2.Control Muto motion

Before running this program, you need to bind the port number of the voice board and the port number of the ROS expansion board on the host machine. You can refer to the previous chapter for binding;
When entering the docker container, you need to mount the voice board to recognize the voice board in the docker container.

1. Program function description

After the program is started, say "Hello, yahboom" to the module, and the module will reply "Yes" to wake up the voice board.Next, you can issue voice commands to it, which can control Muto's basic motion control, such as the car moving forward, backward, left/right turn, and the motion control will stop after 5 seconds.

2. Program code reference path

After entering the docker container, the location of the source code of this function is:

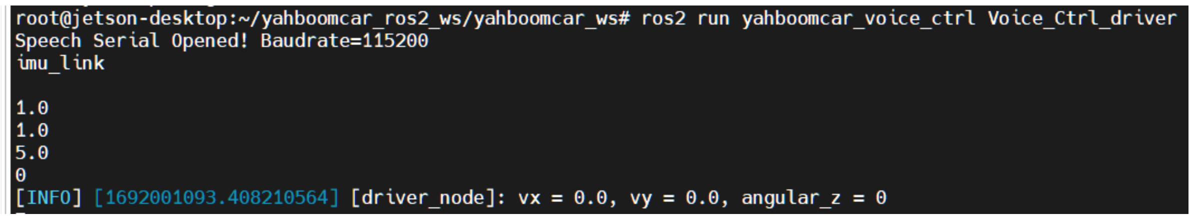
```
/root/yahboomcar_ros2_ws/yahboomcar_ws/src/yahboomcar_voice_ctrl/yahboomcar_voic
e_ctrl/Voice_Ctrl_driver.py
```

3. Program startup

3.1. Start command

After entering the docker container, enter the terminal according to the actual car model,

```
ros2 run yahboomcar_voice_ctrl Voice_Ctrl_driver
```



```
root@jetson-desktop:~/yahboomcar_ros2_ws/yahboomcar_ws# ros2 run yahboomcar_voice_ctrl Voice_Ctrl_driver
Speech Serial Opened! Baudrate=115200
imu_link

1.0
1.0
5.0
0
[INFO] [1692001093.408210564] [driver_node]: vx = 0.0, vy = 0.0, angular_z = 0
```

4. Core code

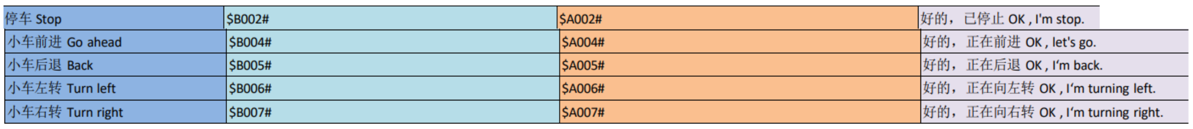
This part of the code is roughly the same as **muto_driver.py** in yahboomcar_bringup. It just adds the process of receiving and parsing voice control instructions. Next, I will mainly talk about how this part is implemented.

```
#Import the corresponding voice library
from Speech_Lib import Speech
#Import the corresponding underlying driver library
from MutoLib import Muto
#Create voice control objects
spe = Speech()
#Create underlying control objects
self.muto = Muto()
```

```
#Read the result of speech board recognition. speech_r is the result of
recognition. The underlying library will return a number after parsing. Use this
number to identify the command.
speech_r = spe.speech_read()
if speech_r == 2 or speech_r == 0 :
    ....
#Write the command and broadcast the voice result. After sending the command
here, the underlying library will package it and send it to the voice board.
After the voice board receives it, it will send out the corresponding audio
file.
spe.void_write(speech_r)
#Control the movement of the car and the light strip, directly connect to the
underlying library, and are not released through ros
self.execute_vel(vx, vy, angular)   #car movement
```

The instruction words in this course correspond to the following,

- Motion control



停车 Stop	$B002#	$A002#	好的，已停止 OK , I'm stop.
小车前进 Go ahead	$B004#	$A004#	好的，正在前进 OK , let's go.
小车后退 Back	$B005#	$A005#	好的，正在后退 OK , I'm back.
小车左转 Turn left	$B006#	$A006#	好的，正在向左转 OK , I'm turning left.
小车右转 Turn right	$B007#	$A007#	好的，正在向右转 OK , I'm turning right.