

Lesson 25 Automatic Obstacle Avoidance

Car

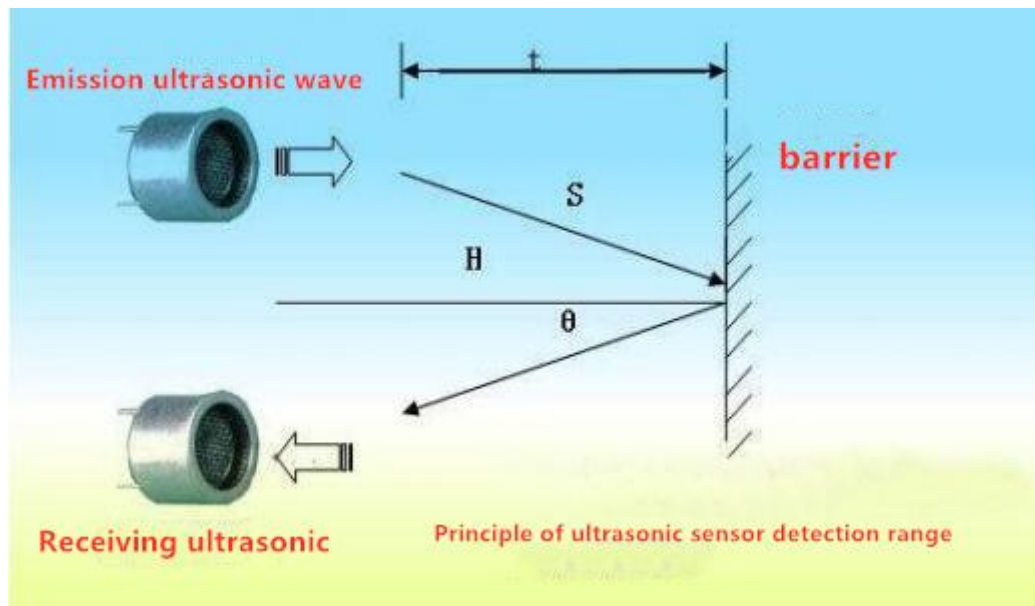
25.1 Overview

In this lesson, we will focus on introducing the automatic obstacle avoidance function of robots based on Raspberry Pi. You will learn about the principles behind ultrasonic obstacle detection and how to use automatic obstacle avoidance on PiCar Pro.

25.2 Introduction to Automatic Obstacle Avoidance

Since the camera used by our Raspberry Pi robot is a monocular camera and cannot collect depth information, many of our robot products use ultrasonic ranging modules to obtain depth information and detect whether there are obstacles in a certain direction to get the distance of the obstacle.

