

# Voice control robot dog tracking

---

## 1. Programme function description

After the program starts, say "Hi, Yahboom" to the module, and the module answers "Hi, I'm here" to wake up the voice module.

Then you can say "track red line" (Or track green line, track blue line, track yellow line).

After receiving the command, the robot dog recognizes the color, loads the processed image.

Next, we can press the L2 key on the handle to start the program. The car will start moving along the color recognition route, and during the automatic driving process of line patrol, it will stop when encountering obstacles.

## 2. Program code reference path

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

```
/home/pi/cartographer_ws2/src/voice_xgo_ctrl_run/voice_xgo_ctrl_run/voice_xgo_follow_line.py
```

## 3. Programme startup

### 3.1 Start command

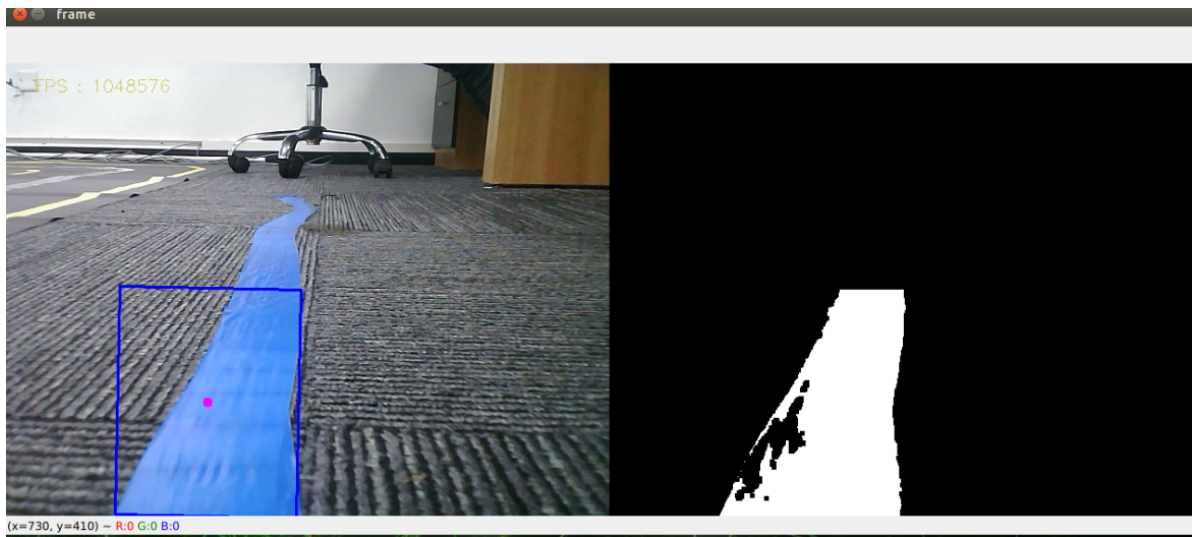
Connect the mechanical dog via vnc and enter it in the terminal:

```
#启动底盘 #Start the chassis
sudo systemctl start YahboomStart.service
cd cartographer_ws2
source install/setup.bash
#启动语音巡线程序 # Start the voice patrol programme
ros2 run voice_xgo_ctrl_run voice_follow_line
```

Reopen a terminal and type in the terminal:

```
cd /home/pi/cartographer_ws2
source install/setup.bash
#启动手柄控制节点 #Start the joystick control node
ros2 run yahboom_ctrl yahboom_joy
ros2 run joy joy_node
```

Break the car's camera down so that it can see the line, then wake up the module ("Hello, Xiaoya"), and when you get a reply, you can, for example, patrol the blue line, say "patrol the blue line" to it.



By default the handle does not post speed messages you need to press start to be able to control it with the joystick. In this routine, the joystick is not used by default.

