1 automatic driving

1.1 the experimental description

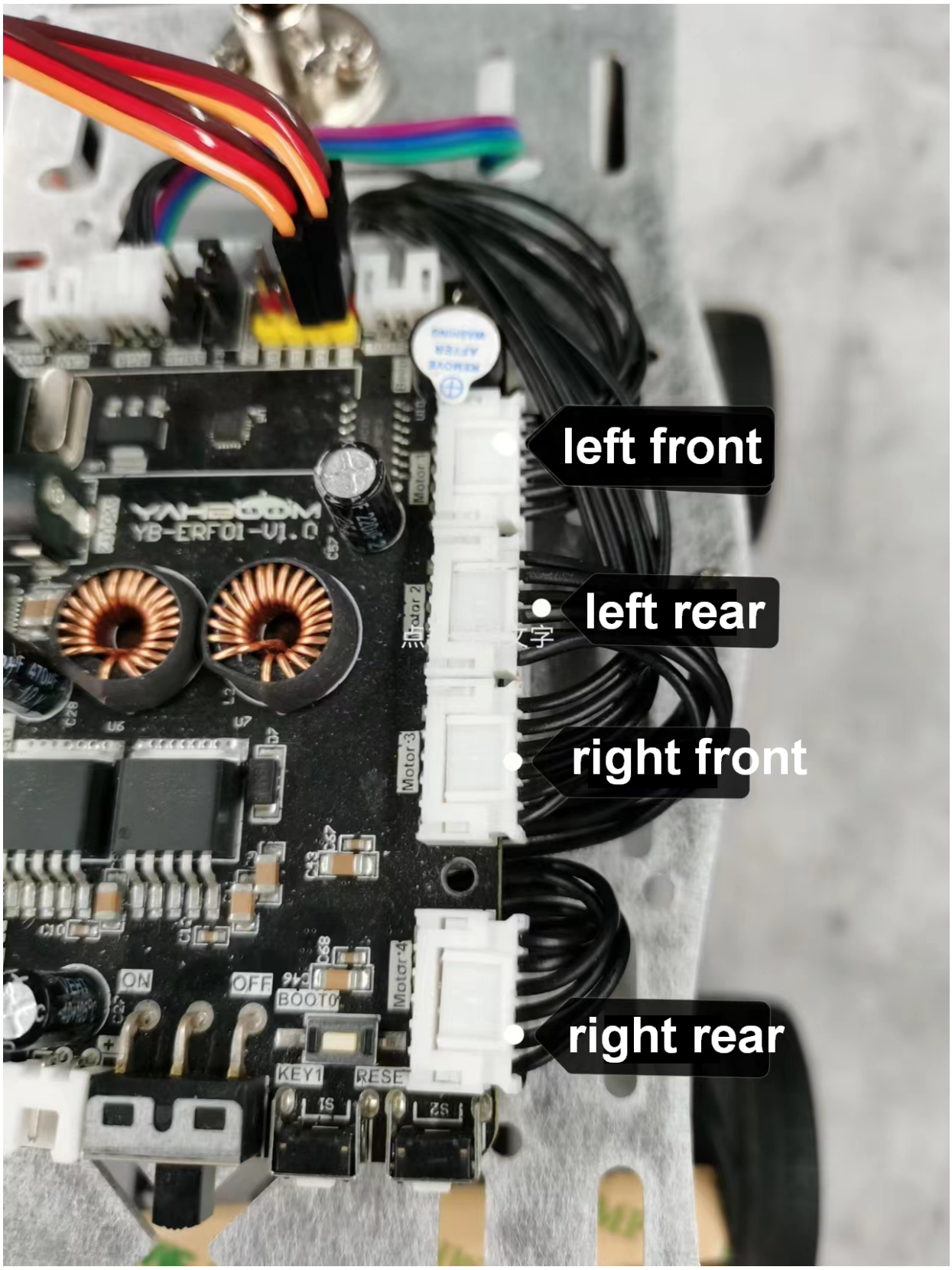
The present experiment was to belong to expand the class of experiments, the need to match other external devices to use, here to the car chassis and the ROS expansion Board is not part of K210 module kit contents, so the present experimental results are for reference only, if there is no corresponding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

