

## 8、HD camera Autopilot

---

### 8、HD camera Autopilot

#### 8.1、Introduction

## 8.1.1、Introduction

The Transbot robot HD camera Autopilot can recognize multiple colors at any time, and autonomously storing the currently recognized colors.

According to the color of the detection and recognition, the function of real-time obstacle avoidance can also be realized in the process of Autopilot .

The color tracking of the Transbot robot can also realize the function of real-time HSV regulation. By adjusting the high and low thresholds of HSV, the interfering colors can be filtered out, so that the square can be identified ideally in a complex environment. If the color picking effect is not ideal At this time, we need to move the car to a different environment to calibrate it, so that we can recognize the color we need in a complex environment.

- **HSV**

H: 0 — 180

S: 0 — 255

V: 0 — 255

Part of the red is classified as the purple range here:

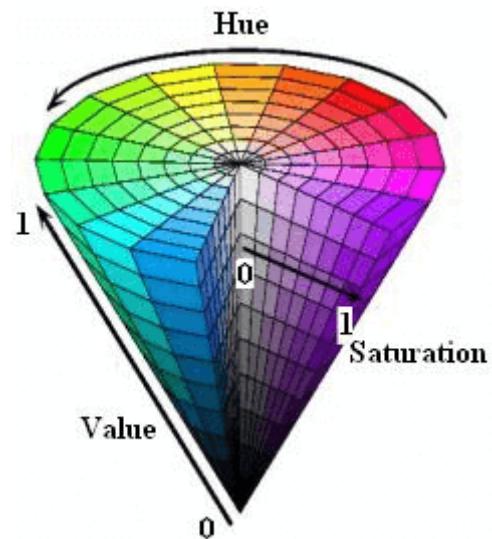
	<b>black</b>	<b>gray</b>	<b>white</b>	<b>red</b>		<b>orange</b>	<b>yellow</b>	<b>green</b>	<b>verdant</b>	<b>blue</b>	<b>purple</b>
<b>hmin</b>	0	0	0	0	156	11	26	35	78	100	125
<b>hmax</b>	180	180	180	10	180	25	34	77	99	124	155
<b>smin</b>	0	0	0	43	43	43	43	43	43	43	43
<b>smax</b>	255	43	30	255	255	255	255	255	255	255	255
<b>vmin</b>	0	46	221	46	46	46	46	46	46	46	46
<b>vmax</b>	46	220	255	255	255	255	255	255	255	255	255

