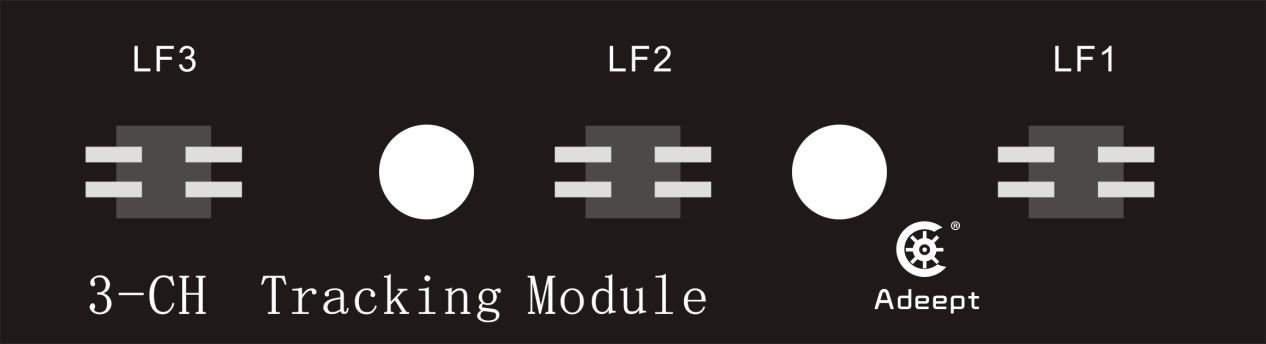
# Lesson 6 How to Use the Tracking Function

**6.1 Tracking module**

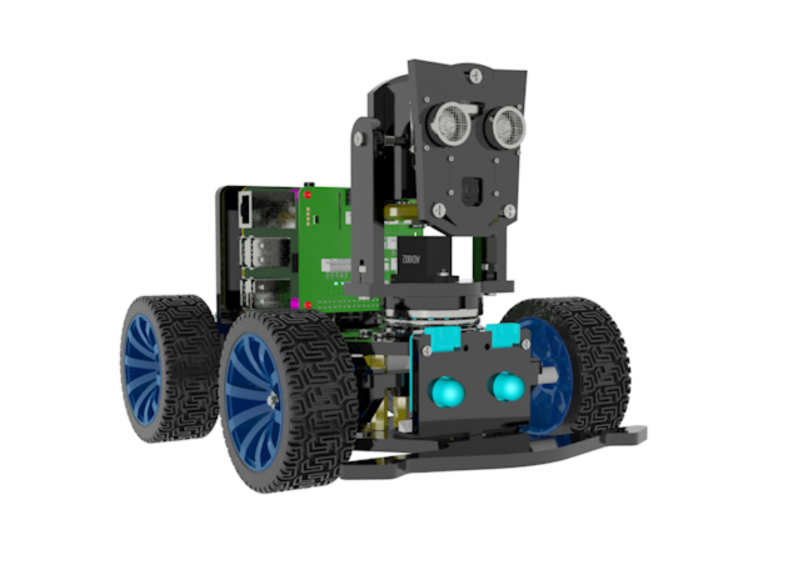
The "road"-the black line can be judged according to the reflected light received on the white paper "road" with black lines because of the different reflection coefficients of light between black lines and white paper. A more common detection method-infrared detection method is adopted in the tracking module.

Infrared detection is the method that makes use of infrared rays with reflection features on different physical surfaces in different colors. When the trolley is running, it continuously emits infrared light to the ground. That the infrared light comes across the white ground will cause  
diffuse reflection, and the reflected light is received by the receiving tube on the trolley; when it coming across a black line, it will be absorbed by the infrared light, and the trolley receiving tube fails to receive the signal.



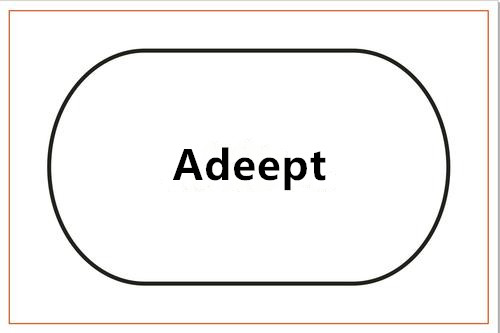
**6.2 Preparation**

1.Assemble the completed PiCar-b. The tracking module is fixed under the center hole of the front wheel of the PiCar Pro chassis by a 15mm nylon column with the M3\*4 screws. Note: please disassemble the tracking module to avoid any damage to the tracking module, when not using the tracking function or driving on the uneven ground. (The tracking module is about 7-10mm away from the ground)



2.Prepare a tracking track. Because of the steering limitation of the front-wheel servo, the curve radius of the track should not be too small.



