LIMO 产品用户使用和开发手册

LIMO User and Development Manual

LIMO 产品用户使用和开发手册

LIMO Use and Development Manual

中 文 | EN Version:

1.0.0

CN | EN Version:

1.0.0

版本 Version	更新信息 Update information	责任人 Responsible
1.0.0	[1]、第一版本开放 [1]. The first version is open	Agilex ROS TEAM

一、LIMO产品简介

1. LIMO Introduction

1.1 产品简介

1.1 Introduction

松灵机器人LIMO是全球首款集四种运动模态于一体的ROS开发平台，提供了适应场景更广泛、更符合行业应用要求的学习平台，适用于机器人教育、功能研发、产品开发。通过创新性的机械设计，能实现四轮差速、阿克曼、履带型、麦克纳姆轮运动模式的快速切换，可在配套的专业沙盘中快速建立多场景实拟教学和测试，LIMO搭载NVIDIA Jeston Nano、EAI XL2激光雷达、深度相机等高性能传感器配置，可实现精确的自主定位、SLAM建图、路线规划和自主避障、自主倒车入库、红绿灯识别等丰富功能。

AgileX Robotics LIMO is the world's first ROS development platform that integrates four motion modes. It provides a learning platform that adapts to a wider range of scenarios and is more in line with industry application requirements. It is suitable for robot education,

function research and development, and product development. Through innovative mechanical design, it can realize the fast switching of four-wheel differential, Ackermann, track-type, and Mecanum wheel motion modes, and can quickly establish multi-scene practical teaching and testing in the supporting professional sand table. LIMO is equipped with NVIDIA Jeston Nano, EAI XL2 LiDAR, depth camera and other high-performance sensor configurations, which can realize rich functions such as precise autonomous positioning, SLAM mapping, route planning, autonomous obstacle avoidance, autonomous reverse stall parking, traffic light recognition and so on.

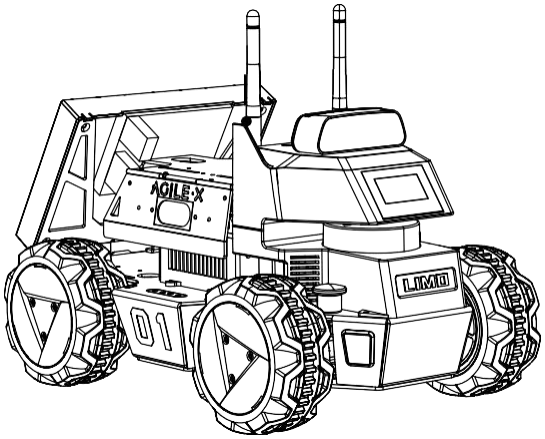
同时，松灵机器人联合国内ROS社区教学开创者古月居，致力于结合企业用人和行业应用需求，打造基于LIMO开发小车的全新ROS精品课程，助力院校科研教学，让学生达到更高的行业应用技术要求。

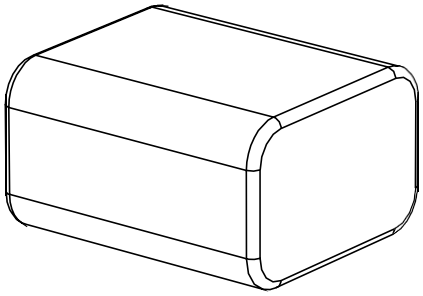
At the same time, AgileX Robotics and Gu Yueju, the domestic ROS community teaching pioneer, are committed to combining enterprise employment and industry application needs to create a new ROS boutique course based on the LIMO-developed car to help colleges and universities in scientific research and teaching, so that students can meet higher technical requirements for industry applications.

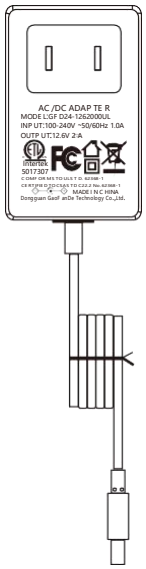
1.2 产品列表

1.2 Component list

名称 Name	数量 Quantity
LIMO高配版主体 LIMO high-end body	X1
电池 Battery	X1
充电器 Charger	X1
越野轮 Off-road wheel	X4
麦克纳姆轮 Mecanum wheel	X4
履带 Tracked wheel	X2
十字螺丝刀 Cross screwdriver	X1
螺丝 Screw	M3*12mmx20*、*M3*5mmx20



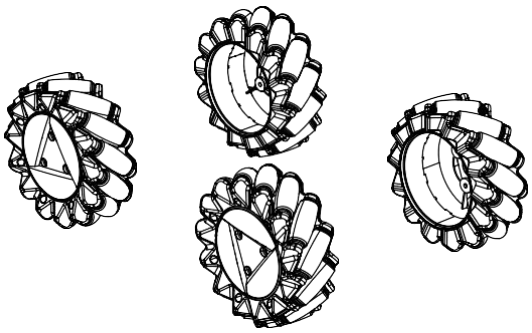


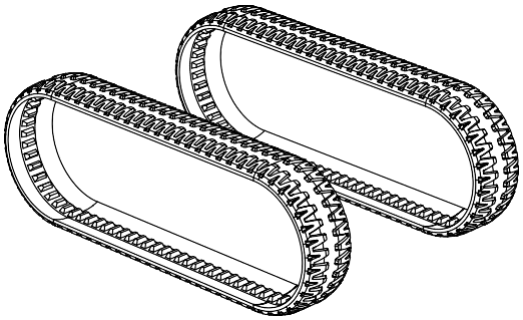


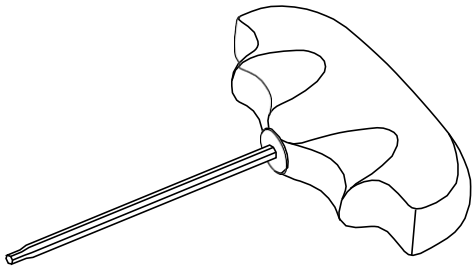
LIMO高配版主体（安装越野轮X4）X1
LIMO high-end body (install off-road wheel X4) X1

电池X1
Battery X1

充电器X1
Charger X1







麦克纳姆轮X4

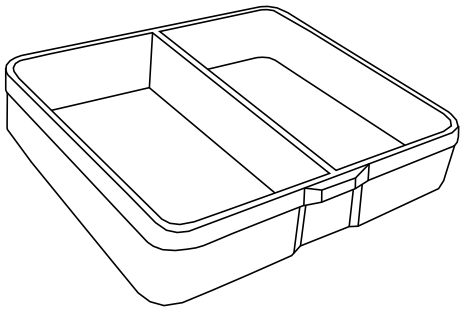
Mecanum wheel X4

履带X2

Track X2

内六角螺丝刀X1

Hexagon screwdriver X1



螺丝盒X1 螺丝盒内容：M3*12mmx20*、*M3*5mmx20

Screw box X1 Screw box content: M312mmx20, M35mmx20

1.3 性能参数

1.3 Tech specifications

参数类型 Parameter Types	项目 Items	指标 Values
机械参数 Mechanical parameters	外形尺寸 Overall dimension	322*220*251mm
	轴距 Wheel base	200mm
	轮距 Tread	175mm
	自重 Dead load	4.8kg
	负载 Load	四轮差速（1kg） Four-wheel differential (1kg)
		阿克曼模式（4kg） Ackermann mode (4kg)
	最小离地间隙 Minimum ground clearance	24mm
	驱动方式 Drive type	轮毂电机(4x14.4W) Hub motor (4x14.4W)
性能参数 Performance parameters	空载最高车速 No-load highest speed	1m/s
	阿克曼最小转弯半径 Ackermann minimum turning radius	0.4m
	工作环境 Work environment	-10~+40℃
	最大爬坡角度 Maximum climbing capacity	40°（履带模式下） 40°(under track mode)
系统参数 System parameters	电源接口 Power interface	DC（5.5x2.1mm)
	系统 System	Ubuntu18.0
	IMU	MPU6050
	CPU	ARM 64位四核@1.43GHz （Cortex-A57） ARM 64-bit 4-core @1.43GHz (Cortex-A57)
	GPU	128核 NVIDIA Maxwell @921MHz 128-core NVIDIA Maxwell @921MHz
	电池	5200mAh 12V

	Battery	
	工作时间 Working time	40min
	待机时间 Stand-by time	2h
	通讯接口 Communication interface	WIFI、蓝牙 WIFI, Bluetooth
传感器 Sensor	激光雷达 LiDAR	EAI X2L
	深度相机 Depth camera	奥比中光 DaBai/RealSense D435 Orbbec DaBai/RealSense D435
	工控机 IPC	NVIDIA Jetson Nano（4G）
	语音模块 Voice module	讯飞语音助手/谷歌助手 iFlytek Voice Assistant/Google Assistant
	扬声器 Speaker	左右双声道（2x2W) Left and right dual channels (2x2W)
	USB-HUB	TYPE-C x1、USB2.0 x2
	前显示器 Front display	1.54寸128x64白色OLED显示屏 1.54 inch 128x64 white OLED display screen
	后显示器 Rear display	7寸1024x600 IPS触控屏 7 inch 1024x600 IPS touch screen
控制参数 Control parameters	控制模式 Control mode	手机APP、指令控制 Mobile APP, command control

	手机APP Mobile APP	蓝牙/极限距离10m Bluetooth/Maximum distance 10m

1.4 Nvidia Jetson Nano介绍

1.4 Nvidia Jetson Nano introduction

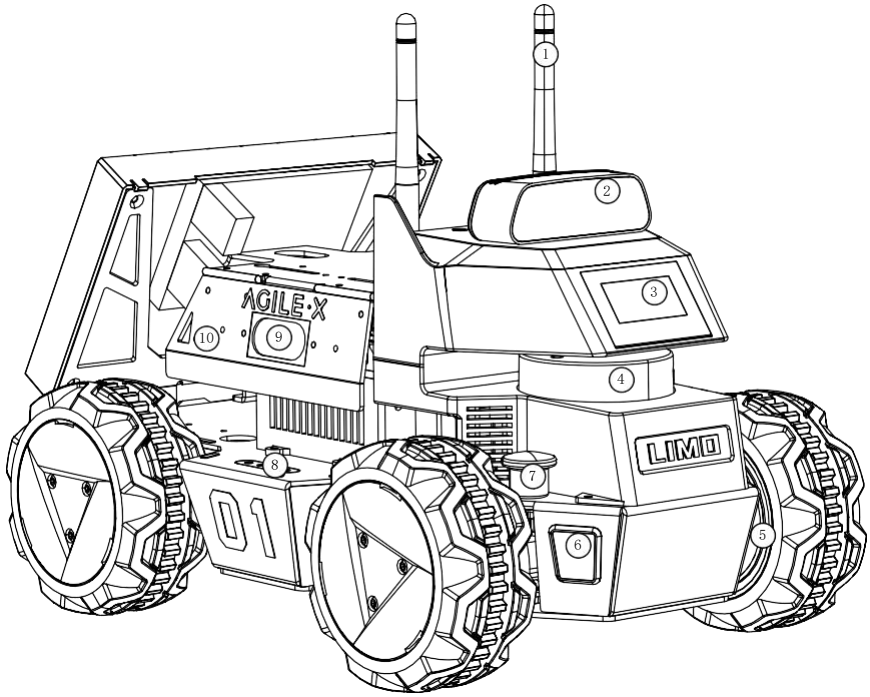
Nvidia Jetson Nano 是一款功能强大的小型计算机，专为支持入门级边缘 AI 应用程序和设备而设计。依托完善的 NVIDIA JetPack™ SDK 包含用于深度学习、计算机视觉、图形、多媒体等方面的加速库。搭载在limo高配版本，可以用于拓展机器人导航定位、图像处理、语音识别等技术的拓展。

Nvidia Jetson Nano is a powerful small computer designed to support entry-level edge AI applications and devices. Relying on the comprehensive NVIDIA JetPack™ SDK, it contains acceleration libraries for deep learning, computer vision, graphics, multimedia, etc. Equipped in the limo high-end version, it can be used for the expansion of robot navigation and positioning, image processing, voice recognition etc.

GPU	**128-Core Maxwell**
CPU	Quad-core ARM57 @1.43Ghz
内存 Memory	4GB 64Bit LPDDR4 25.6GB/s
存储 Storage	Micro SD卡（默认） Micro SD card (default)
视频编码 Video encode	4K@30| 4 X 1080p@30 | 9 X 720p@30(H.264/H.265)
视频解码 Video decode	4K@60| 2X 4K@30 | 8X 1080p@30 | 18 X 720p@30(H.264/H.265)
摄像头 Camera	2 X MIPI CSI-2 DPHY lanes
联网 Networking	千兆以太网，M.2 Key E 接口外扩 Gigabit Ethernet, external expansion of M.2 Key E interface
显示 Display	HDMI X 1, DP X 1
USB	4 X USB 3.0, USB 2.0 Micro-B
扩展接口 Extended interface	GPIO，I2C，I2S，SPI，UART

1.5 部件名称

1.5 Component name



① WIFI/蓝牙天线；

① WIFI/Bluetooth antenna;

② 深度相机；

② Depth camera;

③ 前显示器；

③ Front display;

④ EAI X2L激光雷达;

④ EAI X2L LiDAR;

⑤ 轮毂电机；

⑤ Hub motor;

⑥ RGB车灯；

⑥ RGB vehicle light;

⑦ 四轮差速/阿克曼模式切换插销；

⑦ Four-wheel differential/Ackermann mode switching latch;

⑧ 电量显示；

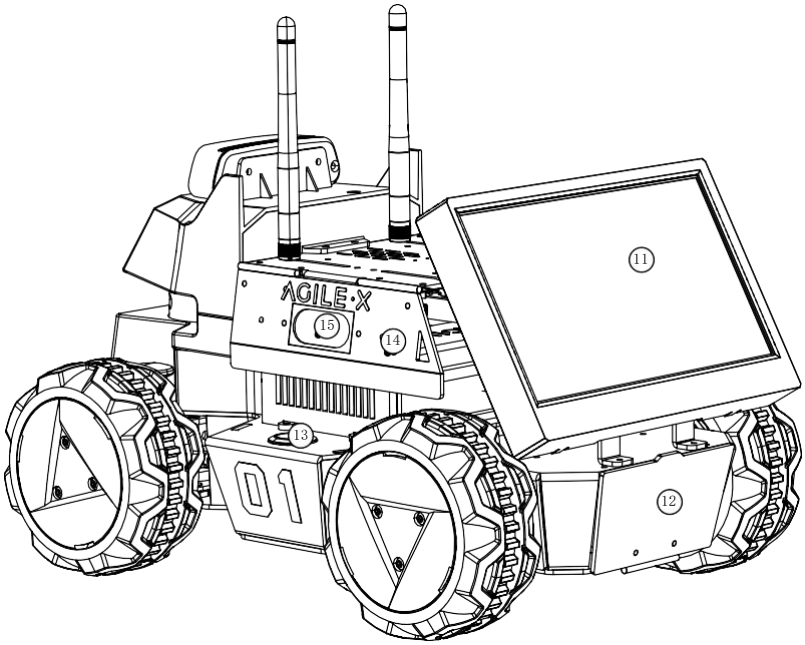
⑧ Power display;

⑨ 左扬声器；

⑨ Left speaker;

⑩ 左海鸥门；

⑩ Left seagull door;



⑪ 后显示器；

⑪ Rear display;

⑫ 电池门；

⑫ Battery door;

⑬ 开 关 ；

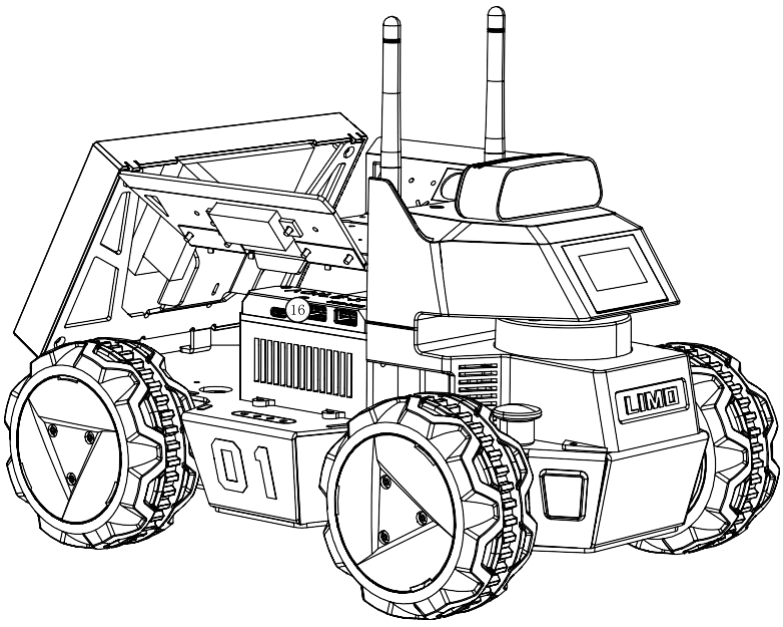
⑬ Switch;

⑭ 右海鸥门；

⑭ Right seagull door;

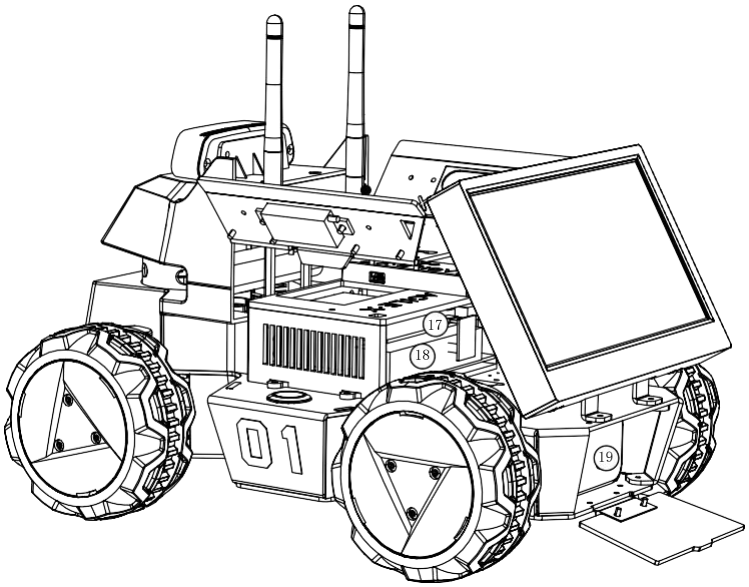
⑮ 右扬声器；

⑮ Right speaker;



⑯ USB-HUB模块；

⑯ USB-HUB module;



⑰ 语音模块；

⑰ Voice module;

⑱ 工控机NVIDIA Jetson Nano（4G）；

⑱ IPC NVIDIA Jetson Nano (4G);

⑲ 电 池 ；

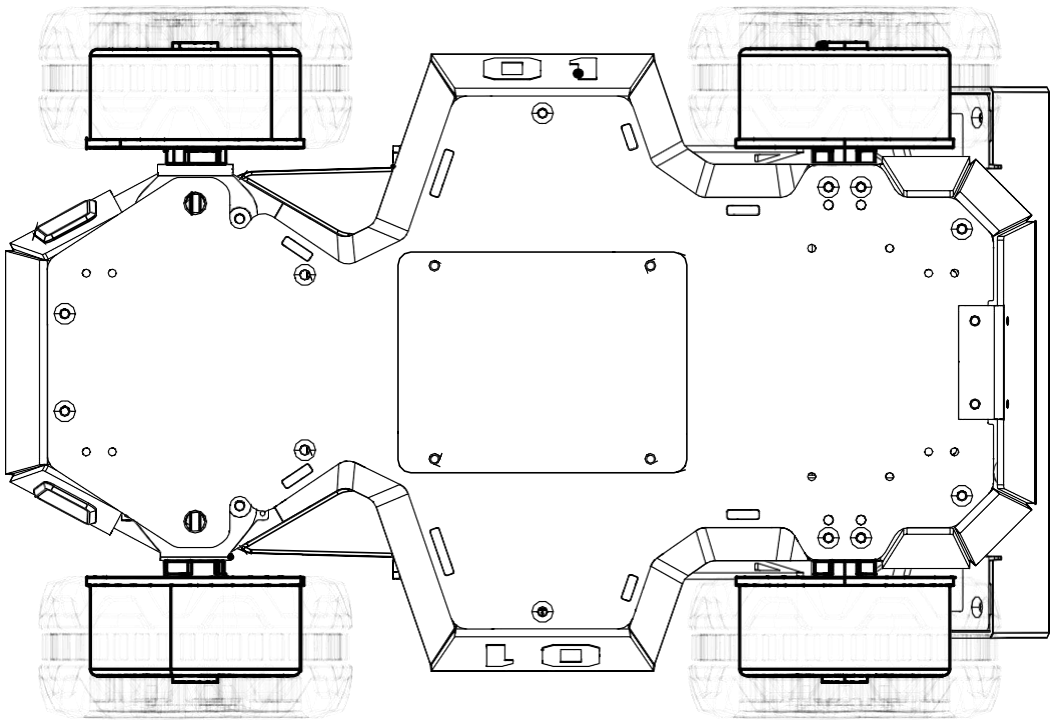
⑲ Battery;

1.6 功能亮点

1.6 Function highlights

（1）使用四个轮毂电机，节省车体内部空间，可在一个车体上实现阿克曼、四轮差速、履带和麦轮这四种模态的快速切换；

(1) Four hub motors are used to save the internal space of the vehicle body, and the four modes of Ackermann, four-wheel differential, track and Mecanum wheel can be quickly switched on one vehicle body;

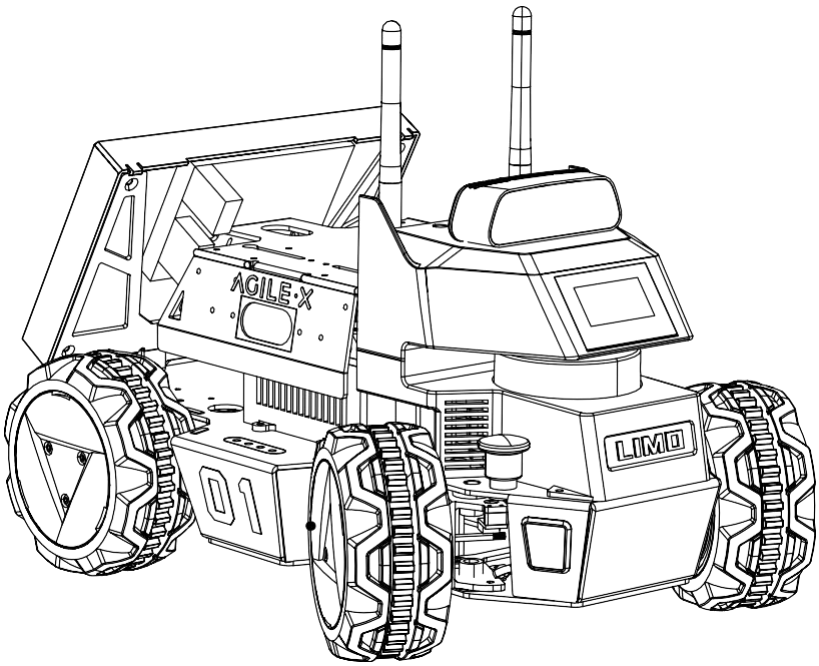


①阿克曼模式：

①Ackermann mode:

一种为了解决交通工具转弯时，内外转向轮路径指向的圆心不同的几何学，依据阿克曼转向几何设计的车辆，沿着弯道转弯时，利用四连杆的相等曲柄使内侧轮的转向角比外侧轮大大约2~4度，使四个轮子 路径的圆心大致上交会于后轴的延长线上瞬时转向中心，让车辆可以顺畅的转弯。

A geometry designed to solve the problem of wheels on the inside and outside of a turn needing to trace out circles of different radii in the steering of vehicles. A vehicle designed according to Ackermann steering geometry, when turning along a curve, use the four-link equal crank to make the steering angle of the inside wheel about 2 to 4 degrees larger than that of the outside wheel, so that the centers of the four wheel paths roughly meet on the extension line of the rear axle; and then the wheels instantly turn to the center, allowing the vehicle to turn smoothly.

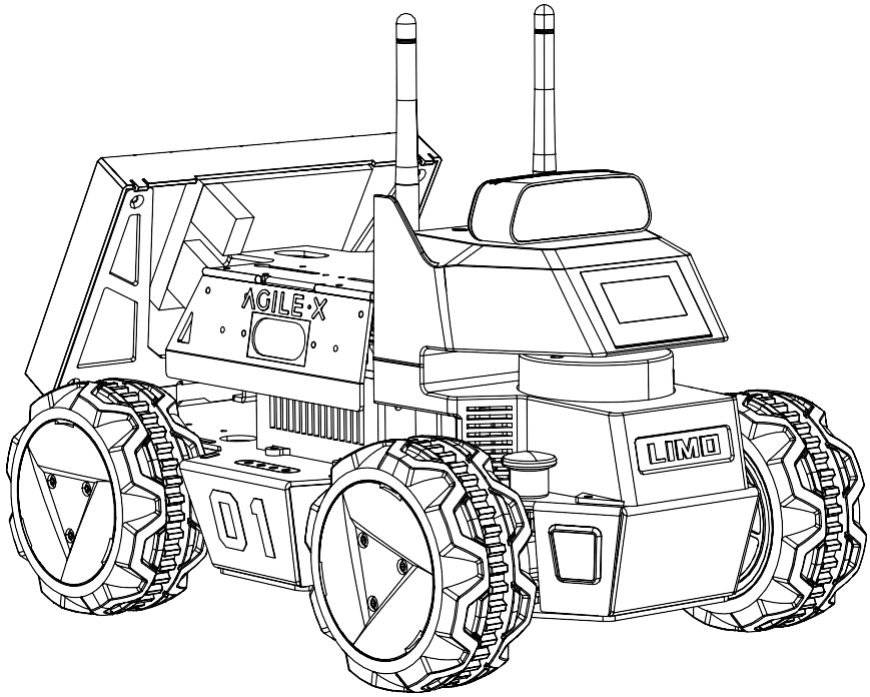


②四轮差速模式：

②Four-wheel differential mode:

四轮驱动，可实现原地自转，但对轮胎磨损严重，请勿长时间原地自转；

Four-wheel drive, which can realize in-situ auto-rotation, but it will cause serious tire wear; please do not auto-rotate in-situ for a long time;

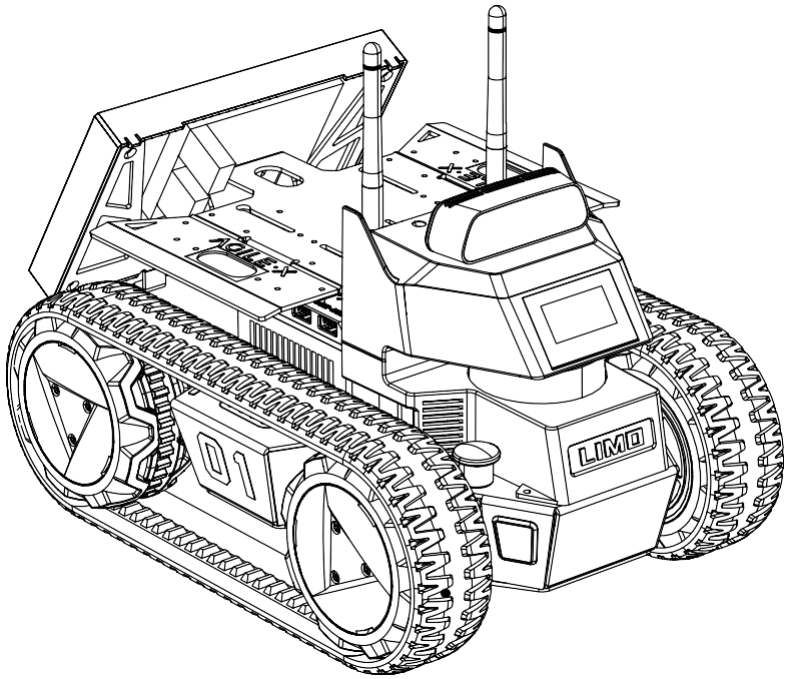


③履带模式：

③Track mode:

具有良好的越野性能，可上40°坡和小台阶；

It has good off-road performance and can climb 40° slopes and small steps;

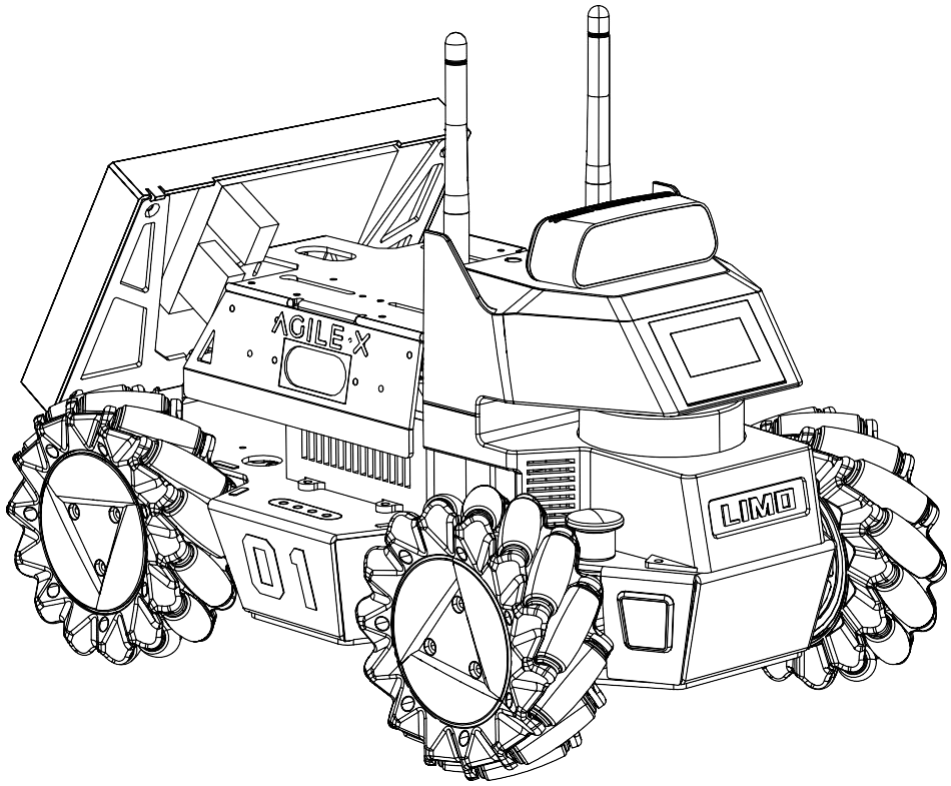


④麦克纳姆轮模式：

④Mecanum wheel mode:

基于麦克纳姆轮技术的全方位运动设备可以实现前行、横移、斜行、旋转及其组合等运动方式。

The omni-directional motion equipment based on Mecanum wheel technology can realize forward, lateral, oblique, rotation and combinations of motion modes.



（2）车灯状态指示：

(2) Indication of vehicle light status:

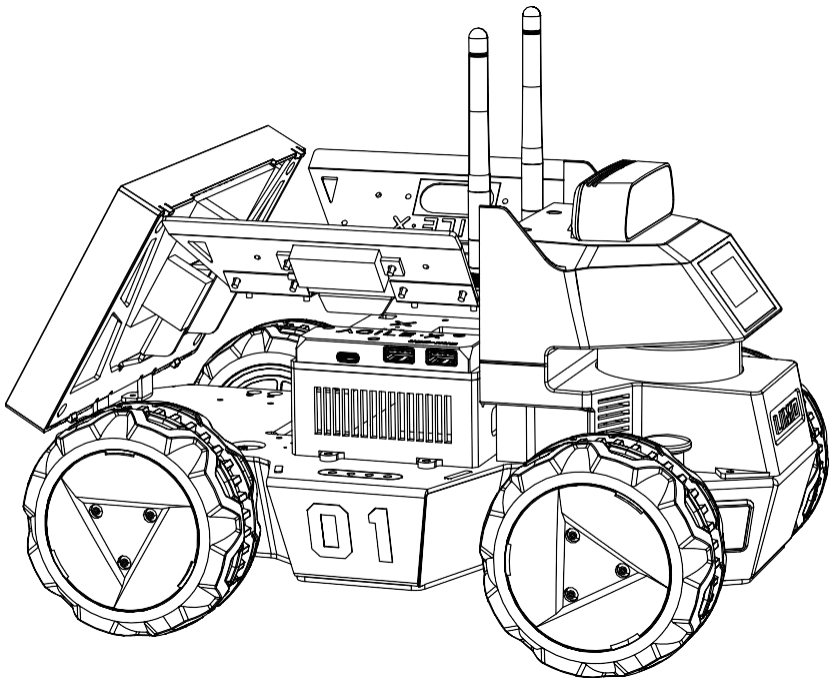
两车灯为RGB LED，选用5种对比度高的颜色作为指示灯，其余颜色可供开发者自定义；

The two vehicle lights are RGB LEDs, and 5 high-contrast colors are used for indicator lights, and the rest of the colors can be customized by the developer;

颜色 **Color**	**状态** **Status**
红色闪烁 Red flashing	低电量/主控报警 Low battery/master control alarm
绿色 Green	阿克曼模式 Ackermann mode
黄色 Yellow	四轮差速/履带模式 Four-wheel differential/track mode
蓝色 Blue	麦克纳姆轮模式 Mecanum wheel mode
白色 White	手动照明 Manual lighting

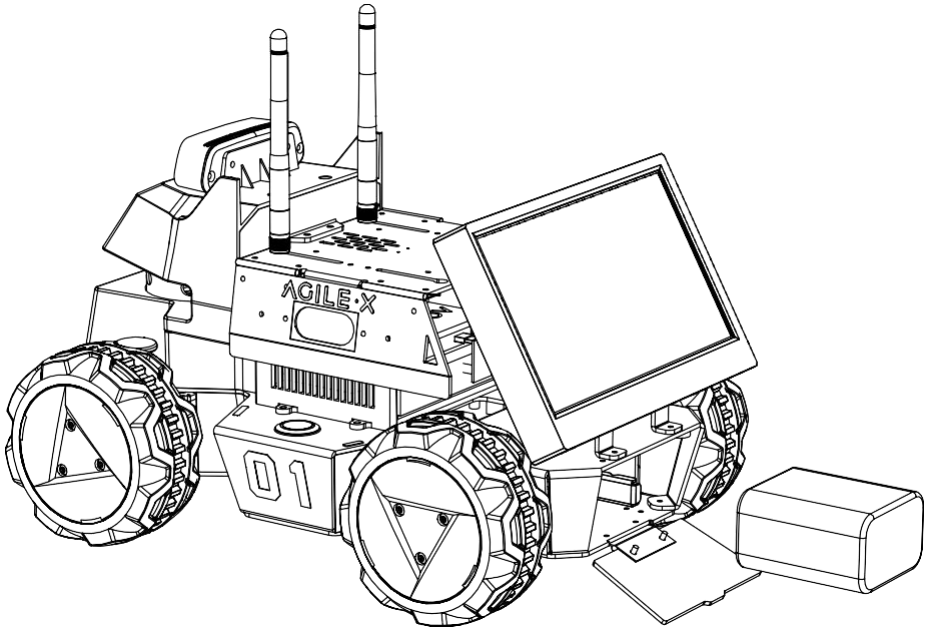
（3）两侧车门可展开，预留一个TYPE-C口和两个USB2.0口，方便调试；

(3) Doors on both sides can be expanded to reserve a Type-C interface and two USB2.0 interfaces for convenient debugging;



（4）电池可拆换；

(4) The battery can be removed and replaced;