- **HSV**

- **Lightness V**

- **Saturation S**

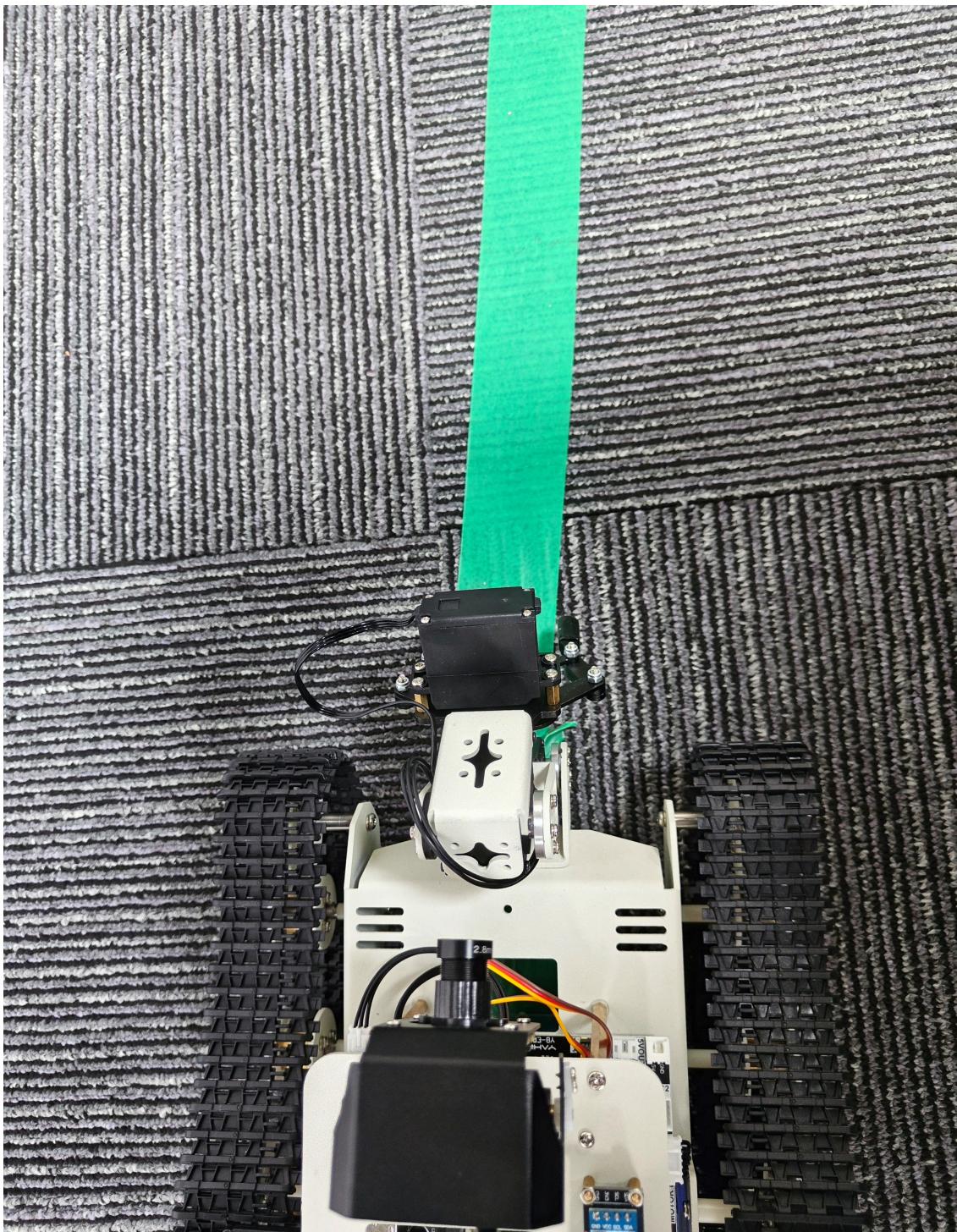
- **Hue H**



- **8.2. Operation steps**

**Note:** [R2] on the remote control handle has the [pause/start] function for all gameplays.

Place the robot into the starting position. as the picture shows



### 8.2.1. Start

#### jetson motherboard/Raspberry Pi

First start the robot driver

```
roslaunch transbot_bringup bringup.launch
```

#### Raspberry Pi 5

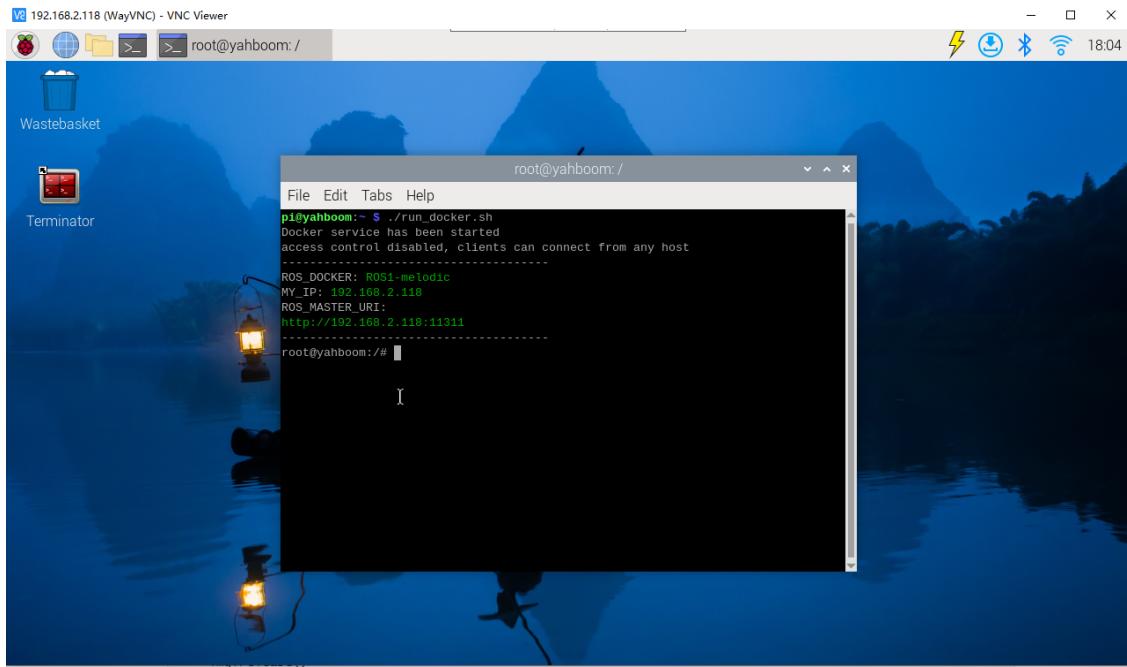
**Before running, please confirm that the large program has been permanently closed**

