

## 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 actions learned and recorded by its movement can be controlled by voice.

### 2.2 Start

#### 2.2.1 ROS package path

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

#### 2.2.2 start

```
python3 ~/yahboomcar_ws/src/yahboomcar_voice_ctrl/scripts/voice_arm_study.py
```

- The code starts, when heard "I'm ready.Let me record some actions", the robotic arm will turn off the torque. At this time, we can manually move each servo. When the angle we need is moved, we can "Hi Yahboom" wake up the module. After hearing "Hi , I'm here" it, we can continue to say "record it", at this time the robotic arm will record the current action, the voice module will also reply, "OK, please set the next action" and the terminal will print out the currently learned pose.

(Here is a tip, **Each servo has an angle limit. When the limit is exceeded, the action of this group will not be recorded, and the buzzer will make a "di" sound, and it will print on the terminal which servo exceeds the limit. Limit, the value beyond the limit will be displayed as a negative number**) Up to **20 groups of actions can be recorded**, and the recording can be stopped by saying "Now is over" to the module.

- After we finish recording the action group, speak to the voice module "Now is over" and reply "OK, Noted it,I've recorded them.", at this time, you can "Display actions" let the robotic arm move the action you just recorded by speaking through the voice;
- 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.

```
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、core 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 can you manually move each servo.

```
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) read the function of each angle value 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 program flow chart