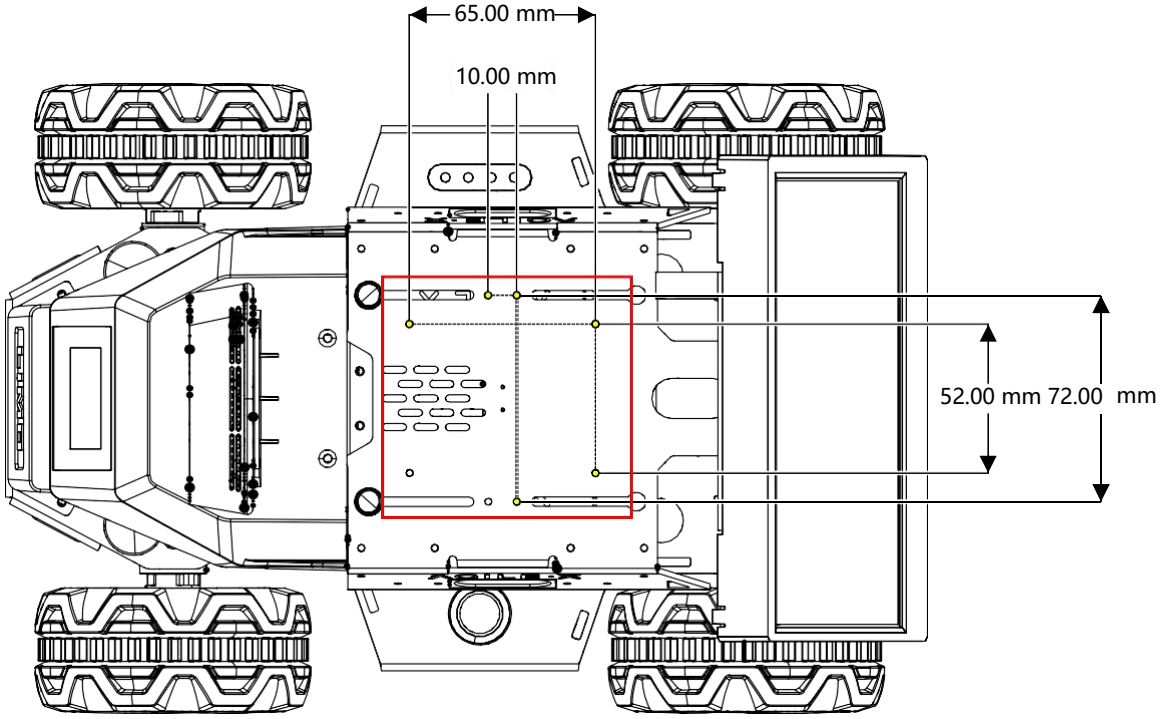


（5）预留丰富的拓展孔位：

(5) Reserve rich expansion holes:

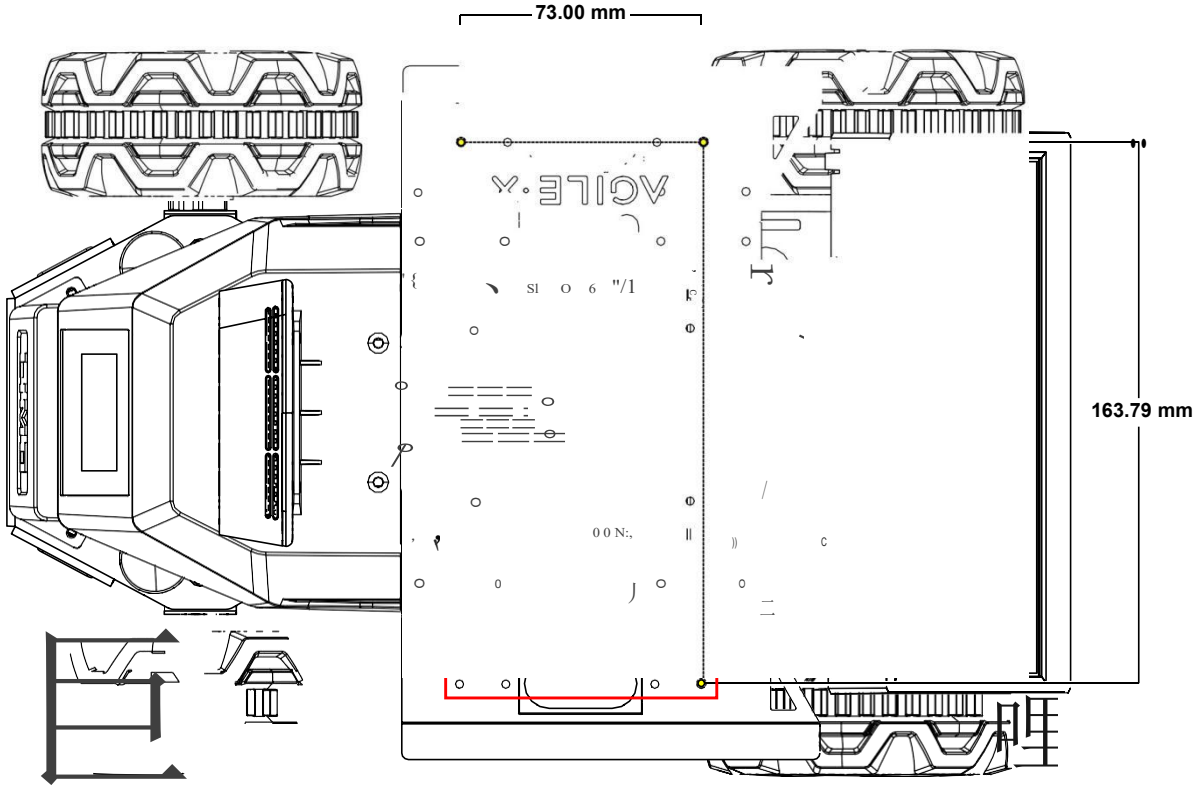
车顶预留8个M3螺丝孔位，以及两条3.2mm宽的槽口；

Eight M3 screw holes and two 3.2mm wide notches are reserved on the roof;



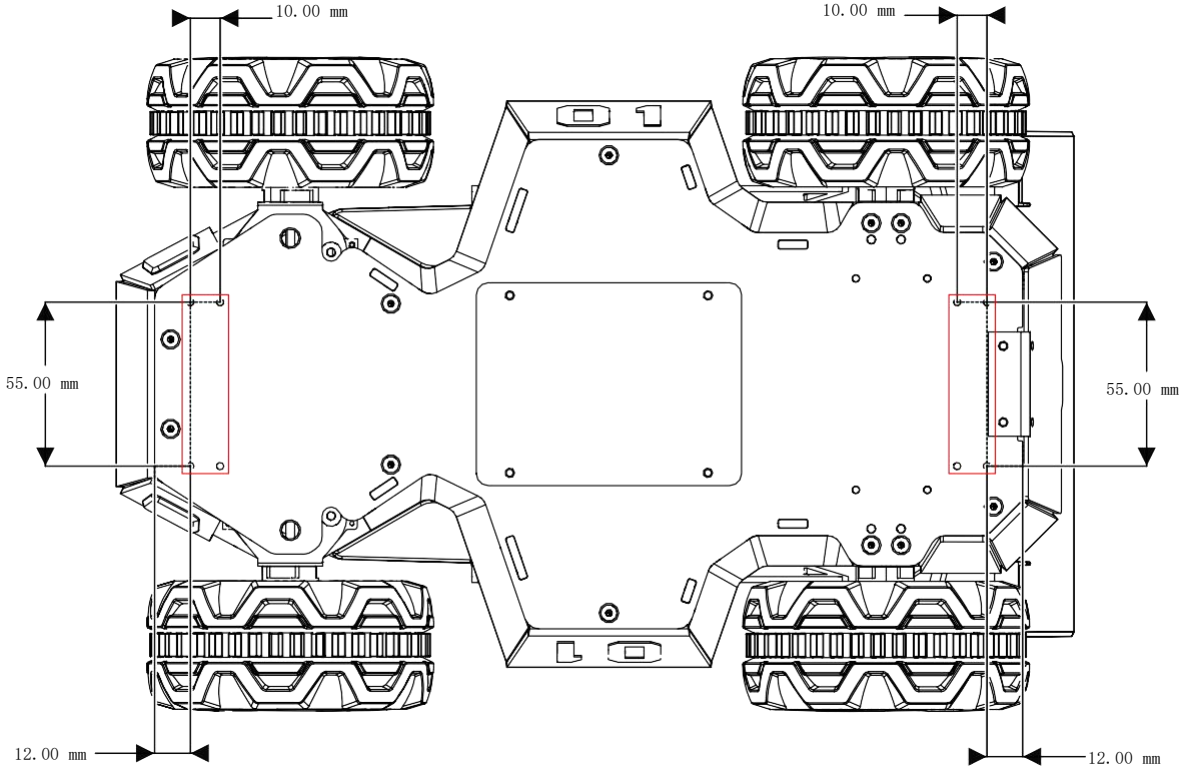
两车门预留4个M3螺丝孔位，水平展开获得更大安装平台；

Four M3 screw holes are reserved for the two doors to expand horizontally to obtain a larger mounting platform;



车底前后各预留四个M3螺丝孔位；

Four M3 screw holes are reserved at the front and rear of the underbody.

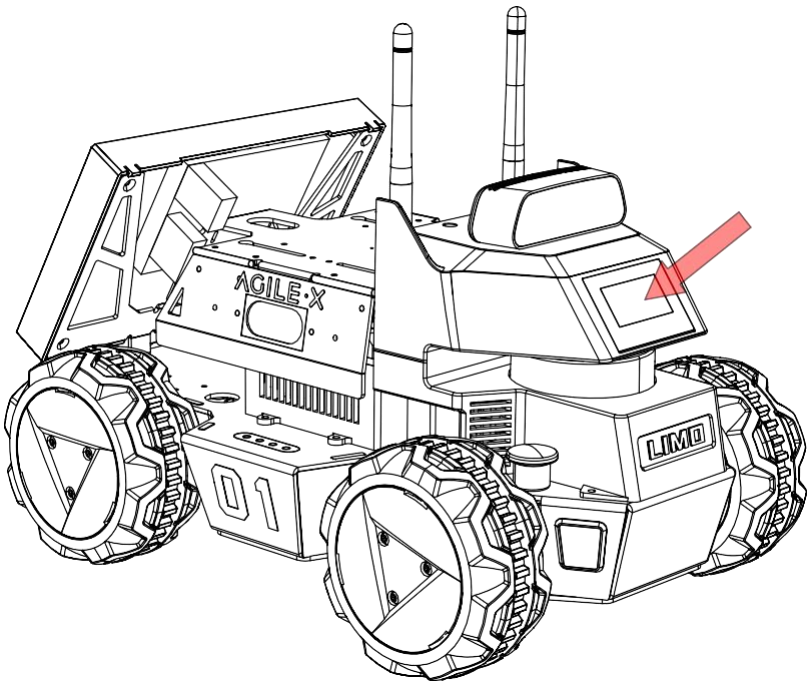


（6）丰富的交互体验：

(6) Rich interactive experience:

摄像头、激光雷达、语音模块、双扬声器配合前显示器可提供丰富的交互体验。

Camera, LiDAR, voice module, dual speakers and front display can provide rich interactive experience.



1.7 模态切换方法

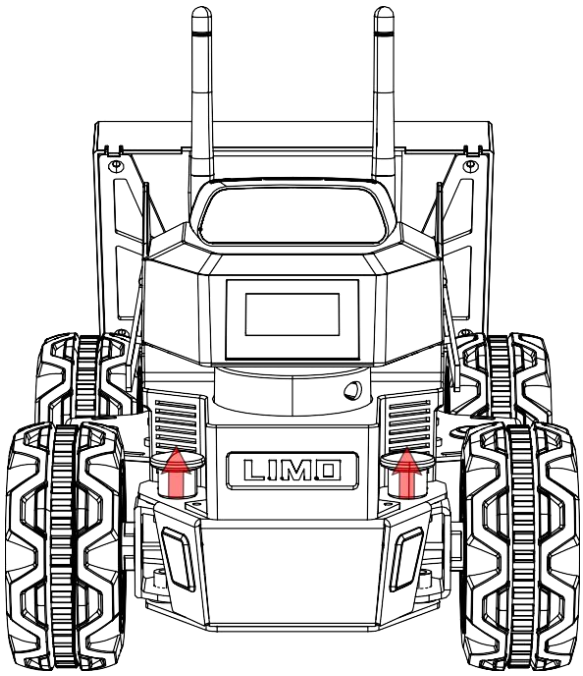
1.7 Mode switching method

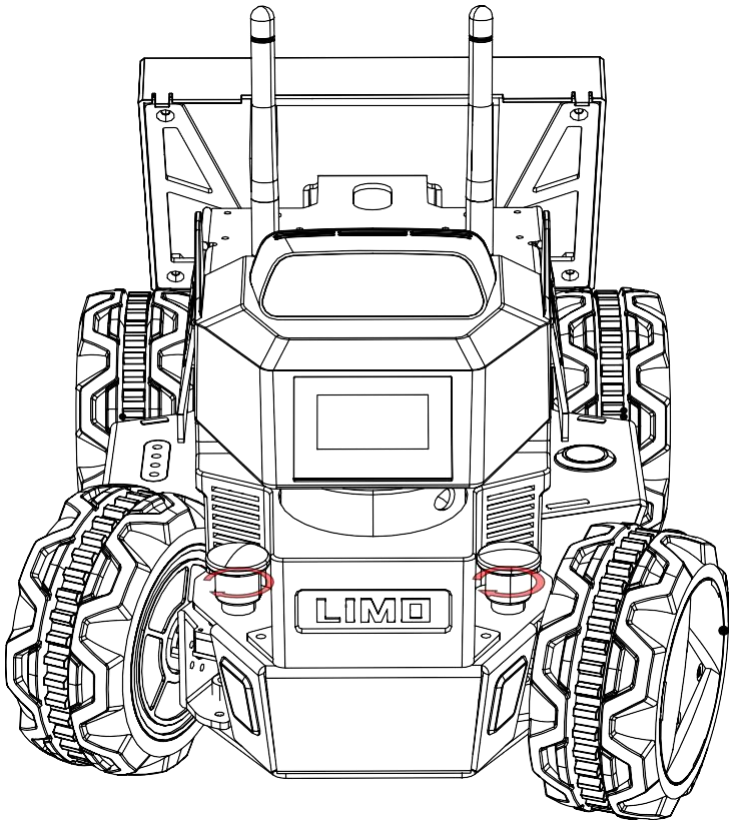
（1）切换阿克曼模式：

(1) Switch to Ackermann mode:

先将两侧插销拔起，顺时针转30度，使两插销上较长的线指向车体正前方  ，即可卡住，车灯变为绿色且常亮时，则切换成功；

First pull up the latches on both sides, and turn 30 degrees clockwise to make the longer line on the two latches points to the front of the vehicle body  , and then they will be stuck. When the vehicle light turns green and is steady on, the switch is successful;


LIMO

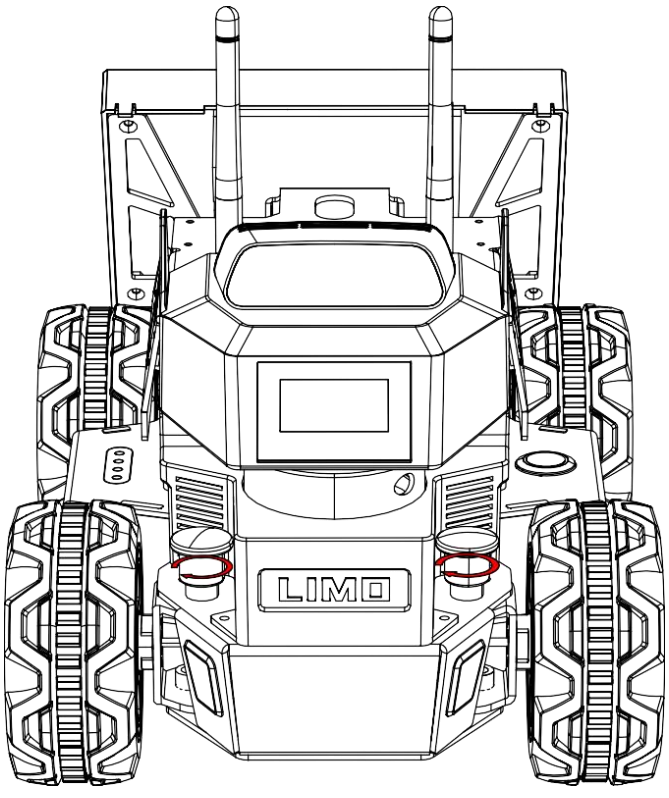


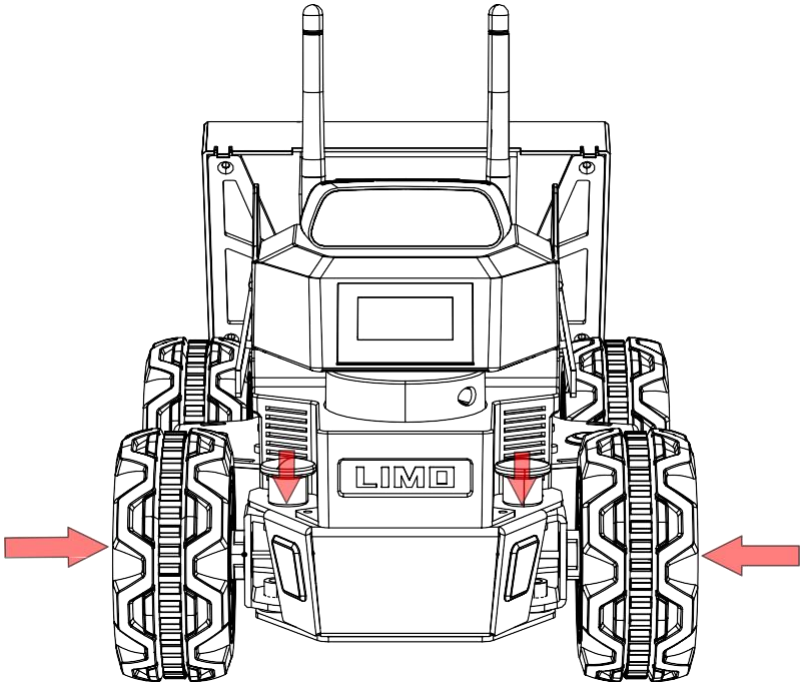
（2）切换四轮差速模式：

(2) Switch to four-wheel differential mode:

拔起来顺时针转30度，使两插销上较短的线指向车体正前  ，此时为插入状态，微调轮胎角度对准孔位让插销插入，车灯变为黄色且常亮时，则切换成功；

Pull up the two latches, and turn 30 degrees clockwise to make the shorter line on the two latches points to the front of the vehicle body  . At this point, it is in insertion state. Fine-tune the tire angle to align the hole so that the latch is inserted. When the vehicle light turns yellow and is steady on, the switch is successful;



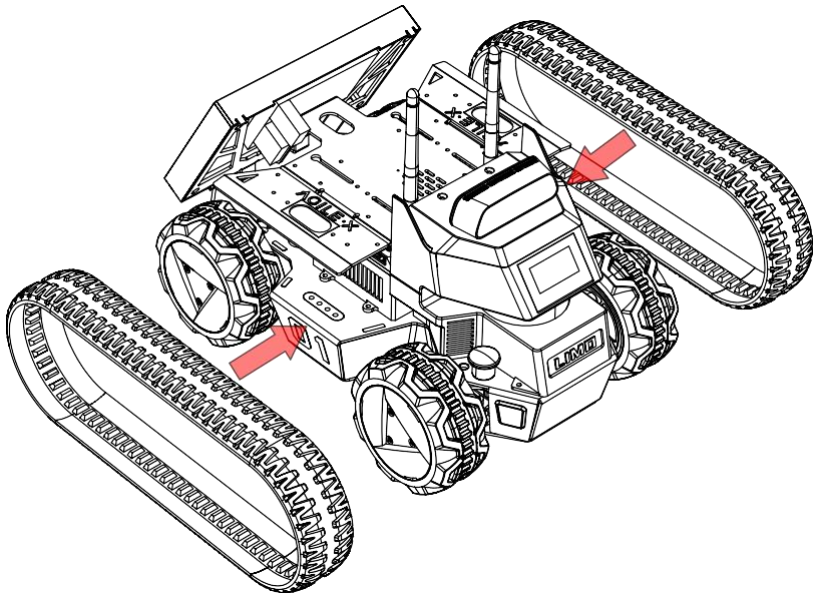


（3）切换履带模式：

(3) Switch to track mode:

在四轮差速模式下将履带直接套上，建议先套空间较小的后轮，并且履带模式下请将两侧车门抬起防止刚蹭；

In the four-wheel differential mode, put the track on directly; it is recommended to put the track on the rear wheel with small space first, and in the track mode, please lift the doors on both sides to prevent scratches;

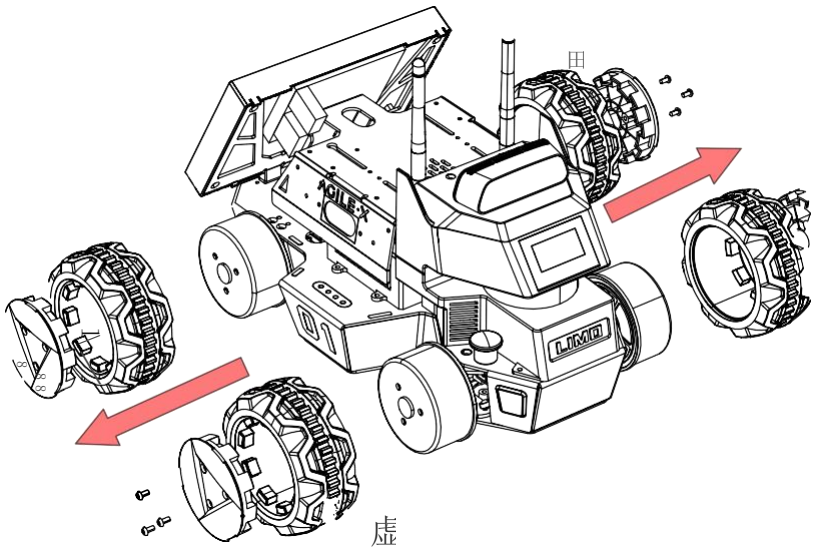


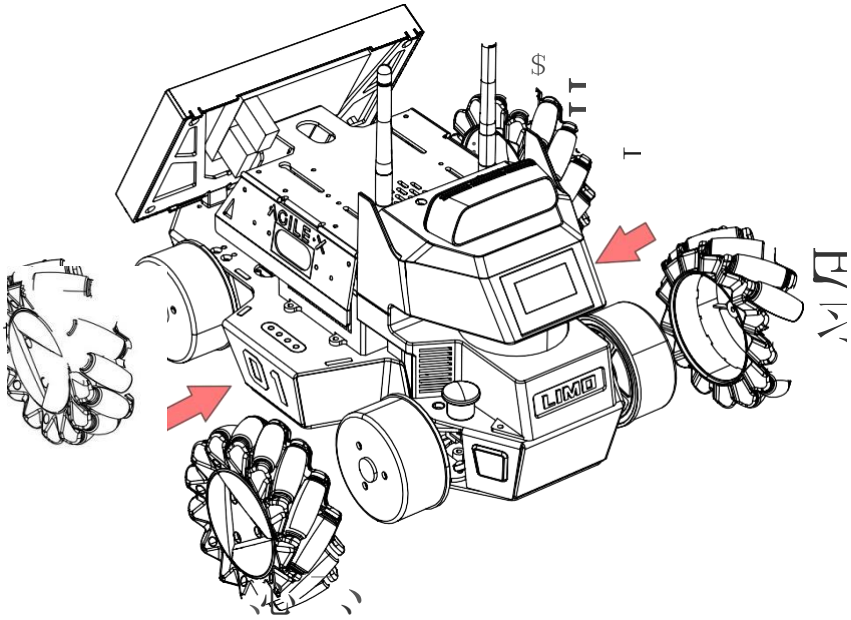
（4）切换麦轮模式：

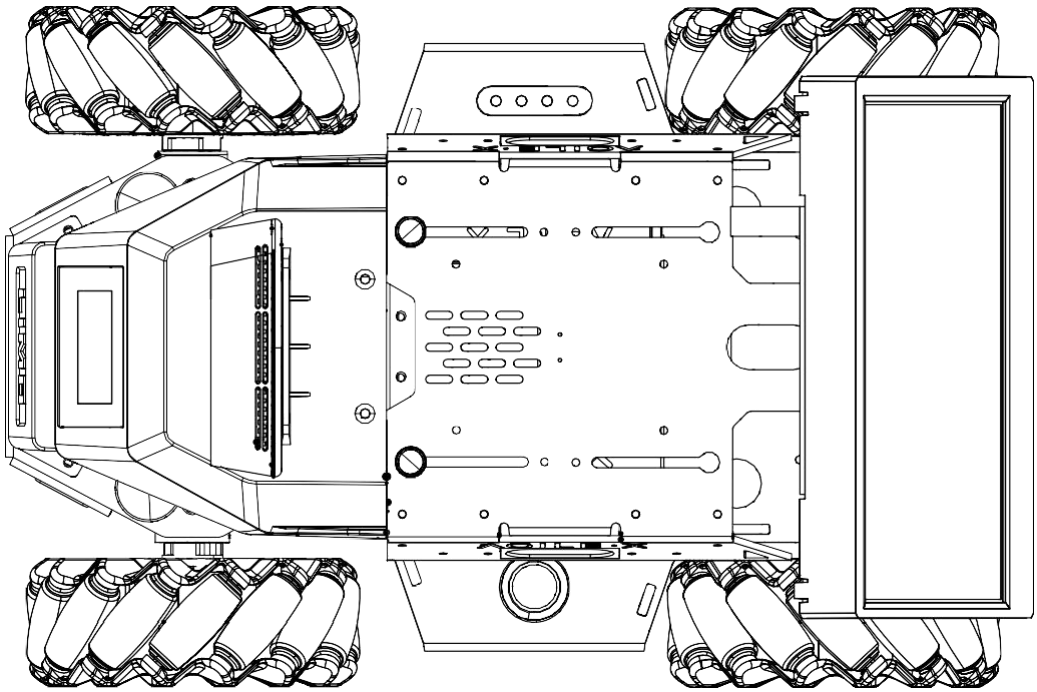
(4) Switch to Mecanum wheel mode:

先将轮毂盖和轮胎拆下，只保留轮毂电机，然后保证每个麦轮的小滚子朝向车体中心，用包装里的M3*5 螺丝将麦轮安装上，最后需要用遥控/APP调整至麦轮模式。

First remove the hubcaps and tires, leaving only the hub motor, then ensure that the small roller of each Mecanum wheel is facing the center of the body, install the Mecanum wheel with the M3*5 screw in the package, and finally adjust to the Mecanum wheel mode with remote control /APP.


AGILE-X
LIMO


AGILE-X
LIMO



注：切换至麦轮模式的时候，请确保每个麦轮安装的角度如上图所示

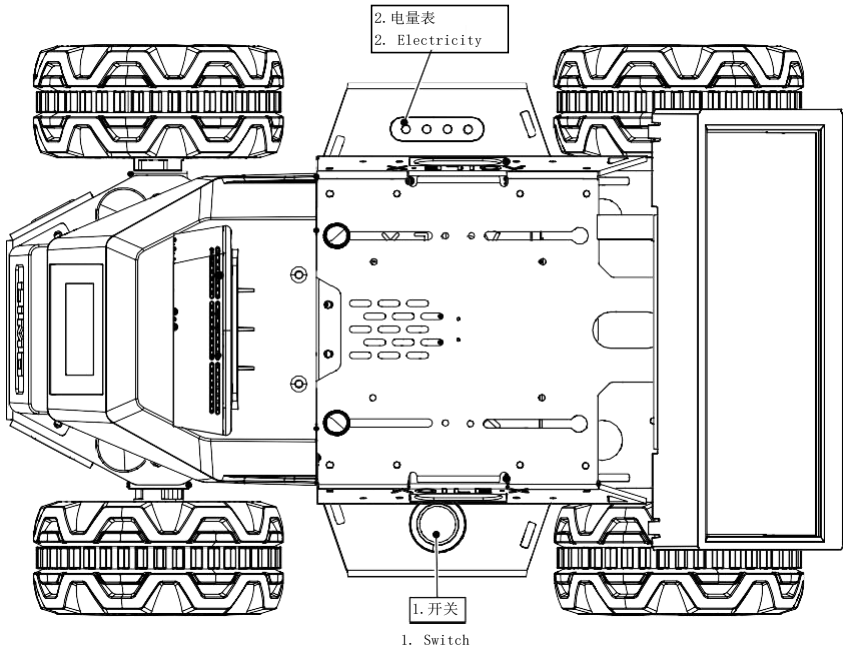
Note: When switching to Mecanum wheel mode, make sure that each Mecanum wheel is installed at the angle shown above

1.8 操作说明

1.8 Instructions on operation

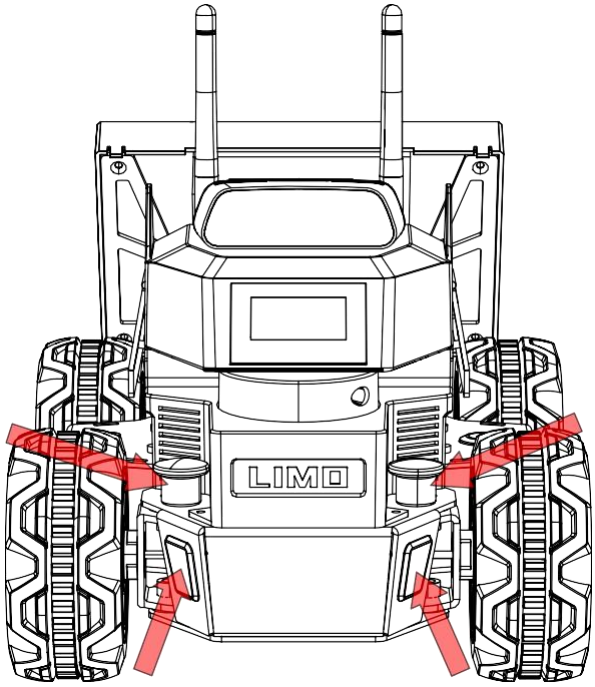
（1）长按开关键开机（短按暂停程序），观察电量表，最后一颗红灯量时请及时充电或更换电池；

(1) Long press the switch to start (short press it to pause the program), observe the electricity meter, and charge or replace the battery in time when the last red light is on;



（2）观察前面插销状态以及车灯颜色判断当前模式：

(2) Observe the status of the front latch and the color of the vehicle light to judge the current mode:



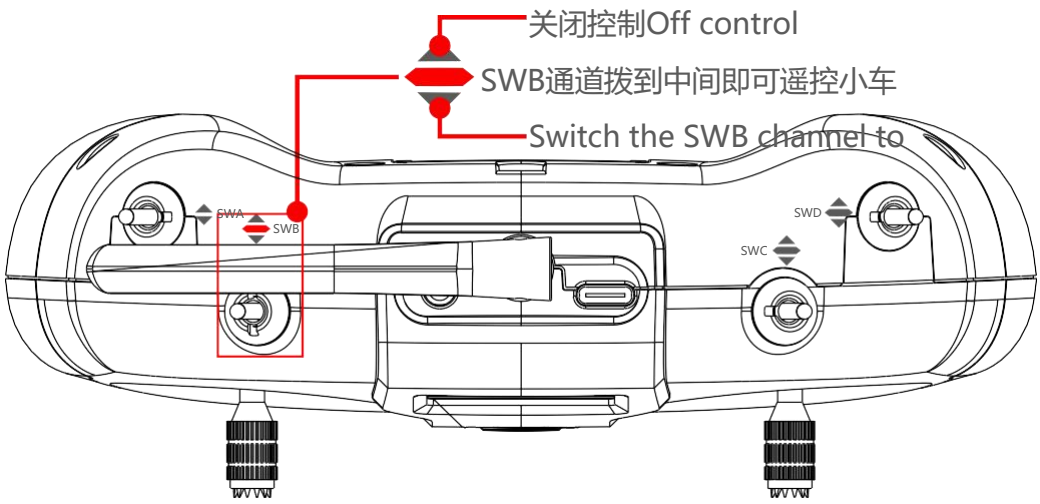
颜色 **Color**	**状态** **Status**
红色闪烁 Red flashing	低电量/主控报警 Low battery/master control alarm
红色常亮 Steady red	程序暂停 Program pause
绿色 Green	阿克曼模式 Ackermann mode
黄色 Yellow	四轮差速/履带模式 Four-wheel differential/track mode
蓝色 Blue	麦克纳姆轮模式 Mecanum wheel mode

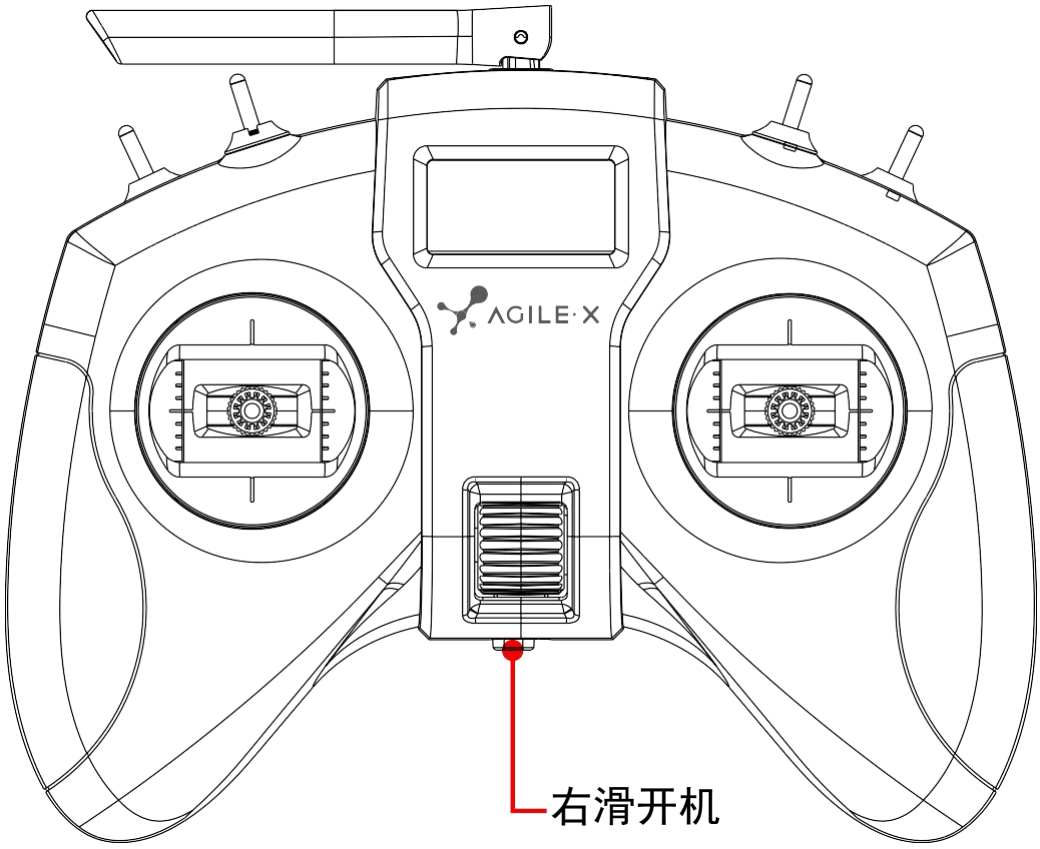
（3）遥控器说明

(3) Instructions on remote control

遥控准备：右滑开机键开机，将SWB通道拨到中间即可遥控控制，下方为指令控制，上方关闭控制；

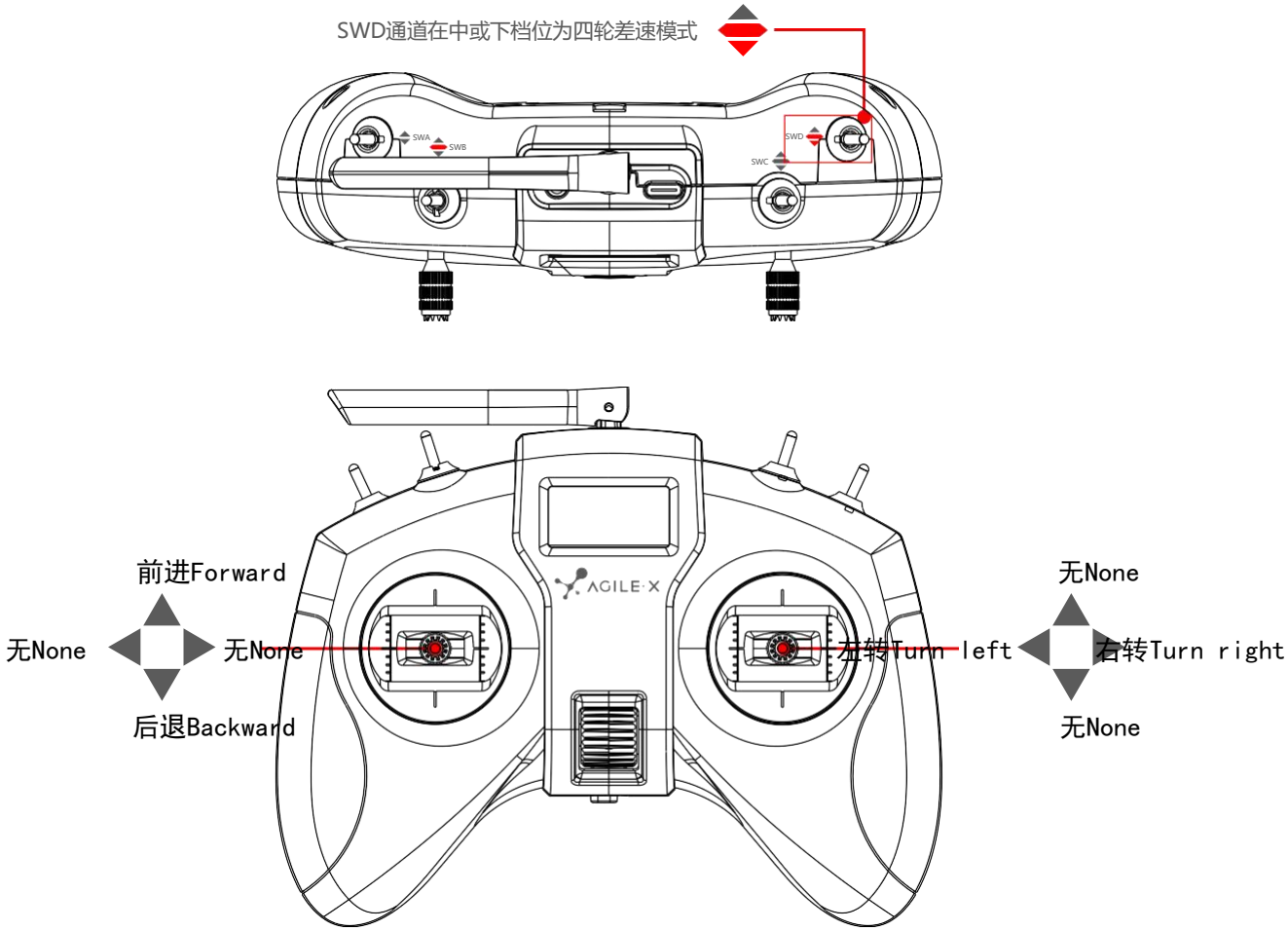
Remote control preparation: right slide on the power button to power on, switch the SWB channel to the middle for remote control; the lower part is command control, and the upper part is off control;





