2. Robot information release

1. Program Function Description

After running the program, combined with the ROS expansion board, you can get the information of the sensors on the ROS expansion board, control the movement of the trolley, control the light strip, buzzer and other functions.

2. Program Code Reference Path

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

```
/root/yahboomcar_ros2_ws/yahboomcar_ws/src/yahboomcar_bringup/yahboomcar_bringup
```

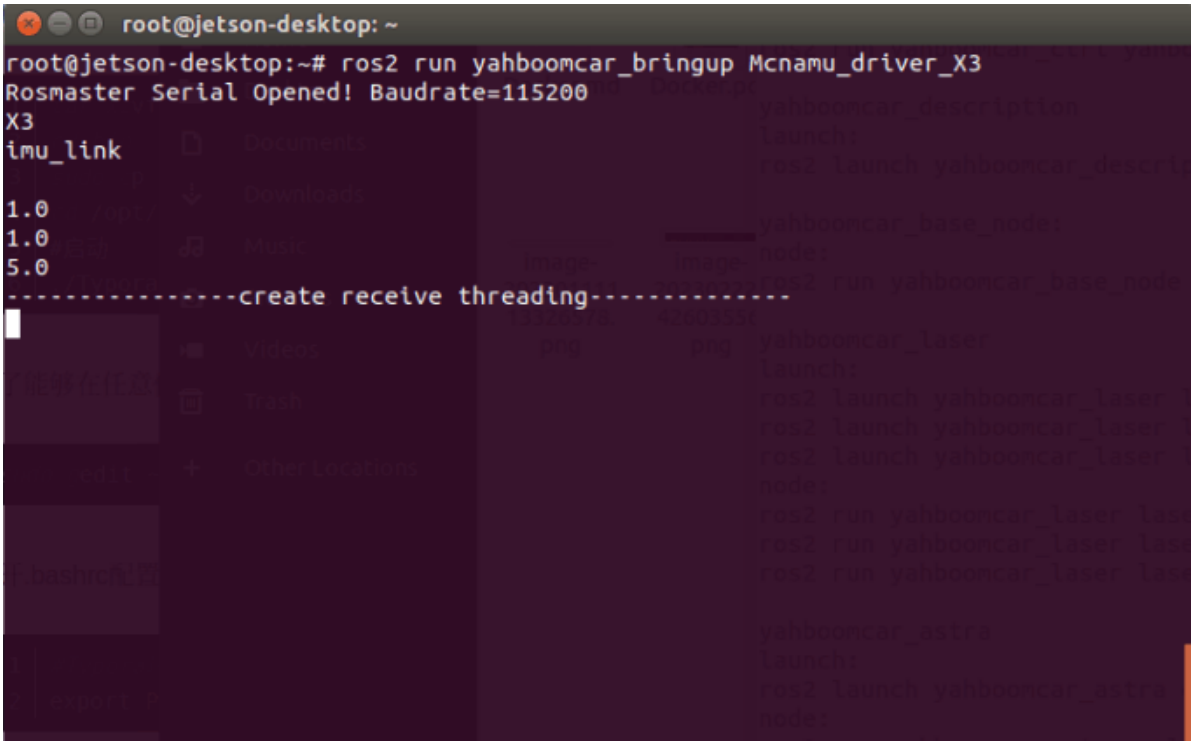
Take X3 model for example, **Mcnamu_driver_X3.py**, is the chassis driver code for X3 model.

3. Program startup

3.1. Startup Commands

After entering the docker container, according to the actual model, terminal input, the

