

## **Lesson 2 Go Up and Down Stair**

### **1. Getting Ready**

Place the map on a flat and smooth surface and set the step in the corresponding position.

The specific placement method for the map and tools can refer to the video and tutorial in “Athletics Lessons/Lesson 1 Tools Assembly and Map Placement”.

### **2. Working Principle**

The working principle of this this lesson is as follow:

The color is recognized through Lab color space firstly. Then convert RGB color into Lab color space, and proceed binarization, and dilation and erosion process to obtain the contour of the target color. Mark the target color with blue line, and obtain the coordinate parameter of the target to complete the color recognition.

Next, control the robot to move toward the step. According to the target coordinate to judge, if the robot is in the middle position of the target red line, it will be controlled to move forward and approach the step. When it reaches to the set range, the action of going up and down step will be executed. Otherwise, control robot to move left or right to the middle of the target, and then execute the action of going up and down the step.

The source code of the program is located in  
`/home/pi/TonyPi/Extend/Stairway.py`

```

5 import cv2
6 import time
7 import math
8 import threading
9 import numpy as np
10 sys.path.append('/home/pi/TonyPi/')
11 import hiwonder.Misc as Misc
12 import hiwonder.Board as Board
13 import hiwonder.PID as PID
14 import hiwonder.ActionGroupControl as AGC
15 import hiwonder.yaml_handle as yaml_handle
16
17 if sys.version_info.major == 2:
18     print('Please run this program with python3!')
19     sys.exit(0)
20
21 # 台阶
22
23 go_forward = 'go_forward'
24 go_forward_one_step = 'go_forward_one_step'
25 turn_right = 'turn_right_small_step_a'
26 turn_left = 'turn_left_small_step_a'
27 left_move = 'left_move_20'
28 right_move = 'right_move_20'
29 go_turn_right = 'turn_right'
30 go_turn_left = 'turn_left'
31
32 lab_data = None
33 servo_data = None
34
35 def load_config():
36     global lab_data, servo_data
37
38     lab_data = yaml_handle.get_yaml_data(yaml_handle.lab_file_path)
39     servo_data = yaml_handle.get_yaml_data(yaml_handle.servo_file_path)
40
41 load_config()
42
43 # 初始位置
44 def initMove():
45     Board.setPWMServoPulse(1, 1000, 500)
46     Board.setPWMServoPulse(2, servo_data['servo2'], 500)
47
48 object_left_x, object_right_x, object_center_y, object_angle = -2, -2, -2, 0

```

### 3. Operation Steps

**i** The entered command must pay attention to case sensitivity and space.

1) Turn on the robot and connect to Raspberry Pi desktop with VNC.

2) Click  or press “Ctrl+Alt+T” to open LX terminal.



3) Enter “cd TonyPi/Extend/” and press “Enter” to come to the directory of game programmings.

```
pi@raspberrypi:~ $ cd TonyPi/Extend/
```

4) Enter “python3 Stairway.py” command, and then press “Enter” to start the game.

```
pi@raspberrypi:~ $ cd TonyPi/Extend/  
pi@raspberrypi:~/TonyPi/Extend $ python3 Stairway.py
```

5) If want to exit the game, press “Ctrl+C” in the LX terminal. Please try multiple times if fail to exit.

## 4. Project Outcome

Place the step within the TonyPi Pro’s vision range and it can adjust the position according to the recognized red line, and then slowly move forward and approach the step. When the robot approaches to the middle of step, it will go up and down step.

## 5. Function Extension

### 5.1 Modify Climbing Position

The program defaults that the robot starts going up and down step at a distance of 1000 from the line after the robot detects the red line on the step. If you want the robot to get closer to the stairs before starting to climbing, the data needs to be decreased; If you want the robot to move further away from the step before starting to climbing, the data needs to be increased. This section takes set the distance from the line at position 1020 where the robot starts climbing the stair. You can refer to the following steps to operate:

1) Enter “cd TonyPi/Extend/” and press “Enter” to come to the directory of game programmings.

```
pi@raspberrypi:~ $ cd TonyPi/Extend/
```

2) Enter command “sudo vim Stairway.py” and press “Enter” to open the game programming file.

```
pi@raspberrypi:~ $ cd TonyPi/Extend/  
pi@raspberrypi:~/TonyPi/Extend $ sudo vim Stairway.py
```

3) Find the code framed in the following figure:

```
44 def initMove():  
45     Board.setPWMServoPulse(1, 1000, 500)  
46     Board.setPWMServoPulse(2, servo_data['servo2'], 500)  
47  
48 object_left_x, object_right_x, object_center_y, object_angle = -2, -2, -2, 0  
49 switch = False  
50 # variable reset  
51 def reset():
```

4) Press “i” on keyboard. When “Insert” appears in the lower left corner, which means it has entered the editing mode.

```
52     global object_left_x, object_right_x  
53     global object_center_y, object_angle, switch  
54  
55     object_left_x, object_right_x, object_center_y, object_angle = -2, -2, -  
56     2, 0  
57 def init():  
58     load_config()  
59     initMove()  
60     reset()  
61  
-- INSERT -- 45,1 17%
```

5) Modify “1000” in “Board.setPWMServoPulse(1, 1000, 500)” to “1020”, as the figure shown below:

```
43 # Initial position  
44 def initMove():  
45     Board.setPWMServoPulse(1, 1020, 500)  
46     Board.setPWMServoPulse(2, servo_data['servo2'], 500)  
47  
48 object_left_x, object_right_x, object_center_y, object_angle = -2, -2, -2, 0  
49 switch = False
```

6) After modification, press “Esc” and then enter “:wq” (Please note that the

colon is in front of wq). Then press “Enter” to save and exit the modified content.

```
53     global object_center_y, object_angle, switch
54
55     object_left_x, object_right_x, object_center_y, object_angle = -2, -2, -
56     2, 0
57 def init():
58     load_config()
59     initMove()
60     reset()
61
:wq
```

## 5.2 Modify Recognition Color

When making step, we put the red insulated rubber tape on it. The robot will climb the step after detecting the red line. Therefore, we can stick other colored tapes such as green, and let the robot climb the step after detecting the green line. The specific steps are as follows:

- 1) Refer to the steps (1), (2) and (4) in folder “5.1 Modify Climbing Position”, enter the program editing interface to find the code framed in the following figure:

```
199     img_h, img_w = img.shape[:2]
200
201     # stair
202     object_left_x, object_right_x, object_center_y, object_angle = color_identify
203     (img, img_copy, target_color = 'red')
204     print('stairway', object_left_x, object_right_x, object_center_y, object_angle)
205     # print the position angle parameter
206
207     return img_copy
208
209 if __name__ == '__main__':
```

- 2) Press “i” on keyboard to enter the editing mode. Modify “red” in “object\_left\_x, object\_right\_x, object\_center\_y, object\_angle = color\_identify(img, img\_copy, target\_color = 'red')” to “green”.

```
201     # stair
202     object_left_x, object_right_x, object_center_y, object_angle = color_identify
    (img, img_copy, target_color = 'green')
203     print('stairway',object_left_x, object_right_x, object_center_y, object_angle)
    )# print the posiont angle parameter
204
205
206
207     return img_copy
```

- 3) After modification, refer to step (6) in “5.1 Modify Climbing Position” to save the modified code.