# Lesson 4 **Introduction to the Ultrasonic Scanning Function**

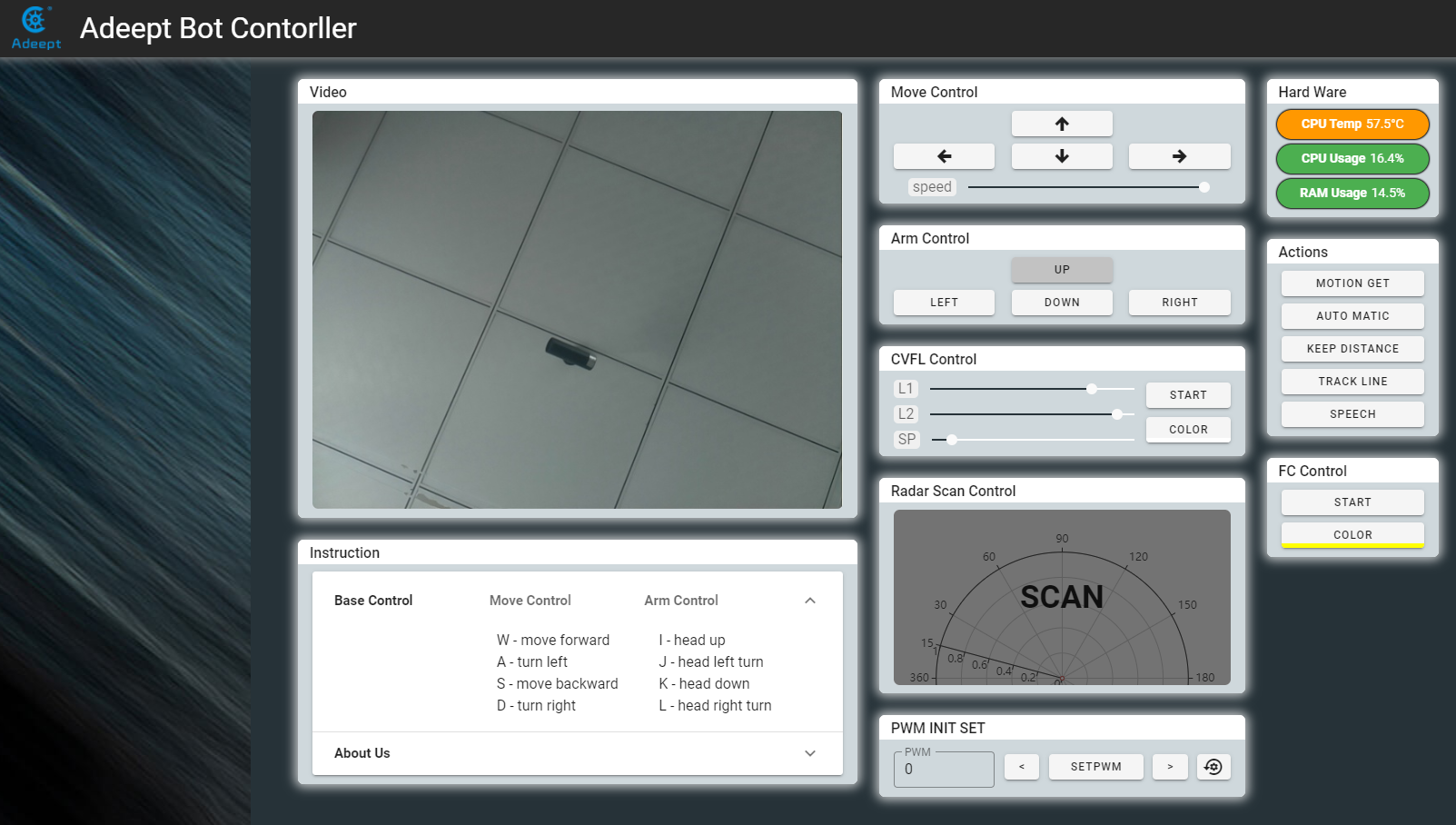
**4.1 Function introduction**

Ultrasonic scanning function refers to that when running the program, PiCar-b will turn the servo to rotate the ultrasonic module, detect the surroundings via the ultrasonic module, test the distance between the robot and surrounding objects, and display the corresponding distance in the WEB control interface.

**4.2 Turn on the ultrasonic scanning function**

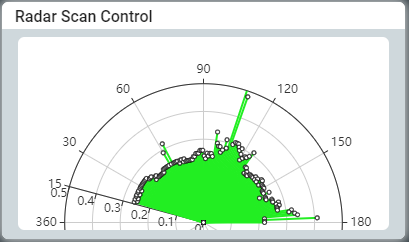
### Run the ultrasonic scanning program

1. Start up Raspberry Pi robot for about 30s-60s.
2. When PiCar-b is started up, enter the IP address of your Raspberry Pi on the Google browser of your 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.



3. Place the trolley at the position to be detected.

4. Click the pattern "scan" , and PiCar-b will rotate and detect the surroundings. The scanning results will be displayed in the pattern "scan" .



**4.2 The main code program**

Please see the complete code in findline.py.

Import dependency and initialize

|  |
| --- |
| 1. **import** time 2. **import** RPi.GPIO as GPIO 3. **import** Adafruit\_PCA9685 5. # Ultrasonic module pin 6. UPin1 = 11 7. Upin2 = 8 9. # Servo pin. Set the servo pin of your product that controls the left and right rotation of the ultrasonic module. 10. servoPin = 1 12. GPIO.setmode(GPIO.BCM) 13. GPIO.setup(UPin1, GPIO.OUT,initial=GPIO.LOW) 14. GPIO.setup(Upin2, GPIO.IN) 16. pwm = Adafruit\_PCA9685.PCA9685() 17. pwm.set\_pwm\_freq(50) 19. pwm0\_init = 300 20. pwm0\_max  = 450 # Set the maximum scanning angle 21. pwm0\_min  = 150 # Set the minimum scanning angle 22. pwm0\_pos  = pwm0\_init |

**Define the scanning function**

|  |
| --- |
| 1. **def** checkdist(): 2. ''' Reference basic module control ultra.Py ''' 4. **def** radarScan(): # Main function of ultrasonic scanning function 5. **global** pwm0\_pos 6. scan\_speed = 3 # Scanning speed 8. # Move the servo to the maximum angle 9. pwm0\_pos = pwm0\_max 10. pwm.set\_pwm(servoPin, 0, pwm0\_pos) 11. time.sleep(0.5) 13. # Rotate the servo and measure the distance continuously to reach the minimum angle 14. **print**(str(checkdist()), 'm') 15. **while** pwm0\_pos>pwm0\_min: 16. pwm0\_pos-=scan\_speed 17. pwm.set\_pwm(servoPin, 0, pwm0\_pos) 18. **print**(str(checkdist()), 'm') 20. # Returning 21. pwm.set\_pwm(servoPin, 0, pwm0\_init) |

**Execute the function**

|  |
| --- |
| 1. **if** \_\_name\_\_ == '\_\_main\_\_': 2. radar\_scan() 3. GPIO.cleanup() |