2.Robot information release

1. Program function description

After the program is run, combined with the ROS expansion board, you can obtain the sensor information on the ROS expansion board, control Muto movement, control the buzzer and other functions.

2. Program code reference path

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

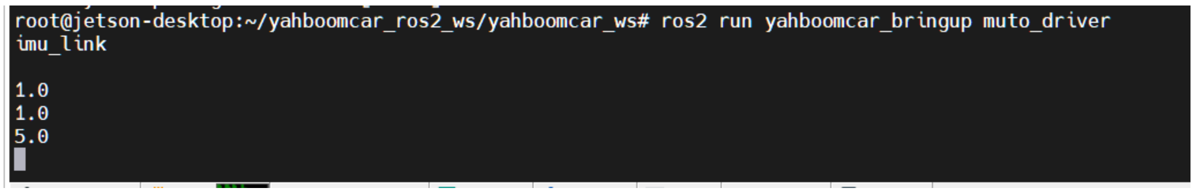
```
/root/yahboomcar_ros2_ws/yahboomcar_ws/src/yahboomcar_bringup/yahboomcar_bringup/muto_driver.py
```

3. Program startup

3.1.Startup command

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

```
ros2 run yahboomcar_bringup muto_driver
```



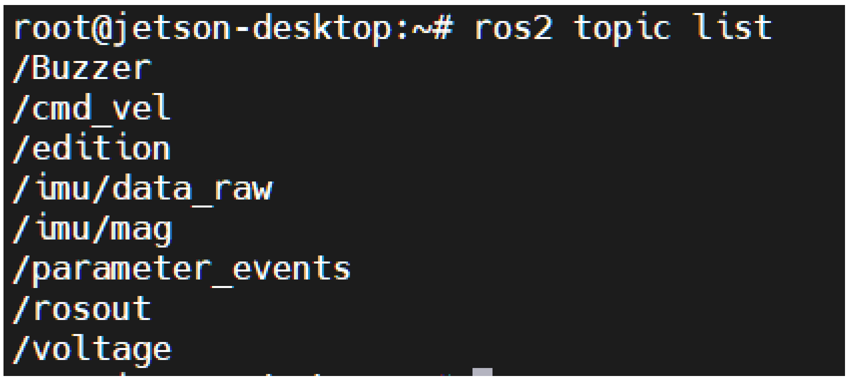
```
root@jetson-desktop:~/yahboomcar_ros2_ws/yahboomcar_ws# ros2 run yahboomcar_bringup muto_driver
imu_link

1.0
1.0
5.0
```

3.2. View node topics

Open the terminal and enter the container and enter,

```
ros2 topic list
```



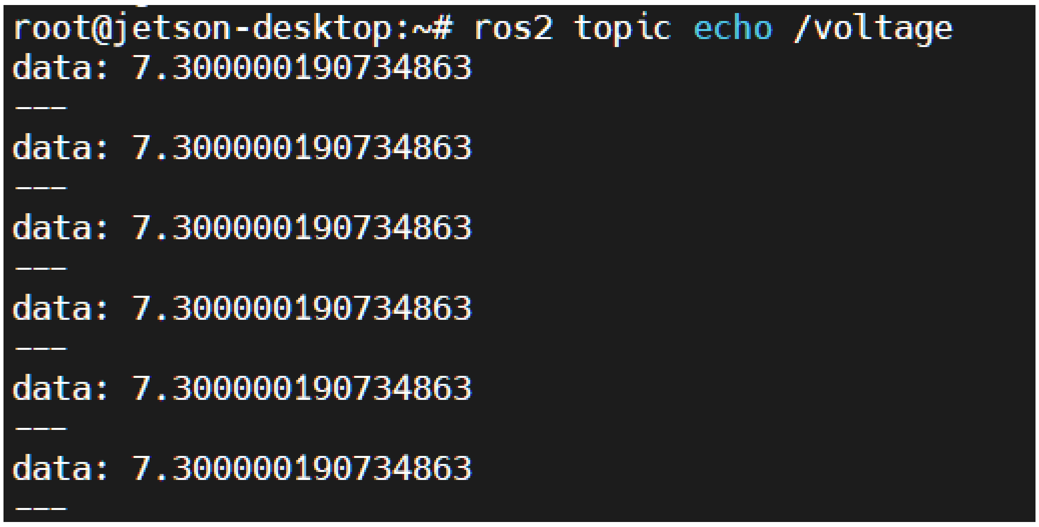
```
root@jetson-desktop:~# ros2 topic list
/Buzzer
/cmd_vel
/edition
/imu/data_raw
/imu/mag
/parameter_events
/rosout
/voltage
```

Topic name	Topic content
/Buzzer	buzzer
/cmd_vel	speed control
/edition	Version Information
/imu/data_raw	IMU sensor data
/imu/mag	IMU-Magnetometer data
/voltage	Battery voltage information

