

Lesson 26 Enable Speech Recognition

26.1 Overview

In this lesson, we will introduce how to configure and enable Speech recognition, and simply implement the function of using Speech to control the robotic arm.

26.2 Introduction to Speech Recognition

In the course <**Lesson 20 Local Speech Recognition**>, we have introduced local speech recognition. Here, we combine speech recognition with the control of the small car and achieve the function of controlling the robotic arm through Speech.

Main logic:

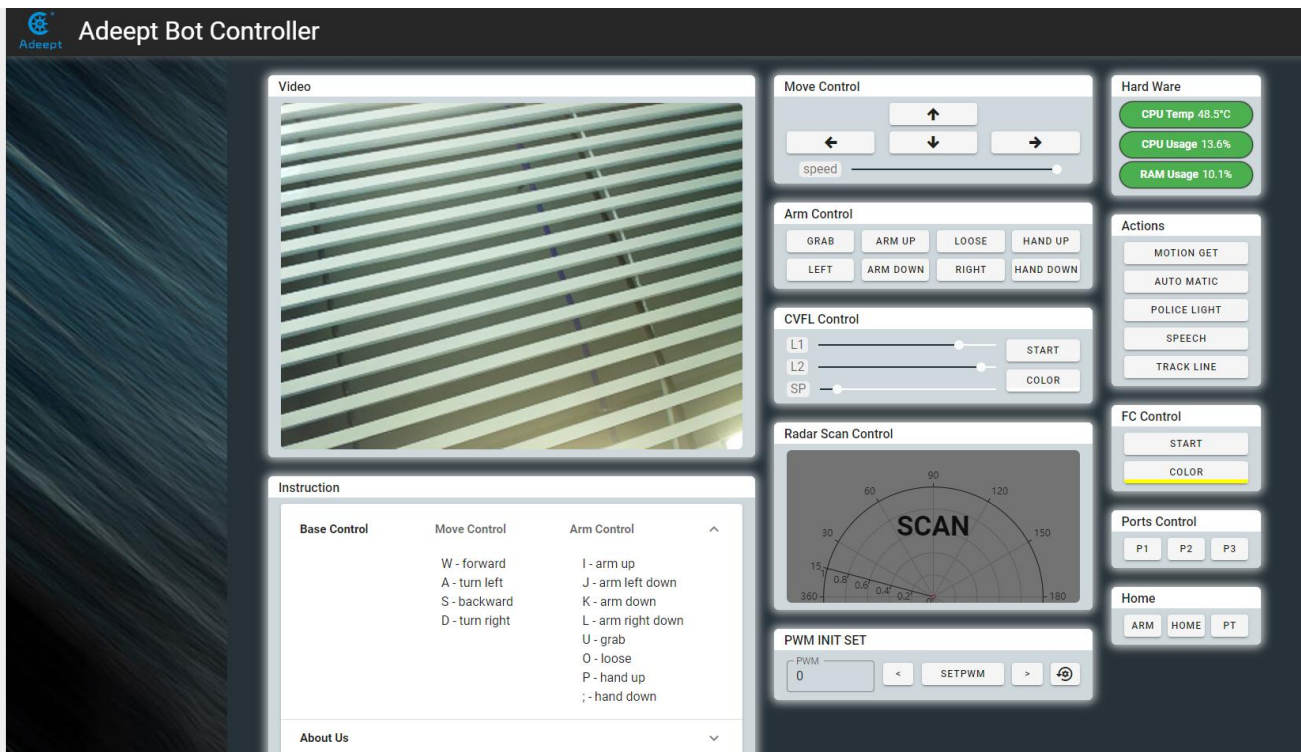
1. Start a separate thread to handle speech recognition. Enable the local speech recognition function during the initialization process and put the thread processing in a paused state.
2. When you click the **SPEECH** button on the page, activate the speech processing thread to perform speech recognition, convert the recognized content into text, and store it in the **output.txt** file.
3. Match the recognized speech content through commands and control the small car to perform corresponding actions.

Note: The speech recognition function depends on **sherpa-ncnn**. Therefore, before enabling the speech recognition function, please follow the steps in <**Lesson 20 Local Speech Recognition**> to download **sherpa-ncnn** and install the relevant dependent libraries.

26.3 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 "**SPEECH**", the robot will enable the speech recognition function.

For example, when the word "**lookleft**" is spoken, the corresponding information will be printed on the command line, and then the robotic arm will perform the corresponding action. Then, when the word "**stop**" is spoken, the servo of the robotic arm will return to the 90-degree position.

```
speech
get_info
get_info
get_info
The information recognized by the speech recognition is:: look
The information recognized by the speech recognition is:: lookright
Your command is lookright
get_info
The information recognized by the speech recognition is:: stop
get_info
Your command is stop
get_info
speechOff
get_info
```

4. When you want to terminate the speech recognition function, you can click "**SPEECH**" again. "**speechOff**" will appear on the page.

