9 PTZ tracking face

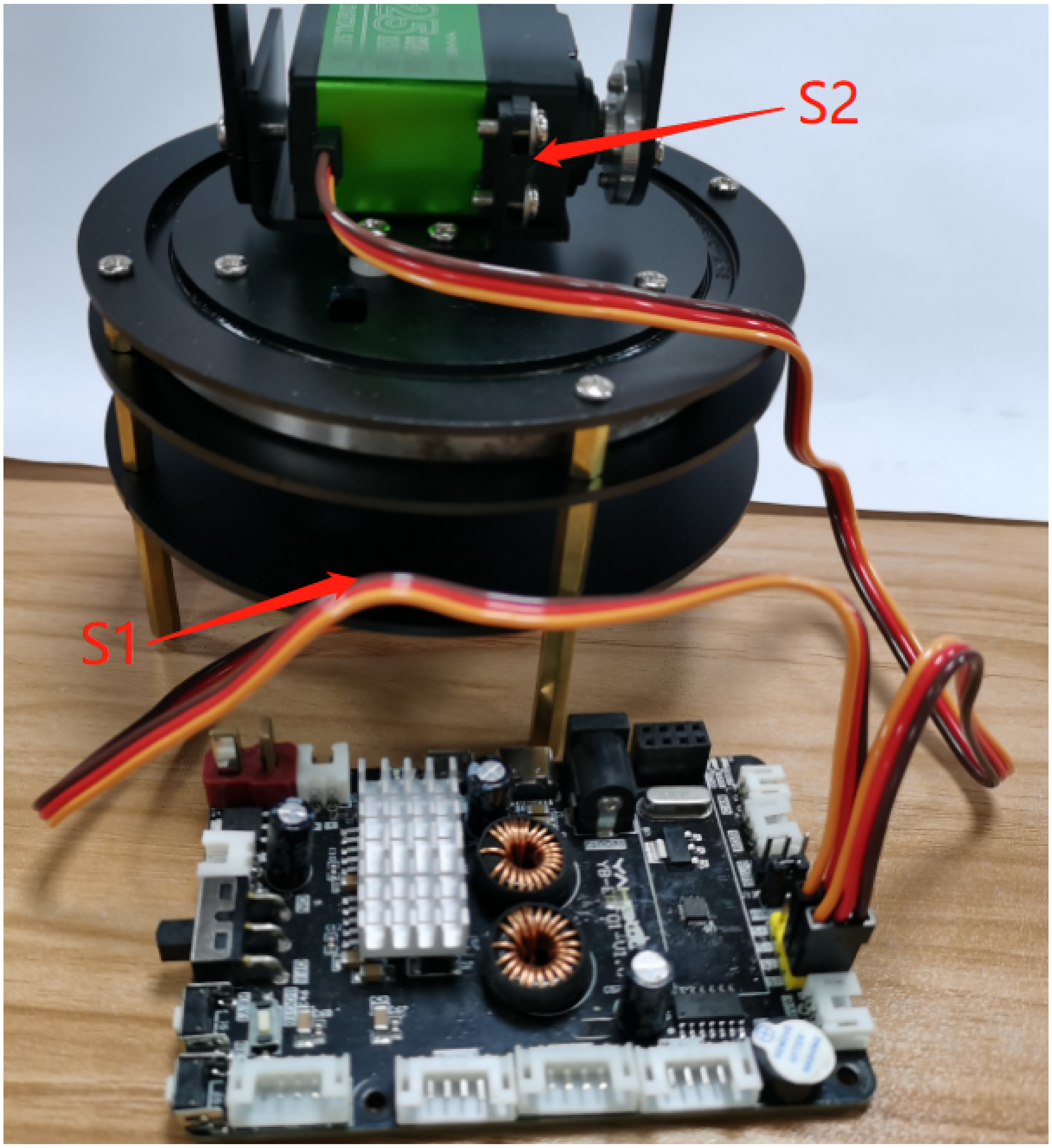
9.1 experiment description

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

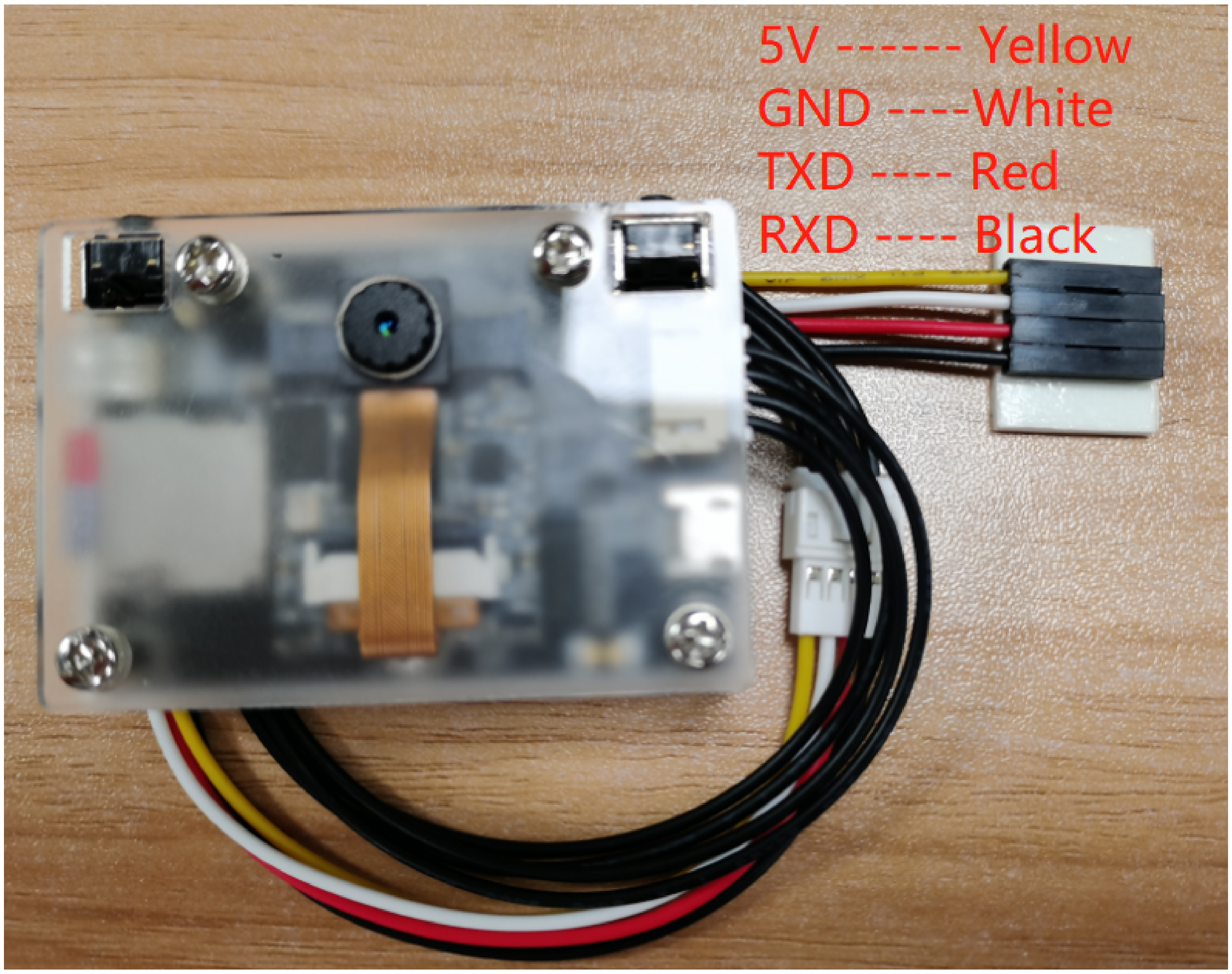
The gimbal servos connected to the ROS expansion Board on the corresponding interface:

S1 connected to the X-axis servos, S2 is connected to the Y-axis servos. Yellow for signal, red for VCC, black for GND.