Enter docker

**Note: If there is a terminal that automatically starts docker, or there is a docker terminal that has been opened, you can directly enter the docker terminal to run the command, and there is no need to manually start docker**

Start docker manually

```
./run_docker.sh
```



Start the robot driver

```
roslaunch transbot_bringup bringup.launch
```

method one

### jetson motherboard/Raspberry Pi 4B

(Robot side) Start the monocular camera

```
roslaunch usb_cam usb_cam-test.launch
```

(Virtual machine) Start automatic driving control

```
roslaunch transbot_linefollow follow_line.launch videoSwitch:=False
```

To achieve remote control in this way under the same LAN, multi-machine communication needs to be configured.

### Raspberry Pi 5

Enter the same docker from multiple terminals

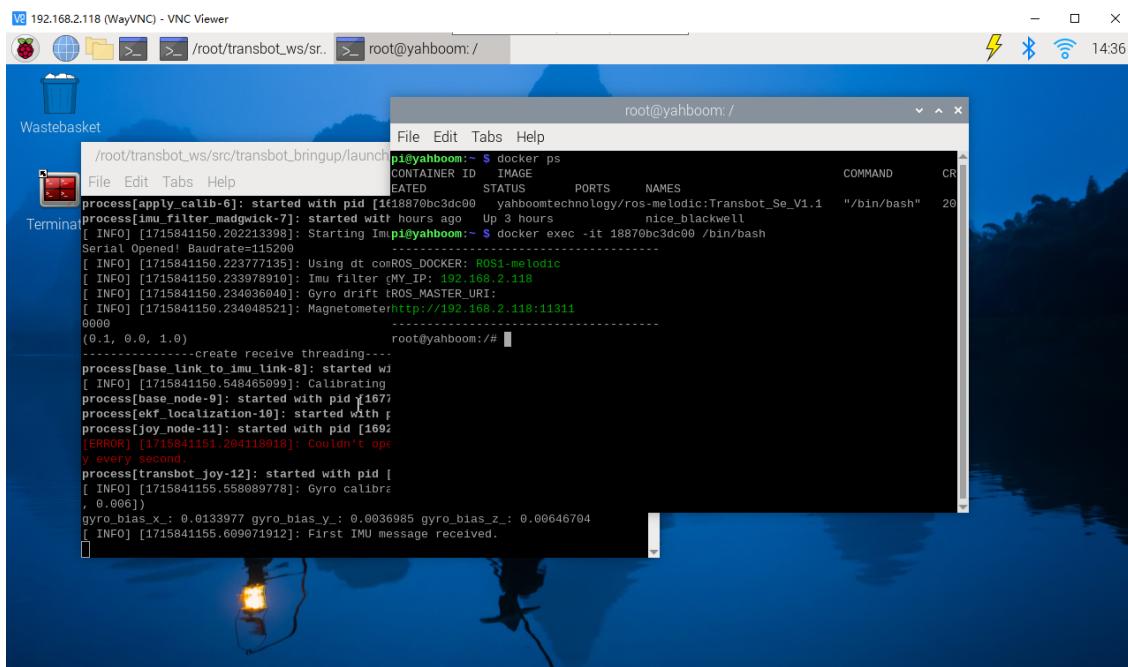
Keep the program of the previous docker terminal running and open a new terminal

