

# AI Voice Interaction: Intelligent Interaction

---

## AI Voice Interaction: Intelligent Interaction

[Function Introduction](#)

[Function Experience](#)

[Program source code](#)

[Functional principle](#)

## Function Introduction

---

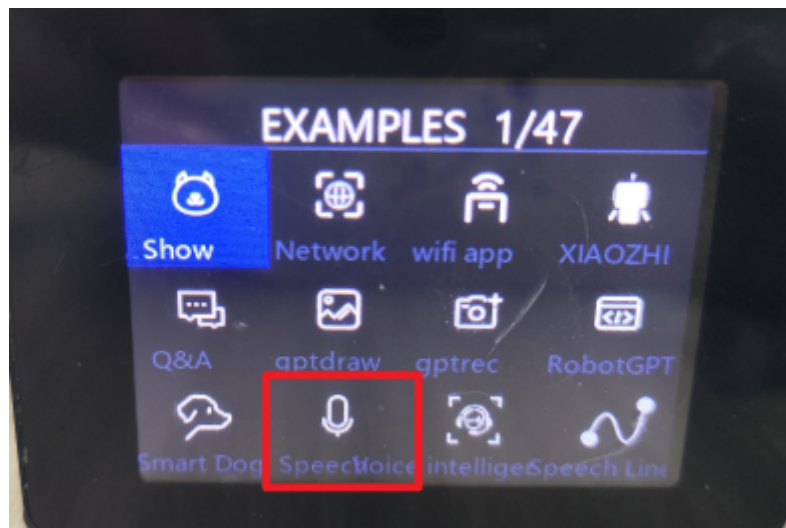
This case is the gameplay of the startup program. You can use fixed voice command words to make the robot dog do corresponding actions.