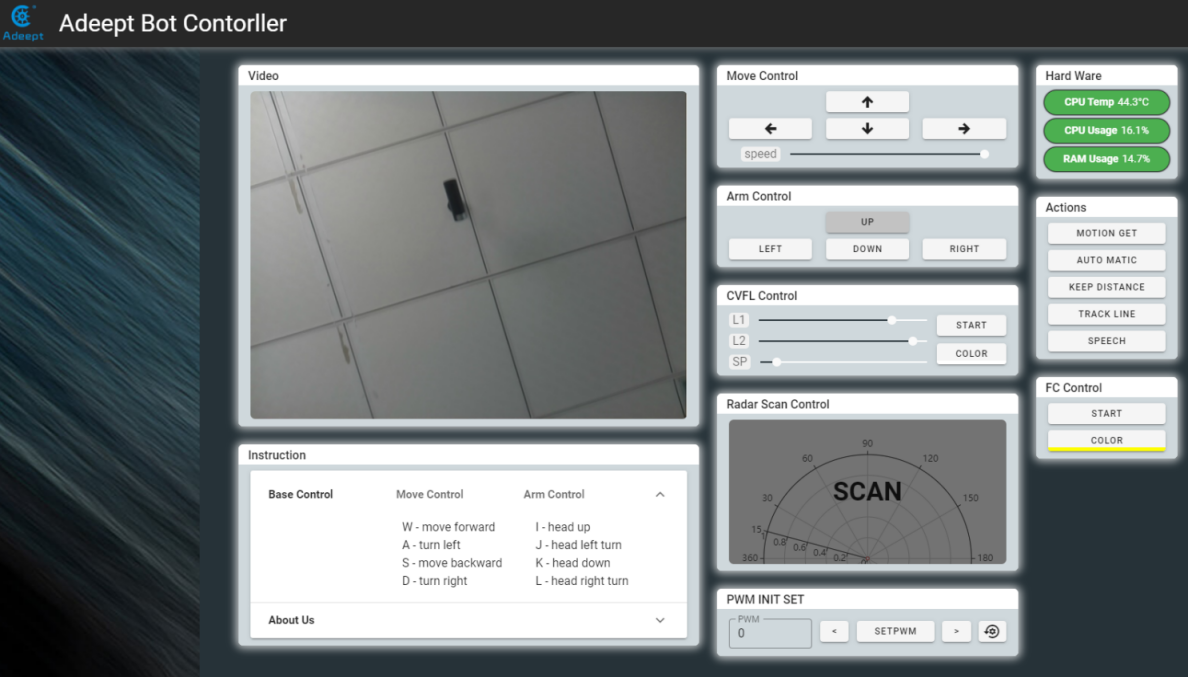


**6.3 Turn on the tracking function**

### Run the tracking program

1. Start up PiCarPro for about 1 minute.

2. When PiCarPro is started up, enter the IP address of your Raspberry Pi on the Google browser of the mobile phone or computer, and access Port 5000, for example: 192.168.3.44:5000. Then the web controller will be displayed on the browser.



1. Place the car on the completed tracking track.

2. Click "TRACK LINE", and PiCarPro will drive along the black line.

3. Click "TRACK LINE" for a second time, if you want to terminate the tracking function.

4. The height of the tracking module and the ground will have effects on the accuracy of the detection line of the tracking module. If the tracking function fails, the tracking module should be adjusted.

**6.4 The main code program**

**Please see the complete code in findline.py.**

**Import dependency and initialize**

|  |
| --- |
| 1. **import** RPi.GPIO as GPIO 2. **import** time 4. line\_pin\_right = 20 5. line\_pin\_middle = 16 6. line\_pin\_left = 19 8. **def** setup(): 9. GPIO.setwarnings(False) 10. GPIO.setmode(GPIO.BCM) 11. GPIO.setup(line\_pin\_right,GPIO.IN) 12. GPIO.setup(line\_pin\_middle,GPIO.IN) 13. GPIO.setup(line\_pin\_left,GPIO.IN) |

**Define the main function of the line searching module**

|  |
| --- |
| 1. **def** run(): 2. status\_right = GPIO.input(line\_pin\_right) 3. status\_middle = GPIO.input(line\_pin\_middle) 4. status\_left = GPIO.input(line\_pin\_left) 6. #  Detect whether the line searching module senses the lines 7. **if** status\_middle == 1: 8. **print**('forward') 9. **elif** status\_left == 1: 10. **print**('left') 11. **elif** status\_right == 1: 12. **print**('right') 13. **else**: 14. **print**('stop') |

**Execute the function**

|  |
| --- |
| 1. **if** \_\_name\_\_ == '\_\_main\_\_': 2. setup() 3. **while** 1: 4. run() |