2、color follows

2.1、Experiment description

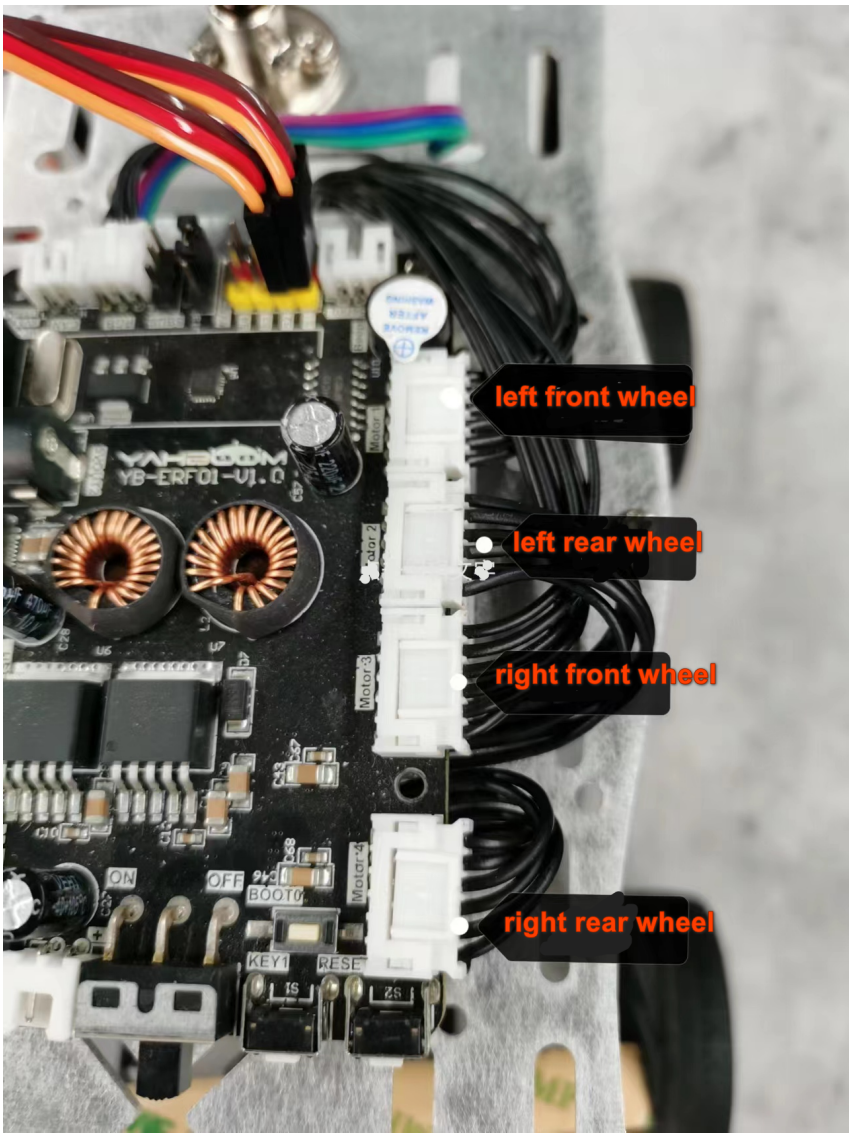
Note: This experiment is an expansion experiment and needs to be used with other external devices. The car chassis and ROS expansion board used here are not part of the K210 development board kit, so the effect of this experiment is for reference only. If there is no corresponding device, it cannot be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