26.4 Code

Speech_Command.py

```
001  #!/usr/bin/env python3
002  # File name   : Voice_Command.py
003  # Author      : Aadept
004  # Date        : 2025/03/12
005  import time
006  from board import SCL, SDA
007  import busio
008  from adafruit_motor import servo
009  from adafruit_pca9685 import PCA9685
010
011  import threading
012  import os
013  import json
014  import Ultra as ultra
015  import Kalman_Filter as Kalman_filter
016  import Move as move
017  import RPiServo
018  import subprocess
019  from gpiozero import InputDevice
020
021  scGear = RPiServo.ServoCtrl()
022  scGear.start()
023  scGear.moveInit()
024
025  move.setup()
026
027  posUD = 0
028
029
030  def clear_output():
031      try:
032          with open("output.txt", "w") as file:
033              file.write("")
034              print("Output file cleared.")
035      except Exception as e:
036          print(f"Error clearing output file: {e}")
037
038
039  class Speech(threading.Thread):
040      def __init__(self, *args, **kwargs):
041          self.SpeechMode = 'none'
042          super(Speech, self).__init__(*args, **kwargs)
043          self.__flag = threading.Event()
044          self.__flag.clear()
045
046      def setup(self):
047          clear_output()
048
049      def run_sherpa_ncnn():
050          try:
051              print('run_sherpa_ncnn')
```

```

052         result = subprocess.run(['sudo', 'python', 'VoiceIdentify.py'], capture_output=True,
053 text=True)
054         if result.stdout:
055             print(result.stdout)
056         if result.stderr:
057             print(result.stderr)
058     except Exception as e:
059         print(f"Error executing command: {e}")
060
061     #os.system('sudo python VoiceIdentify.py')
062
063     threading.Thread(target=run_sherpa_ncnn).start()
064
065     def pause(self):
066         self.SpeechMode = 'none'
067         move.motorStop()
068         self.__flag.clear()
069
070     def resume(self):
071         self.__flag.set()
072
073     def speech(self):
074         clear_output()
075         self.SpeechMode = 'speech'
076         self.resume()
077
078     def functionGoing(self):
079         if self.SpeechMode == 'none':
080             self.pause()
081         elif self.SpeechMode == 'speech':
082             self.SpeechProcessing()
083
084     def run(self):
085         while True:
086             self.__flag.wait()
087             self.functionGoing()
088             time.sleep(0.1)
089
090     def SpeechProcessing(self):
091         file_position = 0
092         try:
093             with open("output.txt", "r") as file:
094                 file.seek(file_position)
095                 new_lines = file.readlines()
096                 if new_lines:
097                     print(f"Read {len(new_lines)} new lines from output.txt\n")
098                     for line in new_lines:
099                         line = line.replace(" ", "").lower()
100                         for keyword in ["lookleft", "lookright", "armup", "armdown", "handup",
101 "handdown",
102                                     "grab", "loose", "stop"]:
103                             if keyword in line:
104                                 if keyword == 'lookleft':
105                                     scGear.singleServo(1, 1, 60)
106                                     time.sleep(0.1)
107                                     print('Your command is lookleft ')

```

```
108         elif keyword == 'lookright':
109             scGear.singleServo(1, -1, 60)
110             time.sleep(0.1)
111             print('Your command is lookright ')
112         elif keyword == 'armup':
113             scGear.singleServo(2, -1, 60)
114             time.sleep(0.1)
115             print('Your command is armup ')
116         elif keyword == 'armdown':
117             scGear.singleServo(2, 1, 60)
118             time.sleep(0.1)
119             print('Your command is armdown ')
120         elif keyword == 'handup':
121             scGear.singleServo(3, 1, 60)
122             time.sleep(0.1)
123             print('Your command is handup ')
124         elif keyword == 'handdown':
125             scGear.singleServo(3, -1, 60)
126             time.sleep(0.1)
127             print('Your command is handdown ')
128         elif keyword == 'grab':
129             scGear.singleServo(4, -1, 60)
130             time.sleep(0.1)
131             print('Your command is grab ')
132         elif keyword == 'loose':
133             scGear.singleServo(4, 1, 60)
134             time.sleep(0.1)
135             print('Your command is loose ')
136         elif keyword == 'stop':
137             scGear.stopWiggle()
138             time.sleep(0.3)
139             print('Your command is stop ')
140         file_position = file.tell()
141     except Exception as e:
142         print(f"Error reading output file: {e}")
143         time.sleep(0.1)
144
145
146 if __name__ == '__main__':
147     try:
148         fuc = Speech()
149         fuc.setup()
150         fuc.start()
151         fuc.speech()
152         while True:
153             pass
154     except KeyboardInterrupt:
155         move.motorStop()
```