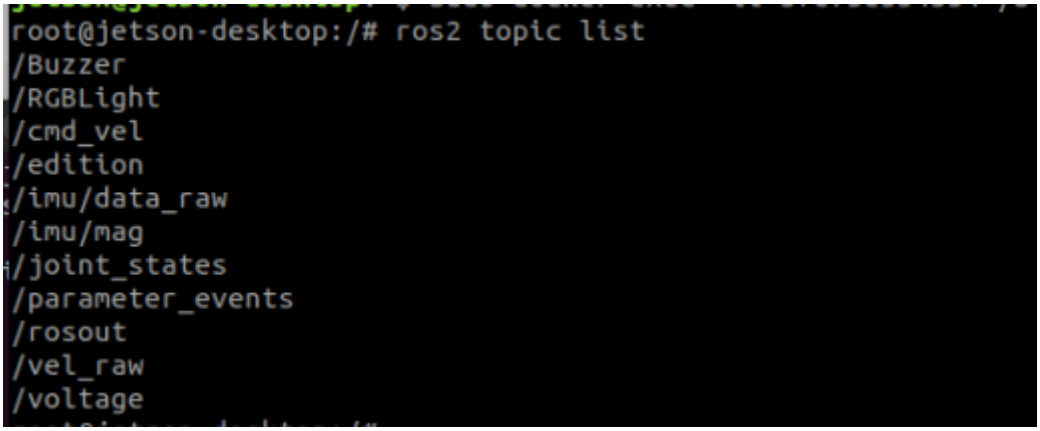
```
ros2 run yahboomcar_bringup Mcnamu_driver_X3 #X3车型#X3 model
ros2 run yahboomcar_bringup Ackman_driver_R2 #R2车型#R2 model
```



3.2. Viewing Node Topics

Open a terminal and enter the container and type, the

```
ros2 topic list
```

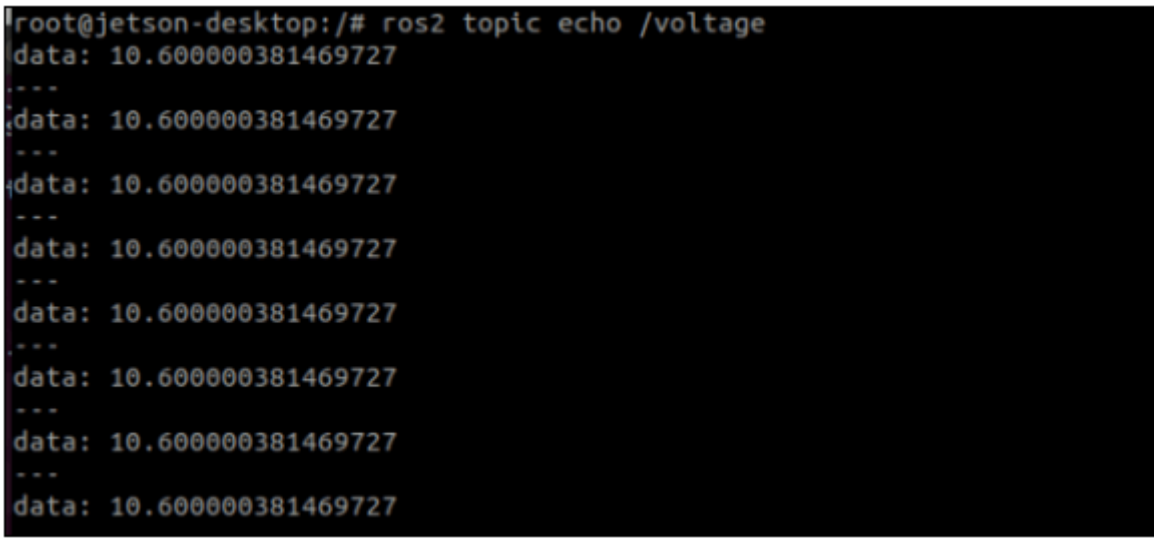


topic name	Topic content
/Buzzer	buzzer
/RGBLight	Light strip effect control
/cmd_vel	speed control
/edition	version information
/imu/data_raw	IMU sensor data
/imu/mag	MU-Magnetometer datal
/vel_raw	Trolley speed information
/voltage	Battery voltage information

3.3. Reading topic data

Open the terminal to enter the container and enter, for example, to read the voltage magnitude