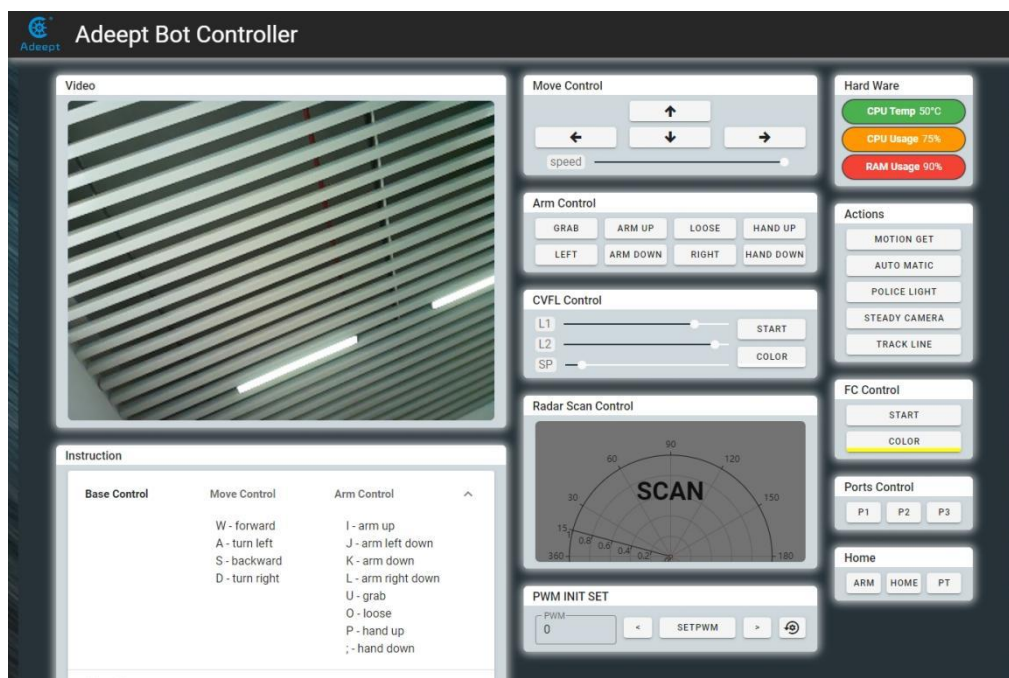
The principle of distance detection by ultrasonic ranging sensor: the method of detecting distance by ultrasonic is called echo detection method, that is, the ultrasonic transmitter emits ultrasonic waves in a certain direction, and the timer starts timing at the same time as the launch time. The ultrasonic waves propagate in the air and encounter obstacles on the way. When the object surface (object) is blocked, it will be reflected back immediately, and the ultrasonic receiver will immediately stop timing when the reflected ultrasonic wave is received. The propagation speed of ultrasonic waves in the air is 340m/s. According to the time t recorded by the timer, the distance s from the launch point to the obstacle surface can be calculated, namely: $s=340t/2$. Using this principle of ultrasound, the ultrasonic ranging module is widely used in practical applications, such as car reversing radar, unmanned aerial vehicle, and smart car.



25.3 Turning on Automatic Obstacle Avoidance

Running the Automatic Obstacle Avoidance program

1. Start the PiCar-Pro Robot. It may take about 30-50s to boot.
2. After PiCar-Pro is turned on, open the Chrome browser on your mobile or computer, enter the IP address of your Raspberry Pi and access port ":5000" into the IP address bar, like this: 192.168.3.31:5000. The web controller will then be displayed on the browser.



3. After clicking "**AUTO MATIC**", the robot will automatically avoid obstacles when encountering obstacles. .

4. When you want to terminate the Automatic Obstacle Avoidance function, you can click "**AUTO MATIC**" again.

25.4 Code

The main code is as follows. For the complete code, please check [Function.py](#).

```
01 def automaticProcessing(self):
02     scGear.moveAngle(1, 0)
03     dist = self.distRedress()
04     time.sleep(0.2)
05     if dist >= 40:
06         scGear.moveAngle(0, 0)
07         time.sleep(0.2)
08         move.move(35, 1, "mid")
09     elif dist > 20 and dist < 40:
10         scGear.moveAngle(1, 30)
11         move.move(0, 1, "mid")
12         time.sleep(0.3)
13         distLeft = self.distRedress()
14         self.scanList[0] = distLeft
15         scGear.moveAngle(1, -30)
16         time.sleep(0.3)
17         distRight = self.distRedress()
18         self.scanList[1] = distRight
19         print(self.scanList)
20         scGear.moveAngle(1, 0)
21         if self.scanList[0] >= self.scanList[1]:
22             scGear.moveAngle(0, -30 * Dv - Angular_deviation)
23             time.sleep(0.3)
24             move.move(35,1,"left")
25             time.sleep(1)
26         else:
27             scGear.moveAngle(0, 30 * Dv - Angular_deviation)
28             time.sleep(0.3)
29             move.move(35, 1, "right")
30             time.sleep(1)
31     else:
32         move.move(35, -1, "mid")
33         time.sleep(1)
```