

3. Voice-controlled trolley patrol line autopilot

Before running the program, you need to bind the port number of the voice board and the port number of the ROS expansion board in the host computer, you can refer to the previous chapter for binding; when you enter the docker container, you need to mount this voice board to be able to recognize the voice board in the docker container.

1. Program function description

After the program starts, it says "Hello, Xiaoya" to the module, and the module replies "Yes" to wake up the voice board, and then it can say "Patrol red line/green line/blue line/yellow line" to it, and then it will recognize the color and load the processed image screen, and then it will load the processed image screen. After receiving the command, the program recognizes the color and loads the processed image screen, then press the R2 key of the handle to start the program. The car will start moving along the route of the recognized color, and in the process of cruising the line automatically, it will also "beep, beep, beep" the buzzer and stop when it meets the obstacles.

2. Program code reference path

After entering the docker container, the source code of this function is located at.

```
/root/yahboomcar_ros2_ws/yahboomcar_ws/src/yahboomcar_voice_ctrl/yahboomcar_voice_ctrl/Voice_Ctrl_follow_line_a1_R2.py  
/root/yahboomcar_ros2_ws/yahboomcar_ws/src/yahboomcar_voice_ctrl/yahboomcar_voice_ctrl/Voice_Ctrl_follow_line_4ROS_R2.py
```

View the source code according to the actual lidar purchased.

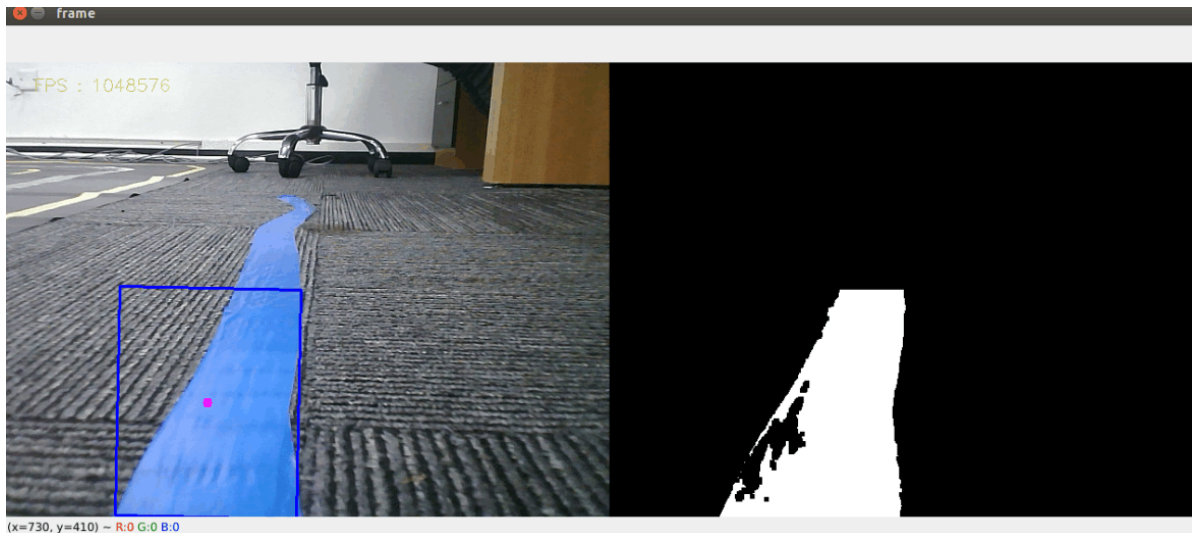
3. Program startup

3.1. Start command

After entering the docker container, according to the actual model and lidar model, terminal input, the

```
#启动语音巡线程序 A1雷达  
# Activate voice patrol program A1 lidar  
ros2 run yahboomcar_voice_ctrl Voice_Ctrl_follow_line_a1_R2  
#启动语音巡线程序 4ROS2雷达  
#Start voice patrol program 4ROS2 lidar  
ros2 run yahboomcar_voice_ctrl Voice_Ctrl_follow_line_4ROS_R2  
#启动手柄控制节点  
#Start the joystick control node  
ros2 run yahboomcar_ctrl yahboom_joy_R2  
ros2 run joy joy_node  
#启动A1雷达 # Activate A1 lidar  
ros2 launch sllidar_ros2 sllidar_launch.py  
#启动4ROS雷达 # Activate 4ROS lidar  
ros2 launch ydlidar_ros2_driver ydlidar_launch.py  
#启动小车底盘 #Start the trolley chassis  
ros2 run yahboomcar_bringup Ackman_driver_R2
```

Break the car's camera down so that it can see the line, then first wake up the module ("Hello, Xiaoya"), get a reply to patrol the blue line can be said to it "patrol the blue line".



Press the R2 key of the handle to start the patrol.

3.2. Adjusting HSV values

The hsv value loaded in the program is not used in all scenes, as we all know, the light on the results of image processing will be quite big, so if the loaded hsv value is not good for image processing, then you need to re-calibrate the value of hsv, the method of calibration is as follows.

- Run the line patrol program and the dynamic parameter adjuster.

```
ros2 run yahboomcar_linefollow follow_line_a1_R2.py
ros2 run rqt_reconfigure rqt_reconfigure
```

- Press the r key on the keyboard, enter the color selection mode, in the need to patrol the line of the place, box out a region, and then, click reconfigure_GUI interface interface of the interface of the blank interface, you will find that the value of the HSV inside the value of the change, the value of these values, corresponds to the Voice_Ctrl_follow_line_a1_R2.py of self.hsv_range, and be careful not to mix up the colors.
- Finally, after modifying the Voice_Ctrl_follow_line_a1_R2.py code, you need to go back to the yahboomcar_ws directory and use **colcon build** to compile and **source install/setup.bash**.

4. Core code

Here with the previous "eleven, opencv series course -5, patrol line autopilot" patrol line of the same principle, are through the center of the processed image coordinates of the calculation of speed, the only extra part is through the voice to load the value of hsv, before the need to manually use the mouse to select the color of the line, while the voice module, we only need to follow the command, load the hsv value. We only need to load the corresponding hsv value according to the command, the core content is as follows.

```
def process(self, rgb_img, action):

    binary = []
    rgb_img = cv.resize(rgb_img, (640, 480))

    if self.img_flip == True: rgb_img = cv.flip(rgb_img, 1)
    #这里开始接收语音命令以及发布指令和加载hsv的值
```

```

# Start receiving voice commands here as well as issuing commands and loading
hsv values
self.command_result = self.spe.speech_read()
self.spe.void_write(self.command_result)
if self.command_result == 23:
    self.model = "color_follow_line"
    print("red follow line")
    #红色HSV #Red HSV
    self.hsv_range = [(0, 84, 131), (180, 253, 255)]
    .....
#以下部分就是把hsv的值传进去，图像处理，得到一个self.circle的值，最后传入self.execute的函数，计算速度
# The following part is to pass in the value of hsv, the image is processed to
get a value of self.circle, and finally the function self.execute is passed in to
compute speed
if self.model == "color_follow_line":
    rgb_img, binary, self.circle = self.color.line_follow(rgb_img,
self.hsv_range)
    if len(self.circle) != 0:
        threading.Thread(target=self.execute, args=(self.circle[0],
self.circle[2])).start()

```

The command word correspondences for this section of the course are as follows.

command word	Speech Recognition Module Results
Turn off line patrols	22
patrol a red line	23
patrol green line	24
patrol blue line	25
patrol yellow line	26