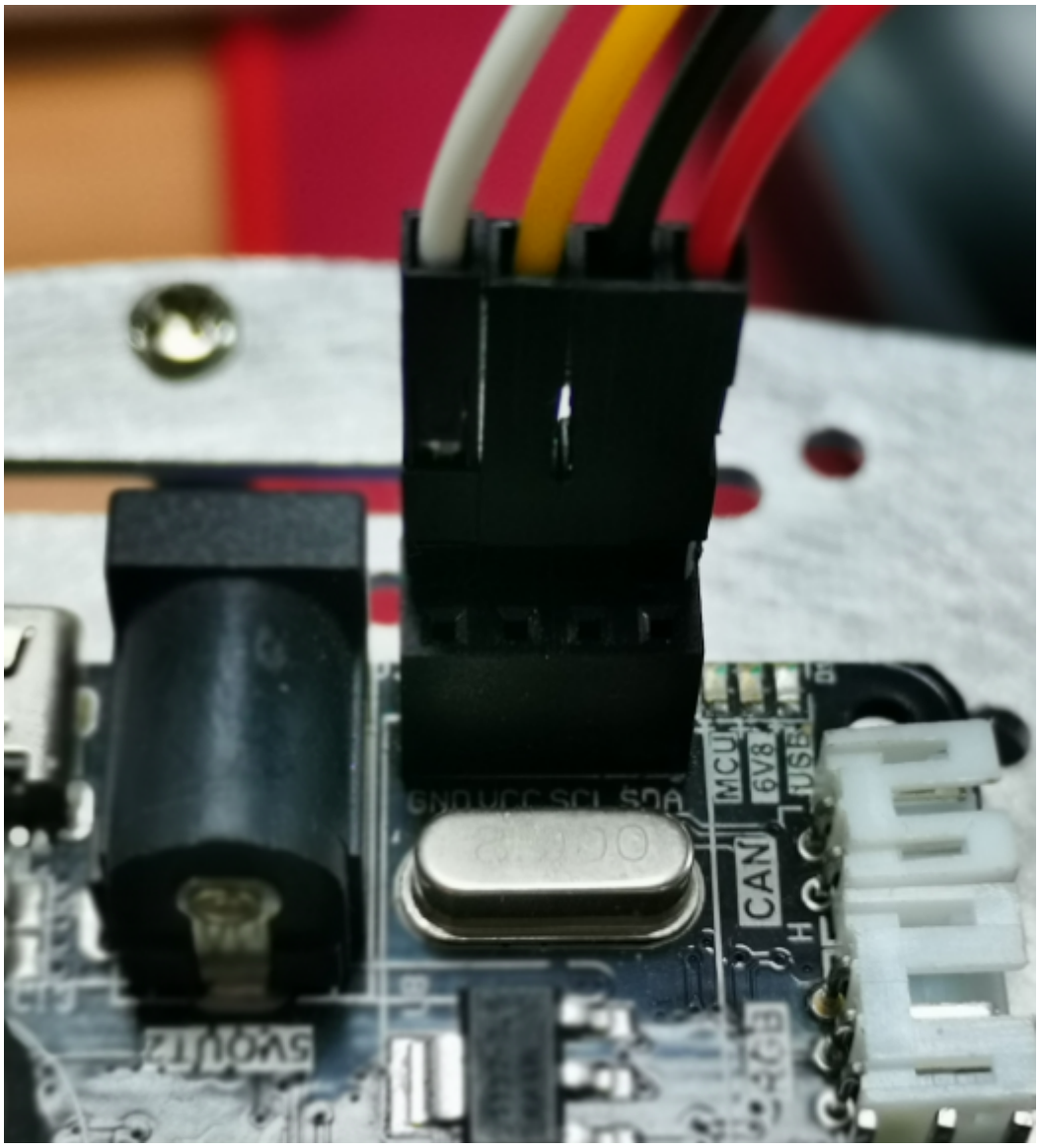
The motor Motor 1 is connected to the left front wheel, the motor Motor 2 is connected to the left rear wheel, the motor Motor 3 is connected to the right front wheel, and the motor Motor 4 is connected to the right rear wheel.



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.



2.2、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\follow_color.py

2.3、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. Connect the K210 development board to the ROS expansion board through the 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. After the system initialization is complete, the LCD will display the camera screen, and there is a white box in the middle of the screen. Please move the car to fill the white box with the color to be recognized. Wait for the white box to turn green to start collecting colors. After the collection is completed, the green box Disappears and starts running the program.

2.4、Experimental effect

After the system initialization is completed, the LCD will display the camera screen, and there is a white box in the middle of the screen. Please put the color to be recognized in the white box, and wait for the white box to turn green to start collecting colors. After the collection is completed, the green box disappears. Start running the program.

The car will move with the color identified by the green box in the previous step. When the color is far away from the car, the car moves forward. When the color is close to the car, the car moves backward. When the color moves to the left/right, the car also moves to the left/right, keeping the recognized color in the middle of the screen Location.

If you find that the car often cannot follow the object, please place a solid color object on a white background to follow the car. If the car responds too slowly or too fast, you can adjust the values of PID_x and PID_y appropriately.

follow_color.py

```
import sensor, image, time, lcd

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

PIDx = (30, 1, 3)
PIDy = (5, 0, 2)
SCALE = 1000.0

PID_x = PID(
    160,
    PIDx[0] / 1.0 / (SCALE),
    PIDx[1] / 1.0 / (SCALE),
    PIDx[2] / 1.0 / (SCALE))

PID_y = PID(
    120,
    PIDy[0] / 1.0 / (SCALE),
    PIDy[1] / 1.0 / (SCALE),
    PIDy[2] / 1.0 / (SCALE))
```

2.5、Experiment summary

The color following of the car mainly relies on obtaining the position of the color in the image after identifying the color, and then judging the offset from the center point of the screen according to the center point of the position, combined with the PID algorithm, to calculate whether the car should move forward, backward, or turn left Turn right and other states to keep the following color in the middle of the screen.