Lesson 4 Introduction to the Ultrasonic Scanning Function

4.1 Function introduction

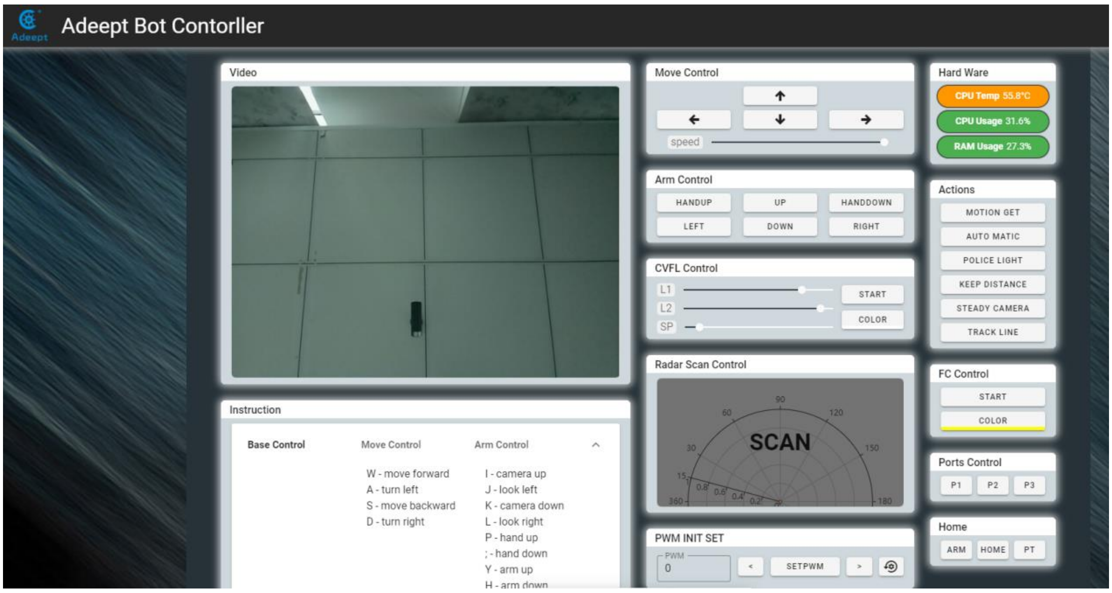
Ultrasonic scanning function refers to that when running the program, PiCarPro will turn the servo to rotate the ultrasonic module, detect the surroundings via the ultrasonic module, test the distance between the robot and surrounding objects, and display the corresponding distance in the WEB control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

4.2 Turn on the ultrasonic scanning function

4.2.1 Run the ultrasonic scanning program

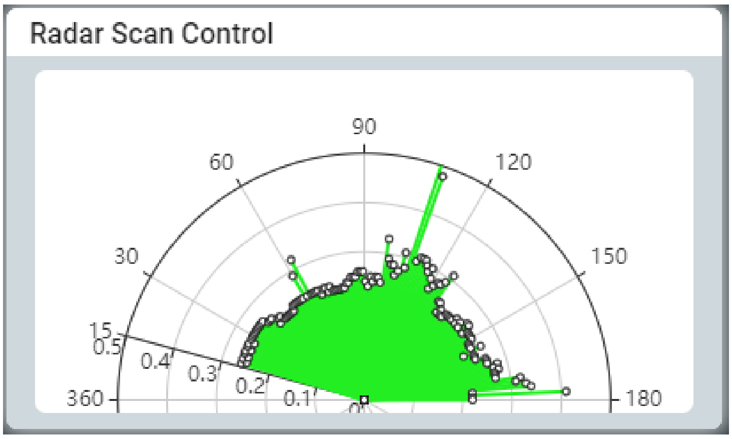
1. Start up Raspberry Pi robot for about 30s-60s.

2. When PiCarPro 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 PiCarPro will rotate and detect the surroundings. The scanning results will be displayed in the pattern "scan" .



4.3 The main code program

Please see the complete code in findline.py.

Import dependency and initialize

```
import time
import RPi.GPIO as GPIO
import Adafruit_PCA9685

# Ultrasonic module pin
UPin1 = 11
Upin2 = 8

# Servo pin. Set the servo pin of your product that controls the left and right rotation of the ultrasonic module.
servoPin = 1

GPIO.setmode(GPIO.BCM)
GPIO.setup(UPin1, GPIO.OUT,initial=GPIO.LOW)
GPIO.setup(Upin2, GPIO.IN)

pwm = Adafruit_PCA9685.PCA9685()
pwm.set_pwm_freq(50)

pwm0_init = 300
pwm0_max  = 450 # Set the maximum scanning angle
```

```
pwm0_min  = 150 # Set the minimum scanning angle
pwm0_pos  = pwm0_init
```

Define the scanning function

```
def checkdist():
    ''' Reference basic module control ultra.Py  '''

def radarScan(): # Main function of ultrasonic scanning function
    global pwm0_pos
    scan_speed = 3 # Scanning speed

    # Move the servo to the maximum angle
       pwm0_pos  =  pwm0_max
    pwm.set_pwm(servoPin, 0, pwm0_pos)
    time.sleep(0.5)

    # Rotate the servo and measure the distance continuously to reach the minimum angle
    print(str(checkdist()), 'm')
    while pwm0_pos>pwm0_min:
        pwm0_pos-=scan_speed
        pwm.set_pwm(servoPin, 0, pwm0_pos)
        print(str(checkdist()), 'm')

    # Returning
    pwm.set_pwm(servoPin, 0, pwm0_init)
```

Execute the function

```
if __name__ == '__main__':
    radar_scan()
    GPIO.cleanup()
```