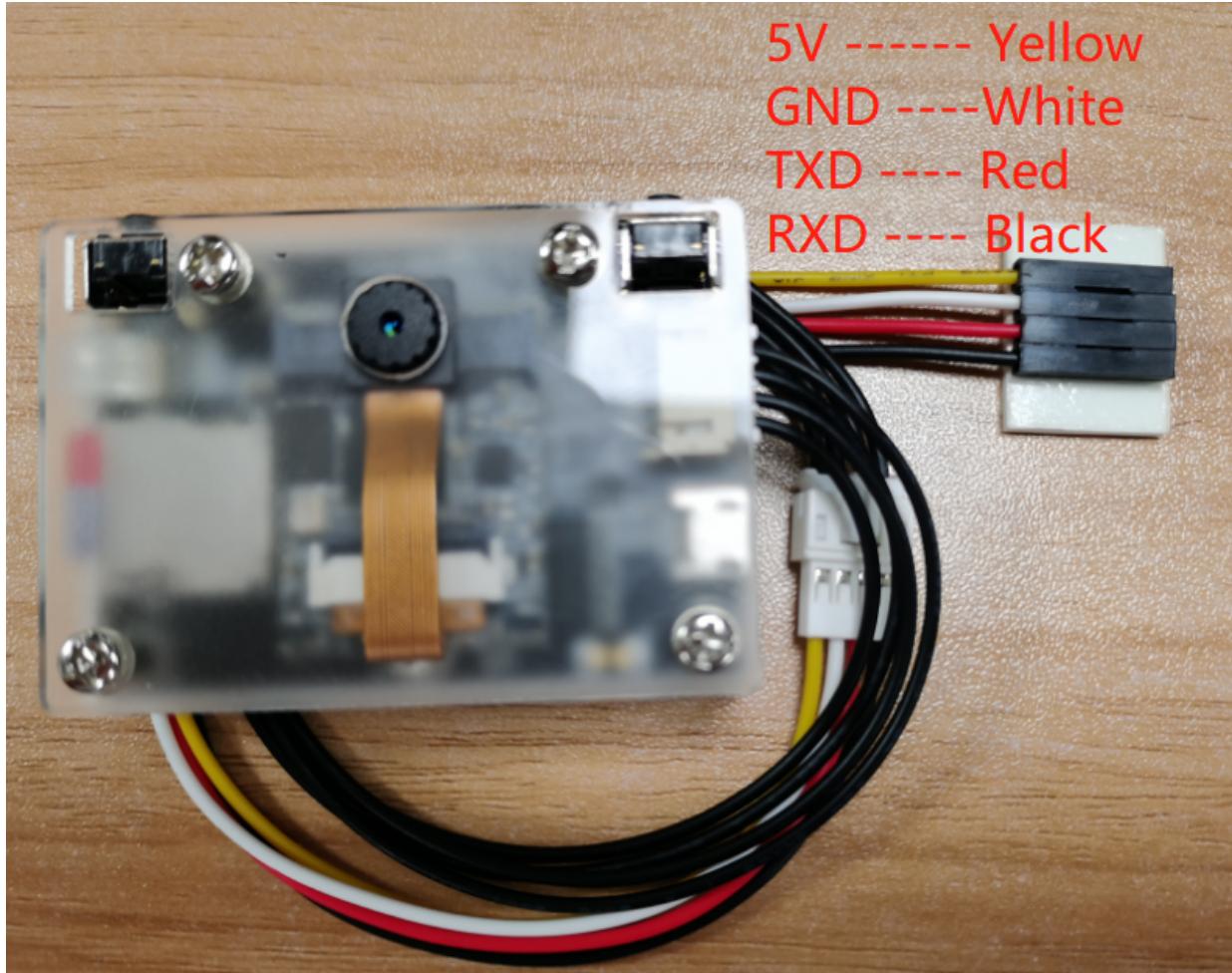
Trolley motor connected to the line as shown below:

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.

