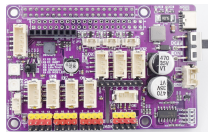



Lesson 12 How to use the Line Tracking Module

In this lesson, we will learn how to use the Line Tracking Module.

12.1 Components used in this Course

Components	Quantity	Picture
Raspberry Pi	1	
Adeept Robot HAT V3.1	1	
Line Tracking module	1	

12.2 The Introduction of the Line Tracking Module

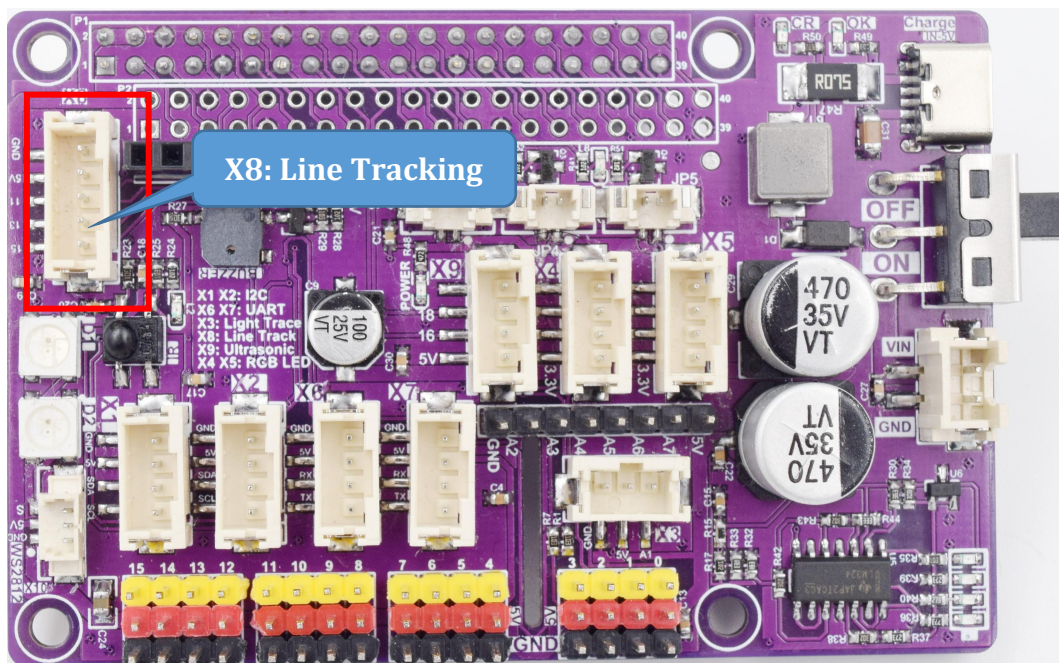
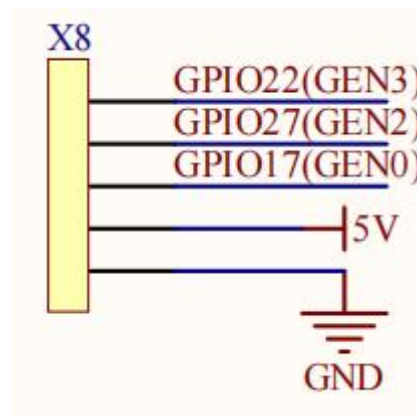
"Line Tracking"--- On the white paper with black lines drawn, the black lines can be judged according to the intensity of the reflected light received due to the different reflection coefficients of the black lines and the white paper to light.

A more common detection method, infrared detection method, is used in the line tracking module. Infrared detection method is to use the characteristics of infrared rays that have different reflection properties on physical surfaces of different colors. During the running of the program, infrared light is continuously emitted to the ground. When the infrared light

encounters the white ground, diffuse emission occurs, and the reflected light is received by the receiving tube; if it encounters a black line, the infrared light is absorbed, and the receiver of the line tracking module Signal cannot be received. The line tracking module we provide is a three-channel infrared tracking module, which contains 3 sets of sensors, each of which consists of an infrared emitting LED and an infrared receiver.

12.3 Wiring Diagram

Figure as below:



12.4 How to use Line Tracking Module

Run the code

1. Remotely log in to the Raspberry Pi terminal.

```
Linux raspberrypi 4.19.118-v7l+ #1311 SMP Mon Apr 27 14:26:42 BST 2020 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat Aug 29 08:17:49 2020 from 192.168.3.208

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set
a new password.

pi@raspberrypi:~ $
```

2. Enter the command and press Enter to enter the folder where the program is located:

```
cd adeept_picar-b2/examples/
```

```
pi@raspberrypi:~ $
pi@raspberrypi:~ $ cd adeept_picar-b2/examples/
pi@raspberrypi:~/adeept_picar-b2/examples $
```

3. View the contents of the current directory file:

```
ls
```

```
pi@raspberrypi:~/adeept_picar-b2/examples $ ls
01_LED.py      03_servo.py  05_RGB.py      07_ultra.py      09_lightTracking.py
02_buzzer.py  04_motor.py  06_ws2812.py  08_lineTracking.py
pi@raspberrypi:~/adeept_picar-b2/examples $
```

4. Enter the command and press Enter to run the program:

```
sudo python3 08_lineTracking.py
```

```
pi@raspberrypi:~/adeept_picar-b2/examples $ sudo python3 08_lineTracking.py
left: 0   middle: 1   right: 1
left: 0   middle: 1   right: 1
left: 0   middle: 1   right: 1
left: 0   middle: 0   right: 1
left: 0   middle: 1   right: 1
left: 0   middle: 1   right: 1
left: 0   middle: 1   right: 1
left: 0   middle: 1   right: 1
left: 0   middle: 0   right: 0
left: 0   middle: 0   right: 1
```

5. After running the program successfully, Try to cover the three detectors of the line tracking module with your hands, and different 0s or 1s will appear on the screen. By detecting three different 0s and 1s, you can determine the approximate position of the line. This module will be covered in more depth in subsequent courses. .

6. When you want to terminate the running program, you can press the shortcut key "[Ctrl + C](#)" on the keyboard.

12.5 Code

```
1. import time
2. import argparse
3. from gpiozero import InputDevice
4.
5. line_pin_left = 22
6. line_pin_middle = 27
7. line_pin_right = 17
8.
9. left = InputDevice(pin=line_pin_right)
10. middle = InputDevice(pin=line_pin_middle)
11. right = InputDevice(pin=line_pin_left)
12.
13. def run():
14.     status_right = right.value
15.     status_middle = middle.value
16.     status_left = left.value
17.     print('left: %d   middle: %d   right: %d' %(status_right,status_middle,status_left))
18.
```

```
19.  
20. if __name__ == '__main__':  
21.     try:  
22.         while 1:  
23.             run()  
24.             time.sleep(0.3)  
25.     except KeyboardInterrupt:  
26.         pass
```