Enter the following command

```
docker ps
```

Enter the same docker and use the following 18870bc3dc00 to modify the ID displayed on the actual terminal.

```
docker exec -it 18870bc3dc00 /bin/bash
```



(Robot side) Start the monocular camera

```
roslaunch usb_cam usb_cam-test.launch
```

(Virtual machine) Start automatic driving control

```
roslaunch transbot_linefollow follow_line.launch videoSwitch:=False
```

## Method 2

(robot side)

**Note: [q] key to exit.**

**jetson motherboard/Raspberry Pi 4B**

```
roslaunch transbot_linefollow follow_line.launch videoSwitch:=true
```

## Raspberry Pi 5

Enter the same docker from multiple terminals

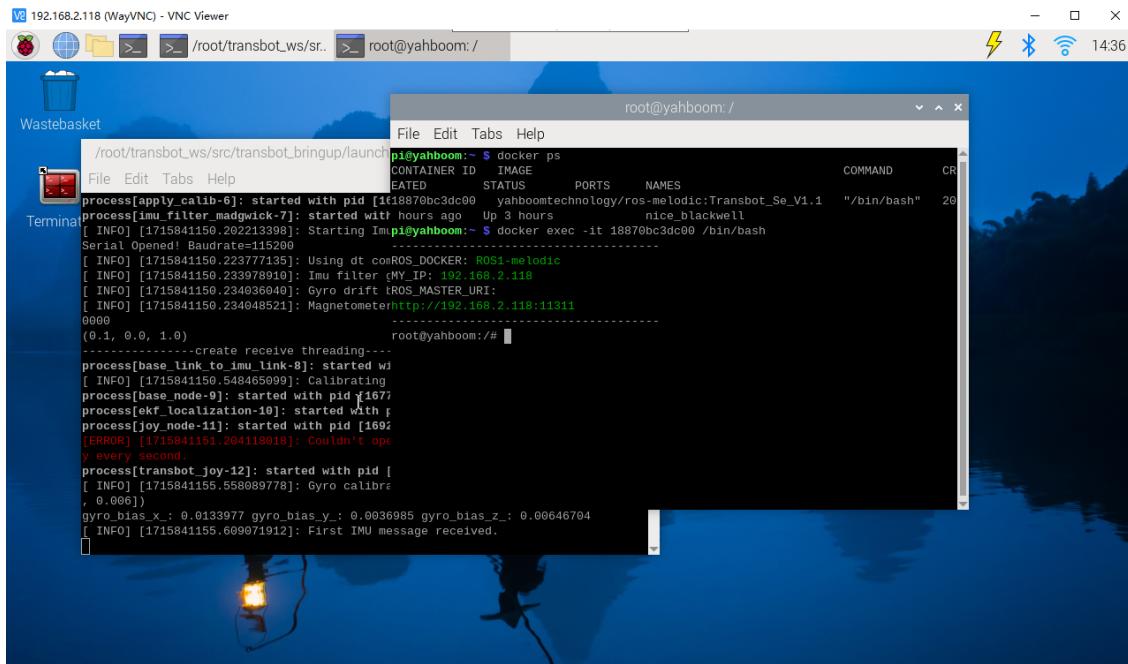
Keep the program of the previous docker terminal running and open a new terminal

Enter the following command

```
docker ps
```

Enter the same docker and use the following 18870bc3dc00 to modify the ID displayed on the actual terminal.

```
docker exec -it 18870bc3dc00 /bin/bash
```



(robot side)

```
roslaunch transbot_linefollow follow_line.launch videoSwitch:=true
```

