3. Robot information release

According to different models, just set the purchased model in [.bashrc], X1 (normal four-wheel drive) X3 (Mailun) X3plus (Mailun robotic arm) R2 (Ackerman differential) etc. , this section takes X3 as an example

```
#Raspberry Pi 5 master needs to enter docker first, please perform this step
#If running the script into docker fails, please refer to ROS/07, Docker tutorial
~/run_docker.sh
#Multiple ros commands require multiple terminals to be executed in the same
docker container. Please refer to the tutorials in Sections 07/5 and 5.8.
```

Open the [.bashrc] file

```
sudo vim .bashrc
```

Find the [ROBOT_TYPE] parameter and modify the corresponding model

```
export ROBOT_TYPE=X3 # ROBOT_TYPE: X1 X3 X3plus R2 X7
```

This section takes X3, Mecanum wheel vehicle as an example.

3.1. Node subscription and publishing topics

3.1.1. Function package path:

```
~/yahboomcar_ws/src/yahboomcar_bringup
```

Functions that ROSMASTER needs to implement: speed control, speed information feedback, battery voltage feedback, buzzer control, and running water light control. (Note: **In the version with a robotic arm, robotic arm control, robotic arm status feedback and PTZ control also need to be implemented**)

3.2. View node data

3.2.1. Start

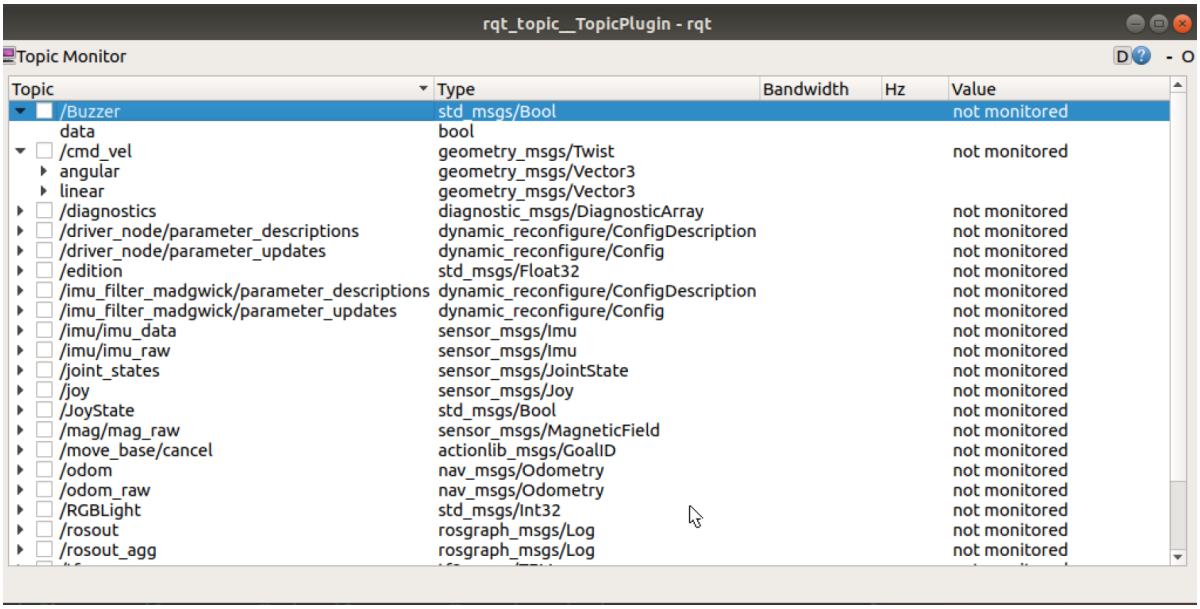
```
roslaunch yahboomcar_bringup yahboomcar.launch
```