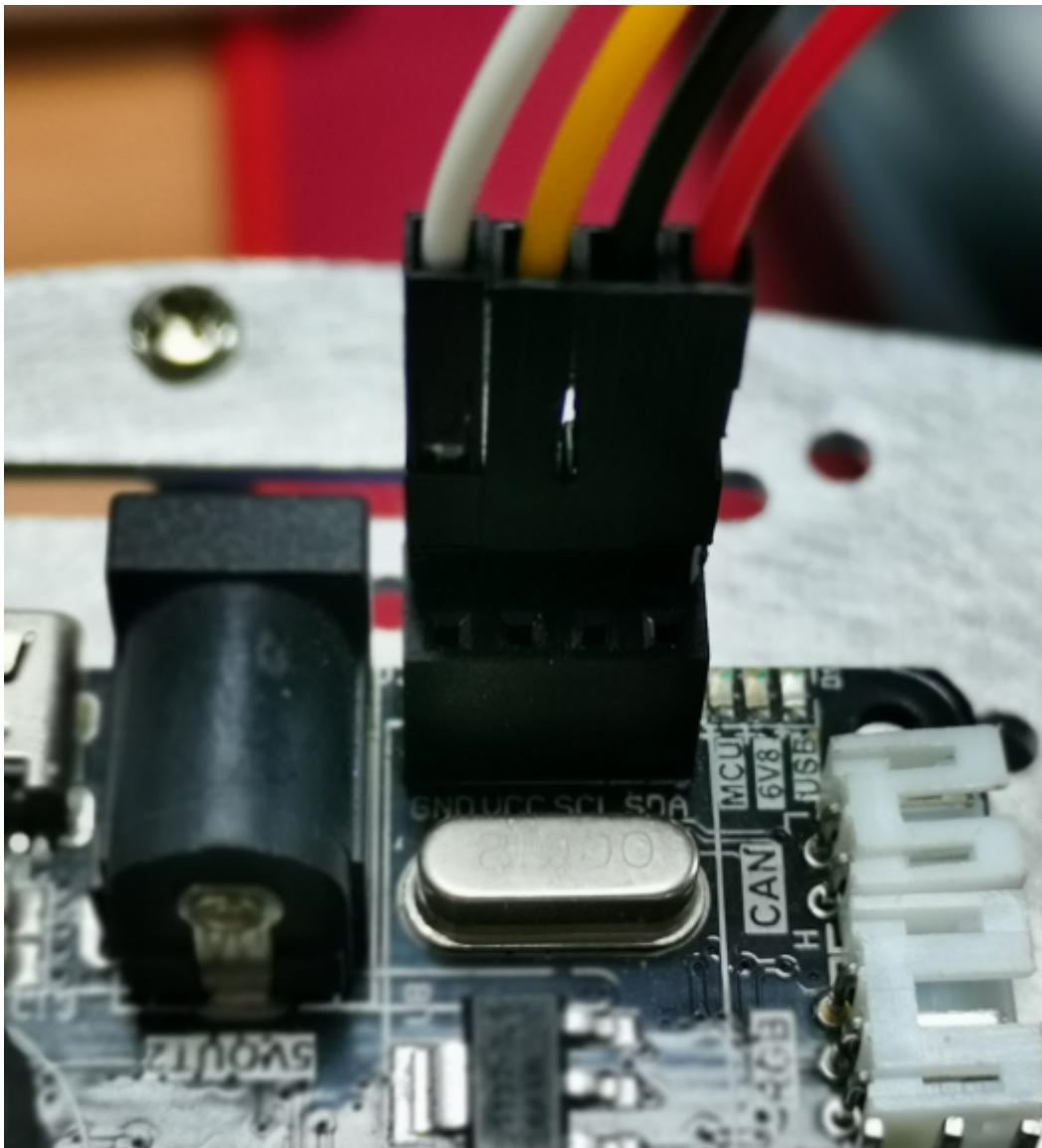


K210 module with the ROS expansion Board connected to the line sequence as shown below:

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.



3.2 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\mnist_car.py

3.3 experimental operation

1. ROS expansion Board to burn the firmware: ROS-CAR. hex
2. The bottom plate of the motor connected to the ROS expansion Board, in accordance with the M1 is connected to the Left front motor M2 is connected to the left rear of the motor, the M3 is connected to the right front of the motor, the M4 is connected to the right rear of the motor.
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.