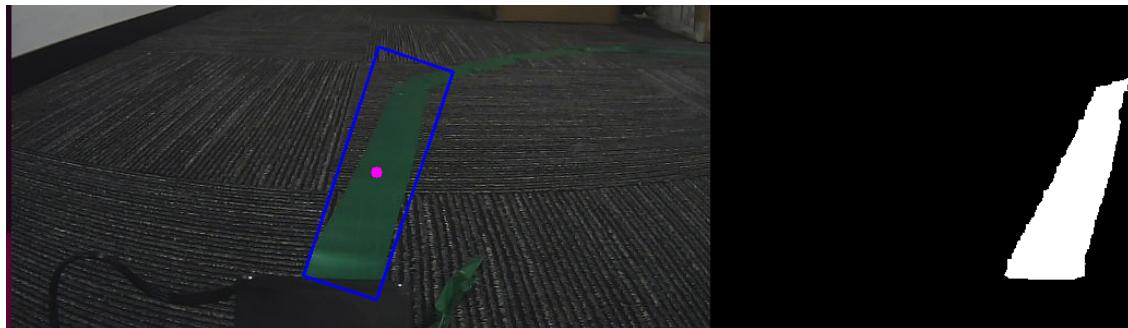
This method can only be started in the main control connected to the camera

- VideoSwitch parameter: whether to use the camera function package to launch; for example: to launch usb\_cam-test.launch, this parameter must be set to True; otherwise, it is False.

Set parameters according to needs, or modify the launch file directly, so there is no need to attach parameters when starting.

### 8.2.2. Identification

After startup, the system defaults to [Target Detection Mode], as shown below:



Keyboard key control:

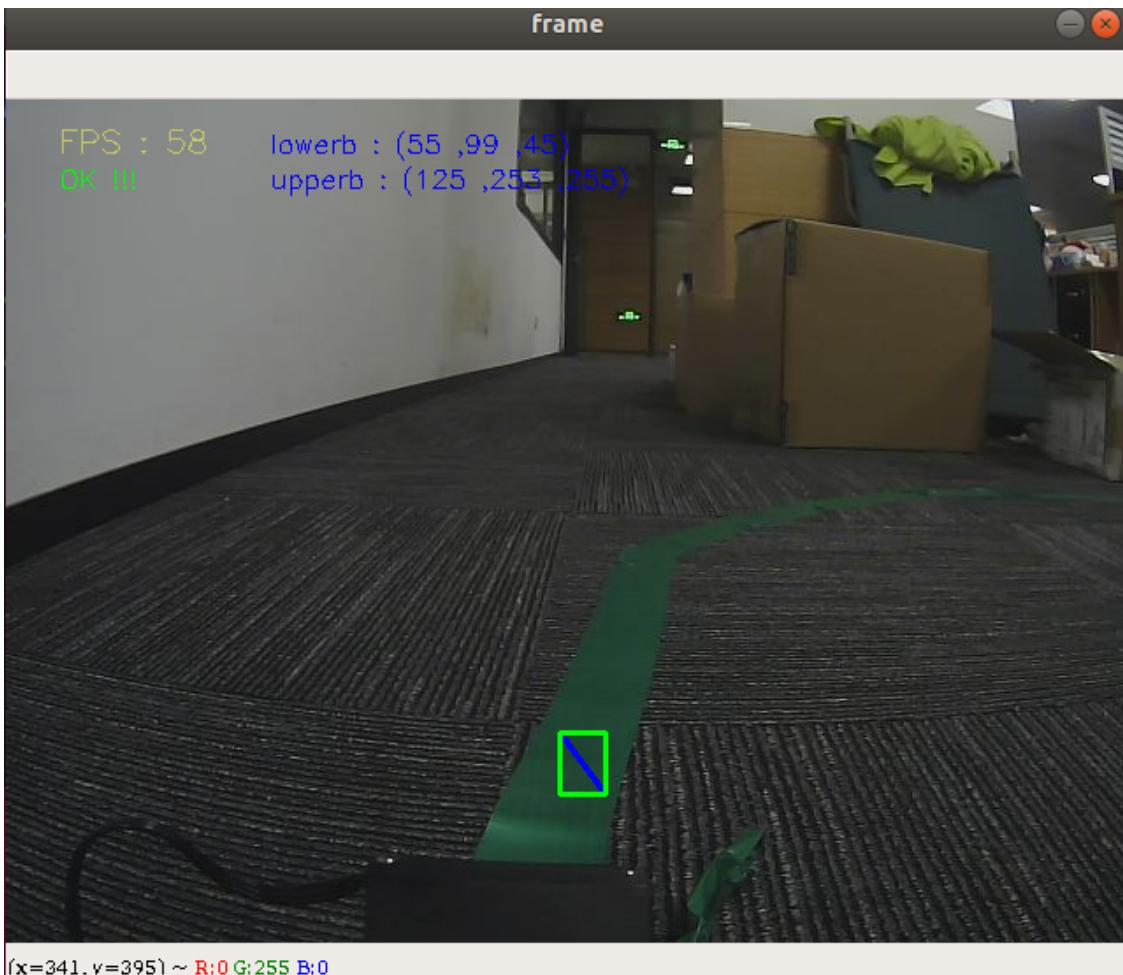
**[r]** : Color selection mode.