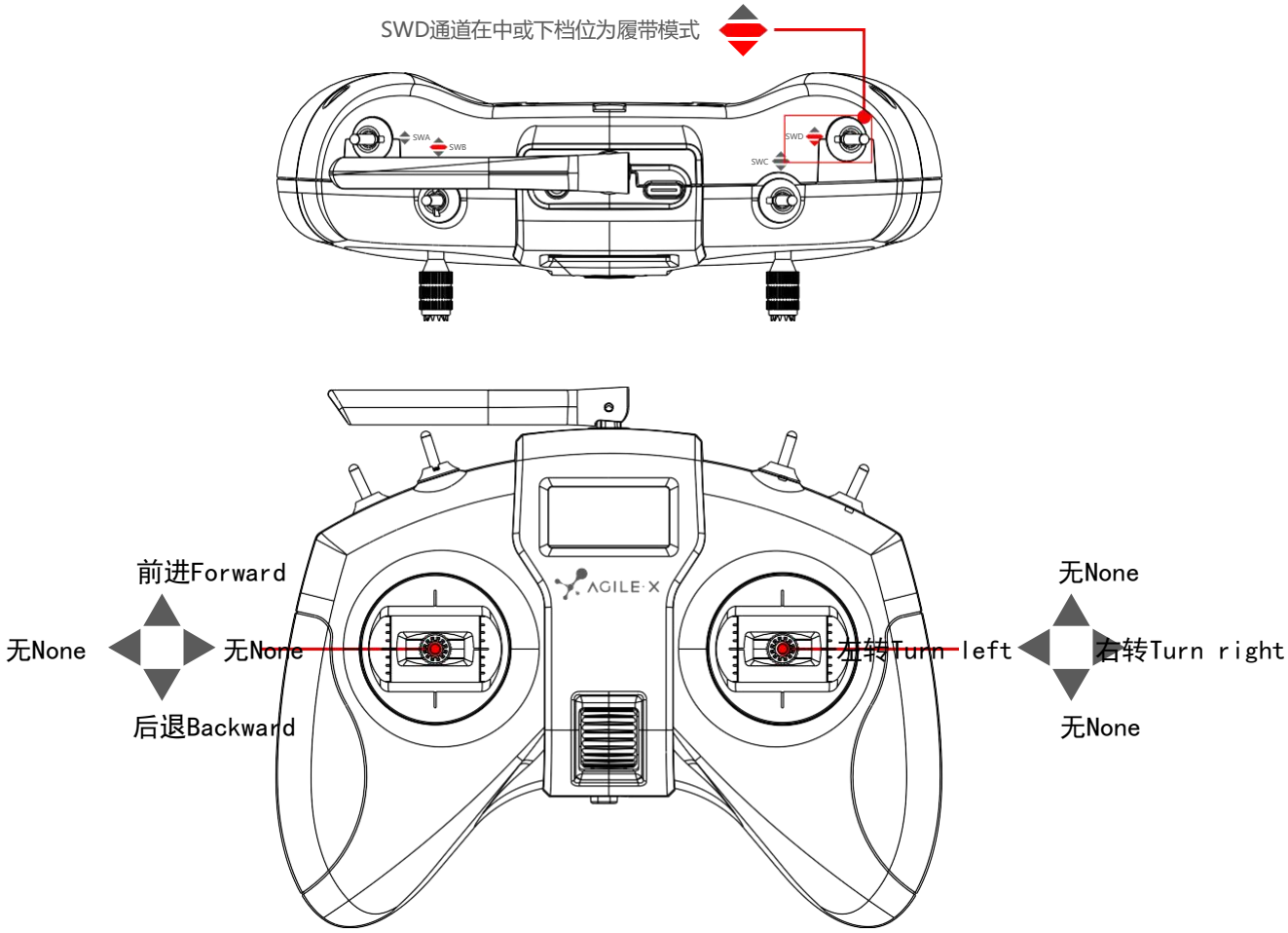
差速模式：将SWD通道拨到中下档位为四轮差速模式，左摇杆控制前进后退，右摇杆控制原地左右转；

Differential mode: switch the SWD channel to the middle and lower gear for four-wheel differential mode; the left joystick controls forward and backward, and the right joystick controls left and right turns in place;



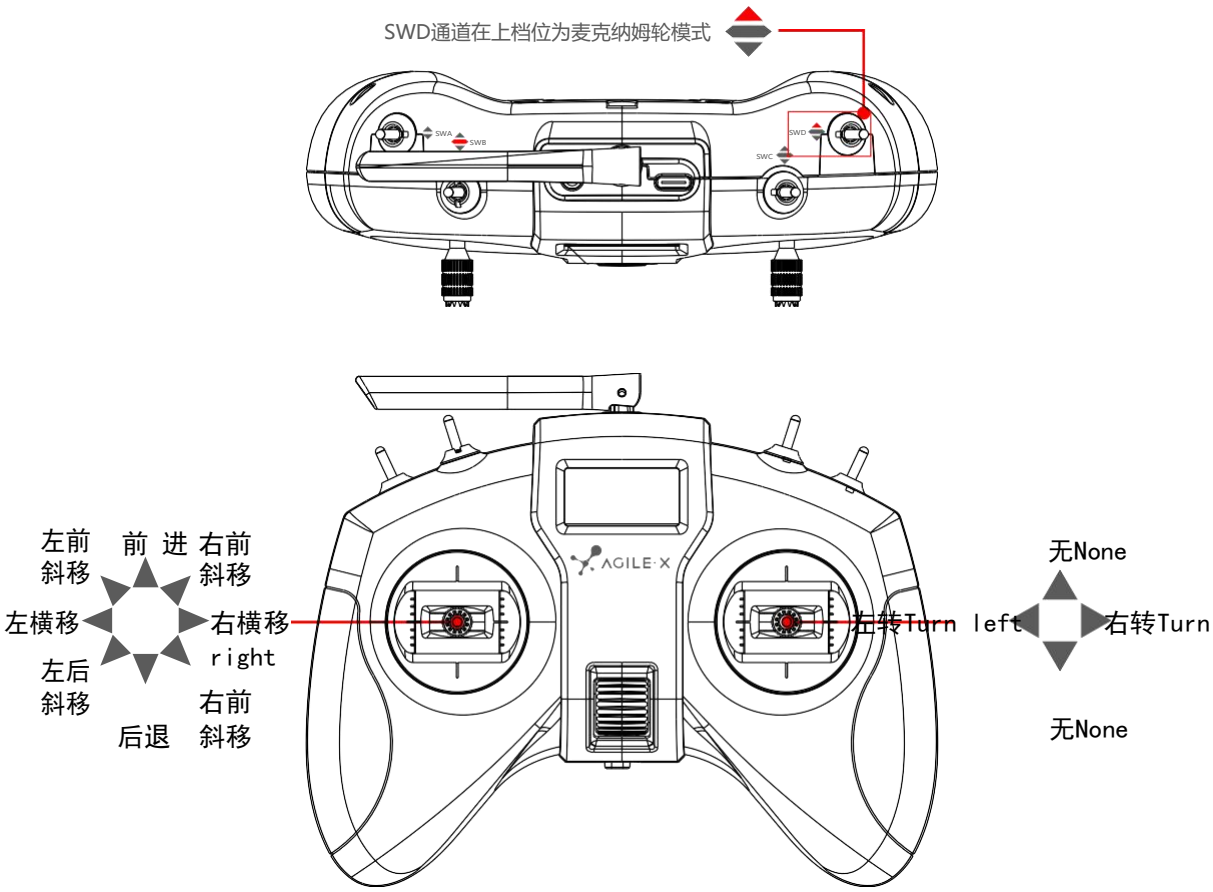
履带模式：与四轮差速模式的运动模型一样；

Track mode: same as the motion model of four-wheel differential mode;



麦轮模式：SWD通道在上档位时为麦克纳姆轮模式，左摇杆控制运动方向，右摇杆控制左右原地转；

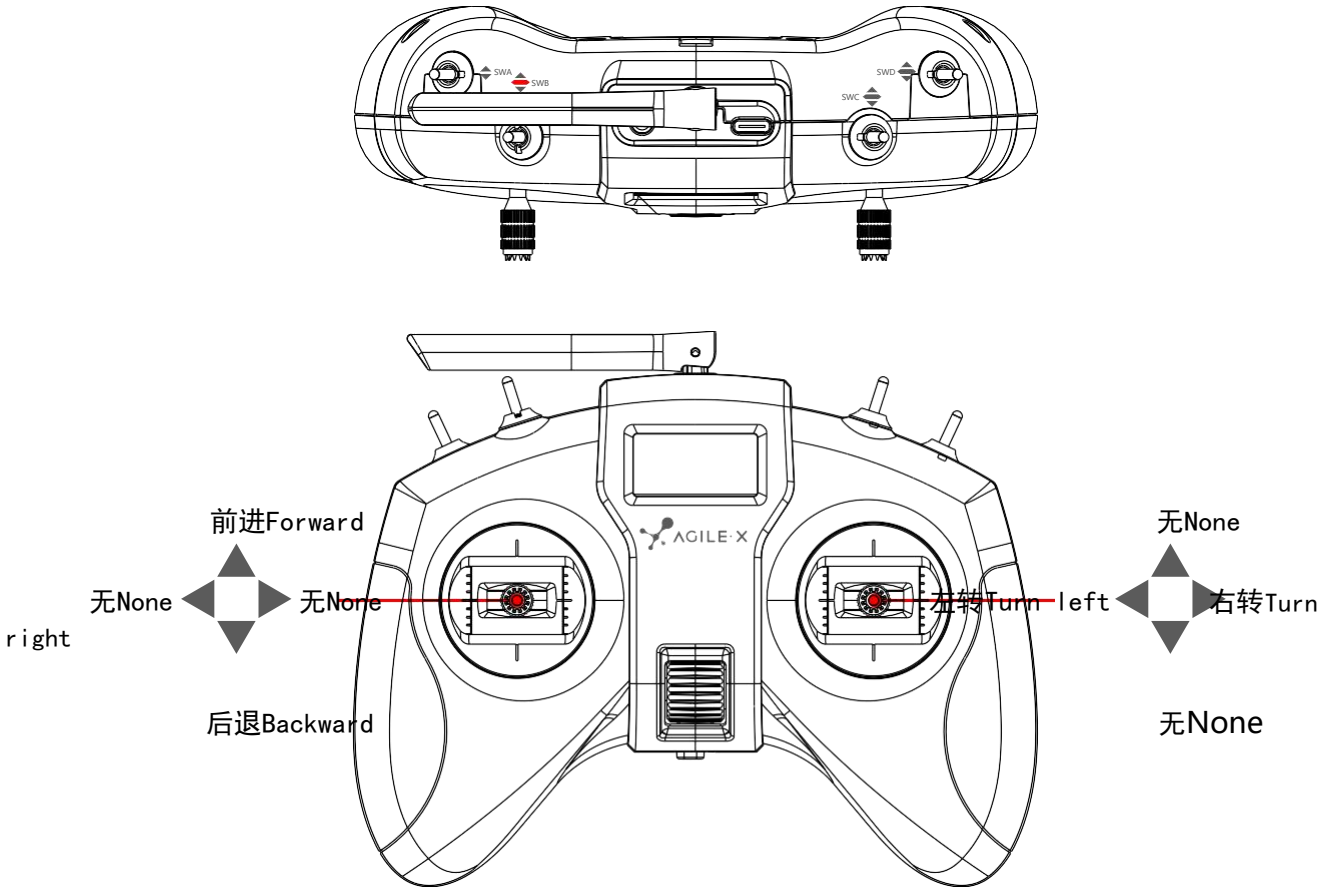
Mecanum wheel mode: it is in Mecanum wheel mode when the SWD channel is in the upper gear. The left joystick controls the direction of motion and the right joystick controls the left and right turn in place.



原文	译文
前进	Forward
后退	Backward
左前斜移	Move left anterior oblique
右前斜移	Move right anterior oblique
左横移	Move left transverse
右横移	Move right transverse
左后斜移	Move left posterior oblique
右后斜移	Move right posterior oblique

阿克曼模式：在车体上切换为阿克曼模式，开启遥控器控制即可，左摇杆控制前进后退，右摇杆控制左右方向；

Ackermann mode: switch to Ackermann mode on the vehicle body, and open the remote control; the left joystick controls forward and backward, the right joystick controls left and right direction;



（4）APP遥控说明

(4) Instructions on APP remote control

1、 首先在手机上下载我司提供的APP--Nexus，下载方式如下：

1. First download the APP--Nexus provided by our company on the mobile phone. The download method is as follows:

IOS 端 下 载： 在 AppStore 搜 索 Nexus 并 下 载

IOS download: search for Nexus in AppStore and download it

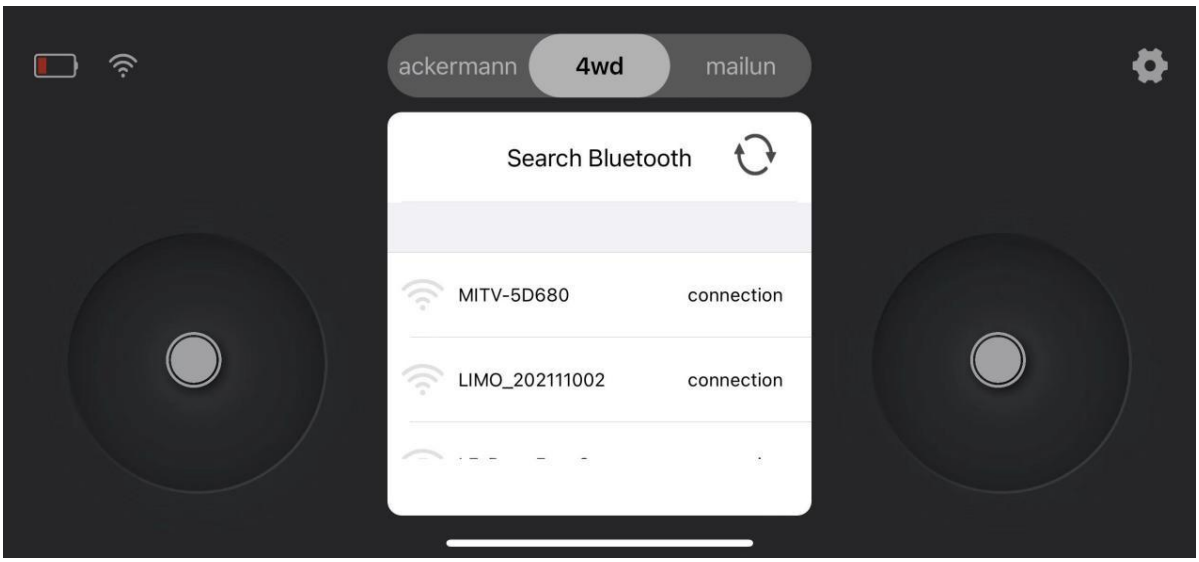
Android端扫描下面二维码：

For Android, scan the following QR code:

下载链接：https://www.pgyer.com/lbDi

Download link: https://www.pgyer.com/lbDi

2、下载App成功后，打开App,如图1-1所示，连接LIMO_xxxxxx的蓝牙；安卓手机点击左上的的蓝牙图 标进入蓝牙扫描界面



2. After downloading the App successfully, open the App; as shown in Figure 1-1, connect to the Bluetooth of LIMO_xxxxxx; click the Bluetooth icon on the upper left of the Android phone to enter the Bluetooth scanning interface

3、遥控limo

3. Remote control limo

左边控制杆：控制limo前进后退

Left lever: control limo forward and backward

右边控制杆：控制limo左转右转

Right lever: Control limo to turn left and right

中间进度条：速度值显示

Middle progress bar: speed value display

模式切换：总共有三种模式，阿克曼运动模式：ackermann，四轮差速运动模式：4wd，麦克纳姆运动 模式：mailun

Mode switching: There are three modes in total, Ackermann motion mode: ackermann; four-wheel differential motion mode: 4wd; Mecanum motion mode: mailun

ackerMann：需要手动将LIMO小车切换阿克曼模式，主要用于校准零点、控制前进后退以及转动角度；

Ackermann: You need to manually switch LIMO to Ackermann mode, which is mainly used to calibrate the zero point, and control forward and backward as well as rotation angle;

4wd：需要手动将LIMO小车切换四轮差速模式，主要控制前进后退、转换方向以及原地旋转；

4wd: You need to manually switch LIMO to the four-wheel differential mode, which mainly controls forward and backward, rotation direction and rotation in place;

Mailun：需要手动将LIMO小车切换麦轮模式，主要控制前进后退、转换方向以及原地旋转

Mailun: You need to manually switch LIMO to the Mecanum wheel mode, which mainly controls forward and backward, rotation direction, and rotation in place

4、APP设置说明

4. Instructions on APP setting

Language switch：通过点击右边的按钮 English/简体，来切换英语和中文

Language switch: Switch between English and Chinese by clicking the button English/Simplified on the right

Left-romte min speed：设置LIMO小车的最小速度

Left-romte min speed: Set the minimum speed of LIMO

Left-romte max speed：设置LIMO小车的最大速度

Left-romte max speed: Set the maximum speed of LIMO

right-romte min speed：设置LIMO小车的最小旋转速度

right-romte min speed: Set the minimum rotation speed of LIMO

right-romte max speed：设置LIMO小车的最大旋转速度

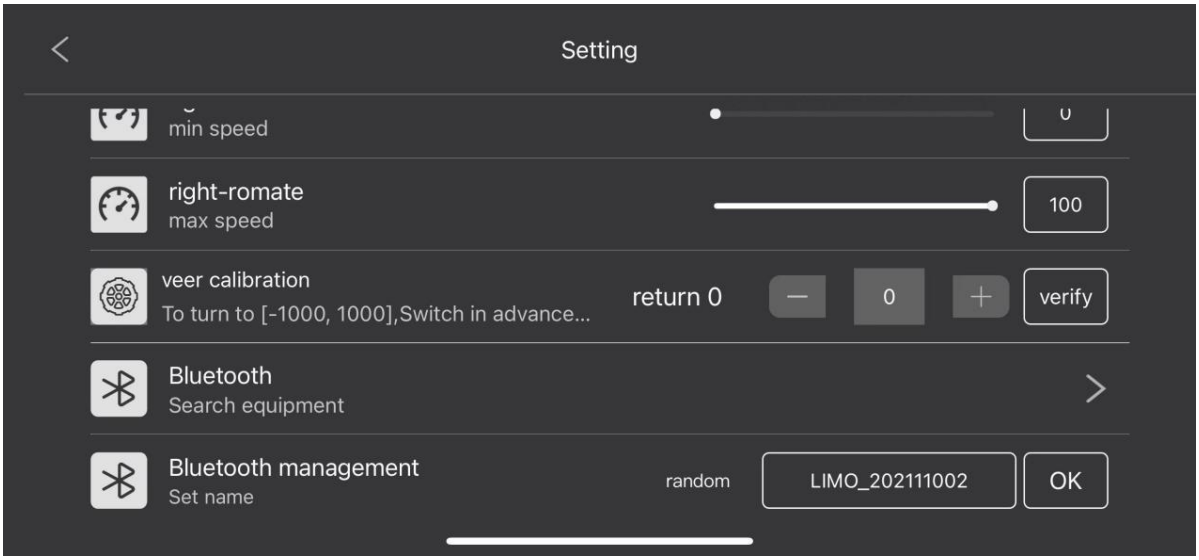
right-romte max speed: Set the maximum rotation speed of LIMO

veer calibration：设置零点矫正，先点击+号，然后点击确定Verify，当校准成功会弹出提醒框Successful calibration
veer calibration: To set the zero point calibration, first click the + sign, and then click Confirm Verify. When the calibration is successful, a reminder box will pop up: Successful calibration

BlueTooth：点击弹出蓝牙扫描界面

BlueTooth: Click to pop up the Bluetooth scanning interface

Bluetooth management：点击random 生成任意以LIMO_xxxx的命名的名字，确认OK并同步修改小车蓝牙的名字，注意此时蓝牙会断开，并会提醒重新连接蓝牙，重新连接蓝牙后可以继续控制小车；当app再次启动，小车的蓝牙名字已经显示修改成功的状态



Bluetooth management: Click random to generate any name named after LIMO_xxxx, confirm OK and modify the Bluetooth name of the vehicle synchronously. Note that the Bluetooth will be disconnected at this time and will remind you to reconnect the Bluetooth. After reconnecting the Bluetooth, you can continue to control the vehicle; when the app is started again, the Bluetooth name of the vehicle has been modified successfully

1.9 远程桌面连接

1.9 Remote desktop connection

1.9.1 下载安装NoMachine

1.9.1 Download and install NoMachine

首先在个人电脑下载相应的软件，下载链接：https://www.nomachine.com/download，根据自己电脑 的操作

系统和架构下载相应的版本。让limo和电脑连接到同一个WIFI下。

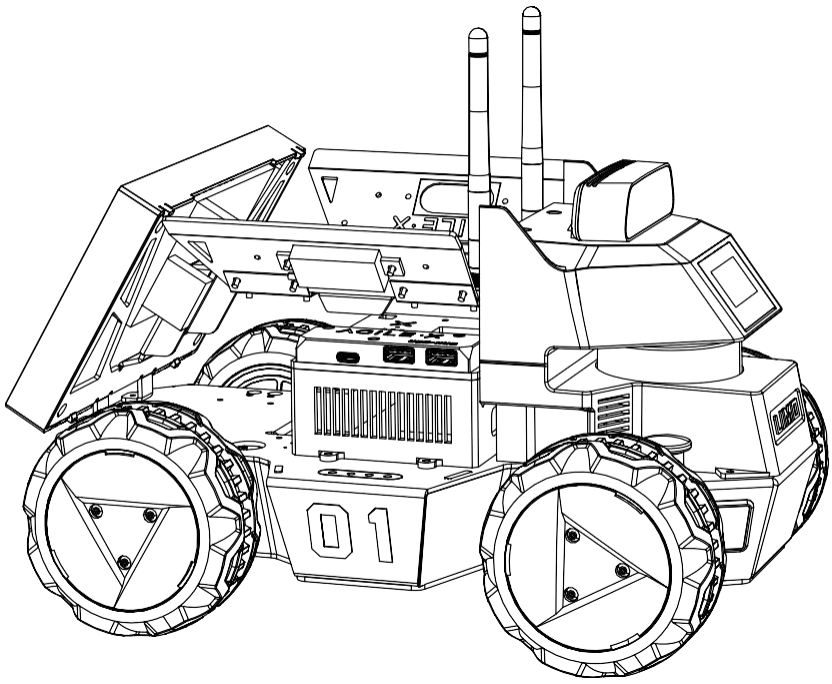
First download the corresponding software on your personal computer; download link: https://www.nomachine.com/download; download the corresponding version according to your computer's operating system and architecture. Let limo and computer connect to the same WIFI.

1.9.2 连接wifi

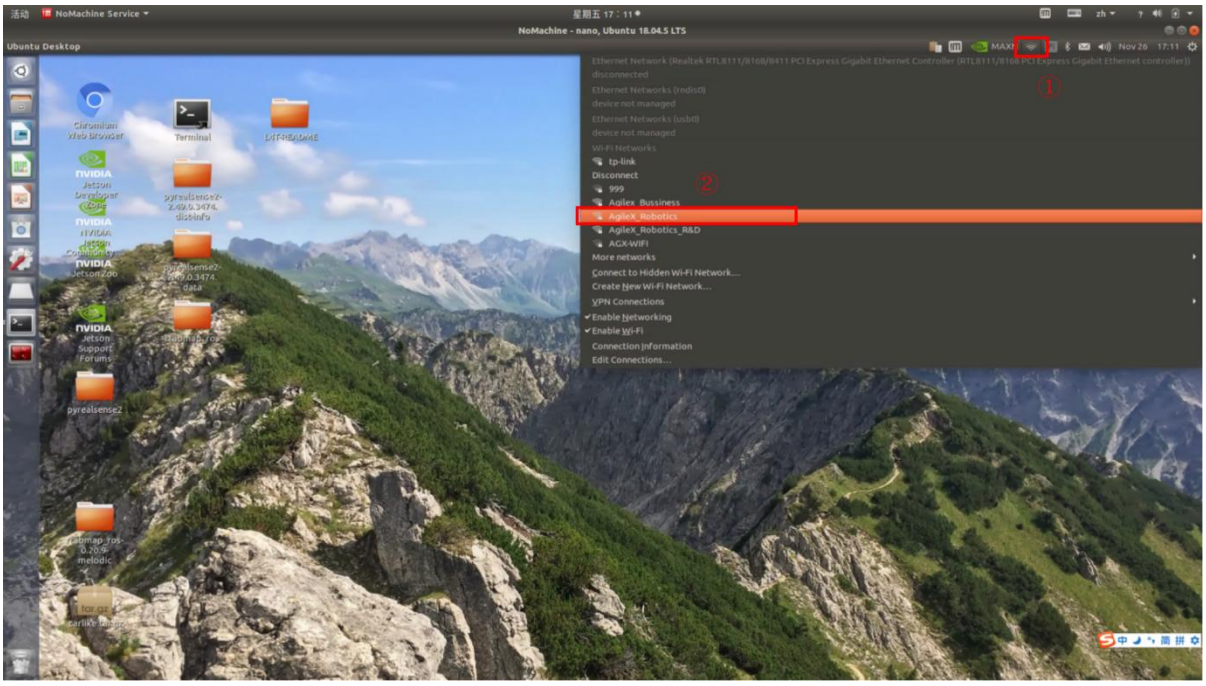
1.9.2 Connect to wifi

打开limo右侧的海鸥门，找到USB-HUB模块，给limo连接上键盘鼠标，USB-HUB模块的位置如下图：

Open the seagull door on the right side of limo, find the USB-HUB module, and connect the keyboard and mouse to limo. The position of the USB-HUB module is as shown in the figure below:



键盘鼠标成功连接之后通过以下操作连接wifi，选择需要连接的wifi。



After the keyboard and mouse are successfully connected, connect to wifi through the following operations, and select the wifi that needs to be connected.

输入wifi的密码

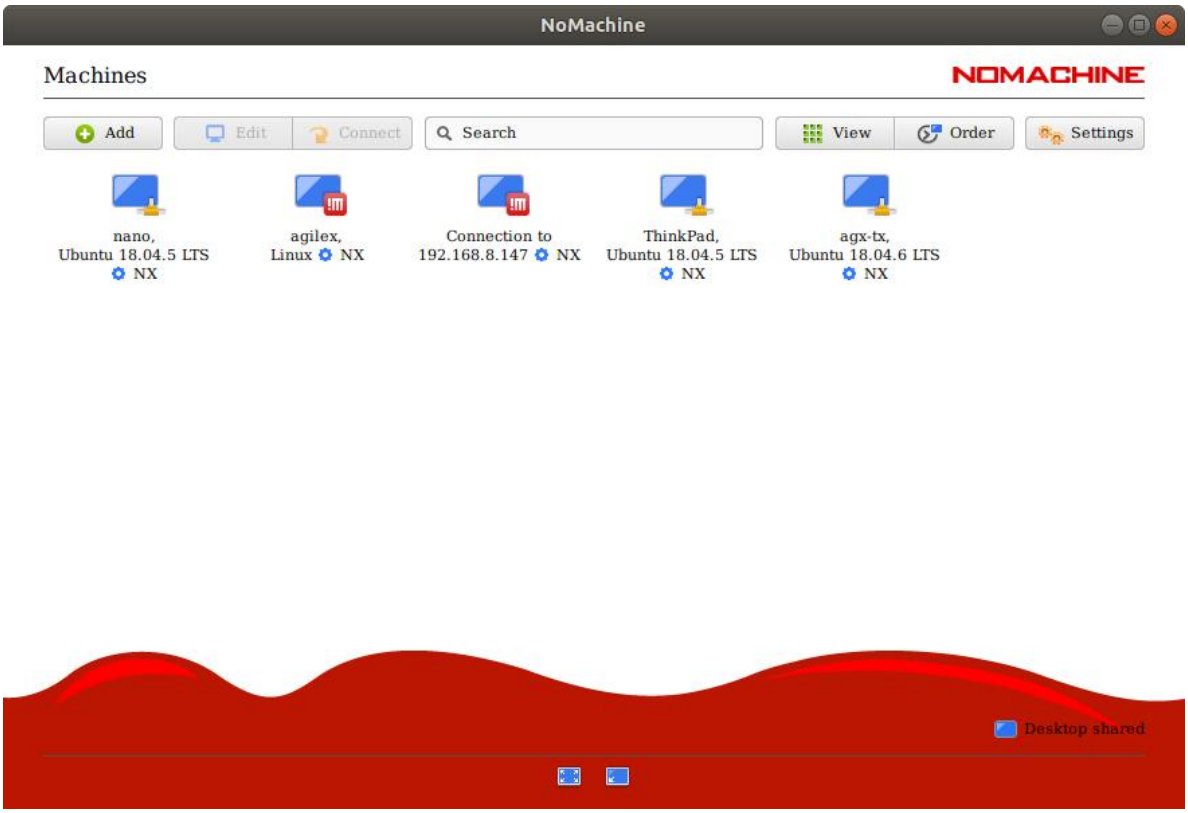
Enter the password of wifi

1.9.3 远程连接limo

1.9.3 Connect limo remotely

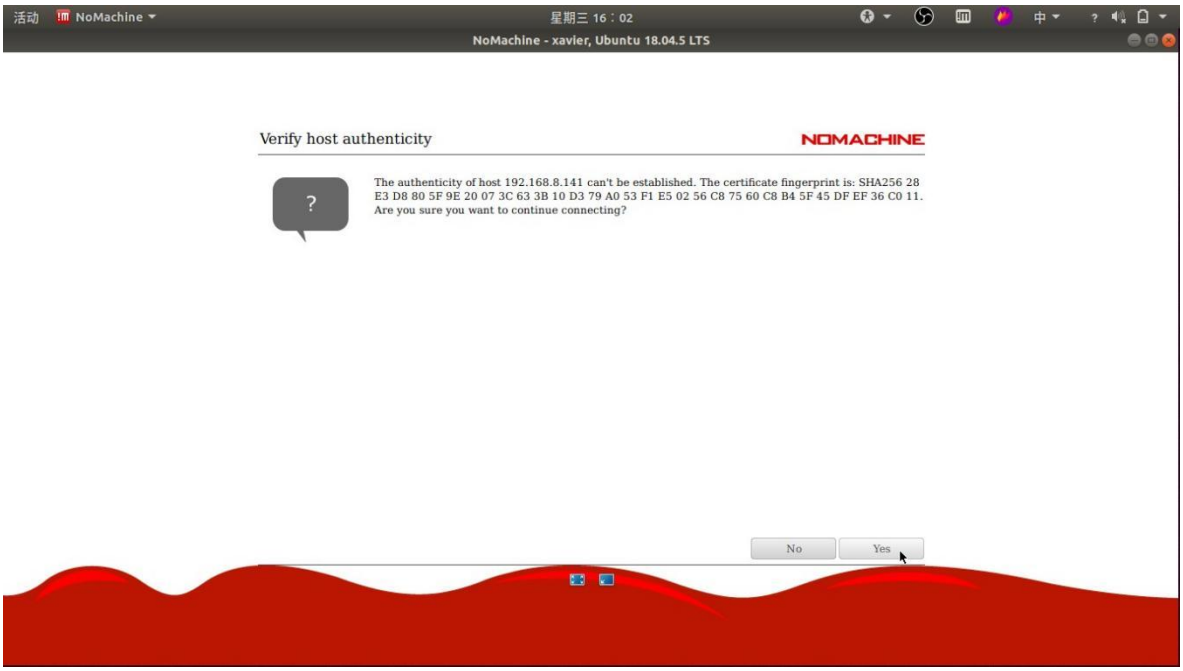
选择连接对象

Select connection object



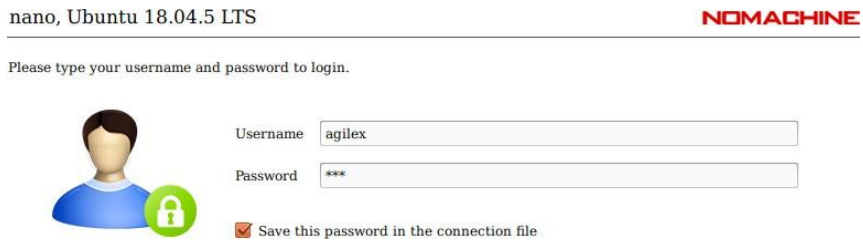
点击Yes

Click Yes

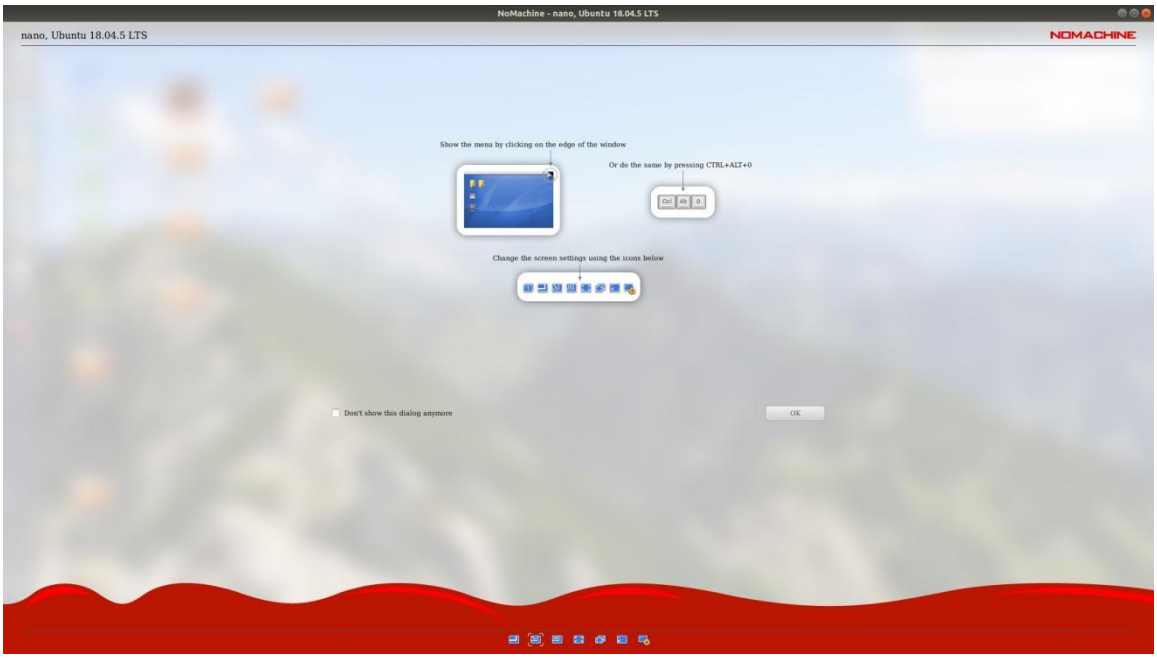


Username：agilex Password：agx 勾选保存密码

Username: agilex Password: agx check to save the password



一路选择默认OK



Always select the default OK

二、底盘电气信息说明

2. Instructions on Chassis Electrical Information

2.1 电池与充电

2.1 Battery and charging

2.1.1 电池基本信息

2.1.1 Basic battery information

LIMO随车配备一个12V的电池，该电池有两个接口。分别为黄色的电池输出接口和黑色的电池充电接口。

LIMO is equipped with a 12V battery with two interfaces. They are the yellow battery output interface and the black battery charging interface.

