

## 2.Voice control robotic arm memory learning

### 2.1 Function description

The robotic arm is controlled by the voice module to record the pose of the robotic arm that we manually set, and the robotic arm can be controlled by voice to perform the recorded actions.

### 2.2 Steps

#### 2.2.1 ROS package path

```
~/yahboomcar_ws/src/yahboomcar_voice_ctrl/scripts
```

#### 2.2.2 Start up

```
#You need to enter docker first, perform this step more  
#If running the script to enter docker fails, please refer to 07.Docker-orin/05,  
Enter the robot's docker container  
~/run_docker.sh  
python3 ~/yahboomcar_ws/src/yahboomcar_voice_ctrl/scripts/voice_arm_study.py
```

1. After entering the above command, the voice module will say "I'm ready. Let me record some actions ", and then the robotic arm will turn off the torque. At this time, we can manually move each servo to change its angle.
2. We can say "Hi Yahboom", the voice module will reply "Hi i'm here".
3. We say "record it", then the robotic arm will record the current action, and the voice module will reply "OK, please set the next action".  
At the same time, we can also see that the terminal will print out the pose of the current robotic arm.

**Tips: Each servo has a limit value. When the action we manually set exceeds the limit value, this group of actions will not be recorded, the buzzer will beep once, and the number of the servo that exceeds the limit value will be printed on the terminal, the value exceeding the limit will be displayed as a negative number.**

The system can record up to 20 sets of actions, if we want to stop recording, we need to say "Now is over".

4. When we finish recording the action group, say "Now is over", the voice module replies with "Noted it,I've recorded them". Next, we can say "Display actions", the voice module replies "OK, Let's do it", and the robotic arm will perform the actions just recorded.
- If you need to re-learn and record the action group, then say "Clear actions", the module replies "OK, actions have been cleared", and then return to the initial position, after the "di" sound, play sound "I'm ready.Let me record some actions.", then you can start to let the robotic arm record again action group.

If you need to re-learn and record the action group, you can say "Clear actions" to it, the module will reply "OK, actions have been cleared ", the robotic arm will return to the initial position.

After the buzzer beeps one time, the voice module will reply "I'm ready. Let me record some actions", at this time, you can start to let the robotic arm record the action group again.

```
jetson@yahboom: ~
jetson@yahboom: ~ 80x24
-----
MY_IP: 192.168.2.111
ROS_MASTER_URI:
http://192.168.2.111:11311
my_robot: X3plus | my_lidar: X3
-----
jetson@yahboom:~$ ls^C
jetson@yahboom:~$ ^C
jetson@yahboom:~$ ^C
jetson@yahboom:~$ python3 ~/yahboomcar_ws/src/yahboomcar_voice_ctrl/scripts/voice_arm_study.py
Speech Serial Opened! Baudrate=115200
Rosmaster Serial Opened! Baudrate=115200
-----create receive threading-----
2
-19
[]
[[90, 130, 2, 52, 90, 33]]
[[90, 130, 2, 52, 90, 33], [90, 134, 30, 52, 89, 33]]
[[90, 130, 2, 52, 90, 33], [90, 134, 30, 52, 89, 33], [90, 112, 55, 80, 89, 33]]
```

### 2.2.3、Code: voice\_arm\_study.py

- code path

```
~/yahboomcar/src/yahboomcar_voice_ctrl/scripts
```

- Core code analysis

1) Turn off/on the torque of the servo. Only when the torque of the servo is turned off, you can manually set each servo angle.

```
driver.car.set_uart_servo_torque(0) # Turn off torque
driver.car.set_uart_servo_torque(1) # Turn on torque
```

2) Two important lists

```
current_joints[] #A list of the current servo data,that is, the data to be
recorded
study_joints[] #learned action group data
```

3) The function to read the angle values of the current servo

```
get_uart_servo_angle_array ()
```

4) Study status flag bit study\_flag

```
study_flag = True #study, record status
study_flag = False #sports state
```

## 2.3 Flow chart