**[i]** : Target detection mode. The color image on the left (Color) and the binary image on the right (Binary).

**[q]**: Exit the program.

**[Spacebar]**: Follow the track.

In the color selection mode, you can use the mouse to select the area of the color to be recognized (cannot exceed the area range), as shown in the figure below, release it to start recognition.



### 8.2.3. Color calibration

Dynamic parameter debugging tool

**jetson motherboard/Raspberry Pi 4B**

```
rosrun rqt_reconfigure rqt_reconfigure
```

**Raspberry Pi 5**

Enter the same docker from multiple terminals

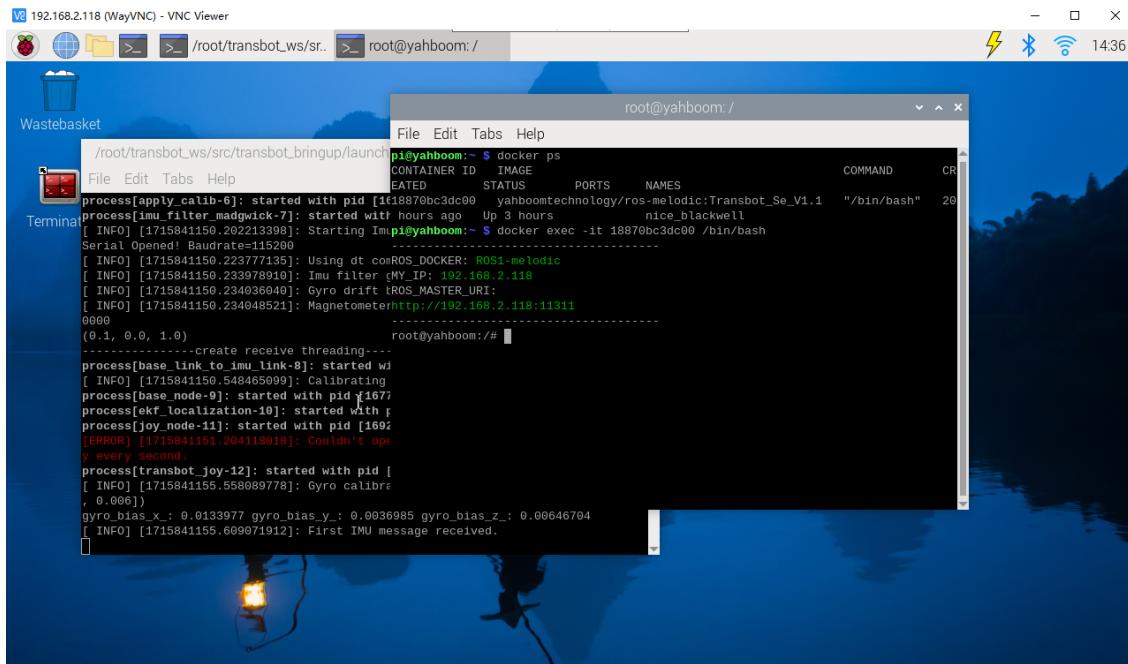
Keep the program of the previous docker terminal running and open a new terminal