该电池参数如下表

The battery’ s parameters are as follows

项目 Items	额定参数 Rated parameters
典型容量 Typical capacity	5200mAH
最小容量 Minimum capacity	5000mAH
标称电压 Nominal voltage	11.1V
充电截至电压 Charge cut-off voltage	12.6V
放电截至电压 Discharge cut-off voltage	8.25V
最大持续放电电流 Maximum continuous discharge current	10A

电池注意事项

Battery precautions

- 为了保证运输存储安全，LIMO配备的电池并不一定处于满电状态 。
  In order to ensure the safety of transportation and storage, the battery supplied with LIMO is not necessarily fully charged.
- 请不要在电池使用殆尽以后再进行充电，当 LIMO提示电量低时，请及时充电。
  Please do not charge the battery after its power has been depleted, and please charge the battery in time when LIMO’ s low battery level alarm is on;
- LIMO在关机状态下仍会产生静态待机电流，为防止电池过放，长时间不使用LIMO时请断开电池与车体的连接。
  LIMO will still generate a quiescent standby current when it is turned off. To prevent

the battery from over-discharging, please disconnect the battery from the vehicle body when you do not use LIMO for a long time.

- 请勿将电池投火中，或对电池加热，请勿在高温下存储电池，电池存储的最佳温度为-10℃~40℃。
  Please do not put the battery in fire or heat up the battery, and please do not store the battery in high-temperature environment. The best temperature for battery storage is -10℃ ~40℃.
- 必须使用原厂配备或认证的电池为LIMO供电。
  LIMO must be charged with the original factory-equipped or certified battery.

2.1.2 充电

2.1.2 Charging

LIMO默认随车配备一个12.6V 2A的充电器，可满足客户的充电需求，且充电器上设有指示灯可显示充电状态。

LIMO is equipped with a 12.6V 2A charger by default to meet customers' charging demand, and there is an indicator light on the charger to show the charging status.

- 充电时请关机取出电池，将电池输出接口与车体分离。
  When charging, please turn off the vehicle and remove the battery, and separate the battery output interface from the vehicle body.
- 将充电器的充电接头与电池连接，再接通充电器电源进行充电。
  Connect the charging connector of the charger to the battery, and then turn on the charger's power supply for charging.
- 充满时请先将电池与充电器分离，再断开充电器电源。
  When fully charged, please separate the battery from the charger first, and then disconnect the charger.

充电器状态如下表：

The charger status is as follows:

充电器指示灯颜色 **Charger indicator light's color**	**电池状态** **Charger status**
红色 Red	充电中 Charging
绿色闪烁 Green flashing	即将充满 Almost fully charged
绿色 Green	已充满 Fully charged

充电注意事项：

Charging precautions:

- 禁止使用非原装充电器对电池进行充电，请勿在0℃以下给电池充电。
  It is forbidden to use non-original chargers to charge the battery, and do not charge the battery below 0°C.
- 充电时必须将电池与LIMO车体分离，禁止在电池充电的同时为LIMO进行供电。
  The battery must be separated from LIMO’ s vehicle body when charging, and it is forbidden to supply power to LIMO while charging the battery.
- 当充电器指示灯变为绿色时表示充电完毕，但为了延长电池寿命，充电器会以0.1A的电流进行涓流充电，持续约0.5小时。
  When the indicator light of the charger turns green, it indicates that the charging is complete, but in order to prolong the battery life, the charger will trickle charge with a current of 0.1A for about 0.5 hours.
- 当前电池从8.25V到充满电状态大约需要2.5小时，电池充满电电压约为12.6V。
  At present, it takes about 2.5 hours for the battery to reach a fully charged state from 8.25V, and the fully charged voltage of the battery is about 12.6V.

2.2 使用环境及安全注意事项

2.2 Operational environment and safety precautions

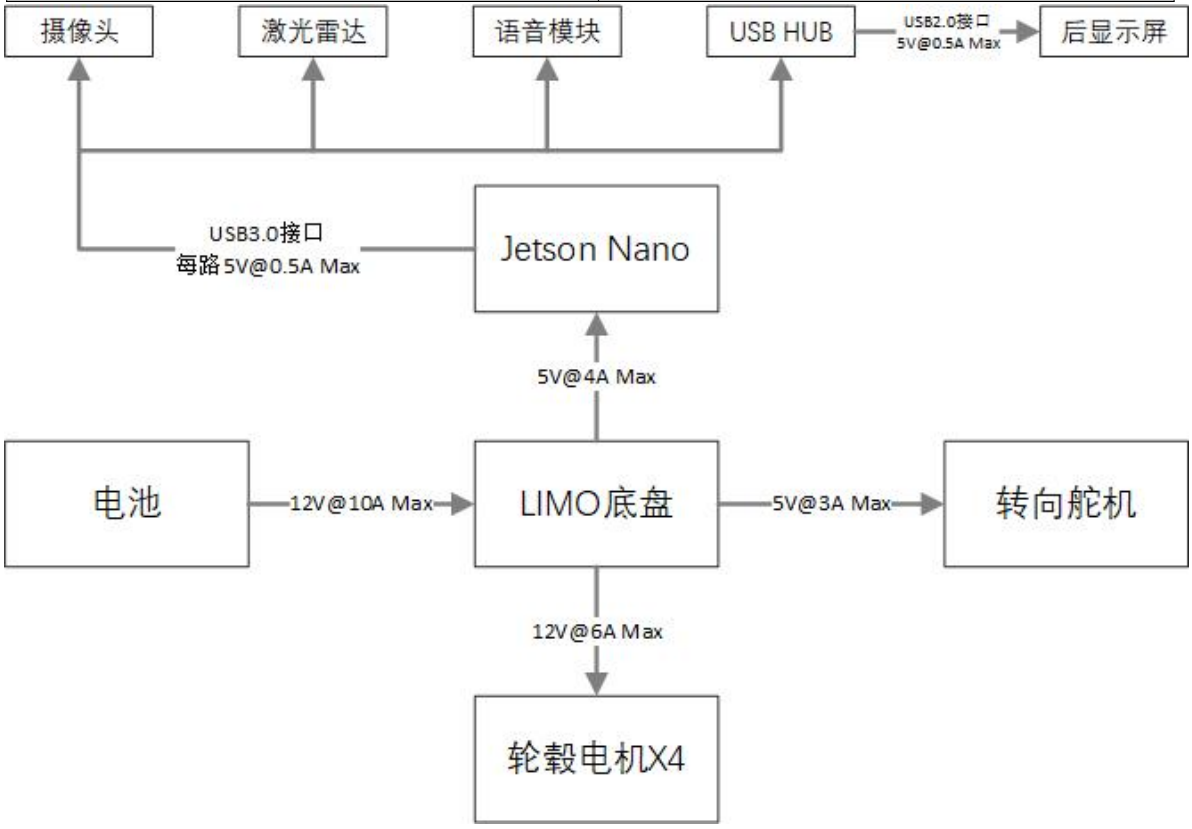
- LIMO的工作温度为-10℃～40℃，请勿在温度低于-10℃、高于40℃环境中使用；
  The operating temperature of LIMO is -10°C ~ 40°C; please do not use it in an environment with a temperature lower than -10°C or higher than 40°C;
- LIMO的使用环境的相对湿度要求是：最大80%，最小30%；
  The relative humidity requirements of LIMO's operational environment are: maximum 80%, minimum 30%;
- 请勿在存在腐蚀性、易燃性气体的环境或者靠近可燃性物质的环境中使用；
  Please do not use it in an environment with corrosive and flammable gas or in an environment near flammable substances;
- LIMO不具有防水功能，请勿在有雨、雪、积水的环境使用；
  LIMO is not waterproof, so please do not use it in an environment with rain, snow, or water;
- 建议使用环境海拔高度不超过1000米、昼夜温差不超过25℃；
  It is recommended that the altitude of the operational environment should not exceed 1000M, and the temperature difference between day and night should not exceed 25°C;
- 使用过程有疑问，请按照相关说明手册进行操作或者咨询相关技术人员；
  In case of any doubts during use, please operate according to the relevant instruction manual or consult related technical personnel;
- 请勿未经技术支持和允许，私自改装内部设备结构。
  Without technical support and permission, please do not personally modify the internal equipment structure.

2.3 供电拓扑

2.3 Power supply topology

原文	译文
摄像头	Camera
激光雷达	LiDAR

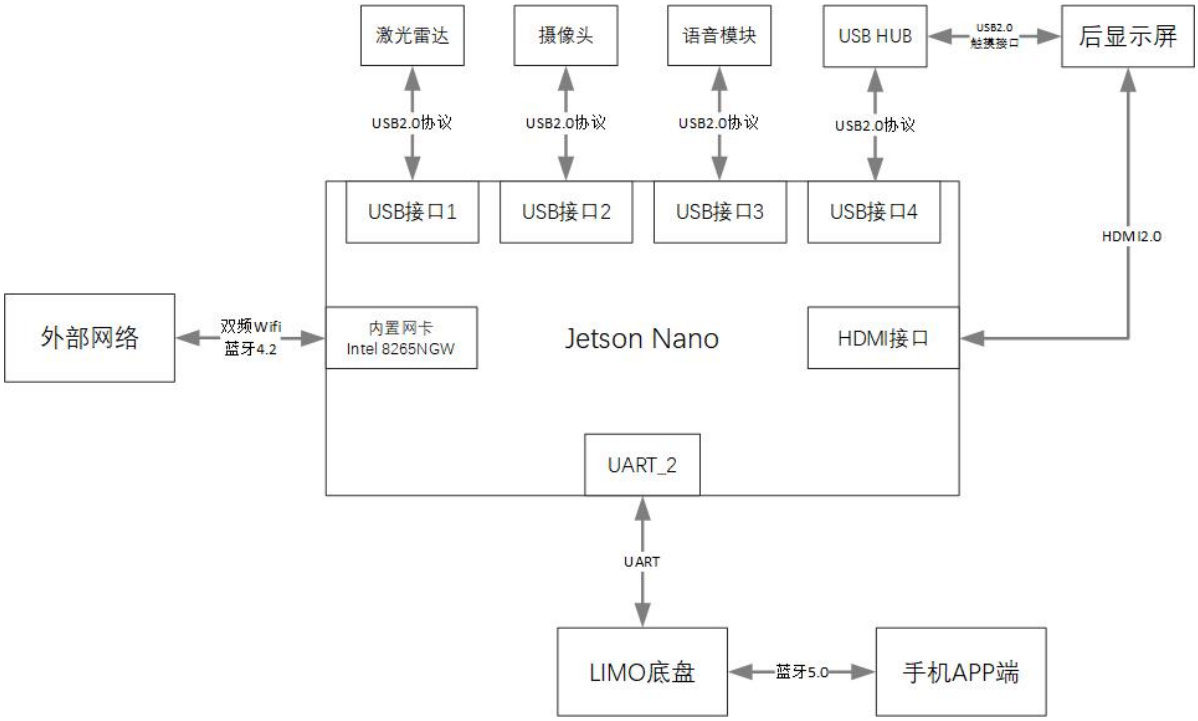
语音模块	Voice module
USB2.0接口	USB2.0 interface
后显示屏	Rear display screen
USB3.0接口	USB3.0 interface
每路5V@0.5A Max	5V per circuit@0.5A Max
电池	Battery
LIMO底盘	LIMO chassis
转向舵机	Steering gear
轮毂电机	Hub motor



- LIMO的电池能提供最大10A的电流给底盘、Nano及传感器等系统供电，当系统检测到工作电流大 于10A时，为了保护电池和电机会进入过流保护模式。

  LIMO's battery can provide a maximum current of 10A to power the chassis, Nano, and sensors. When the system detects that the operating current is greater than 10A, it will enter an overcurrent protection mode to protect the battery and motor.
- USB HUB 的三个USB接口合计输出电流最大为0.5A。
  The total output current of the three USB interfaces of USB HUB is 0.5A at most.

2.4 通信拓扑



2.4 Communication topology

原文	译文
外部网络	External network
双屏Wifi	Dual-screen Wifi
蓝牙4.2	Bluetooth 4.2
激光雷达	LiDAR
摄像头	Camera
语音模块	Voice module
USB2.0触摸接口	USB2.0 touch interface
后显示屏	Rear display screen
USB2.0协议	USB2.0 protocol
USB接口	USB interface
内置网卡	Built-in network card
HDMI接口	HDMI interface
LIMO底盘	LIMO chassis
蓝牙5.0	Bluetooth 5.0
手机APP端	Mobile APP

LIMO底盘内置了蓝牙5.0模块可以与手机端的APP连接，实现遥控功能。

The LIMO chassis has a built-in Bluetooth 5.0 module that can be connected to the APP on the mobile phone to realize the remote control function.

- LIMO与Nano通过UART接口直接连接，Nano通过该接口可实现对底盘的控制。
- LIMO and Nano are directly connected through a UART interface, and Nano can control the chassis through this interface.
  USB HUB提供2个USB和1个Type C接口，3个接口均工作在USB2.0协议下。
  USB HUB provides 2 USB interfaces and 1 Type C interface; all 3 interfaces work under the USB2.0 protocol.
- 后显示屏通过USB2.0接口与USB HUB相连，起触摸功能。
  The rear display screen is connected to the USB HUB through the USB2.0 interface and has a touch function.

三、底盘驱动程序驱动

3. Chassis Driver Drive

移动底盘需要通过程序驱动才能实现limo的导航，limo的底盘驱动程序分为两个版本，分别为C++版本 和Python版本，两个版本都可以控制limo运动。

The mobile chassis needs to be driven by a program to realize the navigation of limo. The chassis driver of limo is divided into two versions, namely the C++ version and the Python version. Both versions can control the movement of limo.

3.1 C++底盘驱动

3.1 C++ chassis driver

C++版本的驱动程序所在文件夹为~/agilex_ws/src/limo_ros/limo_base，可以通过以下命令进入到该文 件夹中
The folder where the C++ version of the driver is located is ~/agilex_ws/src/limo_ros/limo_base, which can be accessed by the following command

```
cd agilex_ws/src/limo_ros/limo_base
```

以下是limo_base功能包的文件列表：

The following is the file list of the limo_base package:

```
├── limo_base
    ├── CMakeLists.txt
    ├── include
    |   ├── limo_driver.h
    |   ├── limo_protocol.h
    |   └── serial_port.h
    ├── launch
    |   └── limo_base.launch
    ├── msg
    |   └── LimoStatus.msg
    ├── package.xml
```

```
    └── src
        ├── limo_base_node.cpp
        ├── limo_driver.cpp
        └── serial_port.cpp
```

limo_base下有四个文件夹，分别为include、launch、msg、src。include文件夹下存放着驱动程序所 调用的库文件；launch文件夹下存放着驱动程序的启动文件；msg文件夹下存放着驱动程序所需要的消 息文件；src文件夹下存放着驱动程序源代码。

There are four folders under limo_base, namely include, launch, msg, and src. The include folder stores the library files called by the driver; the launch folder stores the startup files of the driver; the msg folder stores the message files needed by the driver; the src folder stores the driver source code.

文件夹 **Folder**	**存放文件** **Stored files**
include	驱动程序所调用的库文件 Library files called by the driver
launch	驱动程序的启动文件 Startup files of the driver
msg	驱动程序所需要的消息文件 Message files needed by the driver
src	驱动程序源代码 Driver source code

可以通过一段简单的指令控制limo向前运动

You can control the forward movement of limo with a simple command

注：在运行命令之前，请确保其他终端中的程序已经终止，终止命令为：Ctrl+c

Note: Before running the command, please make sure that the programs in other terminals have been terminated. The termination command is: Ctrl+c

1、启动底盘，打开一个终端，在终端中输入命令：

1. Launch the chassis, open a terminal, and enter the command in the terminal:

```
roslaunch limo_base limo_base.launch
```

2、输入控制指令，打开一个终端，在终端中输入命令：

2. Enter the control command, open a terminal, and enter the command in the terminal:

```
rostopic pub /cmd_vel geometry_msgs/Twist "linear:
  x: 0.2
  y: 0.0
  z: 0.0
angular:
  x: 0.0
  y: 0.0
  z: 0.0"
```

注：整个命令复制到终端中输入即可，不要手动输入

Note: Copy the entire command to the terminal and enter it, and don’ t enter it manually

在终端输入命令之后，limo会先前行走一小段距离，然后停下。

After entering the command in the terminal, limo will walk a short distance before stopping.

驱动程序源码中使用到的函数：

Functions used in the driver source code:

函数名称 Function's name	函数作用 Function's function
connect()	连接底盘 Connect the chassis
readData()	读取数据，获取底盘反馈的信息 Read the data, and get the information feedback from the chassis
processRxData()	接收串口数据 Receive serial data
parseFrame()	处理串口数据 Process serial data
sendFrame()	发送串口数据 Send serial data
setMotionCommand()	设置limo的控制模式 Set limo's control mode
enableCommandedMode()	使能控制模式 Enable control mode
publishOdometry()	发布里程计数据 Publish odometer data
publishLimoState()	发布limo的状态信息 Publish limo's state information
publishIMUData()	发布IMU的数据 Publish IMU data
processErrorCode()	错误检测 Error detection
twistCmdCallback()	发布速度控制数据 Publish speed control data
normalizeAngle()	输出一个正常的角度 Output a normal angle
degToRad()	把角度转成弧度 Turn the angle to radian
convertInnerAngleToCentral()	将内角转换为中心角 Convert inner angle to central angle
convertCentralAngleToInner()	将中心角转换为内角 Convert central angle to inner angle

3.2 Python底盘驱动

3.2 Python chassis driver

limo的Python版本驱动上传到pypi，可以通过pip指令下载该驱动程序；程序的安装目录为~/.local/lib/python3.6/site-packages/pylimo。它的文件列表为：

limo's driver (Python version) is uploaded to pypi, and the driver can be downloaded through the pip command; the installation directory of the program is ~/.local/lib/python3.6/site-packages/pylimo. Its file list is:

```
├──_init__.py
├── limomsg.py
├── limo.py
└──__pycache____
    ├──_init__.cpython-36.pyc
    ├── limo.cpython-36.pyc
    └── limomsg.cpython-36.pyc
```

Python版本的代码比较简洁，仅有三个文件组成驱动程序，init.py的作用为申明需要使用的文件，limomsg.py的作用为驱动成所需要的消息，limo.py是主程序，它的作用是驱动limo。

The Python version of the code is relatively concise, and only three files make up the driver. The function of init.py is to declare the files to be used, the function of limomsg.py is to drive to the required messages, and limo.py is the main program and its function is driving limo.

文件名称 **File's name**	**文件作用** **File's function**
init.py	申明需要使用的文件 Declare the files to be used
limomsg.py	驱动成所需要的消息 Drive to the required messages
limo.py	主程序，用于驱动limo Main program, used to drive limo

我们提供了一个脚本调用该驱动程序，该脚本所在目录为agilex_ws/src/limo_ros/limo_base/script，脚 本名称为limomove.py。

We provide a script to call the driver. The directory where the script is located is agilex_ws/src/limo_ros/limo_base/script, and the script′ s name is limomove.py.

可以通过以下命令访问此目录，打开终端，在终端中输入命令：

You can access this directory with the following command, open the terminal, and enter the command in the terminal:

注：在运行命令之前，请确保其他终端中的程序已经终止，终止命令为：Ctrl+c

Note: Before running the command, please make sure that the programs in other terminals have been terminated. The termination command is: Ctrl+c

```
cd agilex_ws/src/limo_ros/limo_base/script
```

运行脚本，打开终端，在终端中输入命令：

Run the script, open the terminal, and enter the command in the terminal:

```
python3 limomove.py
```

在终端输入命令之后，limo会先前行走一段距离，然后停下。

After entering the command in the terminal, limo will walk a certain distance before stopping.

驱动程序中所使用的函数名称：

Functions used in the driver:

函数名称 **Function's name**	**函数作用** **Function's function**
EnableCommand()	控制使能 Control enable
SetMotionCommand()	设置移动命令 Set motion command
GetLinearVelocity()	获取线速度 Get linear velocity
GetAngularVelocity()	获取角速度 Get angular velocity
GetSteeringAngle()	获取内转角角度 Get steering angle
GetLateralVelocity()	获取横移速度 Get lateral velocity
GetControlMode()	获取控制模式 Get control mode
GetBatteryVoltage()	获取电池电量 Get battery level
GetErrorCode()	获取错误代码 Get error code

GetRightWheelOdem()	获取左轮里程计 Get left wheel odometer
GetLeftWheelOdem()	获取右轮里程计 Get right wheel odometer
GetIMUAccelData()	获取IMU的加速度 Get IMU acceleration
GetIMUGyroData()	获取陀螺仪的数据 Get gyroscope data
GetIMUYawData()	获取IMU的航向角 Get IMU course angle
GetIMUPichData()	获取俯仰角 Get pitch angle
GetIMURollData()	获取横滚角 Get roll angle

四、底盘运动学分析

4. Chassis Kinematics Analysis

现在移动机器人这么火热，大到无人驾驶车，规矩的有工业上应用得很多的AGV（比如智能物流自动搬运机器人），小到淘宝上面的智能小车，都可以算作移动机器人。移动机器人有各种各样的底盘，有两轮的三轮的四轮的，比如无人车是四轮的阿克曼模型，一般的AGV是两轮差速模型，还有大学生机器人竞赛里面常见的三轮全向轮底盘，四轮全向轮底盘，还有一些AGV是四轮滑移底盘，是不是有点让人眼花缭乱的感觉呢；本节将从运动学方面介绍limo的四种运动模式。

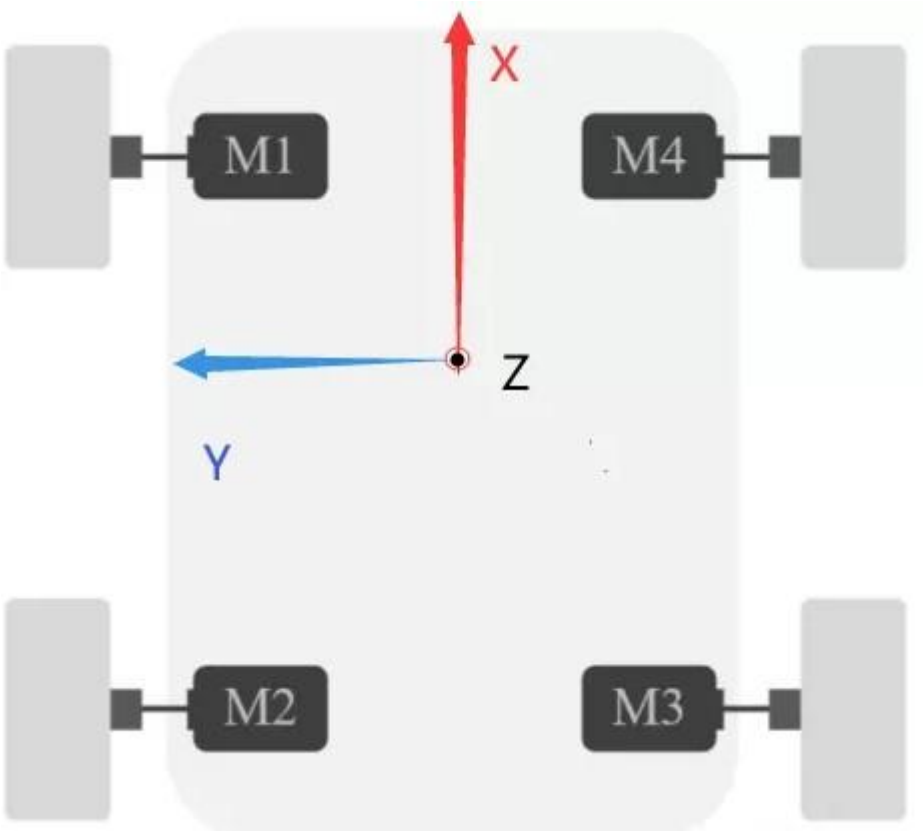
Mobile robots are so popular nowadays, including large robots like unmanned vehicles, regulars robots like AGVs (eg. intelligent logistics automatic handling robots) which are widely used in industry, and small robots like the smart vehicles on Taobao. Mobile robots have a variety of chassis, including two-wheel, three-wheel, and four-wheel ones. For example, unmanned vehicles are four-wheel Ackermann models, and general AGVs are two-wheel differential models. There are also three-wheel omni-wheel chassis and four-wheel omni-wheel chassis that are common in college student robot competitions. There are also some AGVs with four-wheel skid chassis. Is it a bit dazzling? This section will introduce the four motion modes of limo from the kinematics aspect.

4.1 四轮差速运动模式

4.1 Four-wheel differential motion mode

我们把一台四轮差速运动的机器人简化成下图的模型，其中四个车轮由四个单独的电机驱动，只需要控制四个车轮的速度，就可以达到控制机器人的前进、后退、以及转向的运动效果了。

We simplify a four-wheel differential motion robot into the model shown in the figure below. Four wheels are driven by four separate motors. You can control the robot to move forward, backward, and steer only by controlling the velocity of the four wheels.



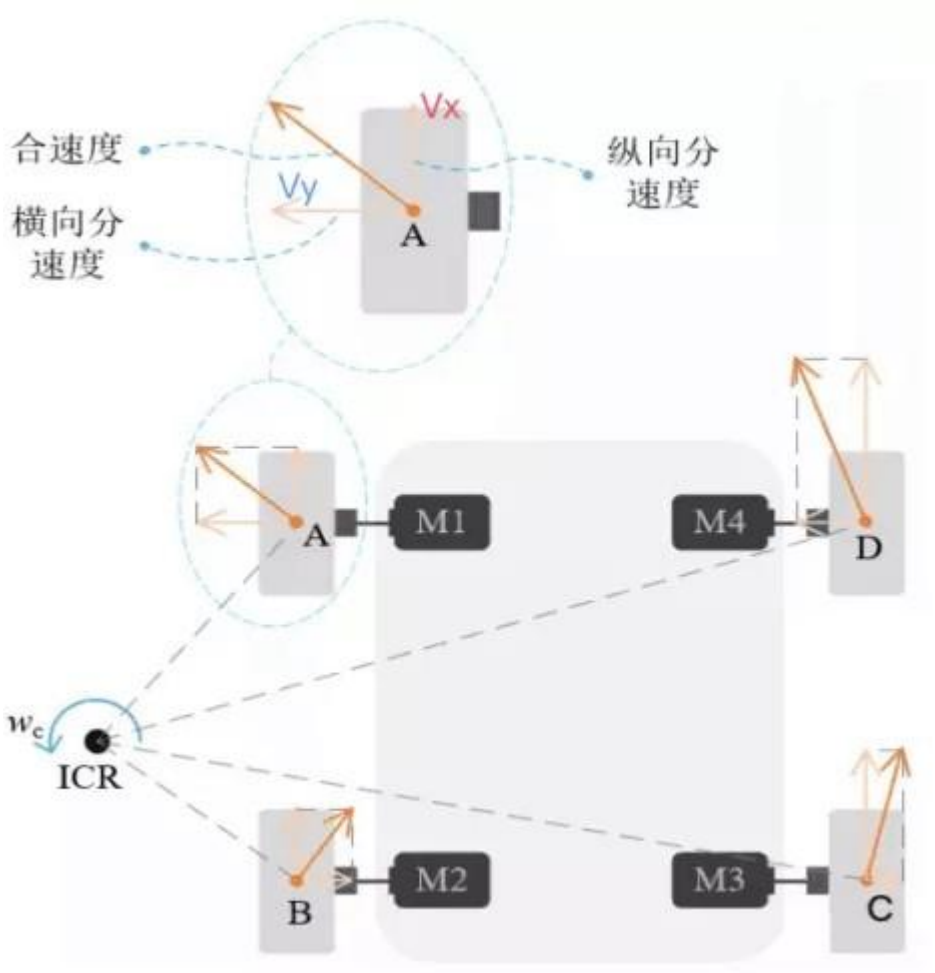
我们给机器人一个参考坐标系，红色箭头为X正方向，蓝色箭头为Y轴正方向，Z轴沿着原点垂直屏幕向外，坐标原点为机器人的质心，坐标系满足安培右手定则。当四个车轮的速度（大小+方向）一样时，机器人就可以实现前进和后退。当四个车轮的速度不一样时，机器人将会产生转向运动。

We give the robot a reference coordinate system. The red arrow is the positive direction of the X axis, the blue arrow is the positive direction of the Y axis, the Z axis is perpendicular to the screen outward along the origin, the coordinate origin is the center of mass of the robot, and the coordinate system satisfies the Ampere’ s right-hand rule. When the velocity (size +

direction) of the four wheels is the same, the robot can move forward and backward. When the velocities of the four wheels are not the same, the robot will turn.

一旦机器人产生转向运动，那么意味着有一个转向中心即下图中的点ICR。以左前轮为例，轮子与地面接触点A的相对运动速度方向如图所示，合速度方向与线段A-ICR相互垂直，而轮胎只能沿着纵向分速度方向转动，做速度分解可知，还存在沿轮子轴向（电机轴向）的横向分速度。

Once the robot turns, it means that there is a turning center, which is the point ICR in the figure below. Taking the left front wheel as an example, the relative velocity direction of the contact point A of the wheel and the ground is shown in the figure. The resultant velocity direction and the line segment A-ICR are perpendicular to each other, and the tire can only rotate along the longitudinal component velocity direction. The velocity resolution shows that there is also a lateral component velocity along the wheel axis (motor axis).



原文	译文
合速度	Resultant velocity
横向分速度	Lateral component velocity
纵向分速度	Longitudinal component velocity

由于四个轮胎的横向分速度大小不同，因此机器人会产生旋转分运动，而纵向分速度产生纵向分运动，合成运动则表现为机器人绕ICR做圆周运动。

Since the lateral component velocities of the four tires are different, the robot will produce rotational sub-motions, while the longitudinal component velocities produce longitudinal sub-motions. The resultant motion is represented by the robot making a circular motion around the ICR.

在机器人转弯时，两侧的轮胎角速度一样，内侧的轮胎转弯半径小，所以线速度更小，外侧的轮胎转弯半径大，所以外侧线速度更大。即右侧速度大时，将会向左转，同理可知，左侧速度大时，将会向右转。

When the robot turns, the angular velocity of the tires on both sides is the same. The inner tire has a smaller turning radius, so the linear velocity is smaller, and the outer tire has a larger turning radius, so the outer linear velocity is greater. That is, when the velocity on the right side is greater, it will turn left. Similarly, when the velocity on the left side is greater, it will turn right.

而且从上图可以看出同侧的轮胎Vx（纵向分速度）一样，上（下）端的轮胎Vy（横向分速度）一样。如果想要绕自身自转的话，只需要左（右）侧的两个车轮速度一样大，且方向相同、右（左）侧的两个车轮速度与另一侧一样，但方向与另一侧相反即可。

Moreover, it can be seen from the figure above that the tires on the same side have the same Vx (longitudinal component velocity), and the tires at the upper (lower) end have the same Vy (lateral component velocity). If you want the vehicle to rotate around itself, you only need the two wheels on the left (right) side to have the same velocity and the same direction, while the two wheels on the right (left) side have the same velocity as the other

side, but the direction is opposite to the other side.

我们可以尝试控制四轮差速模式下的limo，首先把limo调整为四轮差速模式， 拔起来顺时针转30度， 使两插销上较短的线指向车体正前  ，此时为插入状态，微调轮胎角度对准孔位让插销插入，车灯变为黄色则切换成功。

We can try to control limo in four-wheel differential mode. First, adjust limo to four-wheel differential mode, pull up the two latches, and turn 30 degrees clockwise to make the shorter line on the two latches points to the front of the vehicle body  . At this point, it is in insertion state. Fine-tune the tire angle to align the hole so that the latch is inserted. When the vehicle light turns yellow, the switch is successful.

模式切换成功之后，运行以下命令，我们就可以启用键盘或者手柄来控制。

After the mode switch is successful, run the following command, and we can launch the keyboard or handle to control.

启动底盘控制节点

Launch chassis control node

注：在运行命令之前，请确保其他终端中的程序已经终止，终止命令为：Ctrl+c

Note: Before running the command, please make sure that the programs in other terminals have been terminated. The termination command is: Ctrl+c

```
roslaunch limo_base limo_base.launch
```

启动键盘控制节点

Launch keyboard control node

```
roslaunch limo_bringup limo_teletop_keyboard.launch
```

4.2 履带运动模式

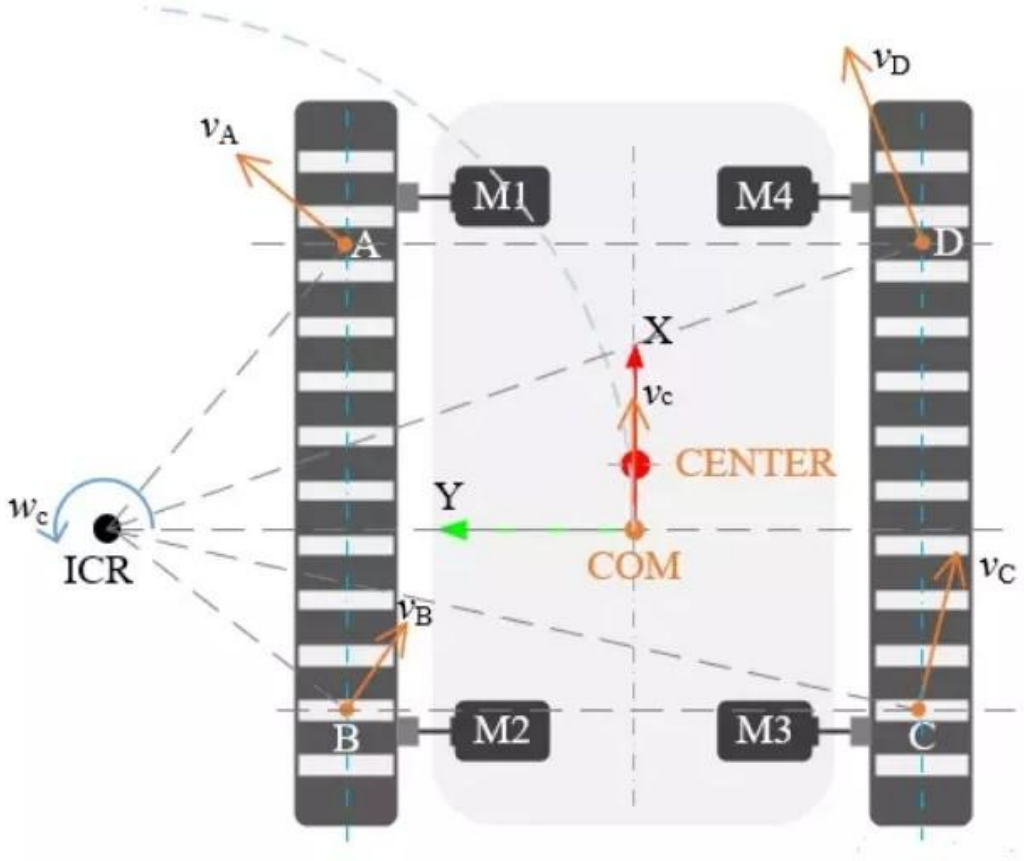
4.2 Track motion mode

履带差速模式下单侧履带可等效视为“无穷多个小轮子”，且这单侧的“无穷多个小轮子”的“转速”是一致的。所以，履带差速模式的转向方式和四轮差速模式的转向方式是一致的，也是滑动转向。

In the track differential mode, a single-sided track can be equivalently regarded as an "infinite number of small wheels", and the "speed" of the single-sided "infinite number of small wheels" is the same. Therefore, the steering mode of the track differential mode is the same as that of the four-wheel differential mode, which is also slide steering.

