2. Voice control trolley movement

Before running the program, you need to bind the port number of the voice board and the port number of the ROS expansion board in the host computer, you can refer to the previous chapter for binding; when you enter the docker container, you need to mount this voice board to be able to recognize the voice board in the docker container.

1. Program function description

After the program starts, say "Hello, Xiaoya" to the module, and the module replies "Yes" to wake up the voice board, then you can issue voice commands to it to control the basic motion control of the car, such as forward, backward, left/right, turn, and so on. It can control the basic motion control of the car, such as forward, backward, left/right turn, the motion control lasts for 5 seconds and then stops. You can also use voice commands to control the light effects, such as bright lights. Turn on the running lights and other operations.

2. Program code reference path

After entering the docker container, the location of the source code of this function is located at.

```
/root/yahboomcar_ros2_ws/yahboomcar_ws/src/yahboomcar_voice_ctrl/yahboomcar_voic
e_ctrl/Voice_Ctrl_Ackman_driver_R2.py
```

3. Program startup

3.1. Startup commands

After entering the docker container, depending on the actual model, terminal input, the

```
ros2 run yahboomcar_voice_ctrl Voice_Ctrl_Ackman_driver_R2
```

4. Core Code

This part of the code and yahboomcar_bringup in the **Ackman_driver_R2.py** roughly similar to the same, just more voice control instructions to receive and parse the process, the next main part of this part of how to realize.

```
#导入相对应的语音库 #Import the corresponding voice library
from Speech_Lib import Speech
#导入相对应的底层驱动库
# Import the corresponding underlying driver libraries
from Rosmaster_Lib import Rosmaster
#创建语音控制对象 #Creating Voice Control Objects
spe = Speech()
#创建底层控制的对象 #Create objects for underlying control
self.car = Rosmaster()
#读取语音板识别的结果，speech_r就是识别的结果，是底层库解析后会返回一个数字，通过这个数字来识别
指令
# Read the results of voice board recognition, speech_r is the result of the
recognition, is the underlying library will return a number after parsing,
through this number to identify the instruction
```

```
speech_r = spe.speech_read()
if speech_r == 2 or speech_r == 0 :
    ....
#写入指令，播报语音结果，这里的发送指令后，底层库会包装打包后发送给语音板，语音板接收到后，发出相对的音频文件
# Write commands to broadcast voice results, here after sending commands, the underlying library will be packaged and packaged to send to the voice board, the voice board receives it and sends out the relative audio file
spe.void_write(speech_r)
#控制小车运动和灯带，直接对接底层库，没有通过ros发布
#Control of cart motion and light strips, direct interface to underlying libraries, no release via ros
self.car.set_car_motion(vx, vy, angular)     #小车运动
#Cart Movement
self.car.set_colorful_effect(6, 6, parm=1)  #灯带效果
#Light Strip Effect
```

The command word correspondences for this section of the course are as follows.

- motion control

function word	Voice module recognition results	Contents of voice announcement
stop	2	Okay, it's stopped.
go forward	4	Okay, it's moving forward.
draw back	5	Okay, it's backing up.
turn left	6	Okay, turning left .
turn right	7	Okay, it's turning right.

- Light Strip Effect

function word	Voice module recognition results	Contents of the voice announcement
turn off the light	10	Okay, light's off.
turn on the red light	11	Okay, red light is on.
turn on the green light	12	Okay, green light is on.
turn on the blue light	13	Okay, blue light is on.
turn on the yellow light	14	Okay, yellow light is on.
Turn on the flowing light.	15	Okay, the flowing light is on.
Turn on the fade light	16	Okay, the fade light is on.
Turn on the breathing light.	17	Okay. Breathing light is on.
Display power level	18	Okay, it's showing the power level.