Press the L2 key of the joystick twice, when the following code appears in the interface, the mechanical dog starts to act.

```
pi@yahboom:~/cartographer_ws2$ ros2 run voice_xgo_ctrl_run voice_follow_line
import finish
cv_edition: 4.6.0
Speech Serial Opened! Baudrate=115200
start it
0
Cancel color_follow_line
25
bule follow line
2
2
0
Cancel color_follow_line
25
bule follow line
按下了 std_msgs.msg.Bool(data=True)
按下了 std_msgs.msg.Bool(data=False)
color_radius: 29 Joy_active: False
0.18
color radius: 29 Joy active: False
```

### 3.2. Adjusting the HSV value

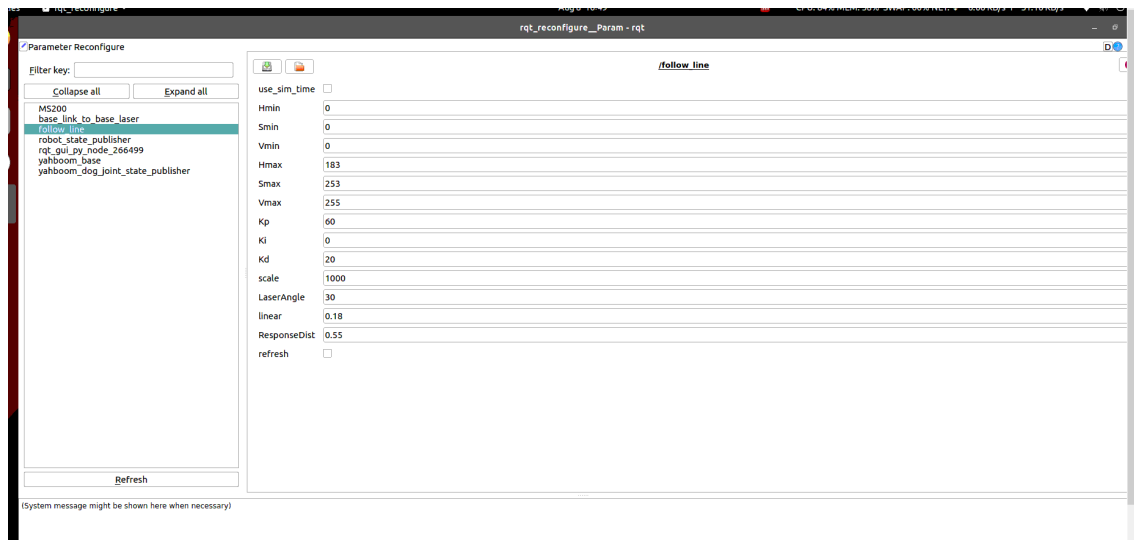
The hsv value loaded in the programme is not used for all scenes, as we all know, the light on the results of image processing will also be quite large, so if the loaded hsv value is not good to deal with the results of image processing, then you need to recalibrate the value of the hsv, the method of calibration is as follows.

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

```
cd cartographer_ws2
source install/setup.bash
ros2 run voice_xgo_ctrl_run follow_line
ros2 run rqt_reconfigure rqt_reconfigure
```

- Mouse click to the camera interface, press the r key on the keyboard, enter the colour selection mode, in the place where you need to patrol the line, box out an area, then, click reconfigure\_GUI interface interface of the interface of the blank interface, you will find that the value of the HSV inside the change, put them these values, corresponding to the

voice\_xgo\_follow\_line.py self. hsv\_range in voice\_xgo\_follow\_line.py and modify it, pay attention to the colours and don't confuse them.



- Finally, after modifying the voice\_xgo\_follow\_line.py code, you need to go back to the cartographer\_ws2 directory and use **colcon build** for compilation and **source install/setup.bash**.

## 4. Core code

Here is the speed is calculated by the centre coordinate of the processed image, and the value of hsv is loaded by voice de-loading, we just need to load the value of the corresponding hsv 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
    calculate 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 |
|---------------------|-----------------------------------|
| Close tracking mode | 22                                |
| track red line      | 23                                |
| track green line    | 24                                |
| track blue line     | 25                                |
| track yellow line   | 26                                |