具体来讲，履带差速运动和四轮差速运动均是通过控制两侧履带（或轮子）的相对速度实现，但二者也有区别：履带对地面产生的剪切和压力分布，是不同于轮子的。此区别体现在车轮速度控制上时影响不大。车轮速度不一致时，我们参照可以以下简化后的模型

Specifically, the track differential motion and the four-wheel differential motion are achieved by controlling the relative velocity of the tracks (or wheels) on both sides, but there are also differences between them: the shear and pressure distribution generated by the track on the ground are different from those of the wheels. This difference has little effect when it comes to wheel speed control. When the wheel speeds are inconsistent, we can refer to the following simplified model



ICR为运动旋转中心，CENTER为机器人几何学中心，COM为机器人质心。转弯时，内侧履带的速度小于外侧履带的速度，如果想要绕自身自转的话，只需要左（右）侧履带速度一样大，且方向相同、右（左）侧的履带速度与另一侧一样，但方向与另一侧相反即可。

ICR is the center of motion rotation, CENTER is the geometric center of the robot, and COM is the center of mass of the robot. When turning, the velocity of the inner track is lower than that of the outer track. If you want the vehicle to rotate around itself, you only need the left (right) side track to have the same velocity and the same direction, while the track velocity on the right (left) side is the same as the other side, but the direction is opposite to the other side.

在四轮差速模式下将履带直接套上，建议先套空间较小的后轮，并且履带模式下请将两侧车门抬起防止刚蹭。

In the four-wheel differential mode, put the track on directly; it is recommended to put the track on the rear wheel with small space first, and in the track mode, please lift the doors on both sides to prevent scratches.

更换完成之后，运行以下命令，我们就可以启用键盘或者手柄来控制。

After the replacement is completed, run the following command, and we can launch the keyboard or handle to control.

启动底盘控制节点

Launch chassis control node

注：在运行命令之前，请确保其他终端中的程序已经终止，终止命令为：Ctrl+c

Note: Before running the command, please make sure that the programs in other terminals have been terminated. The termination command is: Ctrl+c

```
roslaunch limo_base limo_base.launch
```

启动键盘控制节点

Launch keyboard control node

```
roslaunch limo_bringup limo_teletop_keyboard.launch
```

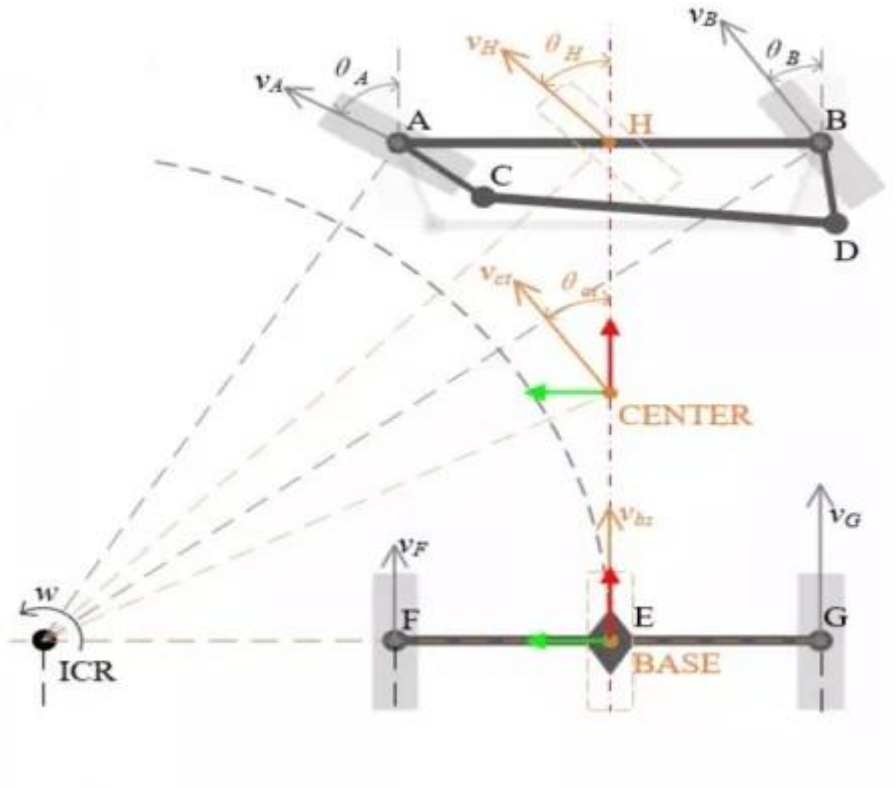
4.3 阿克曼运动模式

4.3 Ackermann motion mode

阿克曼转向结构是现代汽车的转向方式，可以解决汽车在转向时，由于左、右转向轮的转向半径不同所造成的左、右转向轮转角不同的问题。

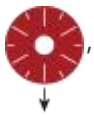
Ackermann steering structure is the steering method of modern cars, which can solve the problem of different steering angles of the left and right steering wheels caused by the different steering radii of the left and right steering wheels when the car is steering.

首先看一下阿克曼运动模态简化之后的模型图。与差速运动相同，如果我们给四个车轮相同的的速度（大小+方向），即可实现机器人的前进和后退。不同之处在于转弯，需要根据前端两个车轮的偏转方向，利用阿克曼转向几何来计算转弯半径。点CENTER为机器人几何中心，点BASE为后杆的中点。此时，机器人将绕ICR做圆周运动，此转弯半径也为最小转弯半径。由图可知，前段两个车轮的偏角并不一致，两个车轮偏角的差值（θA-θB）为阿克曼角。如果机器人在阿克曼运动模式下还是采用四轮驱动的方式，在转弯时，内侧车轮的速度小于外侧车轮。
First look at the simplified model of Ackermann motion mode. Same as the differential motion, if we give the four wheels the same velocity (size + direction), the robot can move forward and backward. The difference lies in turning. It is necessary to use the Ackermann steering geometry to calculate the turning radius based on the deflection direction of the two front wheels. The point CENTER is the geometric center of the robot, and the point BASE is the midpoint of the rear rod. At this time, the robot will make a circular motion around the ICR, and this turning radius is also the minimum turning radius. It can be seen from the figure that the deflection angles of the two wheels in the front section are not the same, and the difference between the deflection angles of the two wheels (θA-θB) is the Ackermann angle. If the robot still uses four-wheel drive in the Ackermann motion mode, the velocity of the inner wheel is lower than that of the outer wheel when turning.



阿克曼进一步简化，就是平时我们骑得自行车模型，上图橘黄色部分是机器人等效的自行车模型。相比于差速运动模态，阿克曼运动模态有转弯半径的限制，移动机器人不能实现自旋，即转弯半径不能为0。

Ackermann can be further simplified as the model of the bicycles we usually ride. The orange part in the figure above is the robot’ s equivalent bicycle model. Compared with the differential motion mode, the Ackermann motion mode has a turning radius limitation, and the mobile robot cannot achieve spin, that is, the turning radius cannot be zero.

先将两侧插销拔起，顺时针转30度，使两插销上较长的线指向车体正前方  ，即可卡住，车灯变为绿色则切换成功，limo就变换为阿克曼运动模态了。

First pull up the latches on both sides, and turn 30 degrees clockwise to make the longer line on the two latches points to the front of the vehicle body, and then they will be stuck. When the light turns green, the switch is successful, and the limo is switched to Ackermann motion mode.

模式切换成功之后，运行以下命令，我们就可以启用键盘或者手柄来控制。

After the mode switch is successful, run the following command, and we can launch the keyboard or handle to control.

切换成功之后，运行以下命令，我们就可以启用键盘或者手柄来控制。

After the switch is successful, run the following command, and we can launch the keyboard or handle to control.

启动底盘控制节点

Launch chassis control node

注：在运行命令之前，请确保其他终端中的程序已经终止，终止命令为：Ctrl+c

Note: Before running the command, please make sure that the programs in other terminals have been terminated. The termination command is: Ctrl+c

```
roslaunch limo_base limo_base.launch
```

启动键盘控制节点

Launch keyboard control node

```
roslaunch limo_bringup limo_teletop_keyboard.launch
```

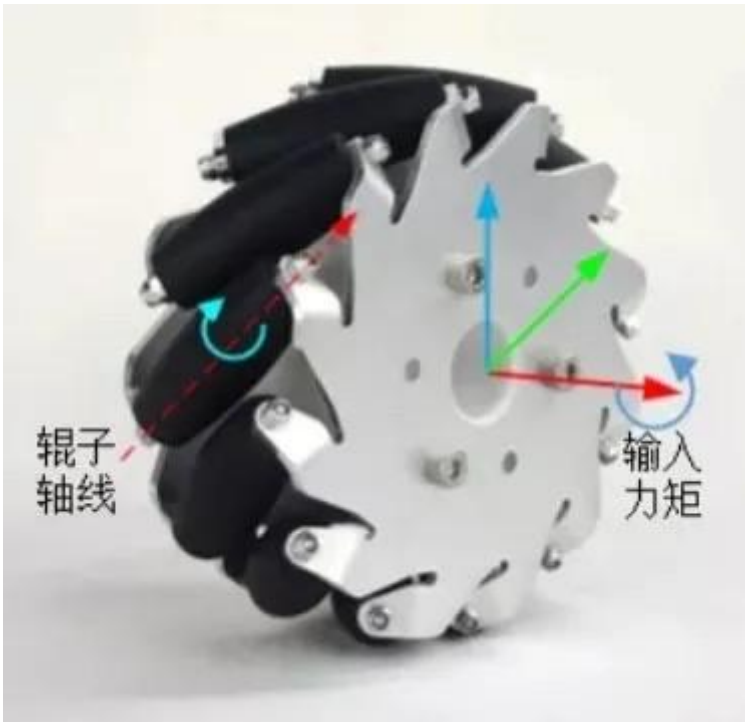
注：当在阿克曼模式下走不直时需要进行舵机校准。

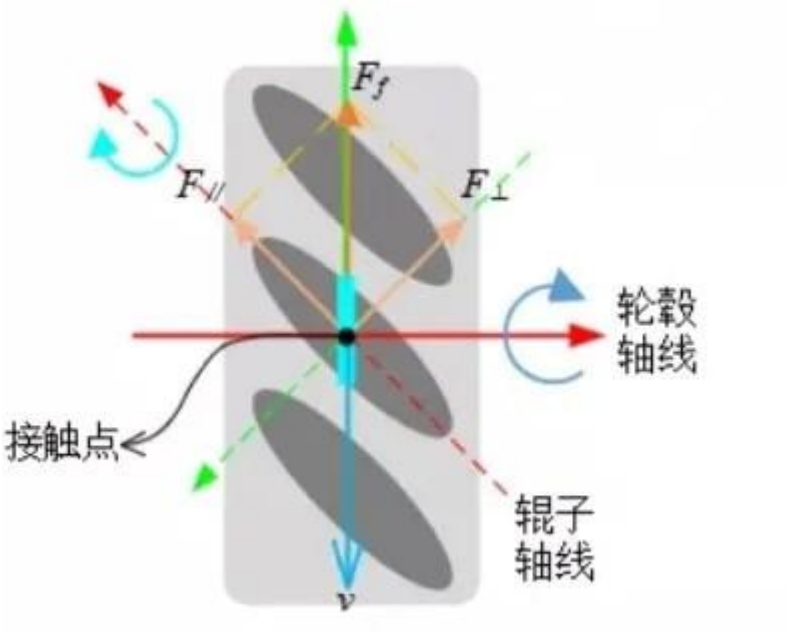
Note: When the vehicle cannot go straight in Ackermann mode, the steering gear calibration is required.

4.4 麦克纳姆运动模式

4.4 Mecanum motion mode

麦克纳姆轮是一种特殊的车轮，由轮毂和辊子组成：轮毂是整个轮子的主体支架，辊子则是安装在轮毂上的被动运动的鼓状物（小轮），两者组成一个完整的大轮。市面上轮毂轴线与辊子转轴夹角大致可分30度、45度、60度三种。为满足全向运动的几何关系，轮毂边缘采用了折弯工艺，为辊子的转轴提供安 装孔。





The Mecanum wheel is a special kind of wheel, which is composed of a hub and rollers: the hub is the main support of the entire wheel, and the rollers are passively moving drums (small wheels) mounted on the hub. The two form a complete big wheel. The angle between the hub axis and the roller shaft on the market can be roughly divided into 30 degrees, 45 degrees, and 60 degrees. In order to meet the geometric relationship of omnidirectional motion, the edge of the hub adopts a bending process to provide mounting holes for the shaft of the roller.

原文	译文
辊子曲线	Roller curve
输入力矩	Input torque

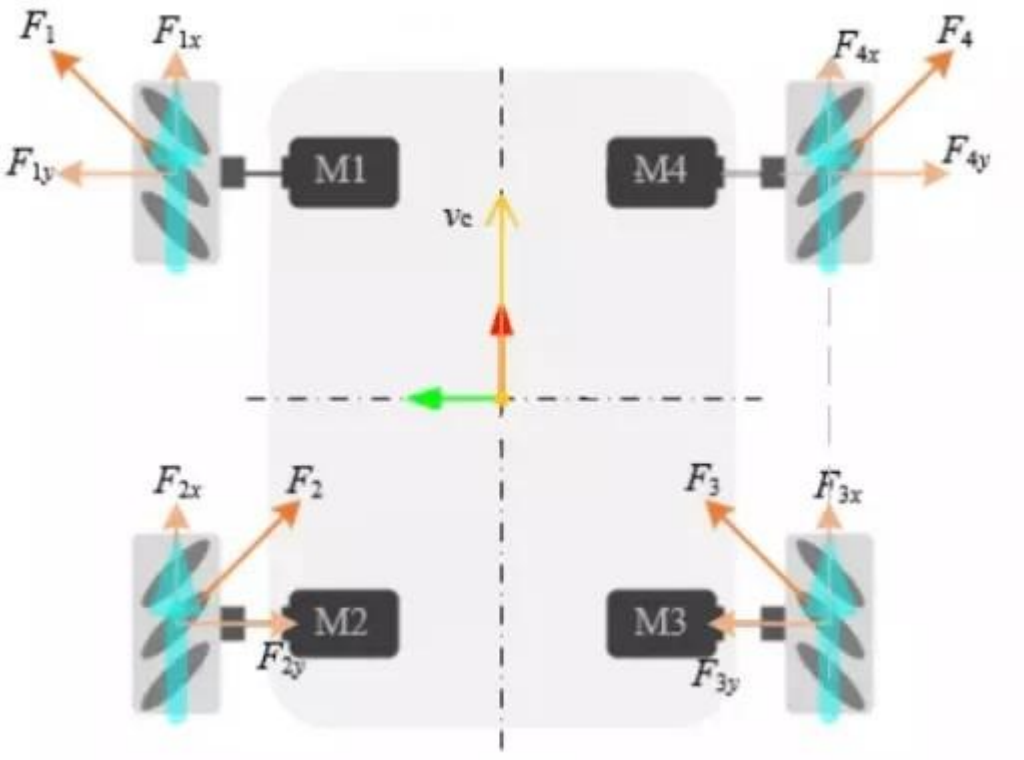
接触点	Contact point
轮毂曲线	Hub curve

假定车轮逆时针转动，对麦克纳姆轮进行受力分析，上图中坐标系红色表示x轴，绿色表示y轴，蓝色表示z轴，辊子坐标系用虚线表示，轮毂坐标系用实线表示；黄色箭头表示麦轮和辊子的受力分析；蓝色箭头表示速度方向。

Assuming that the wheel rotates counterclockwise, the force analysis of the Mecanum wheel is carried out. In the coordinate system in the above figure, red represents the x axis, green represents the y axis, blue represents the z axis, the roller coordinate system is represented by a dotted line, and the hub coordinate system is represented by a solid line; the yellow arrow indicates the force analysis of the Mecanum wheel and roller; the blue arrow indicates the speed direction.

麦轮外围的辊子是与地面接触的，当麦轮绕轮毂轴转动时，辊子会与地面产生摩擦力Ff，其作用力方向为轮毂坐标系Y轴正方向。对Ff沿着辊子坐标系做正交分解，F1沿辊子Y轴负方向，大小为√2/2Ff，FⅡ沿辊子X轴正方向，大小为√2/2Ff。F1为辊子的滚动摩擦力，对辊子造成磨损，并不能改变轮胎的运动方向，FⅡ会迫使辊子沿X轴正方向运动，故FⅡ为静摩擦，促使辊子相对地面运动。

The rollers on the periphery of the Mecanum wheel are in contact with the ground. When the Mecanum wheel rotates around the hub axle, the rollers will generate frictional force Ff with the ground, and the force direction is the positive direction of the Y axis of the hub coordinate system. The orthogonal decomposition of Ff along the roller coordinate system shows that F1 is along the negative direction of the Y axis of the roller, and the size is √2/2Ff, and FII is along the positive direction of the X axis of the roller, and the size is √2/2Ff. F1 is the rolling friction of the roller, which causes wear to the roller and cannot change the direction of movement of the tire. FⅡ will force the roller to move in the positive direction of the X axis, so FⅡ is static friction, which promotes the roller to move relative to the ground.

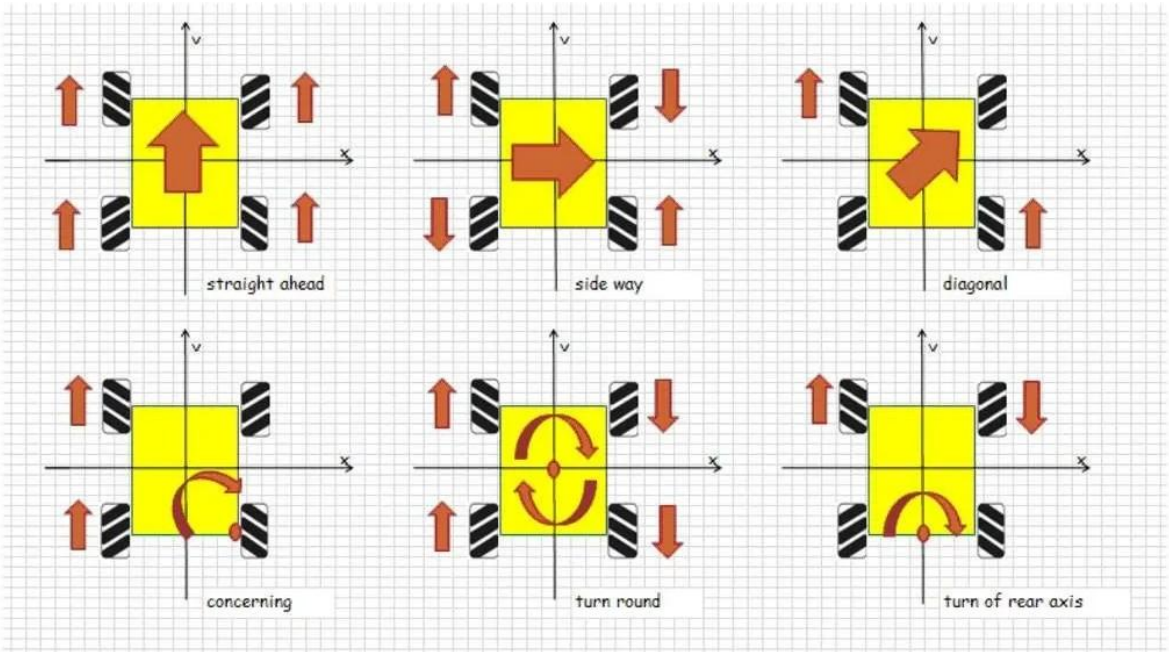


不同的车轮排列方式也需采用不同的控制方式，将麦轮的全向运动简化成上图这个模型，蓝色箭头表示车轮运动方向，橘黄色箭头表示麦轮的受力分析情况。将各力合并，我们将发现移动机器人只有一个向前的力，故此时机器人将向前运动。

Different wheel arrangements also require different control methods. The omnidirectional movement of the Mecanum wheel is simplified to the model shown in the figure above. The blue arrow indicates the direction of wheel movement, and the orange arrow represents the force analysis of the Mecanum wheel. Combining the forces, we will find that the mobile robot has only one forward force, so the robot will move forward at this time.

结合之前四轮差速运动模式，对各个轮胎的摩擦力进行合并，就可使机器人往任意方向去运动。下面列举一些不同方向运动时轮胎的状态。

Combine the friction of each tire based on the previous four-wheel differential motion mode, the robot can move in any direction. Below are some examples of tire conditions when moving in different directions.



先将轮毂盖和轮胎拆下，只保留轮毂电机，然后保证每个麦轮的小滚子朝向车体中心，用包装里的M3*5 螺

丝将麦轮安装上，遥控操作的时候需要将遥控器/APP调整至麦轮模式。

First remove the hub cover and tires, leaving only the hub motor, and then make sure that the small rollers of each Mecanum wheel are facing the center of the vehicle body. Use the M3*5 screws in the package to install the Mecanum wheel. The remote control/APP needs to be adjusted to the Mecanum wheel mode during remote operation.

遥控手柄在四轮差速控制模式下，通道8处于最下档时切换为麦轮控制模式，处于中上档时切回差速模式；

When the remote control handle is in the four-wheel differential control mode, when channel 8 is in the lowest gear, it switches to the Mecanum wheel control mode, and when it is in the middle and upper gear, it switches back to the differential mode;

启动底盘控制节点

Launch chassis control node

注：在运行命令之前，请确保其他终端中的程序已经终止，终止命令为：Ctrl+c

Note: Before running the command, please make sure that the programs in other terminals have been terminated. The termination command is: Ctrl+c

```
roslaunch limo_base limo_base.launch
```

启动键盘控制节点

Launch keyboard control node

```
roslaunch limo_bringup limo_teletop_keyboard.launch
```

移动机器人的运动模态有很多，每一种都有其优缺点，在实际场景中我们可以根据场景特点和需求，选用最合适的运动模态。下表总结了四种运动模式的优缺点，仅供参看。

There are many motion modes of mobile robots, and each has its advantages and disadvantages. In actual scenes, we can choose the most suitable motion mode according to the characteristics and needs of the scene. The following table summarizes the advantages and disadvantages of the four motion modes, for reference only.

模态 Mode	**四轮差速 Four-wheel differential**	**履带 Track**	**阿克曼 Ackermann**	**麦克纳姆 Mecanum**
优点 Advantages	运动性能好，控制简单 Good motion performance and simple control	适应性强，具有良好的越野性能，适用于多种复杂场地 Strong adaptability, good off-road performance, and suitable for a variety of complex sites	与汽车的运动模式相同，便于做深入的自动驾驶研究 Same as the motion mode of a car, which facilitates in-depth research on autonomous driving	机动性好，可以全向运动 Good mobility, and omnidirectional motion
缺点 Disadvantages	转向有滑移，轮胎的磨损比较大 Slippage during steering, and large tire wear	滑动转向阻力大，对履带的磨损较大 Large sliding steering resistance, and large wear on the track	转弯半径有限，轮胎磨损小 Limited turning radius, and low tire wear	场地要求较高，辊子间不连续，运动时有震动，磨损较大 High site requirements, non-continuous rollers, vibration during movement, and large wear

五、雷达建图

5. LiDAR Mapping

5.1 雷达介绍和使用

5.1 Introduction and use of LiDAR

YDLIDAR X2L 激光雷达是深圳玩智商科技有限公司（EAI）研发的一款 360 度二维测距产 品。本产品基于三角测距原理，并配以相关光学、电学、算法设计，实现 高频高精度的距离测量，在测距的同时，机械结构 360 度旋转，不断获取角度信息，从而实 现了 360 度扫描测距，输出扫描环境的点云数据。

YDLIDAR X2L LiDAR is a 360-degree two-dimensional ranging product developed by Shenzhen EAI Technology Co., Ltd. (EAI). This product is based on the principle of trigonometric ranging and is equipped with related optical, electrical, and algorithm design to achieve high-frequency and high-precision distance measurement. While measuring distance, the mechanical structure rotates 360 degrees and continuously obtains angle information, thereby achieving 360 degrees scanning distance measurement, and output point cloud data of the scanning environment.

项目 **items**	**最小值** **Minimum**	**默认值** **Default**	**最大值** **Maximum**	**单位** **Unit**	**备注** **Remarks**
测距频率 Ranging frequency	/	3000	/	Hz	每秒测距 3000 次 3000 ranging times per second
扫描频率 Scanning frequency	5	6	8	Hz	需要接入 PWM 信号，推荐 使用转速 6Hz PWM signal needs to be connected, and the recommended frequency is 6Hz
测距范围 Ranging range	0.12	/	8	m	室内环境，80%反射率物体 Indoor environment, and objects with 80% reflectivity
扫描角度 Scanning angle	/	0-360	/	°	/
绝对误差 Absolute error	/	2	/	cm	测距≤1m 时 When ranging ≤1m
相对误差 Relative error	/	3.5%	/	/	1m <测距≤6m 时 When 1m < ranging ≤ 6m
俯仰角 Pitch angle	0.25	1	1.75	°	/
角度分辨率 Angle resolution	0.60 (5Hz)	0.72 (6Hz)	0.96 (8Hz)	°	不同扫描频率 Different scanning frequency

其使用方式如下：

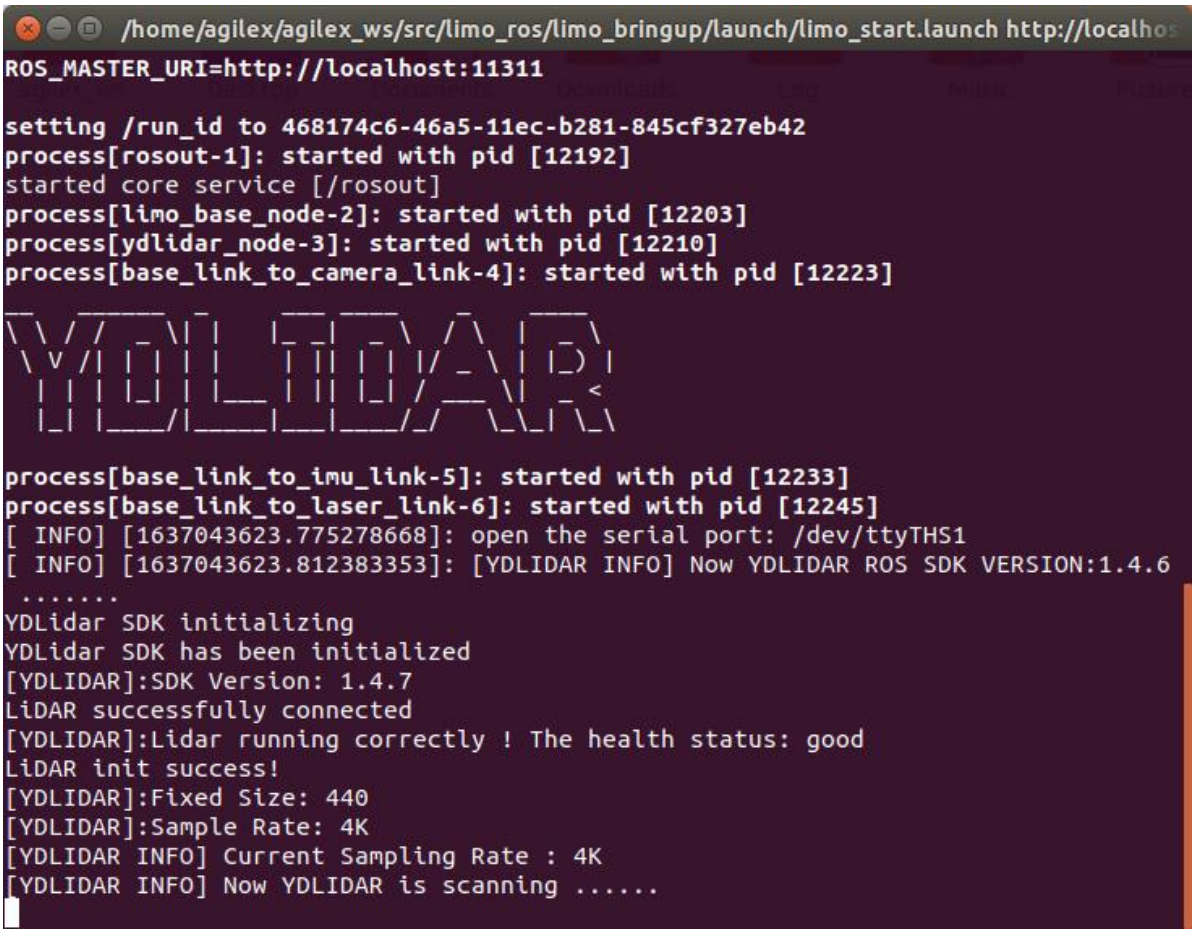
Its usage is as follows:

打开一个新的终端，在终端中输入命令：

Launch a new terminal and enter the command in the terminal:

```
roslaunch limo_bringup limo_start.launch pub_odom_tf:=false
```

成功打开之后，终端会输出以下日志信息，如图：



After launching successfully, the terminal will output the following log information, as shown in the figure:

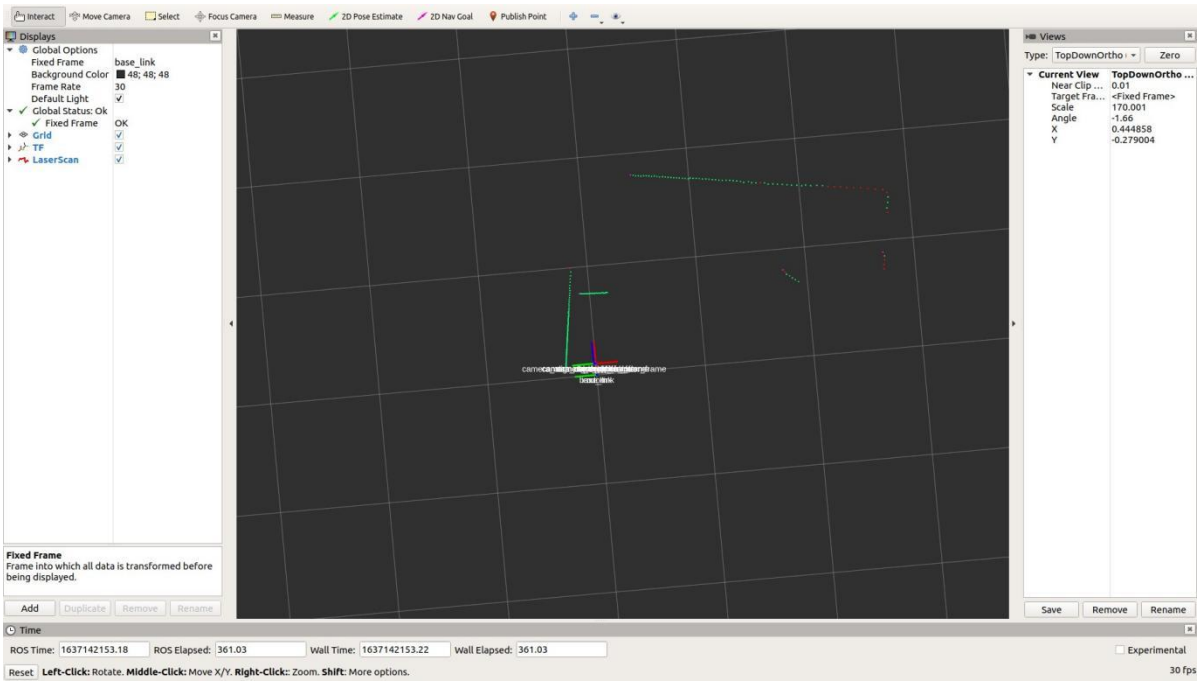
然后新开启一个终端，在终端中输入命令：

Then launch a new terminal and enter the command in the terminal:

```
roslaunch limo_bringup limo_lidar_rviz.launch
```

成功运行之后会看到rviz可视化工具打开，其中显示的绿色数据就是雷达扫描出来的激光数据。

After successfully running, you will see the rviz visualization tool open, and the green data displayed is the laser data scanned by the LiDAR.



这时候可以把遥控器调为遥控模式，遥控小车进行移动，这时会看到激光的数据也会跟着变化。

At this time, you can set the remote control to remote control mode to remote control the vehicle to move. At this time, you will see that the laser data will also change.

5.2 gmapping 建图

5.2 gmapping mapping

5.2.1 gmapping建图算法介绍

5.2.1 Introduction to gmapping mapping algorithm

Gmapping是基于滤波SLAM框架的常用开源SLAM算法。Gmapping有效利用了车轮里程计信息，对激 光雷达的频率要求不高，在构建小场景地图时，所需的计算量较小且精度较高。这里通过使用ROS封装 了的GMapping功能包来实现limo的建图。

Gmapping is a commonly used open source SLAM algorithm based on the filtering SLAM framework. Gmapping effectively utilizes the wheel odometer information and does not require high frequency of laser LiDAR. When building a small scene map, the amount of calculation required is small and the accuracy is high. Here, the GMapping package encapsulated by ROS is used to realize the mapping of limo.

5.2.2 gmapping建图实践操作

5.2.2 Operation of gmapping mapping

注：在运行命令之前，请确保其他终端中的程序已经终止，终止命令为：Ctrl+c

Note: Before running the command, please make sure that the programs in other terminals have been terminated. The termination command is: Ctrl+c

注：建图过程中limo的速度尽量慢点，速度太快会影响建图的效果

Note: The speed of limo should be as slow as possible in the process of mapping. If the speed is too fast, the effect of mapping will be affected

首先需要启动雷达，打开一个新终端，在终端中输入命令：
First, you need to launch the LiDAR; open a new terminal, and enter the command in the terminal:

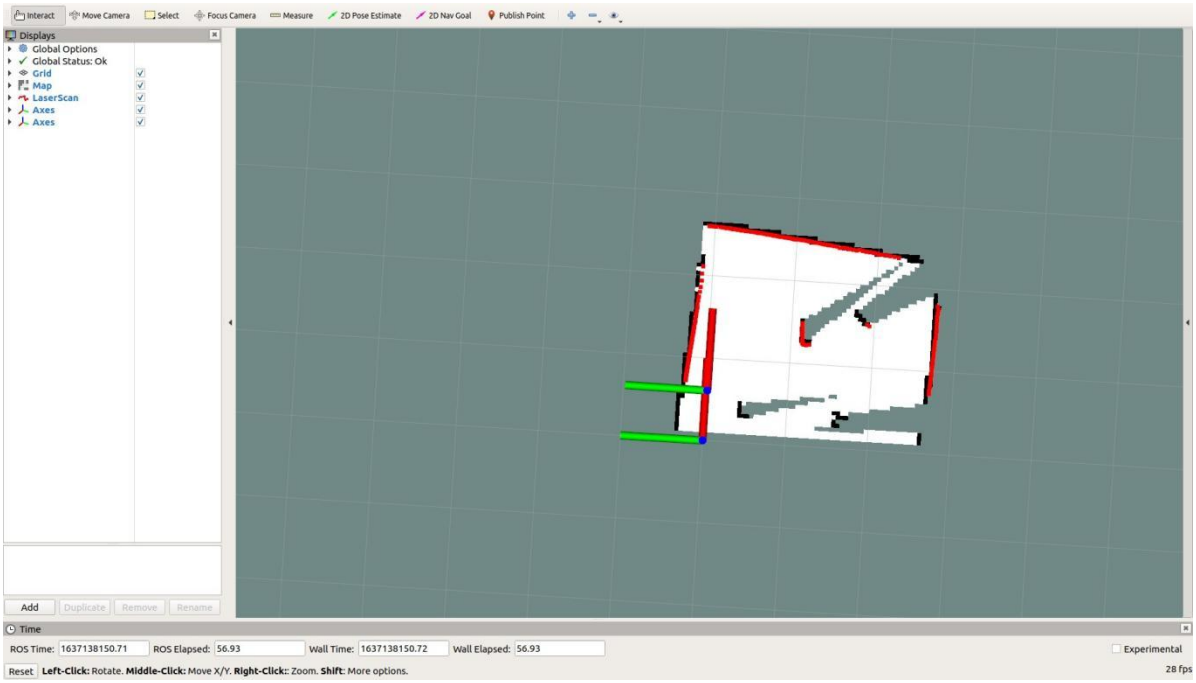
```
roslaunch limo_bringup limo_start.launch pub_odom_tf:=false
```

然后启动gmapping建图算法，打开另一个新终端，在终端中输入命令：

Then launch the gmapping mapping algorithm; open another new terminal, and enter the command in the terminal:

```
roslaunch limo_bringup limo_gmapping.launch
```

成功启动之后会打开rviz可视化工具，这时候看到的界面如图



After launching successfully, the rviz visualization tool will be opened, and the interface you will see at this time is shown in the figure

这时候就可以把手柄调为遥控模式，控制limo建图了。

At this time, you can adjust the handle to remote control mode to control limo to map.