3.2.2, View topic list

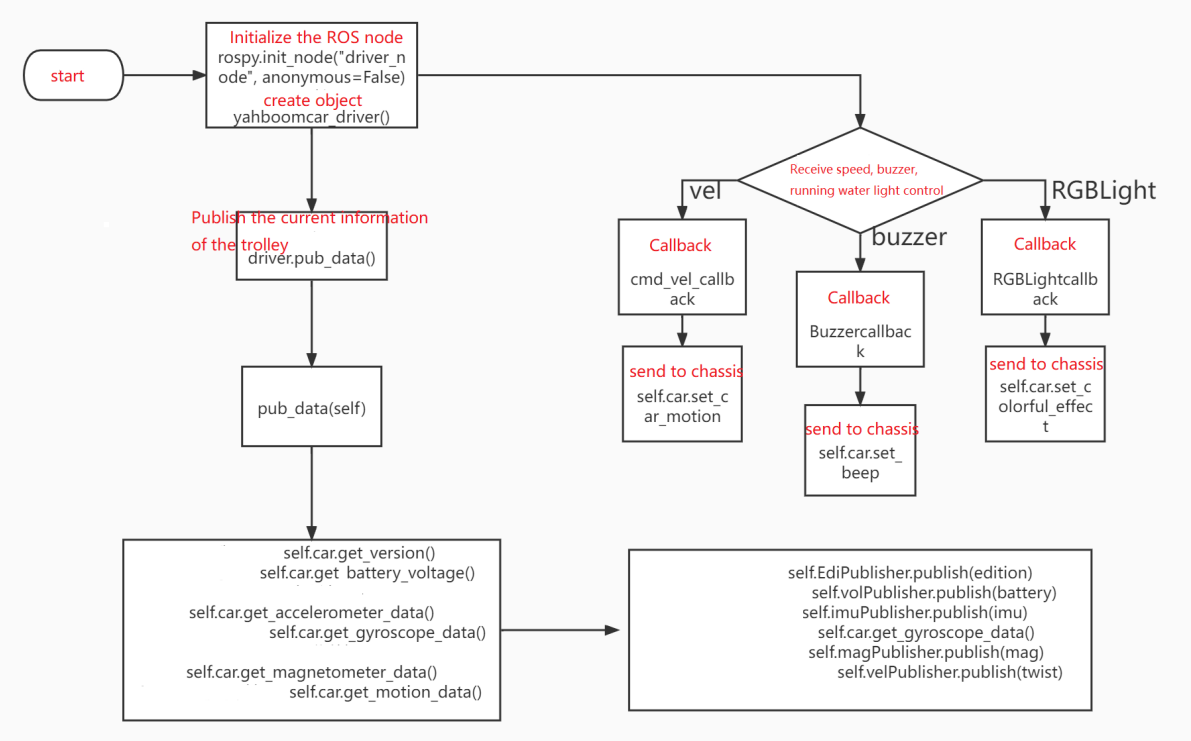
```
rostopic list
```

```
----------------------
MY_IP: 192.168.2.103
ROS_MASTER_URI:
http://192.168.2.103:11311
my_robot: X3 | my_lidar: a1
----------------------
pi@yahboom:~$ rostopic list
/Buzzer
/RGBLight
/cmd_vel
/driver_node/parameter_descriptions
/driver_node/parameter_updates
/edition
/imu/imu_raw
/joint_states
/mag/mag_raw
/rosout
/rosout_agg
/vel_raw
/voltage
pi@yahboom:~$
```

```
rosrun rqt_topic rqt_topic #View topics graphically
```



3.2.3. Program flow chart



3.2.4, core code (Mcunamu_driver.py)

```
Get data (core board -> host computer)
   edition.data = self.car.get_version()
   battery.data = self.car.get_battery_voltage()
   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()
Publish data (host computer -> host computer)
   self.EdiPublisher.publish(edition)
   self.volPublisher.publish(battery)
   self.imuPublisher.publish(imu) Note: The imu data here is a combination of
gyroscope and acceleration data.
   self.magPublisher.publish(mag)
   self.velPublisher.publish(twist)
Processing of acquired data (topic receiving data, transmission of data between
nodes)
   cmd_vel_callback(self, msg)
   RGBLightcallback(self, msg)
   Buzzercallback(self, msg)
Release data (host computer -> core board)
   self.car.set_car_motion(vx, vy, angular)
   self.car.set_colorful_effect(msg.data, 6, parm=1)
   self.car.set_beep(1) or self.car.set_beep(1)
```

3.2.5. Analysis of three callback functions

```
    # Running water light control, server callback function RGBLight control
    '''
    effect=[0, 6], 0: Stop light effect, 1: Running water light, 2: Marquee
light, 3: Breathing light, 4: Gradient light, 5: Starlight, 6: Battery display
```

```
      speed=[1, 10], the smaller the value, the faster the speed changes.
      '''
      # Car motion control, subscriber callback function
      '''
       vx = msg.linear.x
       vy = msg.linear.y
       angular = msg.angular.z
       Note: Because this model is a Mecanum wheel, it can move on the y-axis. It
is not applicable to other models.
       '''
      #Buzzer control, subscriber callback function
      '''
      self.car.set_beep(1): Turn on the buzzer
      self.car.set_beep(0): Turn off the buzzer
      '''
```

3.2.6. Use the rostopic pub command to send running light control, speed control, and buzzer control commands.

```
Running water light control
rostopic pub /RGBLight std_msgs/Int32 1 #Turn on the running water light
speed control
rostopic pub /cmd_vel geometry_msgs/Twist "linear:
   x: 0.0
   y: 0.0
   z: 0.0
angular:
   x: 0.0
   y: 0.0
   z: 0.1" #The car moves at an angular speed of 0.1
Buzzer control
rostopic pub /Buzzer std_msgs/Bool true #Turn on the buzzer, send false if turned
off
```

3.2.7. Use the rostopic echo command to view speed information and battery voltage

```
speed information
rostopic echo /cmd_vel
battery voltage
rostopic echo /voltage
```