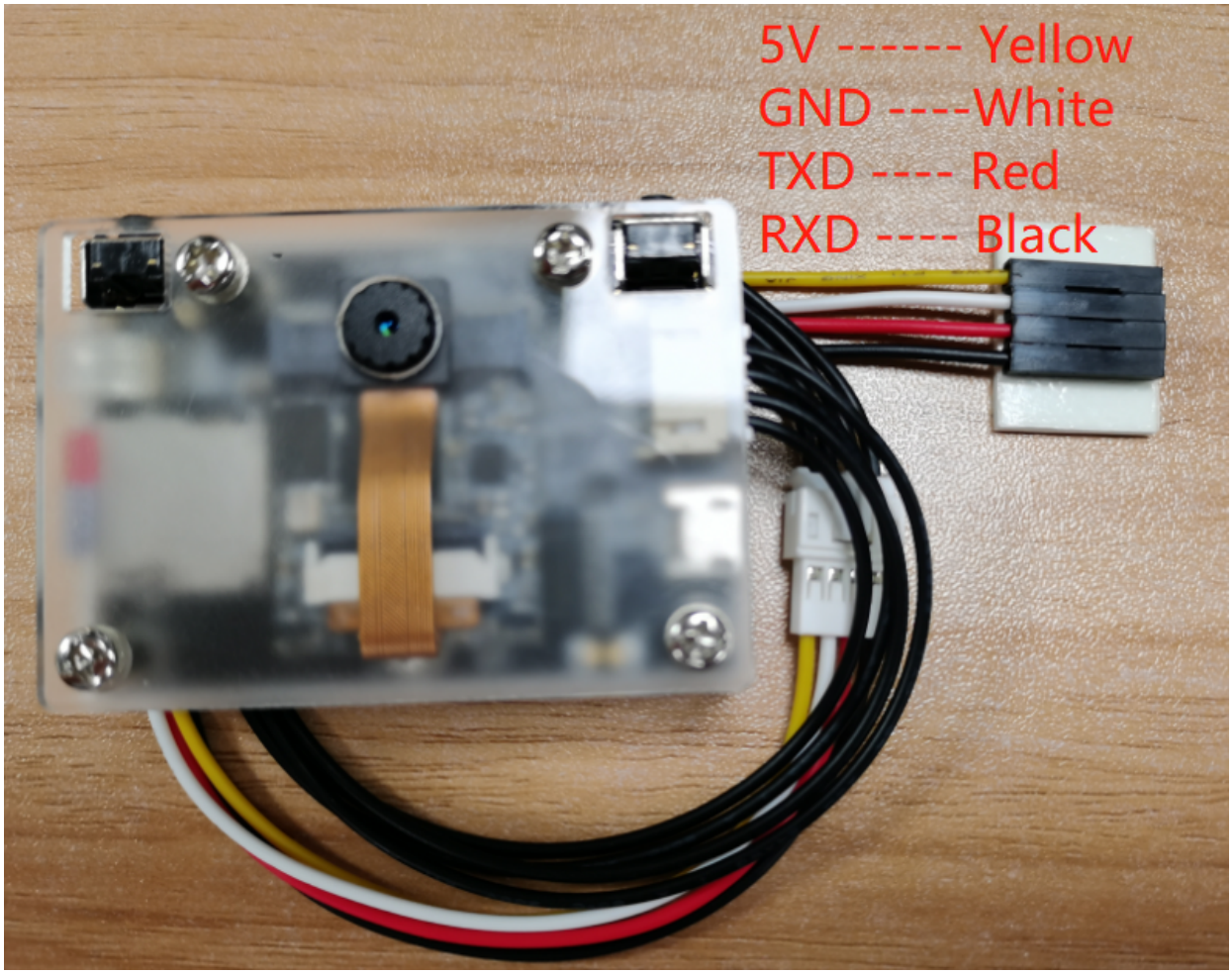
Trolley motor connected to the line as shown below:

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.