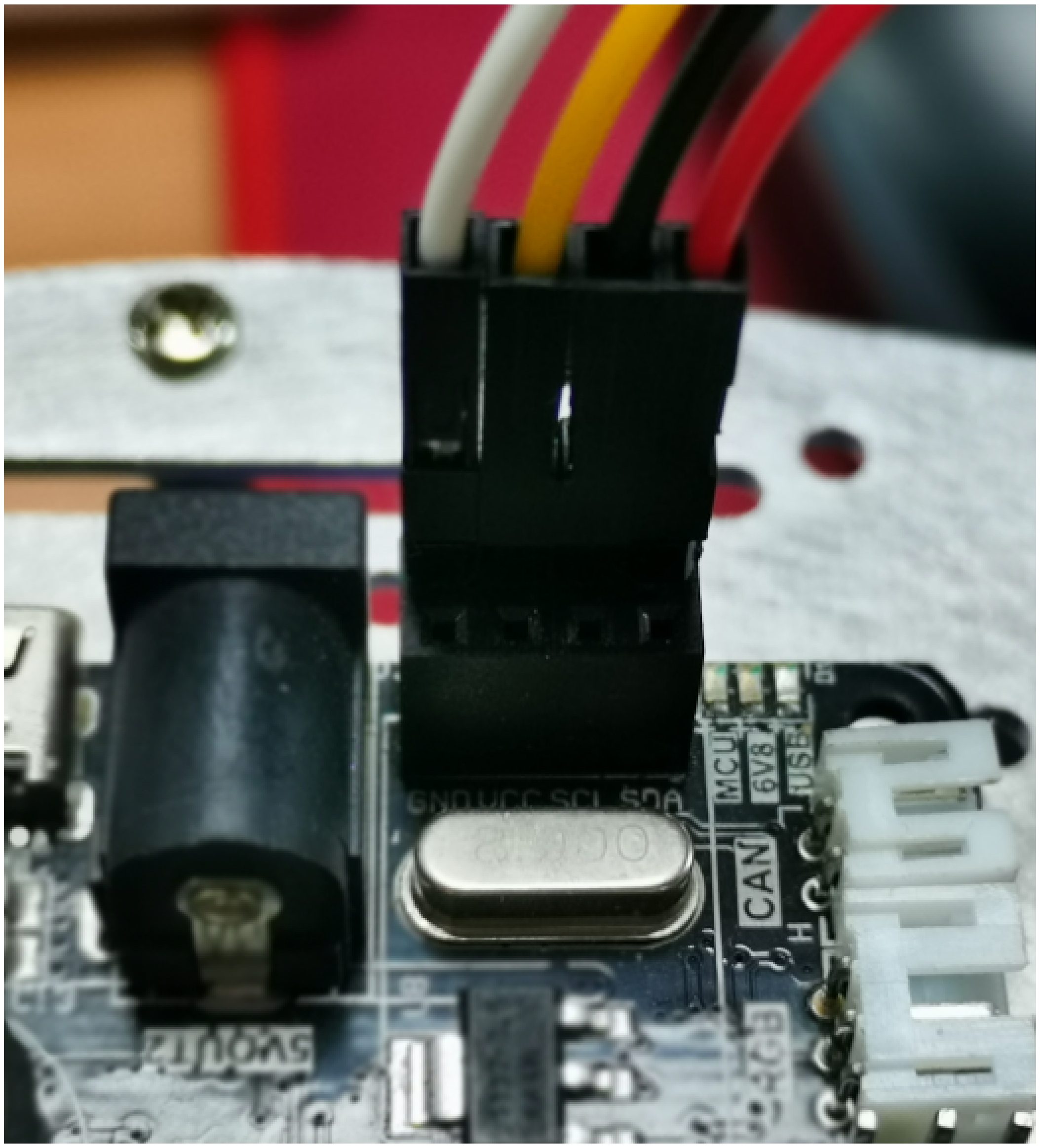


K210 module with the ROS expansion Board connected to the line sequence as shown below:

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.



9.2 experimental target

This lesson is mainly learning K210 module with the car chassis to do a visual inspection of the line features.

The reference code path for this experiment is： CanMV\06-export\tracking_face.py

9.3 experimental operation

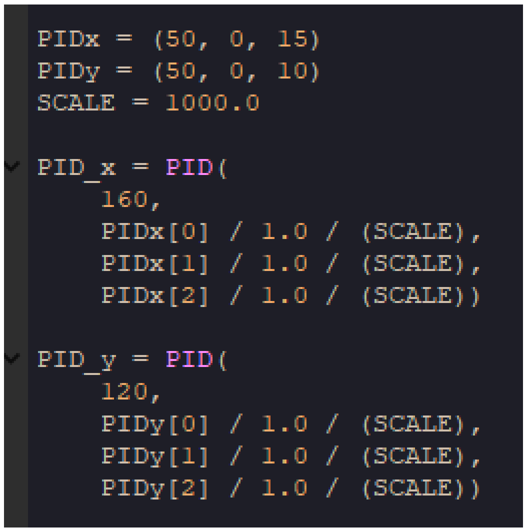
1. ROS expansion Board to burn the firmware: ROS-CAR. hex
2. Insert the RGB light strip into the RGB light interface of the ROS expansion board.
3. Please CanMV\06-export\library directory under the trolley driver library and PID control Library in advance to download to the memory card root directory.
4. Open the CanMV IDE and open the tracking_face.py code and download it to the K210 module.
5. The K210 module via the 4PIN cable is connected to the ROS expansion Board.

6. The gimbal is placed on a white or black background, and then open the ROS expansion Board to the power supply.
7. The face enters the collection range of the K210 module camera, the K210 module will frame the face, and the pan-tilt will track the face.

9.4 experimental results

After the system initialization is completed, the camera of the K210 module will detect whether there is a face in the picture in real time. If there is a face in the picture, modify the Angle of the pan and tilt steering gear to keep the face in the middle of the screen as far as possible, and the camera will track the rotation of the face.

If the tracking of the reaction is too fast or too slow, it may be appropriate to modify the program of PID parameters.



```
PIDx = (50, 0, 15)
PIDy = (50, 0, 10)
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))
```

9.5 the experiments are summarized

The pan-tilt-top face tracking game is developed based on the face detection function. The K210 module camera detects the position coordinates of the face, and calculates the position of the pan-tilt-top need to move through the PID algorithm, so that the pan-tilt-top can track the face in front of the camera. Due to the frame rate and recognition limit, the face movement cannot be too fast, otherwise the pan-tilt may not keep up with the reaction.