**This function needs to be connected to the network to run normally**

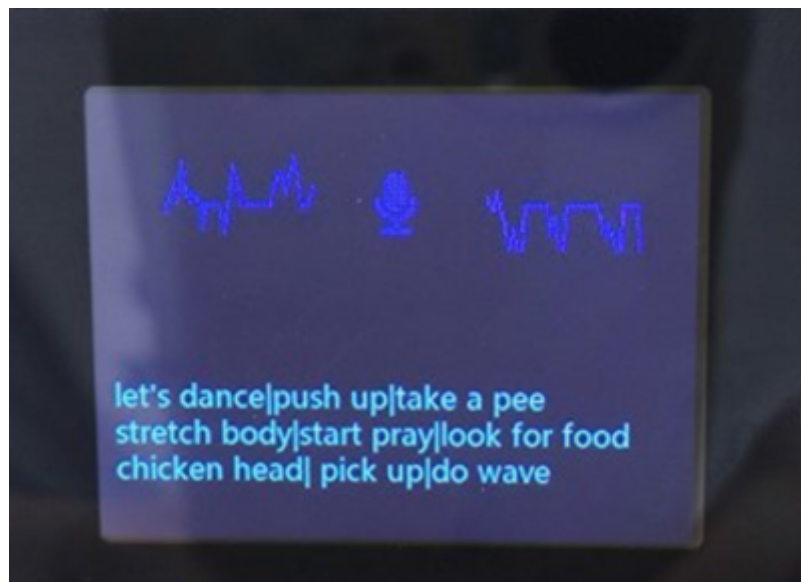
## Function Experience

---

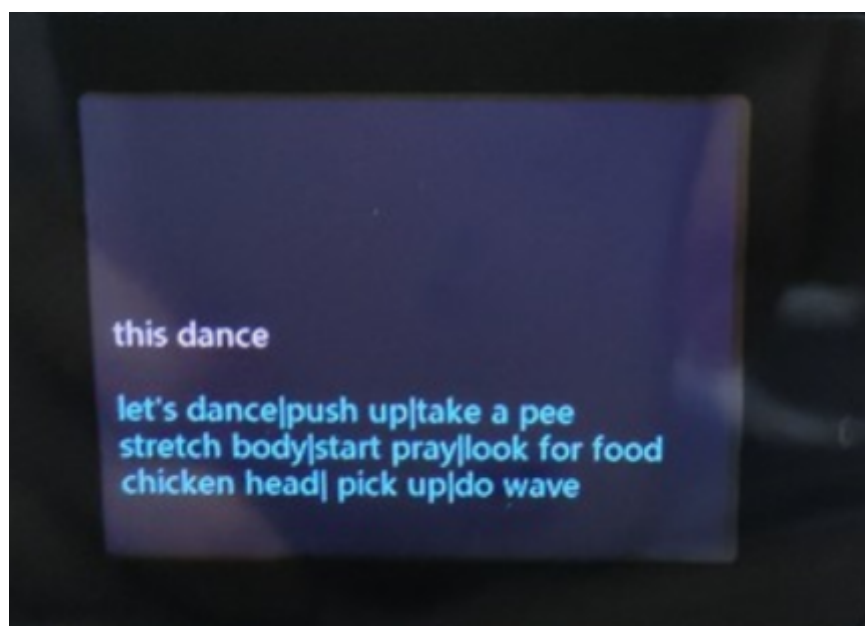
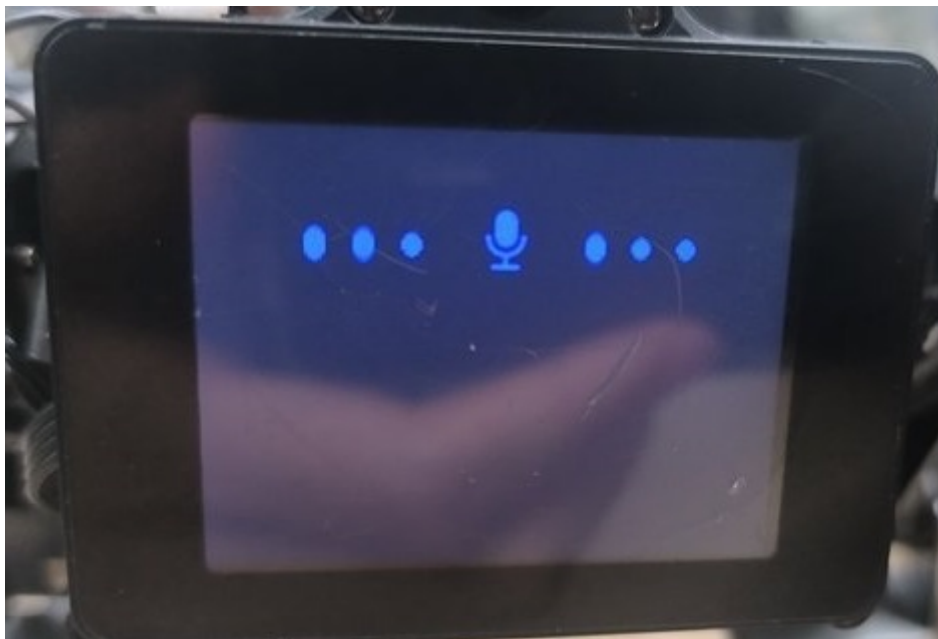
1. Turn on the robot dog first, press the button on the upper right of the "dog head" to enter the sample mode, and then select the voice recognition function.



2. After entering the voice recognition function, wake up with voice first, "lulu".



3. When you hear a ding, you can say the prompt word at the bottom of the screen.



4. The robot dog will recognize the semantics and move.

## Program source code

---

1. First, log in to the robot dog system through VNC
2. Then enter the terminal

```
cd /home/pi/RaspberryPi-CM4-main/demos/speech/  
tree
```

### 3. Directory structure description

- ├─ audio.py #Recording file
- ├─ auto\_platform.py #System environment dependency
- ├─ language\_recognize.py #Speech recognition
- ├─ libnymaya.py #Speech wake-up
- └─ speech.py #Speech interaction main function

python Main function analysis

```
def actions_dog(act):  
  
    command_actions = {  
        "cn": {  
            "跳舞": 23,  
            "俯卧撑": 21,  
            "撒尿": 11,  
            "伸懒腰": 14,  
            "祈祷": 17,  
            "找食物": 18,  
            "鸡头": 20,  
            "向下抓取": 130,  
            "波浪": 15, "拨浪": 15,  
        },  
        "en": {  
            "dance": 23, "let's dance": 23,  
            "push up": 21, "push": 21,  
            "pee": 11, "take a pee": 11, "take": 11,  
            "stretch": 14, "body": 14, "stretch body": 14,  
            "pray": 17, "start": 17, "start pray": 17,  
            "food": 18, "look": 18, "look for food": 18,  
            "head": 20, "chicken": 20, "chicken head": 20,  
            "pick": 130, "pick up": 130,  
            "wave": 15, "do wave": 15,  
        }  
    }  
  
    commands = command_actions.get(1a, command_actions["en"])
```

command\_actions: defines the actions to be performed after speech recognition. If you want to add more command words, you can add one according to the action group in the protocol. For example: the dance action group id mentioned above is 23, which is checked from the action group table. You can also check the action group id according to the section of executing preset actions in Chapter 2.

ID	Actions	Duration/s	ID	Actions	Duration/s	ID	Actions	Duration/s
1	Lie down	3	2	Stand up	3	3	Crawl	5
4	Turn in a circle	5	5	Mini is stepping	4	6	Squat	4
7	Roll	4	8	Turn Pitch	4	9	Yaw	4
10	Three-axis rotation	7	11	Pee	7	12	Sit	5
13	Wave	7	14	Stretch	10	15	Wave	6
16	Sway	6	17	Beg	6	18	Find food	6
19	Shake hands	10	20	Chicken head	9	21	Push-ups	8
22	Look around	8	23	Dance	6	24	Playful	7
128	Grab up	10	129	Catch	10	130	Grab	10

## Functional principle

The specific flow chart is as follows:

