2、Voice control of car movement

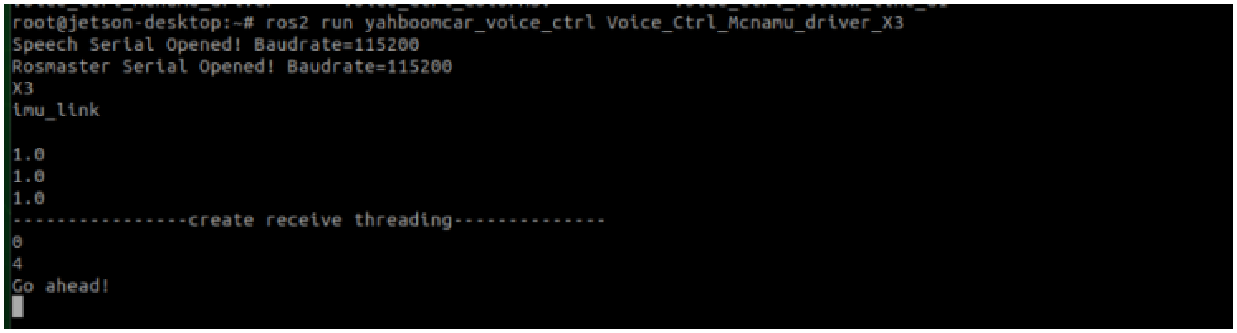
Take our company's product Rosmaster-X3 as an example to illustrate how to call Speech in the program_ The Lib library performs speech recognition to control the motion of the car/robot. This course needs to be combined with the hardware of the Rosmaster-X3 car, and only code analysis will be done here. Firstly, let's take a look at the built-in voice commands，

functional word	Speech recognition results	Voice broadcast content
Stop	2	OK , I'm stop
Go ahead	4	OK , let's go.
Back	5	OK , I‘m back.
Turn left	6	OK , I‘m turning left.
Turn right	7	OK , I‘m turning right.
Close light	10	OK, light is closed.
Red light up	11	OK, red light is on
Green light up	12	OK, green light is on.
Blue light up	13	OK, blue light is on.
Yellow light up	14	OK, yellow light is on.
light A	15	OK, light A is on.
lightB	16	OK, light B is on.
light C	17	OK, light C is on.
display power	18	OK, battery value has been display .

1、Run Case Program

The terminal inputs the following commands，

```
ros2 run yahboomcar_voice_ctrl Voice_Ctrl_Mcnamu_driver_X3
```



```
root@jetson-desktop:~# ros2 run yahboomcar_voice_ctrl Voice_Ctrl_Mcnamu_driver_X3
Speech Serial Opened! Baudrate=115200
Rosmaster Serial Opened! Baudrate=115200
X3
imu_link

1.0
1.0
1.0
----------------create receive threading-------------
0
4
Go ahead!
```

2、Core code

code path：
~/driver_ws/src/yahboomcar_voice_ctrl/yahboomcar_voice_ctrl/Voice_Ctrl_Mcnamu_driver_X3.py

```
#Import the corresponding voice library
from Speech_Lib import Speech
#Import the corresponding underlying driver library
from Rosmaster_Lib import Rosmaster
#Create voice control objects
spe = Speech()
#Create objects for underlying control
self.car = Rosmaster()
#Read the results of speech board recognition, speech_ R is the result of
recognition, which is a number returned by the underlying library after parsing,
which is used to identify instructions

speech_r = spe.speech_read()
if speech_r == 2 or speech_r == 0 :
....
#Write instructions and broadcast voice results. After the instructions are sent,
the underlying library will package and send them to the voice board. After the
voice board receives them, it will send out corresponding audio files
spe.void_write(speech_r)

#Control the movement of the car and the light strip, directly connect to the
underlying library, not released through ROS
self.car.set_car_motion(vx, vy, angular) #Car movement
self.car.set_colorful_effect(6, 6, parm=1) #Light strip effect
```