构建完地图之后，需要运行以下命令，把地图保存到指定目录：

After building the map, you need to run the following command to save the map to the specified directory:

1、切换到需要保存地图的目录下，这里把地图保存到/agilex_ws/limo_bringup/maps，在终端中输入 命令：

1. Switch to the directory where you need to save the map, save the map to /agilex_ws/limo_bringup/maps, and enter the command in the terminal:

```
cd /agilex_ws/limo_bringup/maps/
```

2、切换到/agilex_ws/limo_bringup/maps 之后，继续在终端中输入命令：

2. After switching to /agilex_ws/limo_bringup/maps, continue to enter the command in the terminal:

```
rosrun map_server map_saver -f map1
```

注：map1为保存地图的名称，保存地图时应避免地图的名称重复

Note: map1 is the name of the saved map, and duplicate names should be avoided when saving the map

5.3 cartographer建图

5.3 Cartographer mapping

5.3.1 cartographer建图算法介绍

5.3.1 Introduction to cartographer mapping algorithm

cartographer是google推出的一套基于图优化的SLAM算法。该算法的主要目标是实现低计算资源消耗，达到实时SLAM的目的。该算法主要分为两个部分，第一个部分称为Local SLAM，该部分通过一帧帧的Laser Scan建立并维护一系列的Submap，而所谓的submap就是一系列的Grid Map。算法的第二个部分，称为Global SLAM的部分，就是通过Loop Closure来进行闭环检测，来消除累积误差：当一个submap构建完成，也就是不会再有新的laser scan插入到该submap时，算法会将该submap加入到闭环检测中。

Cartographer is a set of SLAM algorithms based on image optimization launched by Google. The main goal of this algorithm is to achieve low computing resource consumption and achieve the purpose of real-time SLAM. The algorithm is mainly divided into two parts. The first part is called Local SLAM. This part establishes and maintains a series of Submaps through a frame of Laser Scan, and the so-called submap is a series of Grid Maps. The second part of the algorithm, called Global SLAM, is to perform closed-loop detection through Loop Closure to eliminate accumulated errors: when a submap is built, no new laser scans will be inserted into the submap. The algorithm will add the submap to the closed-loop detection.

5.3.2 cartographer建图实践操作

5.3.2 Practical operation of cartographer mapping

注：在运行命令之前，请确保其他终端中的程序已经终止，终止命令为：Ctrl+c

Note: Before running the command, please make sure that the programs in other terminals have been terminated. The termination command is: Ctrl+c

注：建图过程中limo的速度尽量慢点，速度太快会影响建图的效果

Note: The speed of limo should be as slow as possible in the process of mapping. If the speed is too fast, the effect of mapping will be affected

首先需要启动雷达，打开一个新终端，在终端中输入命令：

First, you need to launch the LiDAR; open a new terminal, and enter the command in the terminal:

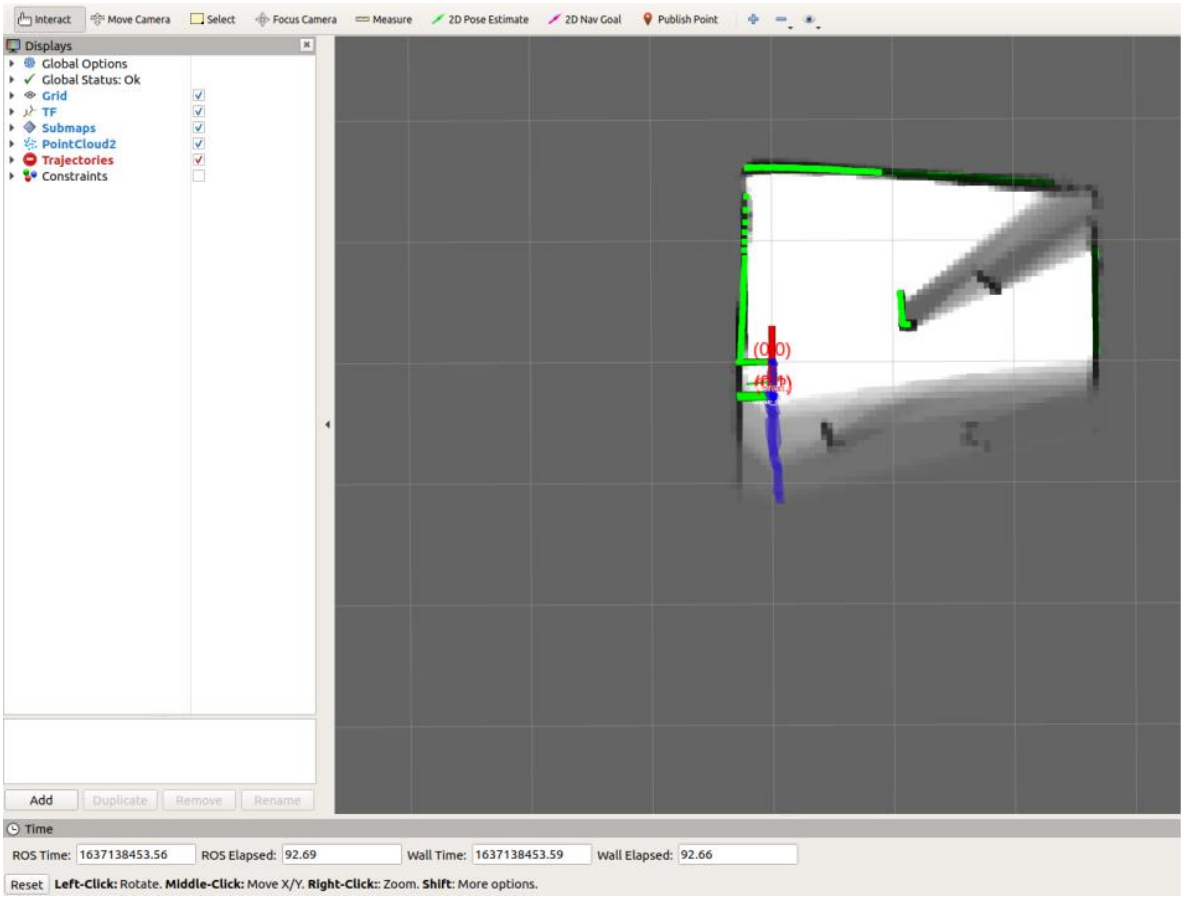
```
roslaunch limo_bringup limo_start.launch pub_odom_tf:=false
```

然后启动cartographer建图算法，打开另一个新终端，在终端中输入命令：

Then start the cartographer mapping algorithm; open another new terminal, and enter the command in the terminal:

```
roslaunch limo_bringup limo_cartographer.launch
```

成功启动之后会弹出rviz可视化界面，如下图：



After launching successfully, the rviz visualization interface will pop up, as shown in the figure below:

在构建完地图之后需要保存地图，需要在终端中输入以下三条命令：

After building the map, you need to save the map. And you need to enter the following three commands in the terminal:

（1）完成轨迹, 不接受进一步的数据。

(1) After completing the trajectory, no further data should be accepted.

```
rosservice call /finish_trajectory 0
```

（2）序列化保存其当前状态

(2) Serialize and save its current state

```
rosservice call /write_state "{filename:
'${HOME}/agilex_ws/src/limo_ros/limo_bringup/maps/mymap.pbstream'}"
```

（3）将pbstream转换为pgm和yaml

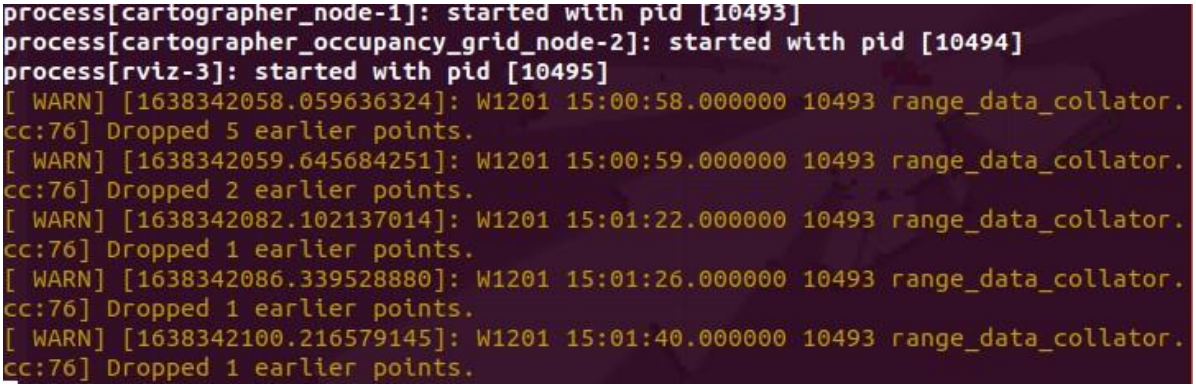
(3) Convert pbstream to pgm and yaml

```
rosrun cartographer_ros cartographer_pbstream_to_ros_map -
map_filestem=${HOME}/agilex_ws/src/limo_ros/limo_bringup/maps/mymap.pbstream -
pbstream_filename=${HOME}/agilex_ws/src/limo_ros/limo_bringup/maps/mymap.pbstream
-resolution=0.05
```

生成对应的pgm和yaml，放于
${HOME}/agilex_ws/src/limo_ros/limo_bringup/maps/mymap.pbstream目录下
Generate the corresponding pgm and yaml, and put them in the directory
${HOME}/agilex_ws/src/limo_ros/limo_bringup/maps/mymap.pbstream

注：在建图过程中，终端中会出现一些警告，这是由于速度过快，数据处理不及时造成的，可以忽略

Note: During the process of mapping, some warnings will appear in the terminal. This is caused by the excessive speed and the delayed data processing, which can be ignored.



六、雷达导航

6. LiDAR Navigation

前面我们使用了两种激光建图方式，接下来利用刚刚构建地图进行导航。

We used two laser mapping methods before, and then we use the map we just built to navigate.

6.1 导航框架

6.1 Navigation framework

导航的关键是机器人定位和路径规划两大部分。针对这两个核心,ROS提供了以下两个功能包。

The key to navigation is robot positioning and path planning. For these two cores, ROS provides the following two packages.

（1） move_base：实现机器人导航中的最优路径规划。

(1) move_base: realize the optimal path planning in robot navigation.

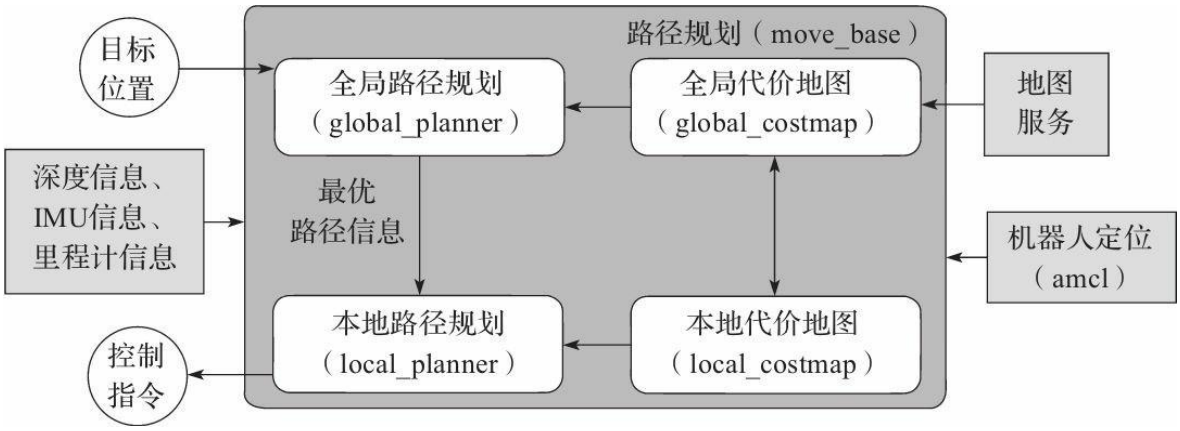
（2） amcl：实现二维地图中的机器人定位。

(2) amcl: realize robot positioning in a two-dimensional map.

在上述的两个功能包的基础上，ROS提供了一套完整的导航框架，

On the basis of the above two packages, ROS provides a complete navigation framework,

原文	译文
目标位置	Goal position
深度信息、IMU信息、里程计信息	Depth information, IMU information, and odometer information
控制指令	Control command
路径规划	Path planning
全局路径规划	Global path planning
全局代价地图	Global cost map
最优路径信息	Optimal path information
本地路径规划	Local path planning
本地代价地图	Local cost map
地图服务	Map service
机器人定位	Robot positioning



机器人只需要发布必要的传感器信息和导航的目标位置,ROS即可完成导航功能。在该框架中,move_base功能包提供导航的主要运行、交互接口。为了保障导航路径的准确性,机器人还要对自己 所处的位置进行精确定位,这部分功能由amcl功能包实现。
The robot only needs to publish the necessary sensor information and navigation goal position, and ROS can complete the navigation function. In this framework, the move_base package provides the main operation and interactive interface of navigation. In order to ensure the accuracy of the navigation path, the robot also needs to accurately locate its own position. This part of the function is implemented by the amcl package.

6.1.1 move_base 功能包

6.1.1Move_base package

move_base是ROS中完成路径规划的功能包,主要由以下两大规划器组成。

move_base is a package for path planning in ROS, which is mainly composed of the following two planners.

全局路径规划(global_planner)。全局路径规划是根据给定的目标位置和全局地图进行总体路径的规划。 在导航中,使用Dijkstra或A*算法进行全局路径的规划,计算出机器人到目标位置的最优路线,作为机器人的 全局路线。

Global path planning (global_planner). Global path planning is to plan the overall path according to a given goal position and global map. In navigation, Dijkstra or A* algorithm is used for global path planning, and the optimal route from the robot to the goal position is calculated as the robot's global path.

本地实时规划(local_planner)。在实际情况中,机器人往往无法严格按照全局路线行驶,所以需要针对地图 信息和机器人附近随时可能出现的障碍物规划机器人每个周期内应该行驶的路线,使之尽量符合全局最优 路径。

Local real-time planning (local_planner). In actual situations, robots often cannot strictly follow the global path, so it is necessary to plan the path that the robot should travel in each cycle according to the map information and obstacles that may appear near the robot at any time, so that it conforms to the global optimal path as much as possible.

6.1.2 amcl 功能包

6.1.2 Amcl package

自主定位即机器人在任意状态下都可以推算出自己在地图中所处的位置。ROS为开发者提供了一种自适应(或kld采样)的蒙特卡罗定位方法(amcl), 这是一种概率定位系统，以2D方式对移动机器人定位。 它实现了自适应（或者KLD-采样）蒙特卡洛定位法，使用粒子滤波跟踪机器人在已知地图中的位姿。

Autonomous positioning means that the robot can calculate its position on the map in any state. ROS provides developers with an adaptive (or kld sampling) Monte Carlo localization (amcl), which is a probabilistic positioning system that locates mobile robots in 2D. It implements an adaptive (or KLD-sampling) Monte Carlo localization, using particle filtering to track the pose of the robot on a known map.

6.1.3 DWA_planner和TEB_planner介绍

6.1.3 Introduction to DWA_planner and TEB_planner

DWA_planner

DWA 的全称为DynamicWindow Approaches，该算法可以搜索躲避和行进的多条路经,综合各评价标准(是否会撞击障碍物,所需要的时间等)选取最优路径,并且计算行驶周期内的线速度和角速度,避免与动态出现的障碍物发生碰撞。
The full name of DWA is Dynamic Window Approaches. The algorithm can search for multiple paths to avoid and travel, select the optimal path based on various evaluation

criteria (whether it will hit an obstacle, the time required, etc.), and calculate the linear velocity and angular velocity during the driving cycle to avoid collisions with dynamic obstacles.

TEB_planner

“TEB”全称Time Elastic Band（时间弹性带）Local Planner，该方法针对全局路径规划器生成的初始轨 迹进行后续修正(modification)，从而优化机器人的运动轨迹，属于局部路径规划。在轨迹优化过程中， 该算法拥有多种优化目标，包括但不限于：整体路径长度、轨迹运行时间、与障碍物的距离、通过中间 路径点以及机器人动力学、运动学以及几何约束的符合性。“TEB方法”明确考虑了运动状态下时空方面的 动态约束，如机器人的速度和加速度是有限制的。

The full name of "TEB" is Time Elastic Band Local Planner. This method performs subsequent modifications to the initial trajectory generated by the global path planner to optimize the robot's motion trajectory and belongs to local path planning. In the process of trajectory optimization, the algorithm has a variety of optimization goals, including but not limited to: overall path length, trajectory running time, distance to obstacles, passing intermediate path points, and compliance with robot dynamics, kinematics, and geometric constraints. The “TEB method” explicitly considers the dynamic constraints of time and space in the state of motion, for example, the velocity and acceleration of the robot are limited.

6.2 limo导航功能

6.2 Limo’s navigation function

注：四轮差速模式和全向轮模式、履带模式下，导航运行的文件一样

Note: In the four-wheel differential mode, the omnidirectional wheel mode and the track mode, the file run for the navigation is the same

注：在运行命令之前，请确保其他终端中的程序已经终止，终止命令为：Ctrl+c

Note: Before running the command, please make sure that the programs in other terminals have been terminated. The termination command is: Ctrl+c

（1）首先启动雷达，在终端中输入命令：

(1) First launch the LiDAR and enter the command in the terminal:

```
roslaunch limo_bringup limo_start.launch pub_odom_tf:=false
```

（2）启动导航功能，在终端中输入命令：

(2) Launch the navigation function and enter the command in the terminal:

```
roslaunch limo_bringup limo_navigation_diff.launch
```

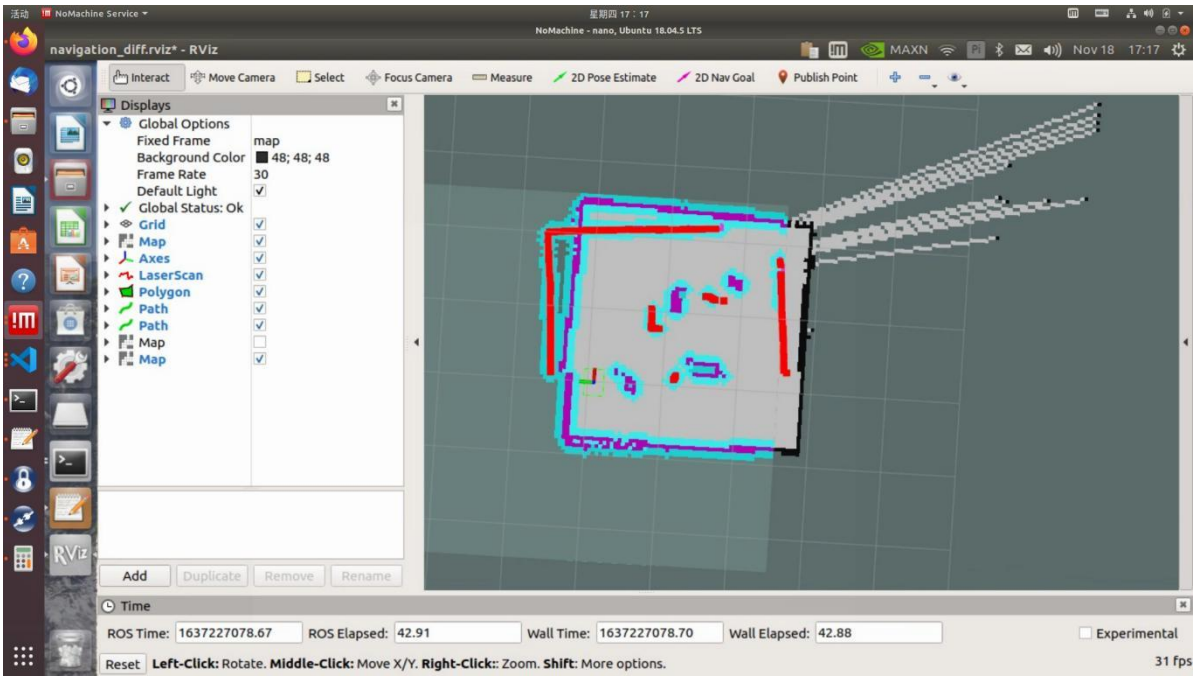
注：如果是阿克曼运动模式，请运行

Note: If it is Ackermann motion mode, please run

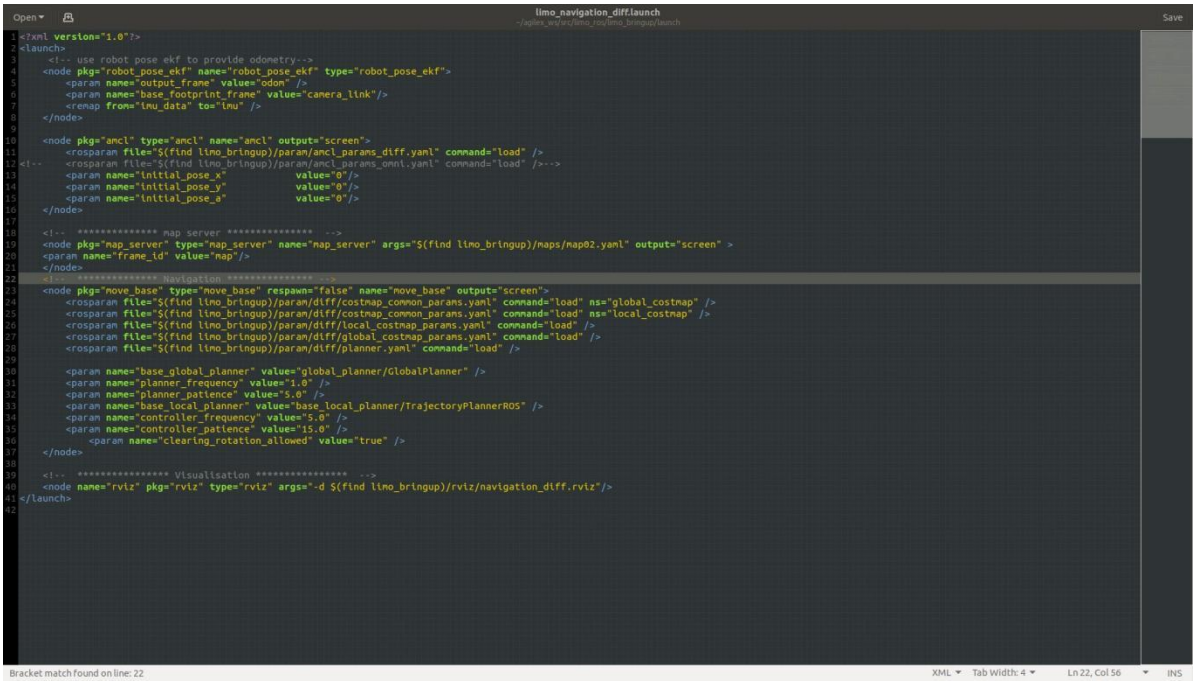
```
roslaunch limo_bringup limo_navigation_ackeman.launch
```

启动成功之后会打开rviz界面，如图

After launching successfully, the rviz interface will be opened, as shown in the figure



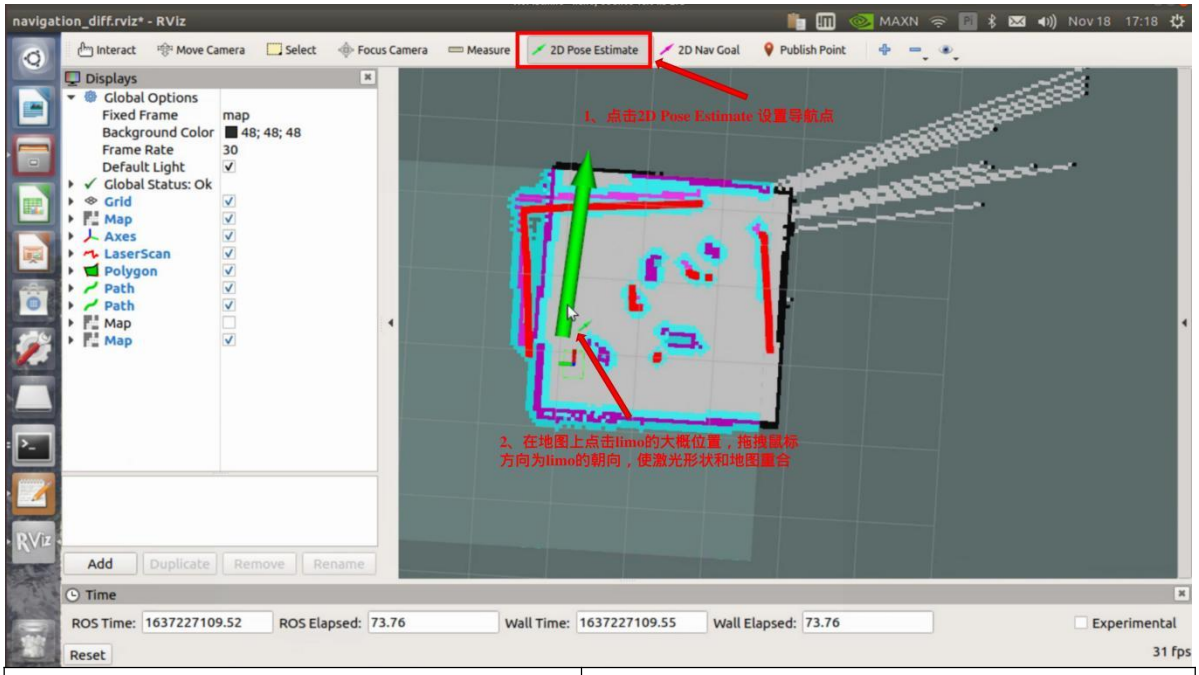
注：如需自定义打开的地图，请打开limo_navigation_diff.launch 文件修改参数，请把map02修改为需要更换的地图名称。



Note: If you need to customize the opened map, please open the limo_navigation_diff.launch file to modify the parameters; please modify map02 to the name of the map that needs to be replaced.

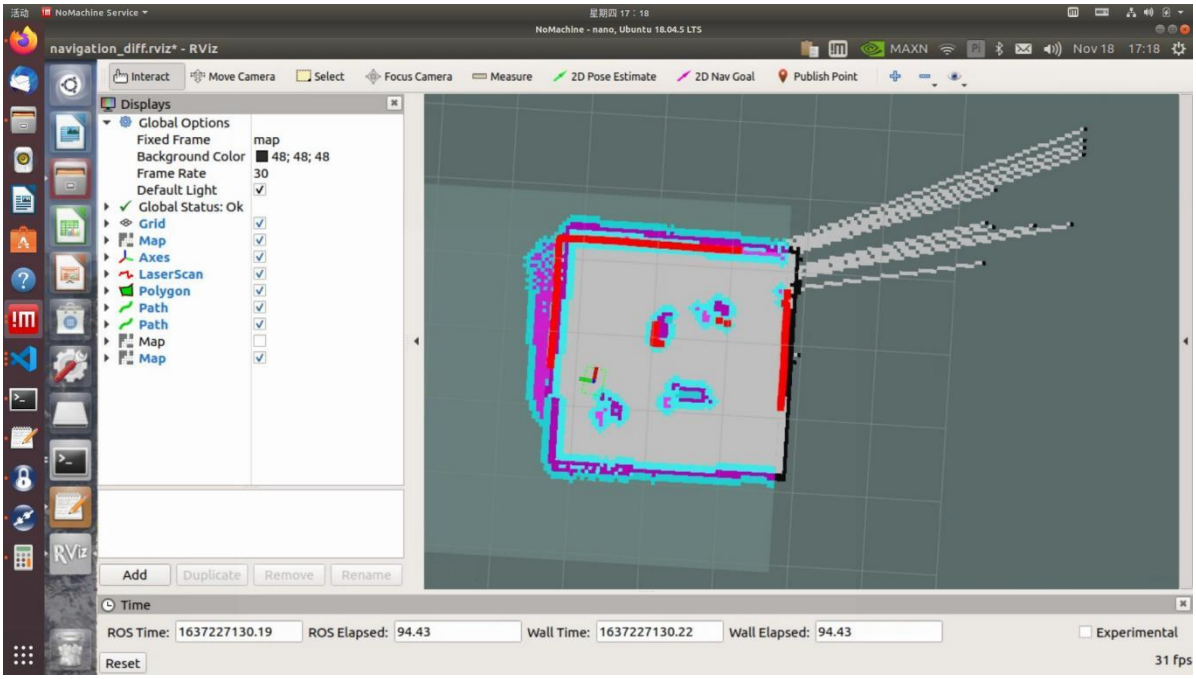
（3）开启导航之后，会发现激光扫描出来的形状和地图没有重合，需要我们手动校正，在rviz中显示的地图上矫正底盘在场景中实际的位置，通过rviz中的工具，发布一个大概的位置，给limo一个大致的位置，然后通过手柄遥控limo旋转，让其自动校正，当激光形状和地图中的场景形状重叠的时候，校正完成。操作步骤如图：

(3) After launching the navigation, you will find that the shape scanned by the laser does not overlap with the map, and we need to manually correct it. Correct the actual position of the chassis in the scene on the map displayed in rviz. Use the tools in rviz to publish an approximate position, give the limo a rough position, and then use the handle to remotely rotate the limo to make it automatically correct. When the laser shape overlaps the scene shape in the map, the correction is complete. The operation steps are as follows:



原文	译文
1. 点击2D Pose Estimate 设置导航点	1. Click 2D Pose Estimate to set the navigation point
2. 在地图上点击limo的大概位置，拖拽鼠标方向为limo的朝向，使激光形状和地图重合	2. Click the approximate position of limo on the map, and drag the mouse to the direction of limo to make the laser shape coincide with the map

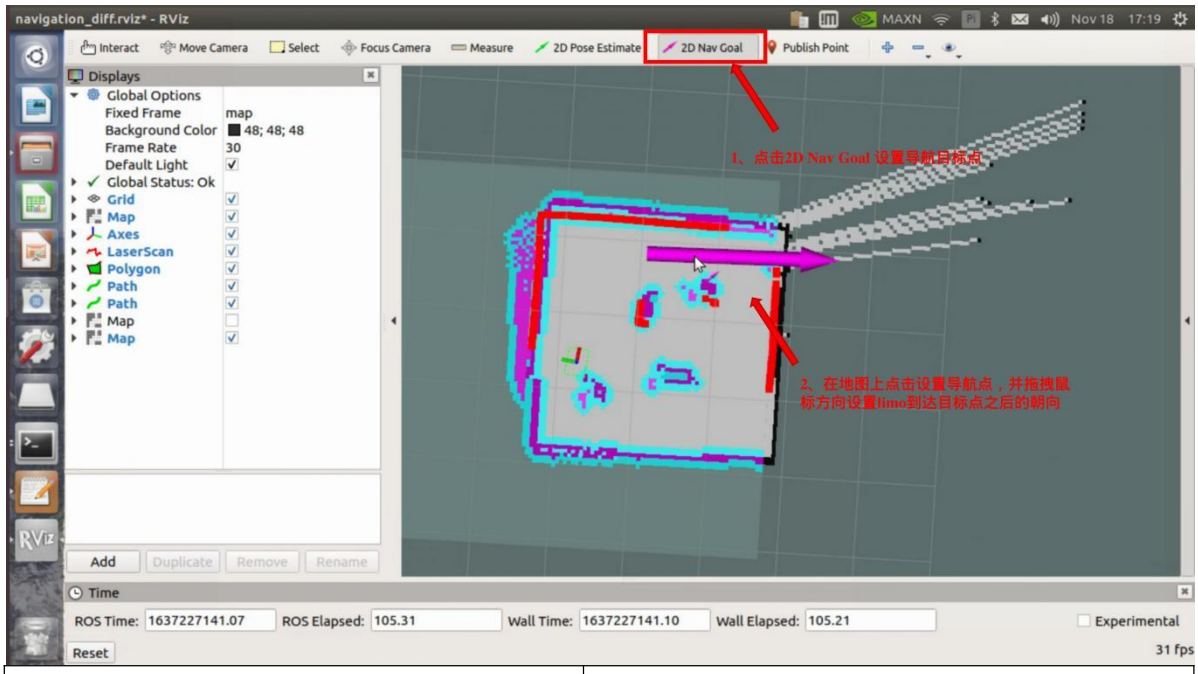
校正完成后



After the correction is complete

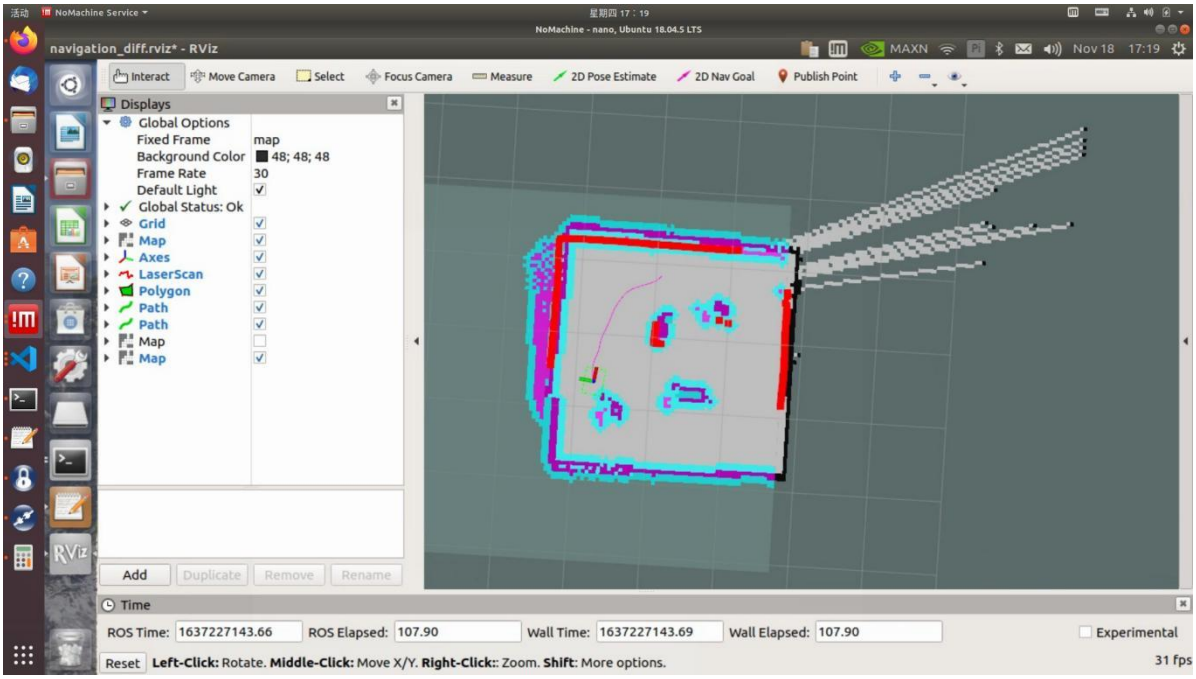
（4）通过2D Nav Goal 设置导航目标点。

(4) Set the navigation goal point through 2D Nav Goal.



原文	译文
1. 点击2D Nav Goal 设置导航目标点	1. Click 2D Nav Goal to set the navigation goal point
2. 在地图上点击设置导航点，并拖拽鼠标方向设置limo到达目标点之后的朝向	2. Click on the map to set the navigation point, and drag the mouse to set the direction of limo after reaching the goal point

地图中将会生成一条紫色的路径，手柄切换至指令模式，limo将自动导航到目标点



A purple path will be generated in the map, the handle will switch to command mode, and limo will automatically navigate to the goal point

6.3 limo路径巡检

6.3 Limo path inspection

（1）首先启动雷达，开启一个新的终端，在终端中输入命令：

(1) First launch the LiDAR, open a new terminal, and enter the command in the terminal:

```
roslaunch limo_bringup limo_start.launch pub_odom_tf:=false
```

（2）启动导航功能，开启一个新的终端，在终端中输入命令：

(2) Launch the navigation function, open a new terminal, and enter the command in the terminal:

```
roslaunch limo_bringup limo_navigation_diff.launch
```

注：如果是阿克曼运动模式，请运行

Note: If it is Ackermann motion mode, please run

```
roslaunch limo_bringup limo_navigation_ackeman.launch
```

（3）启动路径记录功能，开启一个新的终端，在终端中输入命令：

(3) Launch the path recording function, open a new terminal, and enter the command in the terminal:

```
roslaunch agilex_pure_pursuit record_path.launch
```

路径记录结束之后终止路径记录程序，在终端中输入命令为：Ctrl+c

After the path recording is over, terminate the path recording program, and enter the command in the terminal: Ctrl+c

（4）启动路径巡检功能，开启一个新的终端，在终端中输入命令：

(4) Launch the path inspection function, open a new terminal, and enter the command in the terminal:

```
roslaunch agilex_pure_pursuit pure_pursuit.launch
```

注：把手柄调至指令模式

Note: adjust the handle to command mode

七、深度相机+雷达建图

7. Depth Camera + LiDAR Mapping

limo拥有两个版本，一个版本搭配RealSense D435，另一个版本搭配ORBBEC®Dabai，两个深度相机都可以实现视觉+雷达的建图导航功能。下面将介绍两个款深度相机的使用方法。

Limo has two versions, one with RealSense D435 and the other with ORBBEC®Dabai. Both depth cameras can realize the mapping and navigation function of vision + LiDAR. The following will introduce how to use two depth cameras.