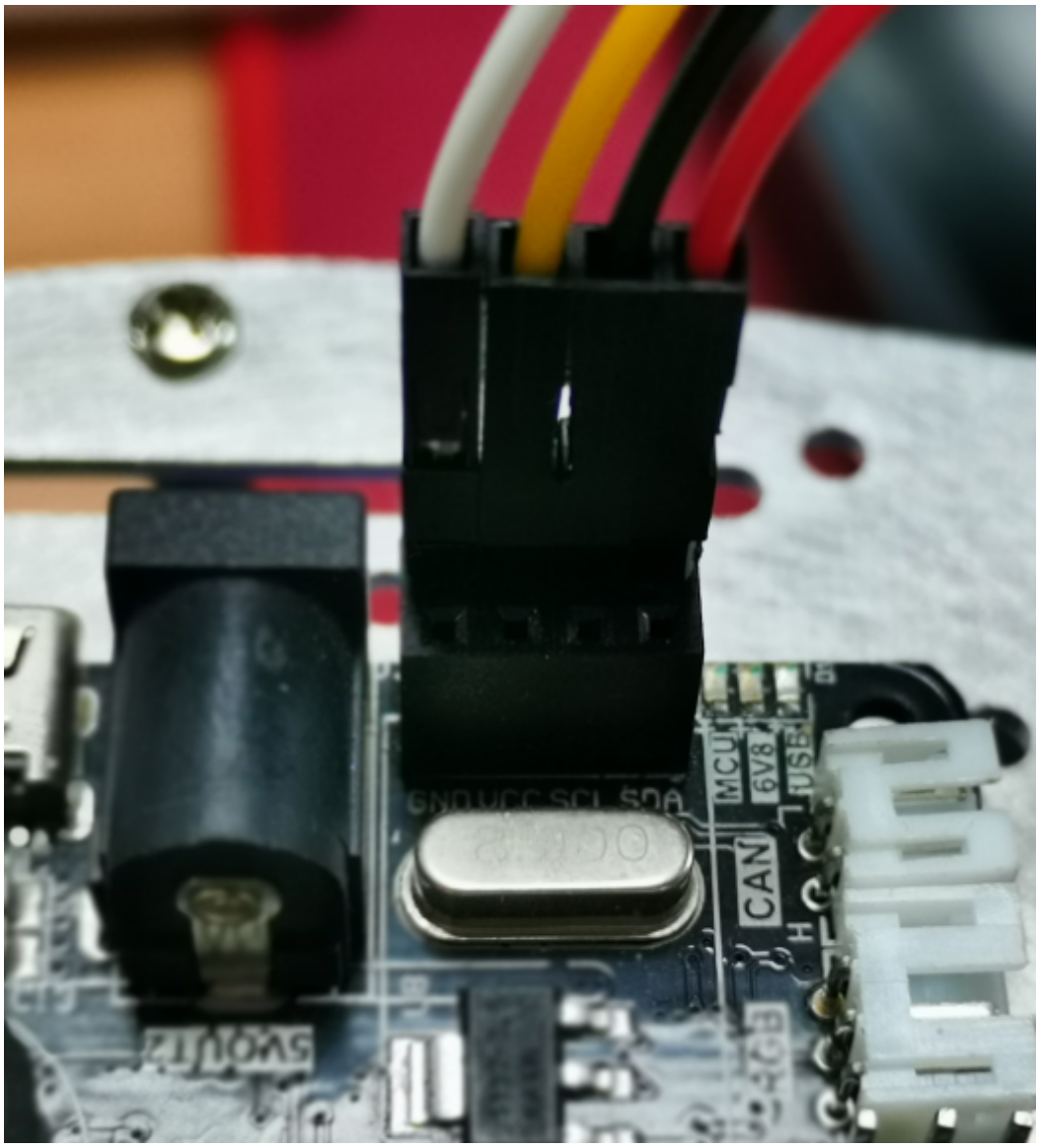


K210 module with the ROS expansion Board connected to the line sequence as shown below:

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.