3.3. Read topic data

Open the terminal and enter the container and enter, taking reading the voltage as an example,

```
ros2 topic echo /voltage
```

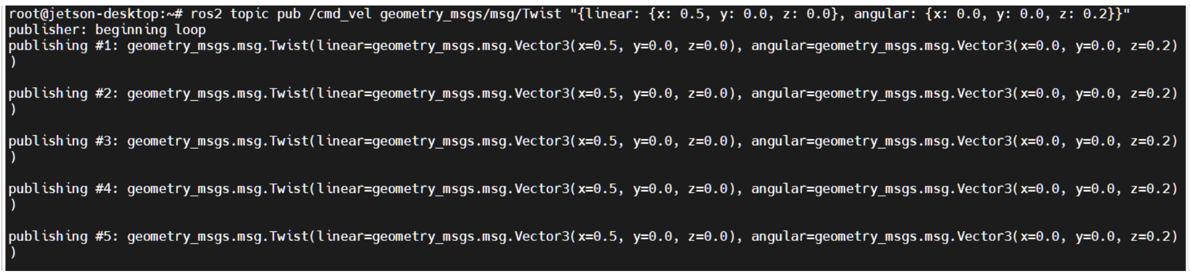


```
root@jetson-desktop:~# ros2 topic echo /voltage
data: 7.300000190734863
---
data: 7.300000190734863
---
data: 7.300000190734863
---
data: 7.300000190734863
---
data: 7.300000190734863
---
data: 7.300000190734863
---
```

3.4. Publish topic data

Open the terminal and enter the container and enter. Take publishing /cmd_vel data to control Muto movement as an example.

```
ros2 topic pub /cmd_vel geometry_msgs/msg/Twist "{linear: {x: 0.5, y: 0.0, z:
0.0}, angular: {x: 0.0, y: 0.0, z: 0.2}}"
```



```
root@jetson-desktop:~# ros2 topic pub /cmd_vel geometry_msgs/msg/Twist "{linear: {x: 0.5, y: 0.0, z: 0.0}, angular: {x: 0.0, y: 0.0, z: 0.2}}"
publisher: beginning loop
publishing #1: geometry_msgs.msg.Twist(linear=geometry_msgs.msg.Vector3(x=0.5, y=0.0, z=0.0), angular=geometry_msgs.msg.Vector3(x=0.0, y=0.0, z=0.2)
)

publishing #2: geometry_msgs.msg.Twist(linear=geometry_msgs.msg.Vector3(x=0.5, y=0.0, z=0.0), angular=geometry_msgs.msg.Vector3(x=0.0, y=0.0, z=0.2)
)

publishing #3: geometry_msgs.msg.Twist(linear=geometry_msgs.msg.Vector3(x=0.5, y=0.0, z=0.0), angular=geometry_msgs.msg.Vector3(x=0.0, y=0.0, z=0.2)
)

publishing #4: geometry_msgs.msg.Twist(linear=geometry_msgs.msg.Vector3(x=0.5, y=0.0, z=0.0), angular=geometry_msgs.msg.Vector3(x=0.0, y=0.0, z=0.2)
)

publishing #5: geometry_msgs.msg.Twist(linear=geometry_msgs.msg.Vector3(x=0.5, y=0.0, z=0.0), angular=geometry_msgs.msg.Vector3(x=0.0, y=0.0, z=0.2)
)
```

4. Program core source code analysis

Take Mcnamu_driver_X3.py as an example,

```
from MutoLib import Muto    #Import driver library

self.muto = Muto()  #Instantiating a Muto object

#create subcriber Create a subscriber
self.sub_cmd_vel =
self.create_subscription(Twist,"cmd_vel",self.cmd_vel_callback,1)
self.sub_BUzzer =
self.create_subscription(Bool,"Buzzer",self.Buzzercallback,100)

#create publisher Create publisher
self.EdiPublisher = self.create_publisher(Float32,"edition",100)
self.volPublisher = self.create_publisher(Float32,"voltage",100)
self.imuPublisher = self.create_publisher(Imu,"/imu/data_raw",100)
self.magPublisher = self.create_publisher(MagneticField,"/imu/mag",100)

#Call the library to read the information of the ros expansion board
version =int(self.muto.read_version(),16)
edition.data = version*1.0
battery.data = self.muto.read_battery()
ax, ay, az , gx, gy, gz, mx, my, mz= self.muto.read_IMU_Raw()

#Publish topic data
self.imuPublisher.publish(imu)
self.magPublisher.publish(mag)
self.volPublisher.publish(battery)
self.EdiPublisher.publish(edition)

#Subscriber callback function
def cmd_vel_callback(self,msg)
def Buzzercallback(self,msg):
```

Please refer to the code for detailed code:muto_driver.py