7.1 ORBBEC®Dabai的介绍与使用

7.1 Introduction and use of ORBBEC®Dabai

ORBBEC®Dabai 是基于双目结构光 3D 成像技术的深度相机，主要包括左红外相机(IR camera1)、右红外相机(IR camera2)、一个红外投影仪(IR projector)以及深度计算处理器(depth processor)。红外投影仪用于向目标场景(Scene)投射结构光图案(散斑图案)，左红外相机以及或红外相机分别采集目标的左红 外结构光图像以及右红外结构光图像，深度计算处理器接收左红外结构光图像、右红外结构光图像后执 行深度计算算法并输出目标场景的深度图像。

ORBBEC®Dabai is a depth camera based on binocular structured light 3D imaging technology. It mainly includes a left infrared camera (IR camera1), a right infrared camera (IR camera2), an IR projector and a depth processor. The IR projector is used to project the structured light pattern (speckle pattern) to the goal scene, the left infrared camera and the right infrared camera respectively collect the left infrared structured light image and the right infrared structured light image of the goal, and the depth processor executes the depth calculation algorithm and outputs the depth image of the goal scene after receiving the left infrared structured light image and the right infrared structured light image.

参数名称 Parameter name	参数指标 Parameter index
左、右红外相机成像中心之间的距离 The distance between the imaging centers of the left and right infrared cameras	40mm
深度距离 Depth distance	0.3-3m
功耗 Power consumption	整机工作平均功耗<2W, 激光开启瞬间峰值 <5W(持续时间 3ms), 待机功耗典型值为<0.7W The average power consumption of the whole machine is <2W, The peak value at the moment the laser is turned on <5W (duration: 3ms), Typical standby power consumption is <0.7W
深度图分辨率 Depth map resolution	640*400@30FPS* *320*200@30FPS
彩色图分辨率 Color map resolution	1920X1080@30FPS 1280X720@30FPS 640X480@30FPS
精度 Accuracy	6mm@1m(81%FOV区域参与精度计算*) 6mm@1m (81% FOV area participates in accuracy calculation*)
深度 FOV Depth FOV	H 67.9° V 45.3°
彩色 FOV Color FOV	H 71° V43.7° @1920X1080
延迟 Delay	30-45ms
数据传输 Data transmission	USB2.0 或以上 USB2.0 or above
支持操作系统 Support operating system	Android / Linux / Windows7/10
供电方式 Power supply mode	USB
工作温度 Operating temperature	10°C ~ 40°C
适用场景 Applicable scene	室内 / 室外(具体以应用场景和相关算法要求为准) Indoor / outdoor (specifically subject to application scenes and related algorithm requirements)
防尘防水 Dustproof and waterproof	基础防尘 Foundation dustproof

安全性 Safety	Class1 激光 Class1 laser
尺寸(毫米) Dimensions (mm)	长59.6X宽17.4X厚11.1mm Length 59.6 X width 17.4 X thickness 11.1mm

了解ORBBEC®Dabai的基本参数之后，开始实践操作

After knowing the basic parameters of ORBBEC®Dabai, start to practice

注：在运行命令之前，请确保其他终端中的程序已经终止，终止命令为：Ctrl+c

Note: Before running the command, please make sure that the programs in other terminals have been terminated. The

```
roslaunch astra_camera dabai_u3.launch
```

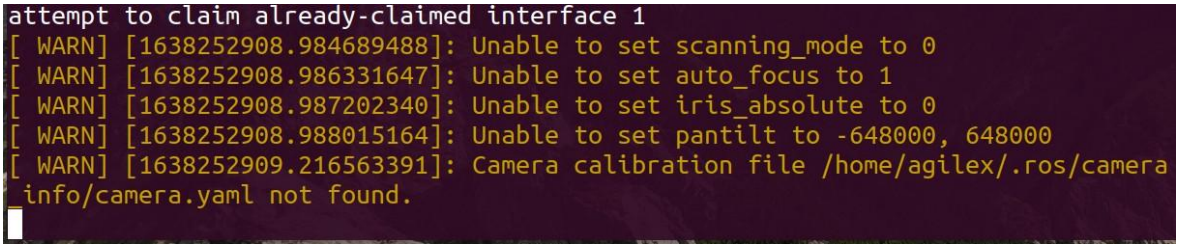
termination command is: Ctrl+c

首先启动ORBBEC®Dabai摄像头，运行下面的命令：

First start the ORBBEC®Dabai camera and run the following command:

运行过程中会出现以下警告，这是由于驱动中的一些参数摄像头不支持，可以忽略。

The following warnings will appear during running. This is because some parameters in the driver are not supported by the camera and can be ignored.



7.2 realsense的介绍与使用

7.2 Introduction and use of realsense

双目视觉传感器，在机器人视觉测量、视觉导航等机器人行业方向中均有大范围的应用场景和需求，目前我们甄选了了在科研教育行业常见的视觉传感器。英特尔实感深度摄像头 D435 配备全局图像快门和宽视野，能够有效地捕获和串流移动物体的深度数据，从而为移动原型提供高度准确的深度感知。

Binocular vision sensors have a wide range of application scenes and requirements in the robot vision measurement, visual navigation and other robotics industries. At present, we have selected the common vision sensors in the scientific research and education industry. The Intel RealSense Depth Camera D435 is equipped with a global image shutter and a wide field of view, which can effectively capture and stream the depth data of moving objects, thereby providing highly accurate depth perception for mobile prototypes.

	型号 **Modal**	**Intel Realsense D435**
基本特征 Basic features	应用场景 Application scenes	户外/室内 Outdoor/indoor
测量距离 Measuring distance	约10米 About 10m	
深度快门类型 Depth shutter type	全局快门/3um X 3um Global shutter/3um X 3um	
是否支持IMU Whether the IMU is supported	否 No	
深度相机 Depth camera	深度技术 In-depth technology	有源红外 Active infrared
FOV	86° x 57°（±3°）	
最小深度距离 Minimum depth distance	0.105m	
深度分辨率 Depth resolution	1280 x 720	
最大测量距离 Maximum measuring distance	约10米 About 10m	
深度帧率 Depth frame rate	90 fps	
RGB	分辨率 Resolution	1280 x 800
FOV	69.4° × 42.5°（±3°）	
帧率 Frame rate	30fps	
其他信息	尺寸	90mm x 25mm x 25mm

Other information	Dimensions	
接口类型 Interface type	USB-C 3.1	

了解realsense的基本参数之后，开始实践操作

After knowing the basic parameters of realsense, start to practice

注：在运行命令之前，请确保其他终端中的程序已经终止，终止命令为：Ctrl+c

Note: Before running the command, please make sure that the programs in other terminals have been terminated. The

```
roslaunch realsense2_camera rs_camera.launch
```

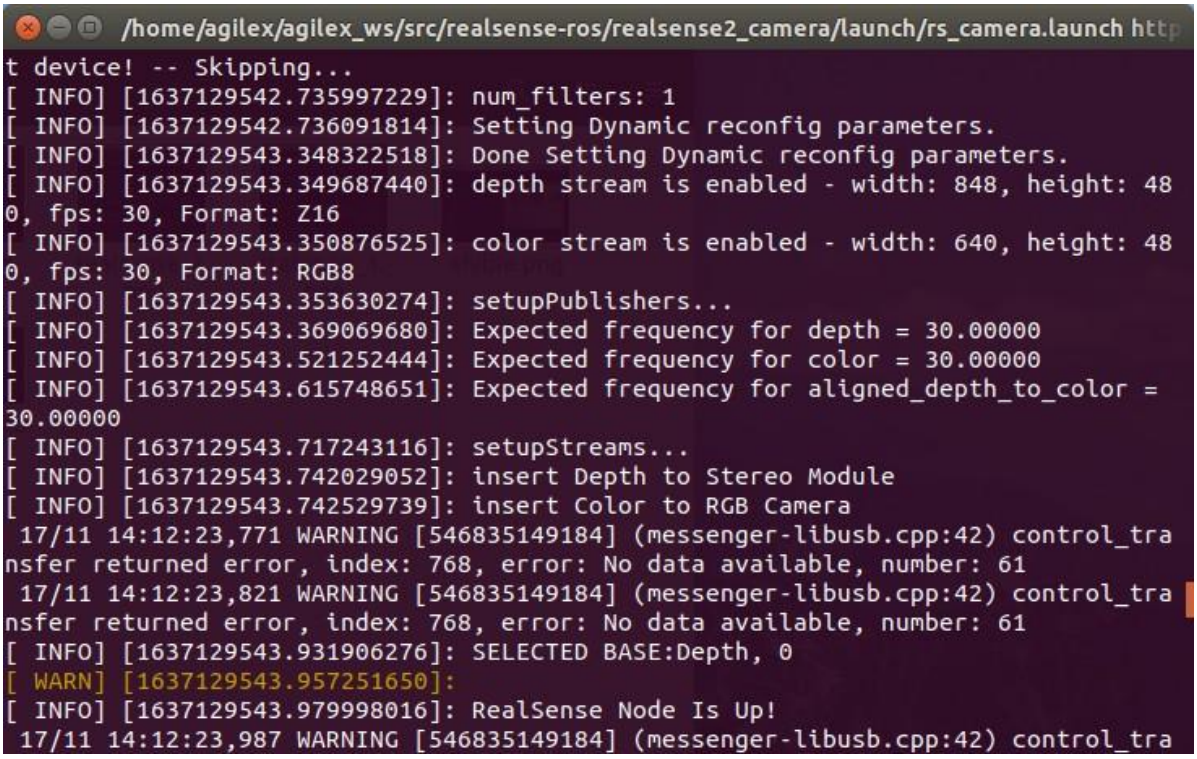
termination command is: Ctrl+c

首先启动realsense摄像头，运行下面的命令：

First start the realsense camera and run the following command:

当终端中出现下面的日志信息，摄像头就启动成功了。

When the following log information appears in the terminal, the camera is launched successfully.



7.3 查看深度相机信息

7.3 View depth camera’s information

成功打开深度相机之后，接下来启动rviz，查看深度相机所拍摄到的图像和采集的深度信息。

After successfully opening the depth camera, launch rviz to view the images captured

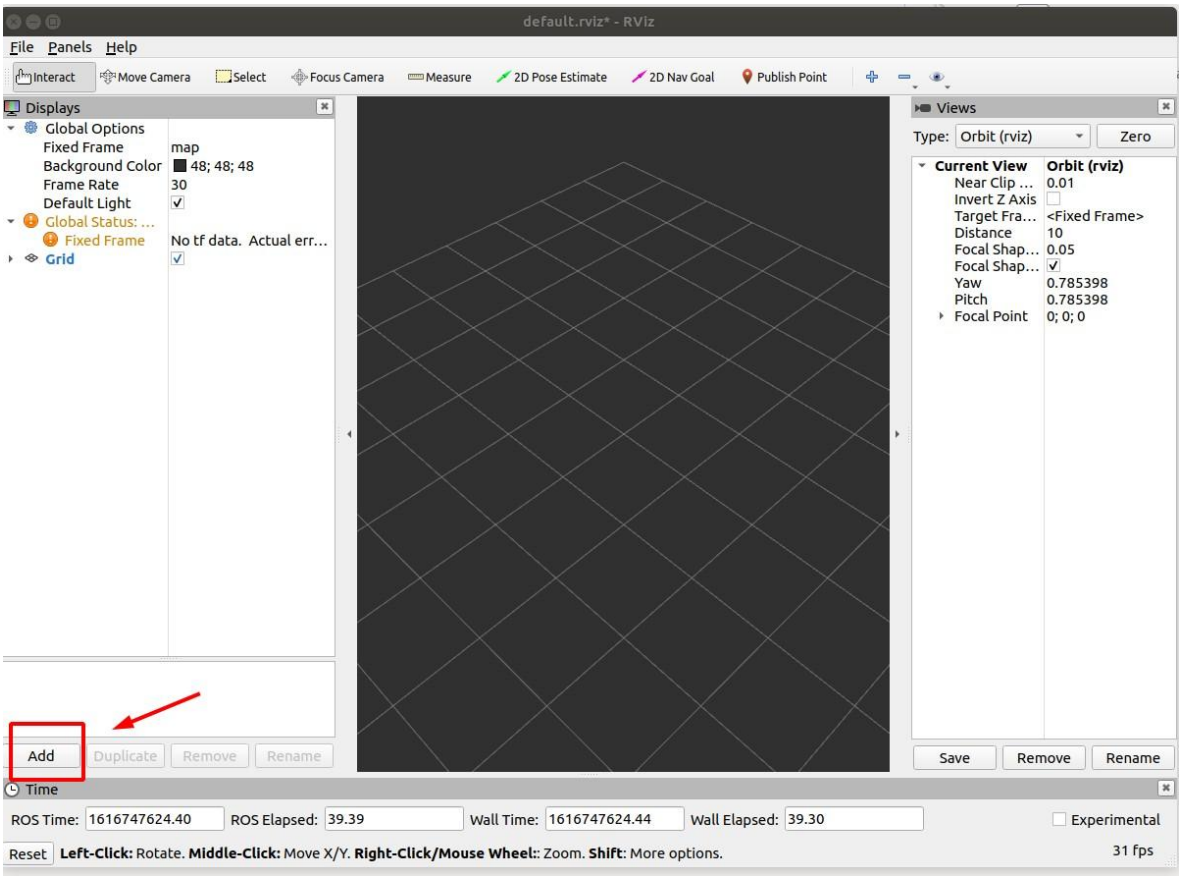
```
rviz
```

by the depth camera and the depth information collected.

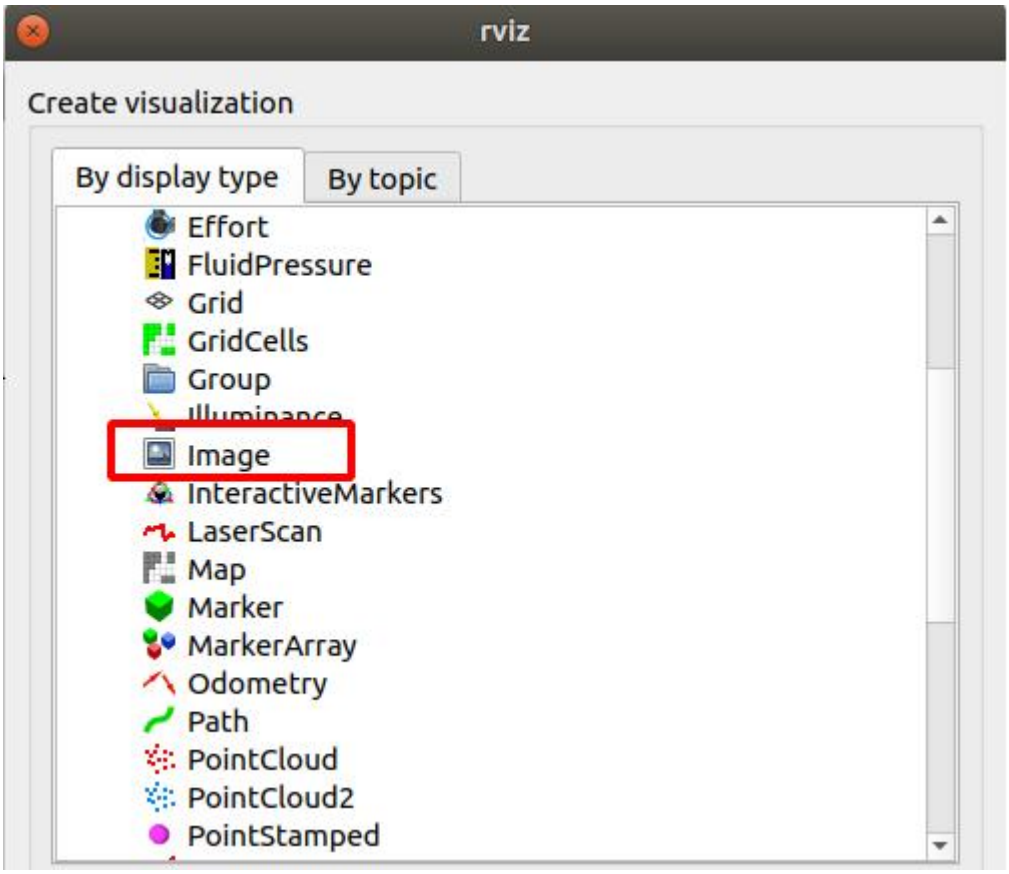
开启一个新终端，输入命令：

Open a new terminal and enter the command:

然后添加Image组件就能看到摄像头所拍摄的画面，操作步骤如下。

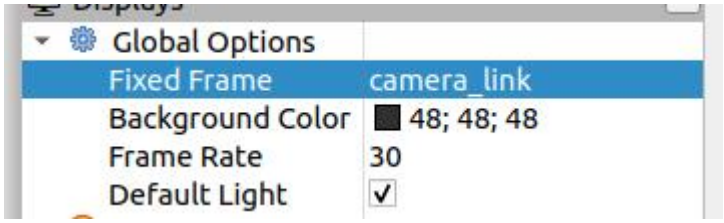


Then add the Image component to see the picture taken by the camera. The operation steps are as follows.



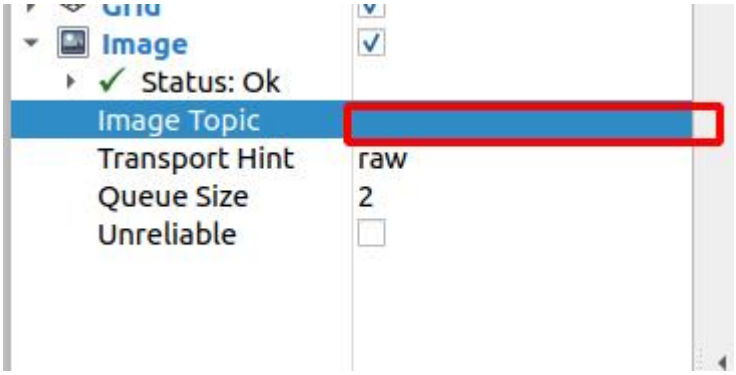
Fixed frame选择camera_link

Select camera_link in fixed frame



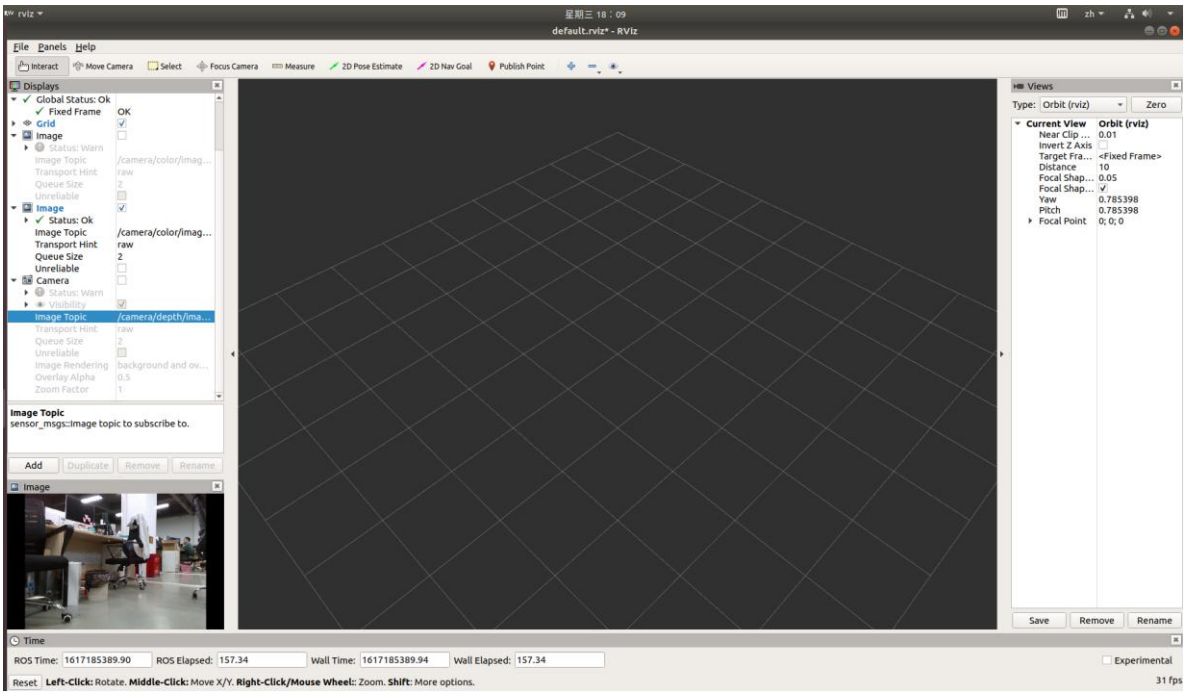
image组件填入对应的话题获取rgb图片

Fill in the corresponding topic in the image component to get the rgb picture

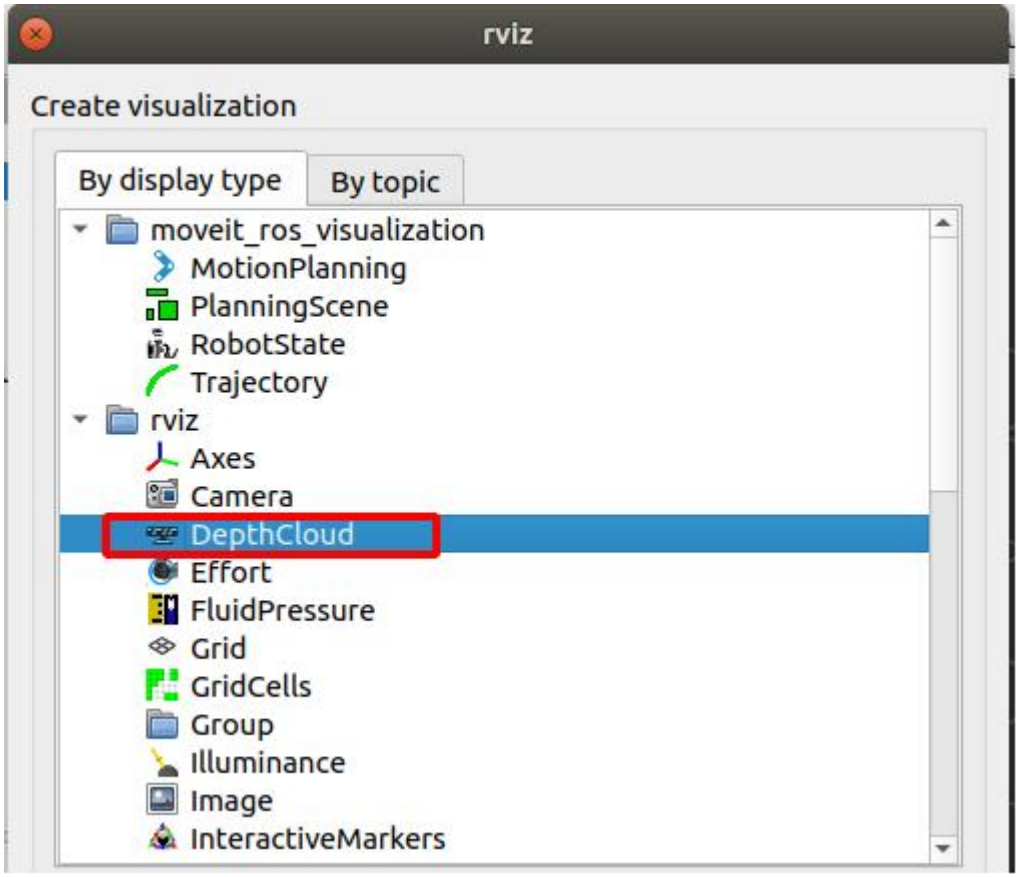


完成上述操作之后就能在Image窗口看到摄像头拍摄的画面了。

After completing the above operations, you can see the picture taken by the camera in the Image window.



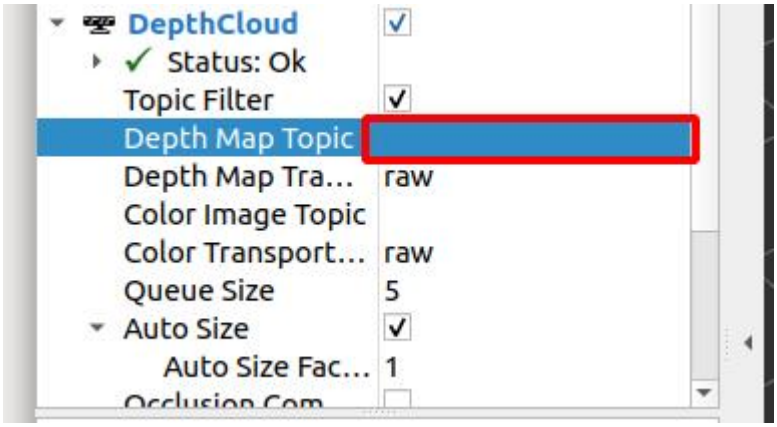
如果想要查看点云数据，点击add添加DepthCloud组件



If you want to view the point cloud data, click “add” to add the DepthCloud component

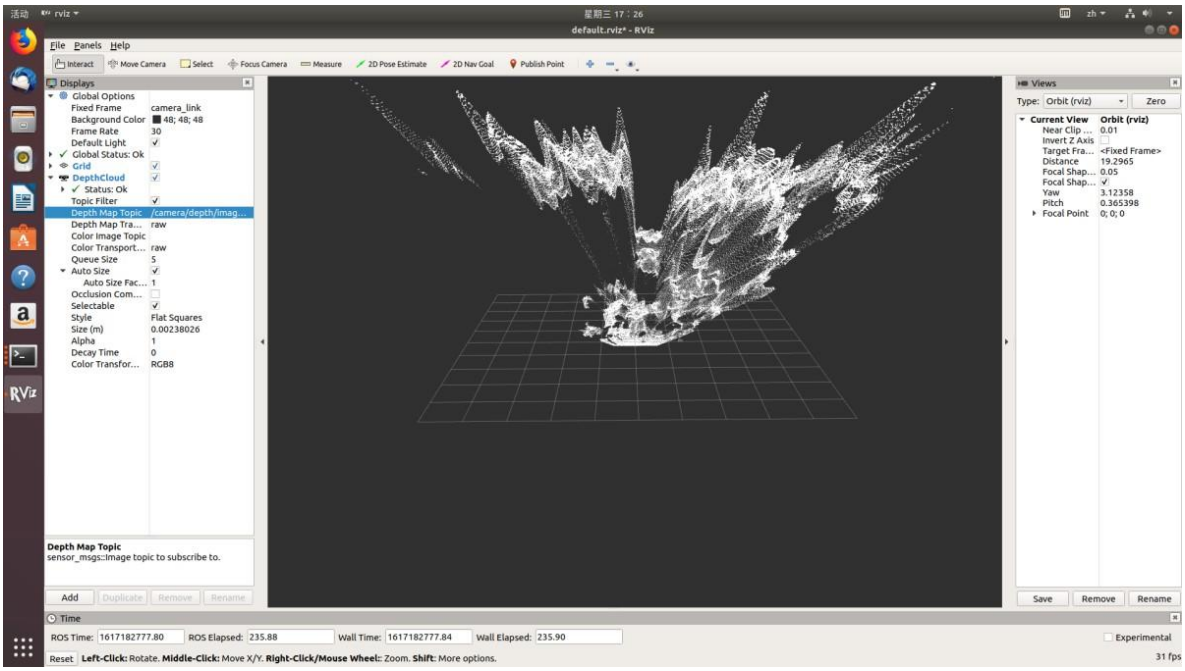
fixed frame选择camera_link, DepthCloud组件选择对应的话题

Select camera_link in fixed frame and select the corresponding topic in DepthCloud component



显示深度图

Show depth map



7.2 rtabmap算法介绍

7.2 Introduction to rtabmap algorithm

rtabmap算法提供一个与时间和尺度无关的基于外观的定位与构图解决方案。针对解决大型环境中的在 线闭环检测问题。方案的思想在于为了满足实时性的一些限制，闭环检测是仅仅利用有限数量的一些定 位点，同时在需要的时候又能够访问到整个地图的定位点。

The rtabmap algorithm provides an appearance-based positioning and mapping solution independent of time and scale. It's aimed at solving the problem of online closed-loop detection in large-scale environments. The idea of the solution is to meet some real-time limitations. Closed-loop detection uses only a limited number of positioning points, while being able to access the positioning points of the entire map when needed.

7.3 rtabmap算法建图

7.3 Rtabmap algorithm mapping

注：在运行命令之前，请确保其他终端中的程序已经终止，终止命令为：Ctrl+c

Note: Before running the command, please make sure that the programs in other terminals have been terminated. The termination command is: Ctrl+c

注：建图过程中limo的速度尽量慢点，速度太快会影响建图的效果

Note: The speed of limo should be as slow as possible in the process of mapping. If the speed is too fast, the effect of mapping will be affected

（1）启动雷达，在终端中输入命令：

(1) First launch the LiDAR and enter the command in the terminal:

```
roslaunch limo_bringup limo_start.launch pub_odom_tf:=true
```

（2）启动realsense，在终端中输入命令：

(2) Launch realsense and enter the command in the terminal:

```
roslaunch realsense2_camera rs_camera.launch align_depth:=true
```

注：如果limo搭载ORBBEC®Dabai，请输入命令：

Note: If limo is equipped with ORBBEC®Dabai, please enter the command:

```
roslaunch astra_camera dabai_u3.launch
```

（3）启动rtabmap算法的建图模式，在终端中输入命令：

(3) Launch the mapping mode of the rtabmap algorithm, and enter the command in the terminal:

```
roslaunch limo_bringup limo_rtabmap_realsense.launch
```

注：如果limo搭载ORBBEC®Dabai，请输入命令：

Note: If limo is equipped with ORBBEC®Dabai, please enter the command:

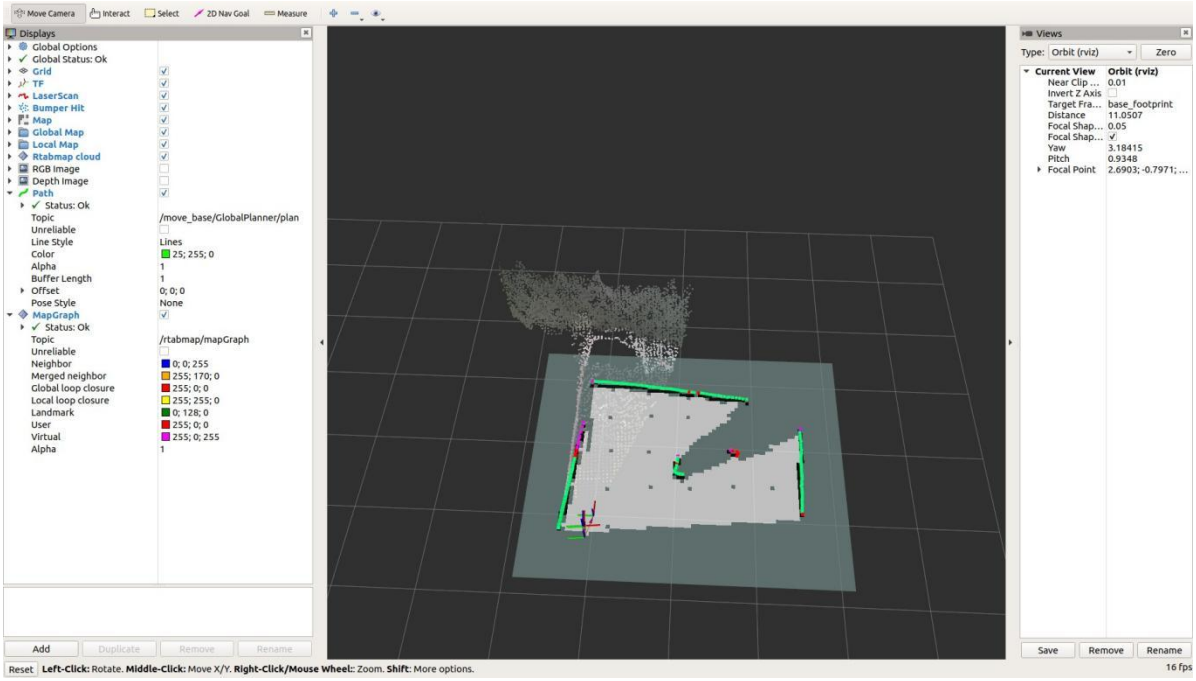
```
roslaunch limo_bringup limo_rtabmap_orbbec.launch
```

（4）启动rviz查看建图效果，在终端中输入命令：

(4) Launch rviz to view the mapping effect, and enter the command in the terminal:

```
roslaunch limo_bringup rtabmap_rviz.launch
```

当rviz界面中出现如图 的画面时，rtabmap算法建图模式成功启动



When the picture shown in the figure appears in the rviz interface, the rtabmap algorithm mapping mode is successfully launched

当构建完地图之后，可以直接终止程序，构建的地图将自动保存在主目录下的.ros文件中，文件名称为rtabmap.db。.ros文件夹为隐藏文件夹，需要通过Ctrl+h指令显示出来。

After the map is built, you can directly terminate the program, and the built map will be automatically saved in the .ros file in the main directory, and the file name is rtabmap.db. The .ros folder is a hidden folder and needs to be displayed through the Ctrl+h command.