1.2 the experimental goals

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\follow_line.py

1.3 experimental operation

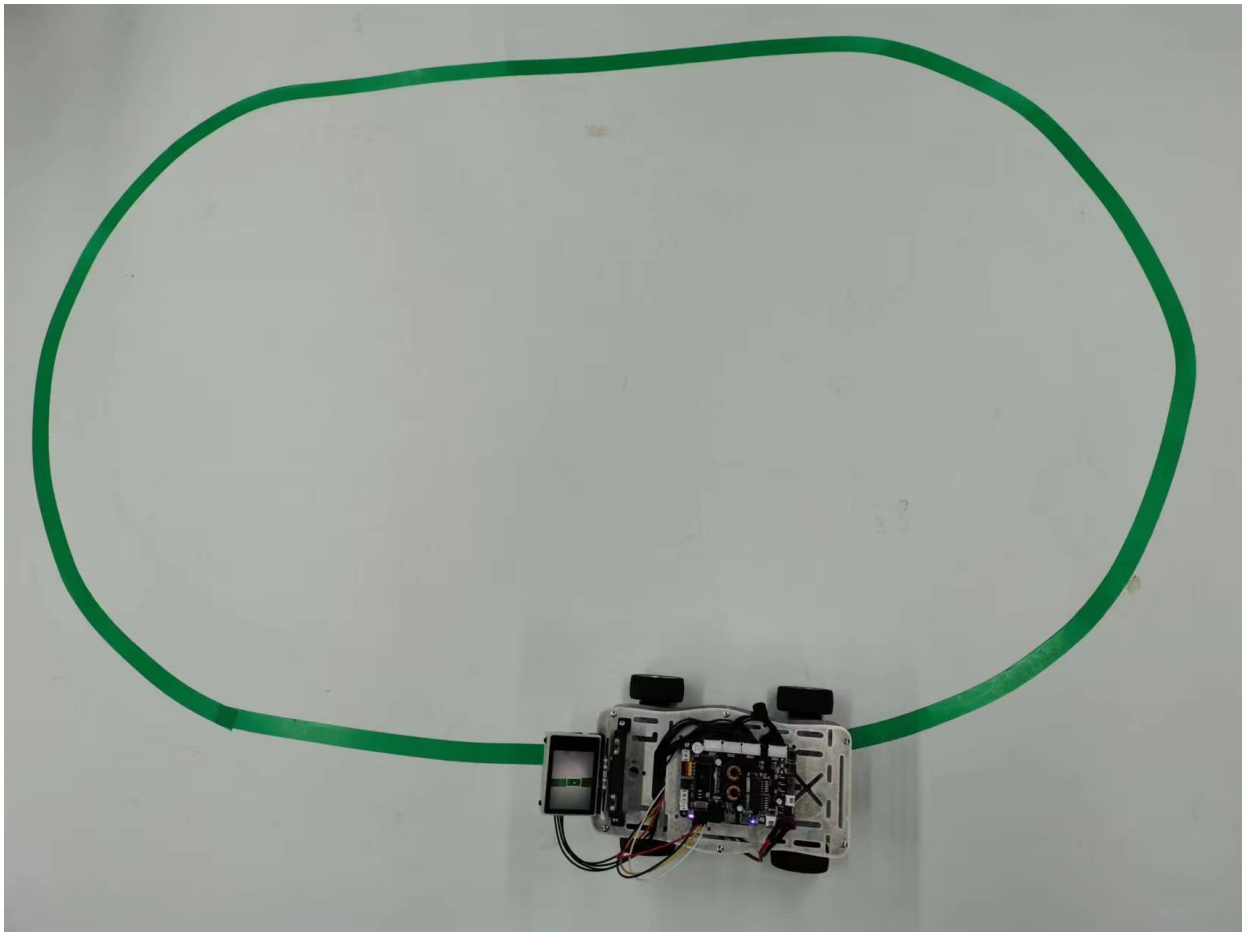
1. ROS expansion Board to burn the firmware: ROS-CAR. hex
2. The bottom plate of the motor connected to the ROS expansion Board, in accordance with the M1 is connected to the Left front motor M2 is connected to the left rear of the motor, the M3 is connected to the right front of the motor, the M4 is 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 follow_line. 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 a visual inspection of the line in the map, breaking off action K210 module bracket to a suitable angle, turn the car switch.
7. Wait for the system initialization is complete, the LCD displays the camera screen, and the middle of the screen there is a white box, move the trolley, to identify the color to fill the White block, waiting for the white box turns green, then start to capture color, the acquisition is complete the green box disappears, start to run the program.

1.4 experimental results

Wait for the system initialization is complete, the LCD displays the camera screen, and the middle of the screen there is a white square box to identify the color into the white box, waiting for the white box turns green, then start to capture color, the acquisition is complete the green box disappears, start to run the program.

The car is going along just the green box to identify the color of forward motion, as shown in the figure is a circular arc, the trolley along a circular motion.



If midway found the car often unable to patrol the line, please view the trolley and then various locations are to be able to identify to the corresponding color, and then if the car is the car of the reaction is too slow or too quick question, adjustable appropriate adjustments FollowLinePID value.

follow_line.py

```
import sensor, image, time, lcd

from modules import ybserial
from robot_Lib import Robot
from simplePID import PID

#FollowLinePID = (22, 0, 2)
FollowLinePID = (15, 0, 2)
SCALE = 1000.0

```

1.5 the experiments are summarized

Color identification of the main function is to analyze the color of the LAB values, the first to identify the color of the placed within the box, then the system will be based on Block read to the color LAB values, and then with the camera to capture the color of the LAB values as the analysis of the comparison, if they meet the requirements of the paintings out of the box represents the identification to the color, and the recognition to the color of the transmission of the location information to the PID controller calculation, determine the recognized color with the trolley in the middle of the offset based on the offset amount to modify the car forward direction, so as to achieve the trolley visual line patrol function.