

Smart Voice Control

Before running the function, you need to close the App large programs. For the closing method, refer to [4. Preparation] - [1. Manage APP control services].

Orin board users can directly open the terminal and enter the tutorial commands to run. Jetson-Nano board users need to enter the docker container first, then enter the tutorial commands in the docker to start the program.

1. Function Description

Control the robotic arm to perform different actions through voice commands.

2. Startup and Operation

2.1. Startup

Open the terminal and enter the following command to start:

```
python3 ~/dofbot_voice/scripts/intelligent_voice_ctrl.py
```

2.2. Operation Steps

After the program runs, the voice module will broadcast "here". Then say "Hello, yahboom" to the voice module. The voice module will reply "here" to indicate successful wake-up. Then say "action a" to the voice module. The voice module will broadcast "ok", and then the robotic arm's gripper will tighten and then release, tighten and then release, simulating a clapping motion. The voice commands that this program can recognize are as follows:

Voice Command	Effect
ACTION-A	Gripper opens and closes (clapping)
ACTION-B	Robotic arm moves up, servo 4 moves up and down (nodding)
ACTION-C	Robotic arm presents praying posture (praying)
ACTION-D	Robotic arm presents kneeling posture (kneeling)
RESET	Return to robotic arm extended state
ACTION-E	Robotic arm extends upward, turns left and right (startled)

3. Core Code Analysis

Jetson-Nano users need to enter the docker container to view

```
Source code path: ~/dofbot_voice/scripts/intelligent_voice_ctrl.py
```

```
#Import robotic arm underlying control library
import Arm_Lib
#Import voice recognition broadcast library
from Speech_Lib import Speech
```

```
#Create voice recognition broadcast object myspeech
mySpeech = Speech()
#Create robotic arm underlying control object Arm
Arm = Arm_Lib.Arm_Device()

#Call speech_read function to get voice recognition result
result = mySpeech.speech_read()

#Call Arm_serial_servo_write6 function to control six servos to rotate to set
angles, 1000 means complete the six servo angle rotation task within 1000
milliseconds
Arm.Arm_serial_servo_write6(90, 90, 90, 90, 90, 90, 1000)
```