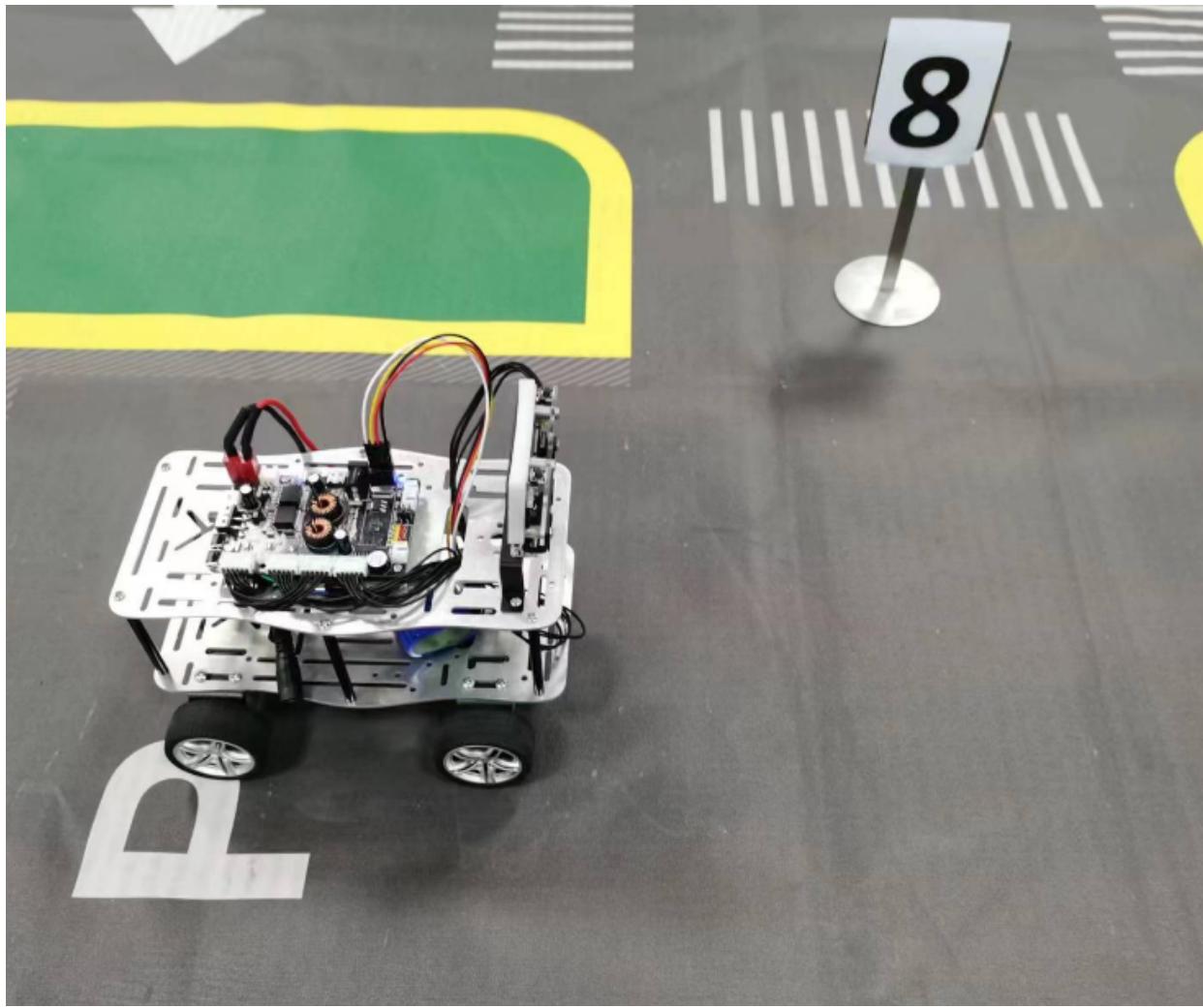
4. Open CanMV IDE mnist_car.py code download into the K210 module.
5. The K210 module via the 4PIN cable is connected to the ROS expansion Board.
6. The trolley into the white background, breaking move K210 module bracket to a suitable angle, turn the car switch.

3.4 experimental results

Wait for the system initialization is complete, the car will start forward movement of a distance, and then stop, the analysis of the current camera picture if there is a figure 8, if the detection of the other numbers, please put the `ACTION_NUM = 8` Modify for other numbers can be.

If three consecutive detection to the number 8, then began to run reversing into the library function.





Reversing into the library features a total of divided into two parts, according to different maps to make changes.

```
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 the experiments are summarized

Trolley handwriting action figure features the main achievement of the car identification number reverse into library functions, since the car is reversing into the library operation for fixed operation, so the map changes after re-commissioning in speed and run time in order to achieve optimal results.