Enter the following command

```
docker ps
```

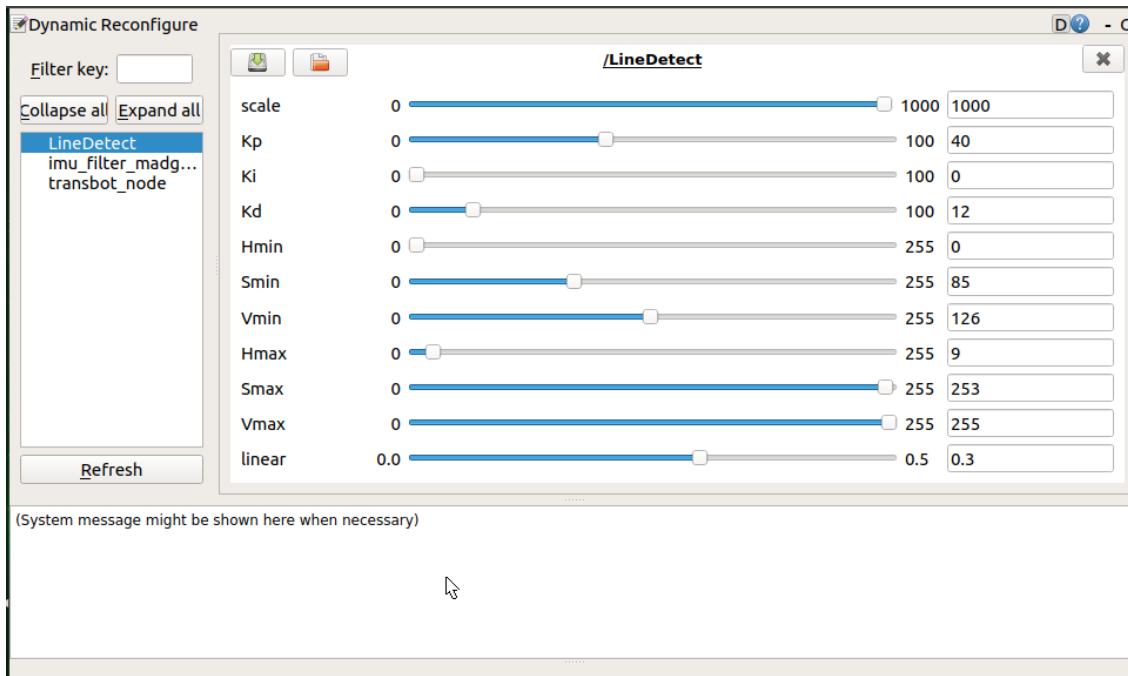
Enter the same docker and use the following 18870bc3dc00 to modify the ID displayed on the actual terminal.

```
docker exec -it 18870bc3dc00 /bin/bash
```



```
rosrun rqt_reconfigure rqt_reconfigure
```

Set the mode to [Target Detection Mode] and start the dynamic parameter debugging tool.



Select the [LineDetect] node. Generally, you only need to adjust [Hmin], [Smin], [Vmin], and [Hmax]. These four parameters can be well identified. The slide bar is always in a dragging state and data will not be transferred to the system until it is released; you can also select a row and then slide the mouse wheel.

Parameter analysis:

[Kp], [Ki], [Kd]: PID control during car driving.

[scale]: PID scaling.

[linear]: Car running speed; range [0, 0.45], unit: meters; set as required.

## 8.2.4. Tracking driving

After identifying that there is no problem, click the [space bar] on the keyboard to execute the tracking program.