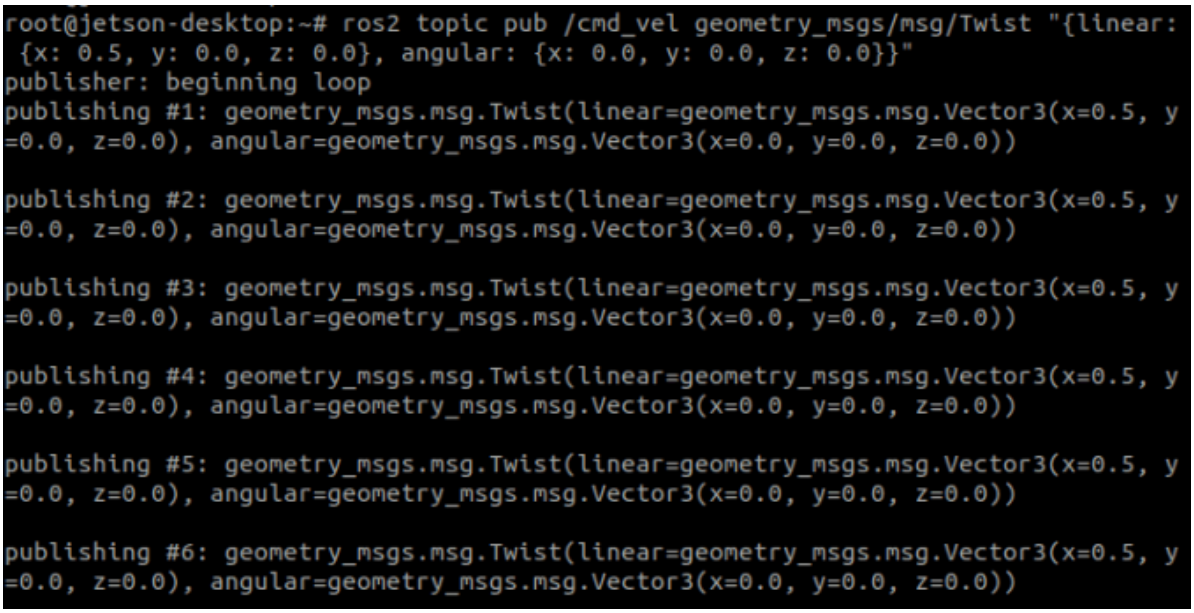
```
ros2 topic echo /voltage
```



3.4. Publishing topic data

Open the terminal to enter the container and enter, to publish /cmd_vel data to control the movement of the cart 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}}"
```



4. Program core source code analysis

Take Mcnamu_driver_X3.py as an example.

```
from Rosmaster_Lib import Rosmaster #导入驱动库#Importing driver libraries

self.car = Rosmaster()  #实例化Rosmaster对象
# Instantiate the Rosmaster object

#create subcriber 创建订阅者
#create subcriber Create subscriber
self.sub_cmd_vel =
self.create_subscription(Twist,"cmd_vel",self.cmd_vel_callback,1)
self.sub_RGBLight =
self.create_subscription(Int32,"RGBLight",self.RGBLightcallback,100)
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.staPublisher = self.create_publisher(JointState,"joint_states",100)
self.velPublisher = self.create_publisher(Twist,"vel_raw",50)
self.imuPublisher = self.create_publisher(Imu,"/imu/data_raw",100)
self.magPublisher = self.create_publisher(MagneticField,"/imu/mag",100)

#调用库，读取ros拓展板的信息
#call library to read ros expansion boards
edition.data = self.car.get_version()*1.0
battery.data = self.car.get_battery_voltage()*1.0
ax, ay, az = self.car.get_accelerometer_data()
```

```
gx, gy, gz = self.car.get_gyroscope_data()
mx, my, mz = self.car.get_magnetometer_data()
vx, vy, angular = self.car.get_motion_data()

#发布话题数据
#Publishing topic data
self.imuPublisher.publish(imu)
self.magPublisher.publish(mag)
self.volPublisher.publish(battery)
self.EdiPublisher.publish(edition)
self.velPublisher.publish(twist)

#订阅者回调函数
#Subscriber callback function
def cmd_vel_callback(self,msg)
def RGBLightcallback(self,msg)
def Buzzercallback(self,msg):
```

Please refer to the code Mcnamu_driver_X3.py for detailed code.