- Node view

**jetson motherboard/Raspberry Pi 4B**

```
rqt_graph
```

### Raspberry Pi 5

Enter the same docker from multiple terminals

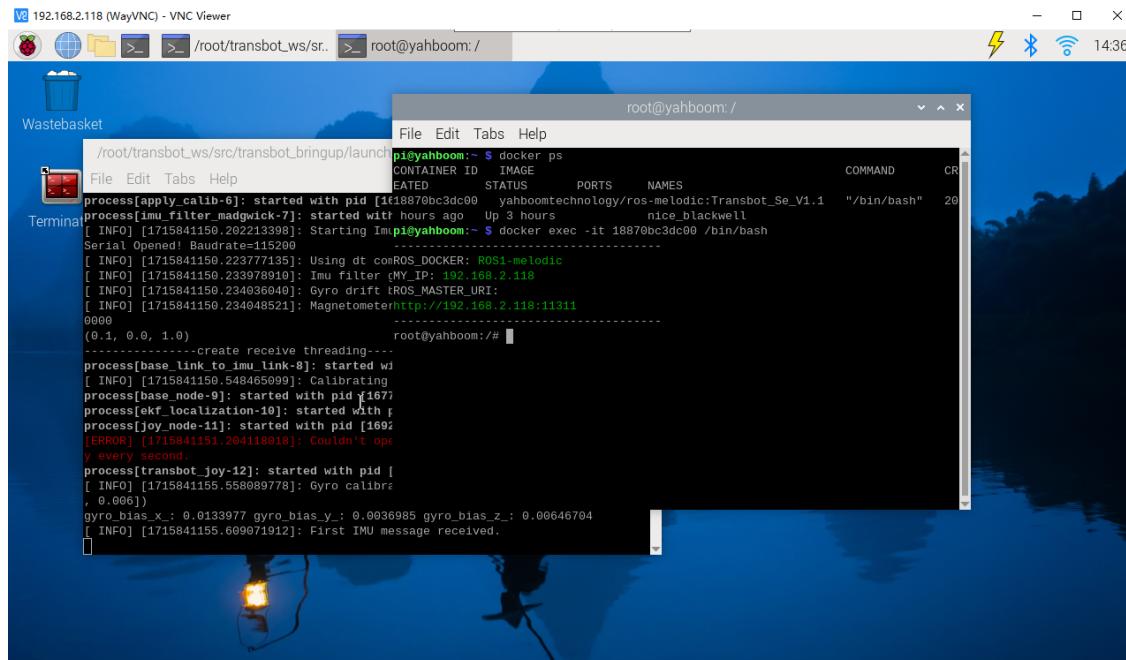
Keep the program of the previous docker terminal running and open a new terminal

Enter the following command

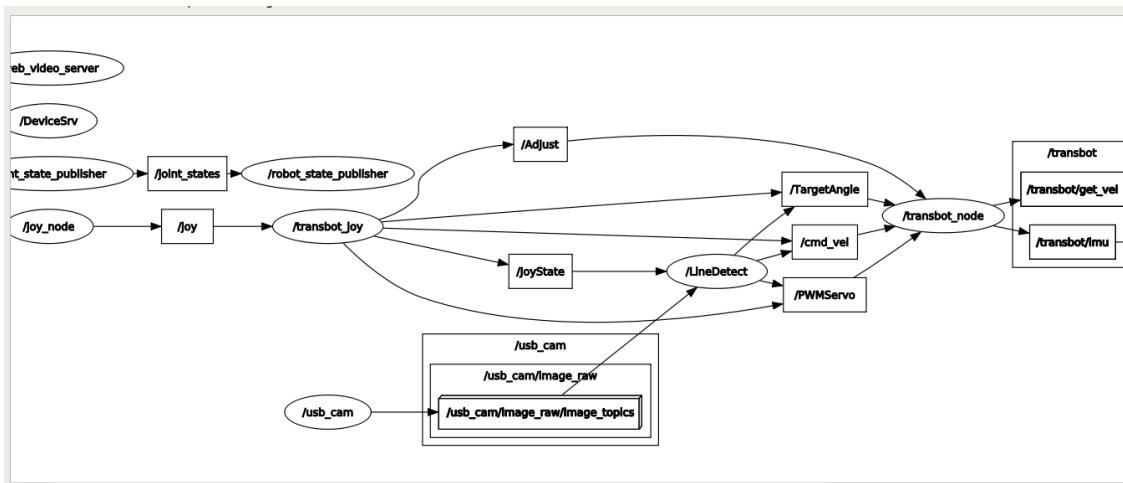
```
docker ps
```

Enter the same docker and use the following 18870bc3dc00 to modify the ID displayed on the actual terminal.

```
docker exec -it 18870bc3dc00 /bin/bash
```



```
rqt_graph
```



### Node 【LineDetect】

- Subscribe
  - images
  - handle
- Publish
  - Car speed
  - PTZ servo
  - Robotic arm