7.4 rtabmap算法导航

7.4 Rtabmap algorithm navigation

注：在运行命令之前，请确保其他终端中的程序已经终止，终止命令为：Ctrl+c

Note: Before running the command, please make sure that the programs in other terminals have been terminated. The termination command is: Ctrl+c

（1）启动雷达，在终端中输入命令：

(1) First launch the LiDAR and enter the command in the terminal:

```
roslaunch limo_bringup limo_start.launch pub_odom_tf:=true
```

（2）启动realsense，在终端中输入命令：

(2) Launch realsense and enter the command in the terminal:

```
roslaunch realsense2_camera rs_camera.launch align_depth:=true
```

注：如果limo搭载ORBBEC®Dabai，请输入命令：

Note: If limo is equipped with ORBBEC®Dabai, please enter the command:

```
 roslaunch astra_camera dabai_u3.launch
```

（3）启动rtabmap算法的定位模式，在终端中输入命令：

(3) Launch the positioning mode of the rtabmap algorithm, and enter the command in the terminal:

```
roslaunch limo_bringup limo_rtabmap.launch localization:=true
```

注：如果limo搭载ORBBEC®Dabai，请输入命令：

Note: If limo is equipped with ORBBEC®Dabai, please enter the command:

```
roslaunch limo_bringup limo_rtabmap_orbbec.launch localization:=true
```

（4）启动move_base，在终端中输入命令：

(4) Launch move_base and enter the command in the terminal:

```
roslaunch limo_bringup limo_navigation_rtabmap.launch
```

注：如果是阿克曼运动模式，请运行

Note: If it is Ackermann motion mode, please run

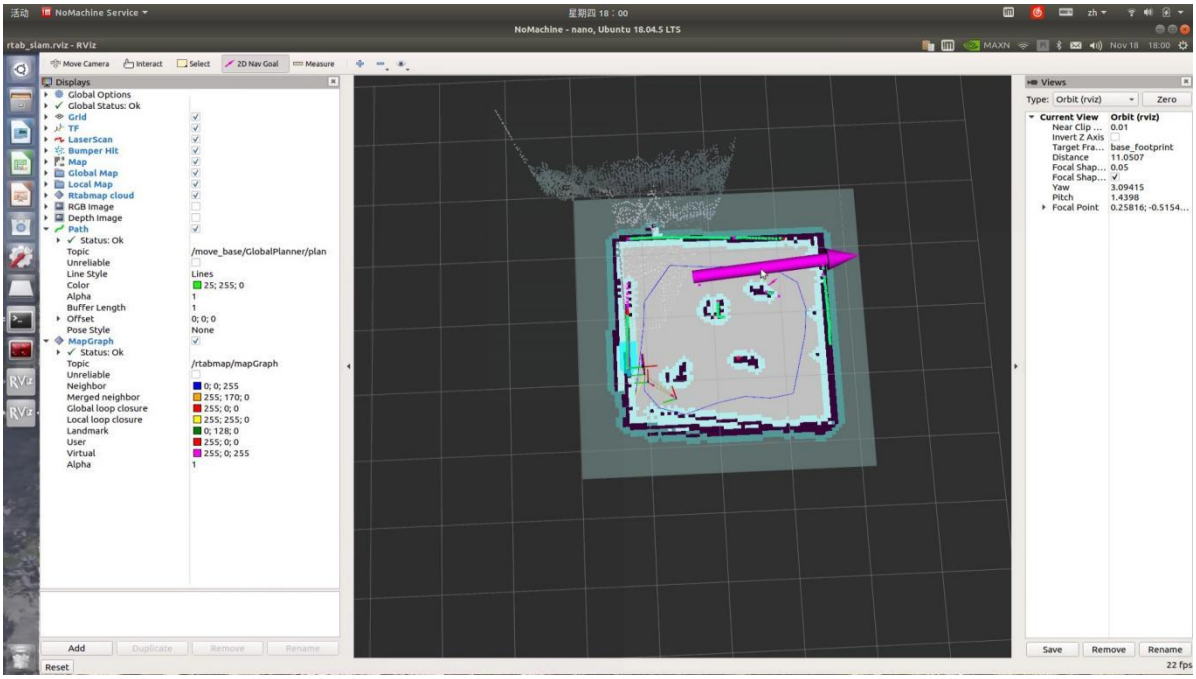
```
roslaunch limo_bringup limo_navigation_rtabmap_ackerman.launch
```

（5）启动rviz查看建图效果，在终端中输入命令：

(5) Launch rviz to view the mapping effect, and enter the command in the terminal:

```
 roslaunch limo_bringup rtabmap_rviz.launch
```

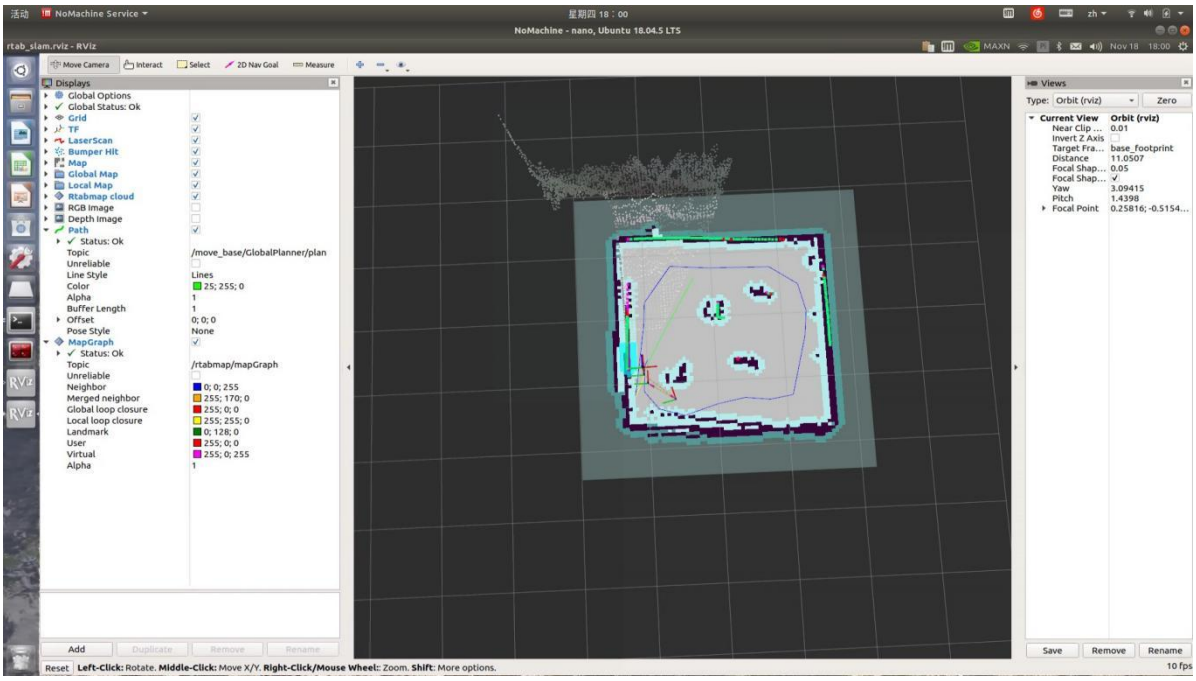
（6）因为我们用到视觉定位，所以在采用rtabmap导航的时候不需要校正，可以直接开始设置目标点进 行导航，操作步骤如图 。



(6) Since we use visual positioning, no correction is needed when navigating with rtabmap. You can directly start to set the goal point for navigation. The operation steps are shown in the figure.

地图中将会生成一条绿色的路径，手柄切换至指令模式，limo将自动导航到目标点

A green path will be generated in the map, the handle will switch to command mode, and limo will automatically navigate to the goal point



八、 视觉模块

8. Vision Module

8.1 识别文字

8.1 Recognize text

8.1.1 功能简介

8.1.1 Function Introduction

通过获取相机的rgb图像，对图像进行灰度化、二值化处理，在利用pytesseract文字识别库对图像的英 文字母或数字进行识别，把识别的结果发布到detect_word_reslut话题中。

Obtain the rgb image of the camera, perform grayscale and binarization processing on the image, and then use the pytesseract text recognition library to recognize the English letters or numbers of the image, and post the recognition result to the detect_word_reslut topic.

8.1.2 运行功能

8.1.2 Running function

注：在运行命令之前，请确保其他终端中的程序已经终止，终止命令为：Ctrl+c

Note: Before running the command, please make sure that the programs in other terminals have been terminated. The termination command is: Ctrl+c

```
roscore
```

以Realsense为例子，启动节点管理，在终端中输入命令：

Take Realsense as an example, launch node management, and enter the command in the terminal:

启动文字识别功能，在终端中输入命令：

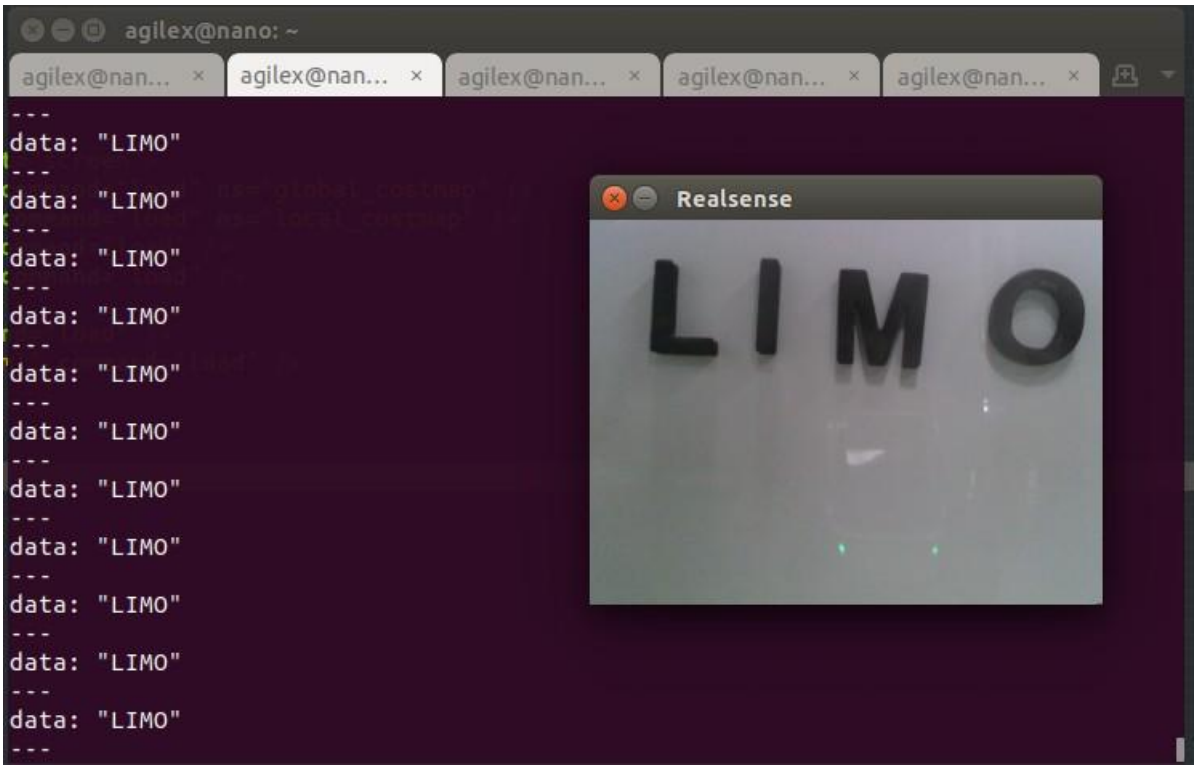
Start the text recognition function and enter the command in the terminal:

```
rosrun vision detect_node.py
```

执行rostopic echo /detect_word_reslut可以查看识别出来的结果

Execute rostopic echo /detect_word_reslut to view the recognized results

```
rostopic echo /detect_word_reslut
```



8.2 识别红绿灯

8.2 Identifying traffic lights

8.2.1 功能简介

8.2.1 Function introduction

通过darknet_ros进行红绿灯目标检测之后，要对红绿灯进行灯色识别并进行在三维空间中的定位，生成 物体相对与摄像机的位置关系。该方式只能实现对红绿灯的识别和定位，无法获得红绿灯姿态。需要使 用深度摄像机，其识别距离取决于深度摄像头范围。

After the traffic lights are detected through darknet_ros, the traffic lights must be identified and positioned in the three-dimensional space to generate the relative position of the object to the camera. This method can only realize the identification and positioning of the traffic lights, and cannot obtain the status of the traffic lights. Depth camera is needed, and its recognition distance depends on the depth camera’ s range.

8.2.2 运行功能

8.2.2 Running function

注：在运行命令之前，请确保其他终端中的程序已经终止，终止命令为：Ctrl+c

Note: Before running the command, please make sure that the programs in other terminals have been terminated. The termination command is: Ctrl+c

```
roslaunch realsense2_camera rs_camera.launch
```

以Realsense为例子，启动realsense深度相机，在终端中输入命令：

Take Realsense as an example, launch the realsense depth camera and enter the command in the terminal:

注：如果limo搭载ORBBEC®Dabai，请输入命令：

Note: If limo is equipped with ORBBEC®Dabai, please enter the command:

```
roslaunch astra_camera dabai_u3.launch
```

启动yolo_v3，在终端中输入命令：

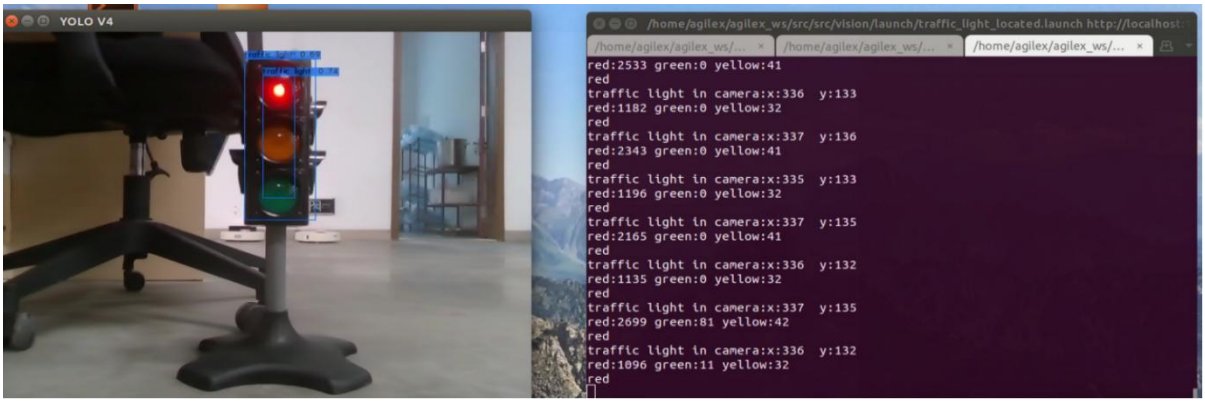
Launch yolo_v3 and enter the command in the terminal:

```
roslaunch darknet_ros  yolo_v3_tiny.launch
```

启动红绿灯识别功能

Launch the traffic light recognition function

```
roslaunch vision traffic_light_located.launch
```



九、语音模块

9. Voice module

9.1 语音转文字

9.1 Speech to text

9.1.1 功能简介

9.1.1 Function introduction

通过nano的外接声卡把声音录制成wav文件，用可以离线识别的语音库pocketsphinx实现语音识别。该 功能对英文的识别率较高，中文的识别率较差。

The voice is recorded into a wav file through the external sound card of the nano, and the voice recognition is realized with the voice library pocketsphinx, which can be recognized offline. This function has a high recognition rate for English, but a poor recognition rate for Chinese.

9.1.2 运行功能

9.1.2 Running function

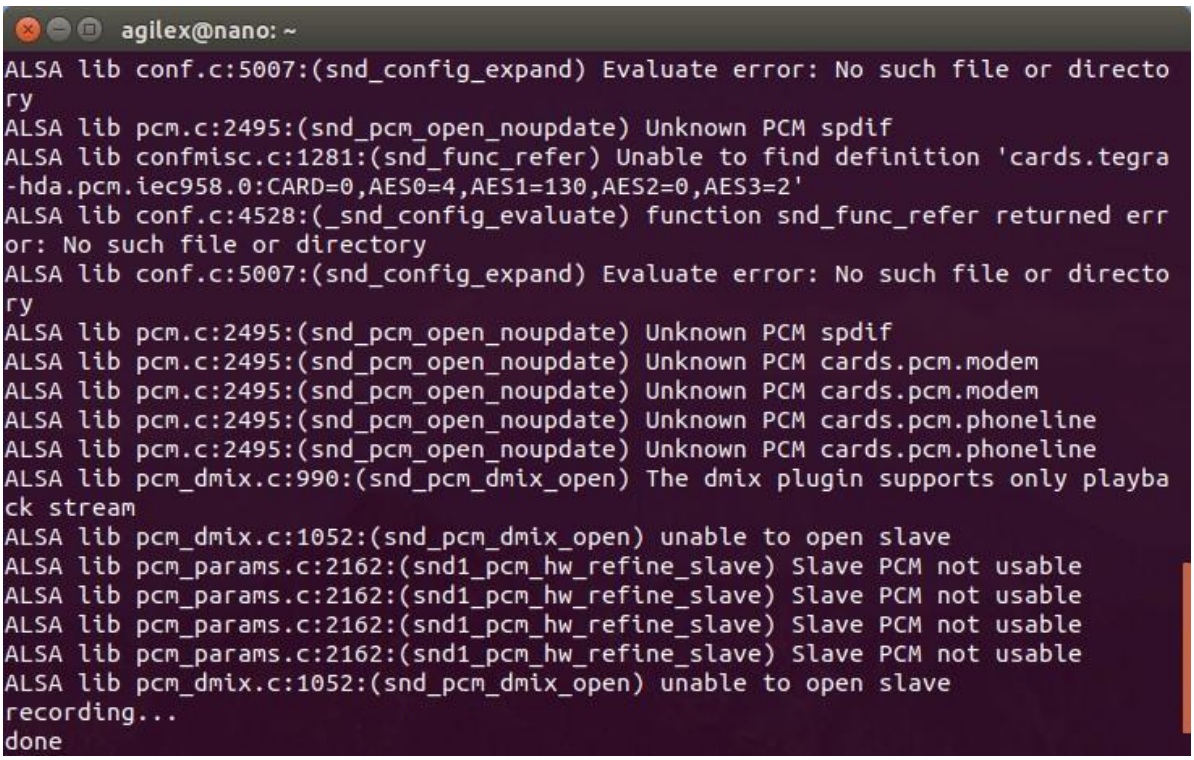
注：在运行命令之前，请确保其他终端中的程序已经终止，终止命令为：Ctrl+c

Note: Before running the command, please make sure that the programs in other terminals have been terminated. The termination command is: Ctrl+c

在终端中输入以下命令，当终端中出现recording时，开始录制声音，3秒中之后录制完毕，终端会出现Done

Enter the following command in the terminal. When “recording” appears in the terminal, start to record the voice. After 3 seconds, the recording is complete, and “Done” will appear on the terminal

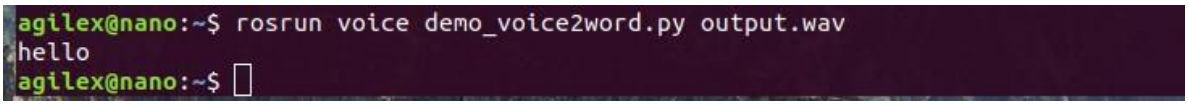
```
rosrun voice  demo_record_voice.py
```



声音录制完毕后，在终端中输入命令：

After the voice is recorded, enter the command in the terminal:

```
rosrun voice  demo_voice2word.py output.wav
```



9.2 语音控制

9.2 Voice control

9.2.1 功能简介

9.2.1 Function Introduction

通过对limo说 ahead back right left，控制limo前进，后退，右转，左转

Control limo to move ahead, move back, turn right, and turn left by saying ahead, back, right and left to limo.

9.2.2 运行功能

9.2.2 Running function

注：在运行命令之前，请确保其他终端中的程序已经终止，终止命令为：Ctrl+c

Note: Before running the command, please make sure that the programs in other terminals have been terminated. The termination command is: Ctrl+c

1、启动底盘节点，底盘节点启动之后，把手柄调至指令模式

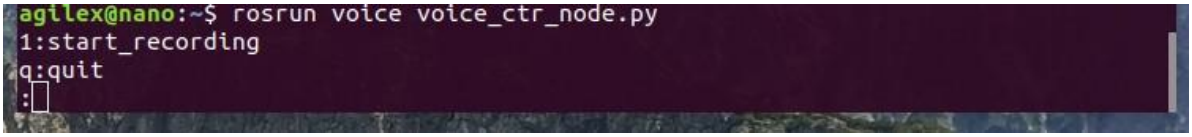
1. Launch the chassis node, and adjust the handle to the command mode after the chassis node is launched

```
roslaunch limo_base limo_base.launch
```

2、启动语音控制节点，启动之后会出现下图的界面，通过输入1，回车之后进入语音录制模式，并控制limo；输入q则退出

2. Launch the voice control node. After launching, the interface shown in the figure below will appear. Enter 1 and press Enter to enter the voice recording mode and control limo; enter q to quit

```
rosrun voice voice_ctr_node.py
```

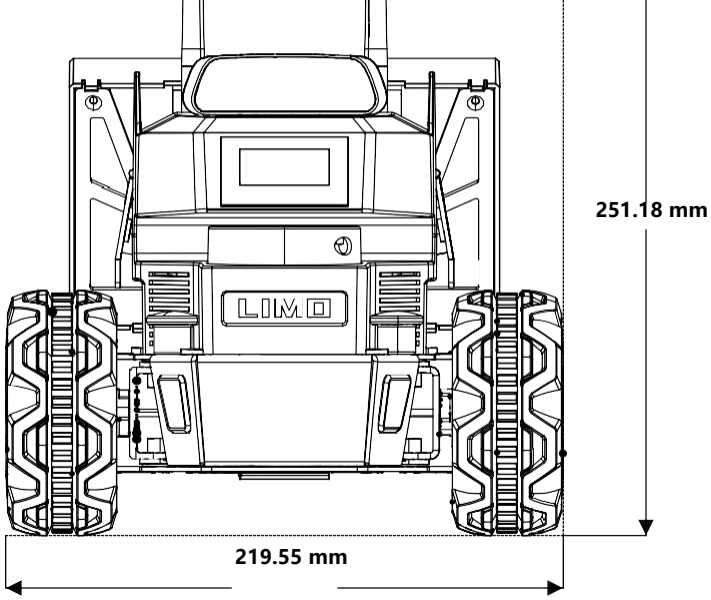


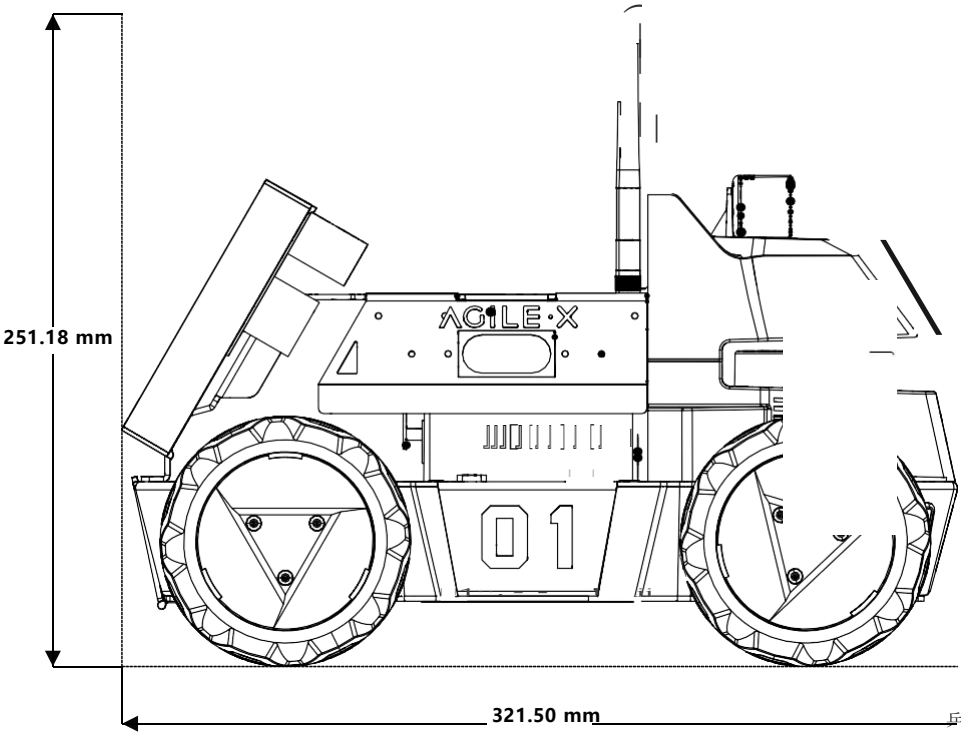
附录

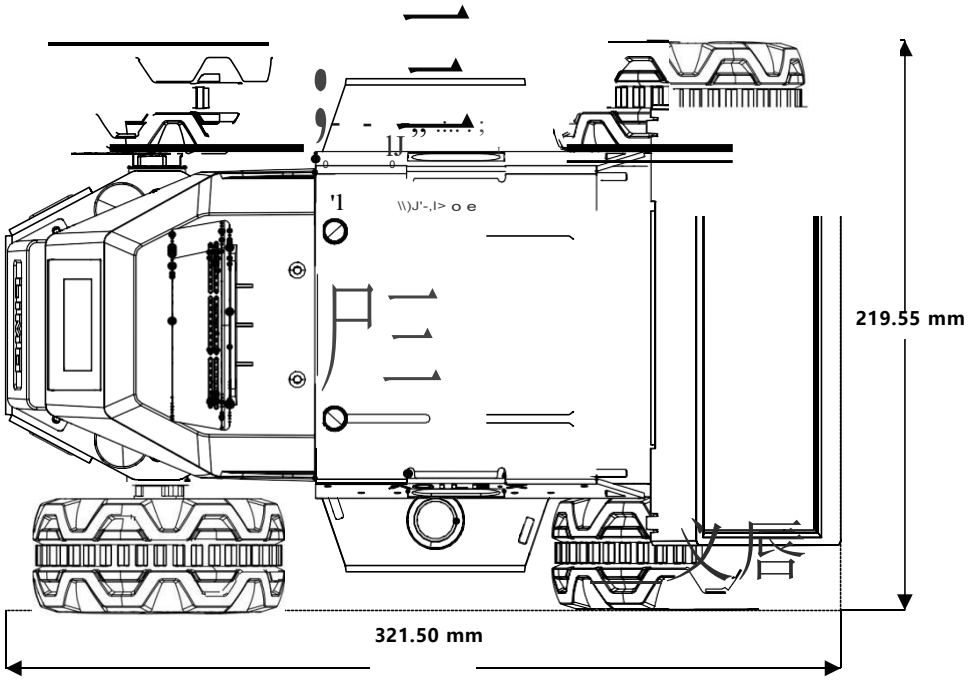
Appendix

附录1、三视图

Appendix 1. Three Views


LIMO
251.18 mm
219.55 mm





附录2、基本操作命令

Appendix 2. Basic Operating Commands

2.1 目录操作命令

2.1 Directory operating commands

(1) 目录切换 cd

(1) Directory switch: cd

① cd /　切换到根目录

① cd /: switch to the root directory

② cd /usr　切换到根目录下的usr目录

② cd /usr: switch to the usr directory under the root directory

③ cd ../　切换到上一级目录 或者 cd ..

③ cd ../: switch to the upper level directory or cd ..

④ cd ~　切换到home目录

④ cd ~: switch to the home directory

⑤ cd -　切换到上次访问的目录

⑤ cd -: switch to the last visited directory

(2) 目录查看 ls

(2) Directory view: ls

① ls　查看当前目录下的所有目录和文件

① ls: view all directories and files in the current directory

② ls -a　查看当前目录下的所有目录和文件（包括隐藏的文件）

② ls -a: view all directories and files in the current directory (including hidden files)

③ ls -l或ll 列表查看当前目录下的所有目录和文件（列表查看，显示更多信息）

③ ls -l or ll: list view all directories and files in the current directory (list view, which displays more information)

④ ls /dir 查看指定目录下的所有目录和文件如：ls /usr

④ ls /dir: view all directories and files in the specified directory, like: ls /usr

(3) 创建目录 mkdir

(3) Create directory: mkdir

① mkdir aaa 在当前目录下创建一个名为aaa的目录

① mkdir aaa: create a directory named aaa in the current directory

② mkdir /usr/aa 在指定目录下创建一个名为aaa的目录

② mkdir /usr/aa: create a directory named aaa in the specified directory

(4) 显示隐藏目录 Ctrl+h

(4) Show hidden directory: Ctrl+h

在文件夹中，通过 Ctrl+h可以显示文件夹中的隐藏文件夹

In the folder, you can display the hidden folders in the folder by Ctrl+h

(5) 终止程序 Ctrl+c

(5) Terminate program: Ctrl+c

在终端中输入命令 Ctrl+c，强制终止正在执行的程序

Enter the command Ctrl+c in the terminal to forcefully terminate the program being executed

2.2 ROS常用命令

2.2 ROS commonly used commands

（1）编译命令 catkin_make

(1) Compile command: catkin_make

用于编译整个工作空间中的功能包

Used to compile packages in the entire workspace

（2）初始化工作空间 catkin_init_workspace

(2) Initialize workspace: catkin_init_workspace

在初次创建工作空间时，用于初始化工作空间

Used to initialize the workspace when creating the workspace for the first time

（3）创建功能包 catkin_create_pkg

(3) Create package: catkin_create_pkg

用于创建一个功能包，其语法为：

Used to create a package, and its syntax is:

```
catkin_create_pkg <package_name> [depend1] [depend2] [depend3]...
```

（4）节点运行命令

(4) Node running command

① rosrun 用于运行.cpp文件和.py文件，其语法为：

① rosrun is used to run the .cpp files and the .py files, and its syntax is:

```
    rosrun package_name node_name
```

② roslaunch 用于运行.launch文件，在launch文件中可以同时调用.cpp文件和.py文件，其语法为：

② roslaunch is used to run .launch files. In the launch files, the .cpp files and the .py files can be called at the same time, and its syntax is:

```
roslaunch package_name node_name
```

附录3、ROS框架

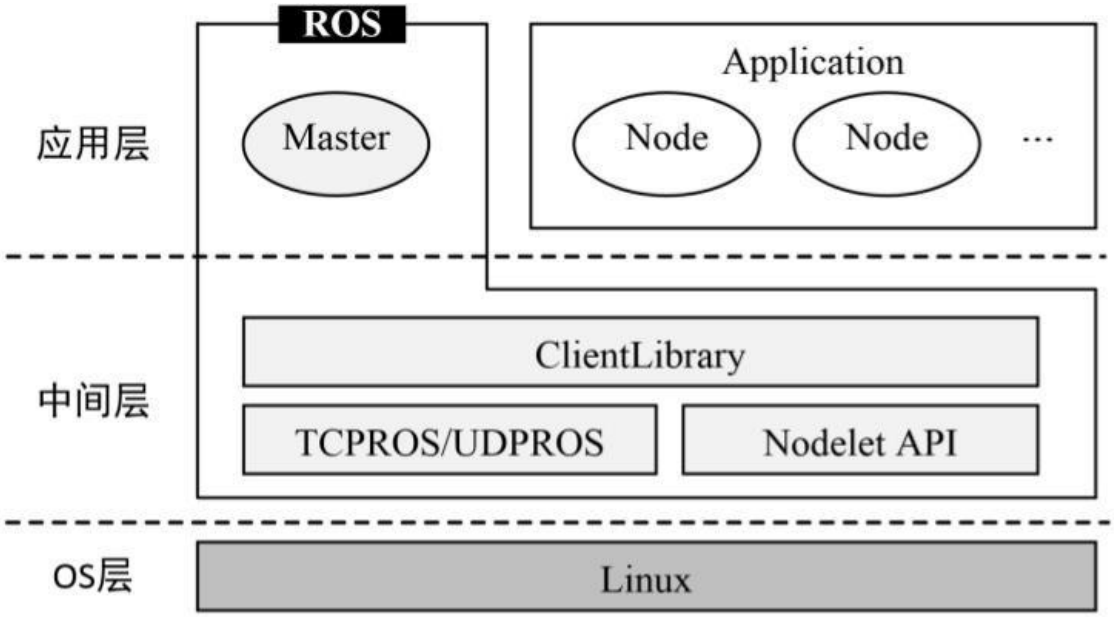
Appendix 3. ROS Framework

3.1 ROS架构设计

3.1 ROS architecture design

ROS架构如下图，可以将其分为三个层次：OS层、中间层和应用层。

The ROS architecture is as shown in the figure below, which can be divided into three layers: OS layer, middle layer and application layer.



原文	译文
应用层	Application layer
中间层	Middle layer
OS层	OS layer

（1）OS层

(1) OS layer

ROS并不是一个传统意义上的操作系统，无法像Windows、Linux一样直接运行在计算机硬件之上，而 是需要依托于Linux系统。所以在OS层，我们可以直接使用ROS官方支持度最好的Ubuntu操作系统，也 可以使用macOS、Arch、Debian等操作系统。

ROS is not an operating system in the traditional sense. It cannot run directly on computer hardware like Windows and Linux. Instead, it needs to rely on the Linux system. So at the OS layer, we can directly use the Ubuntu operating system with the best official support of ROS, or use macOS, Arch, Debian and other operating systems.

（2）中间层

(2) Middle layer

Linux是一个通用系统，并没有针对机器人开发提供特殊的中间件，所以ROS在中间层做了大量工作，其 中最为重要的就是基于TCPROS/UDPROS的通信系统。ROS的通信系统基于TCP/UDP网络，在此之上进 行了

再次封装，也就是TCPROS/UDPROS。通信系统使用发布/订阅、客户端/服务器等模型，实现多种 通信机制的数据传输。

Linux is a general system and does not provide special middleware for robot development, so ROS has done a lot of work in the middle layer, the most important of which is the communication system based on TCPROS/UDPROS. The ROS communication system is based on the TCP/UDP network, on which it is re-encapsulated, that is, TCPROS/UDPROS. The communication system uses publish/subscribe, client/server and other models to realize data transmission through multiple communication mechanisms.

除了TCPROS/UDPROS的通信机制外，ROS还提供一种进程内的通信方法——Nodelet，可以为多进程 通信提供一种更优化的数据传输方式，适合对数据传输实时性方面有较高要求的应用。

In addition to the communication mechanism of TCPROS/UDPROS, ROS also provides an in-process communication method——Nodelet, which can provide a more optimized data transmission method for multi-process communication, which is suitable for applications that have higher requirements for real-time data transmission.

在通信机制之上，ROS提供了大量机器人开发相关的库，如数据类型定义、坐标变换、运动控制等，可以提供给应用层使用。

On top of the communication mechanism, ROS provides a large number of libraries related to robot development, such as data type definition, coordinate transformation, motion control, etc., which can be provided to the application layer.

（3）应用层

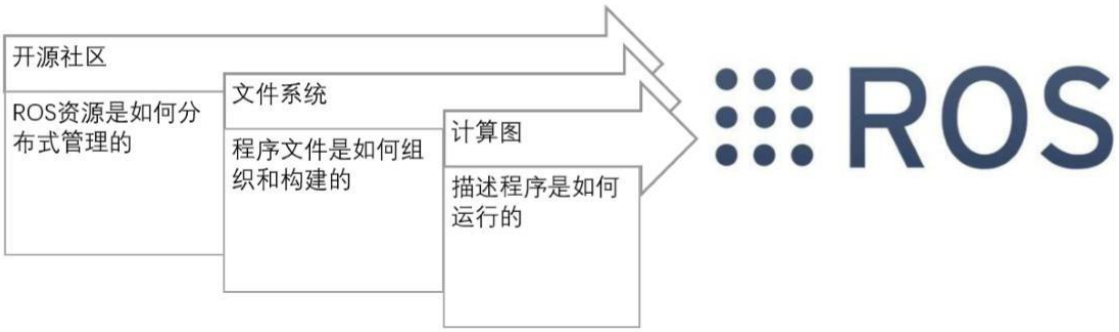
(3) Application layer

在应用层，ROS需要运行一个管理者——Master，负责管理整个系统的正常运行。ROS社区内共享了大 量的机器人应用功能包，这些功能包内的模块以节点为单位运行，以ROS标准的输入输出作为接口，开 发者不需要关注模块的内部实现机制，只需要了解接口规则即可实现复用，极大地提高了开发效率。

At the application layer, ROS needs to run a manager——Master, responsible for managing the normal operation of the entire system. A large number of robot application packages are shared in the ROS community. The modules in these packages run as nodes and use ROS standard input and output as interfaces. Developers do not need to pay attention to the internal implementation mechanism of the modules, but only need to understand the interface rules to achieve reuse, which greatly improves development efficiency.

从系统实现的角度来看，ROS也可以分为如下图所示的三个层次：文件系统、计算图和开源社区。

From the perspective of system implementation, ROS can also be divided into three layers as shown in the following figure: file system, computation graph, and open source community.



原文	译文
开源社区	Open source community
文件系统	File system
计算图	Computation graph
ROS资源是如何分布式管理的	How are ROS resources managed in a distributed manner
程序文件是如何组织和构建的	How are program files organized and constructed
描述程序是如何运行的	How does the description program work

3.2 计算图

3.2 Computation graph

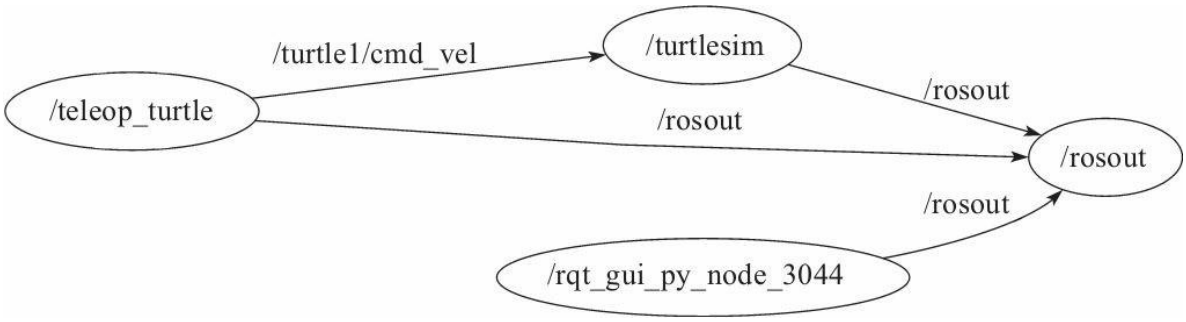
从计算图的角度来看，ROS系统软件的功能模块以节点为单位独立运行，可以分布于多个相同或不同的主机中，在系统运行时通过端对端的拓扑结构进行连接。

From the perspective of the computation graph, the functional modules of the ROS system software run independently in units of nodes, which can be distributed in multiple same or different hosts, and are connected through an end-to-end topology when the system is running.

3.2.1 节点

3.2.1 Node

节点(Node)就是一些执行运算任务的进程，一个系统一般由多个节点组成，也可以称为“软件模块”。节点概念的引入使得基于ROS的系统在运行时更加形象：当许多节点同时运行时，可以很方便地将端对端的通信绘制成如下图所示的节点关系图，在这个图中进程就是图中的节点，而端对端的连接关系就是节点之间的连线。



Nodes are processes that perform computing tasks. A system generally consists of multiple nodes, which can also be called "software modules." The introduction of the node concept makes the ROS-based system more vivid at runtime: When many nodes are running at the same time, it is easy to draw the end-to-end communication into the node diagram as shown below, in which the process is the node, and the end-to-end connection is the connecting line between the nodes.

3.2.2 消息

3.2.2 Message

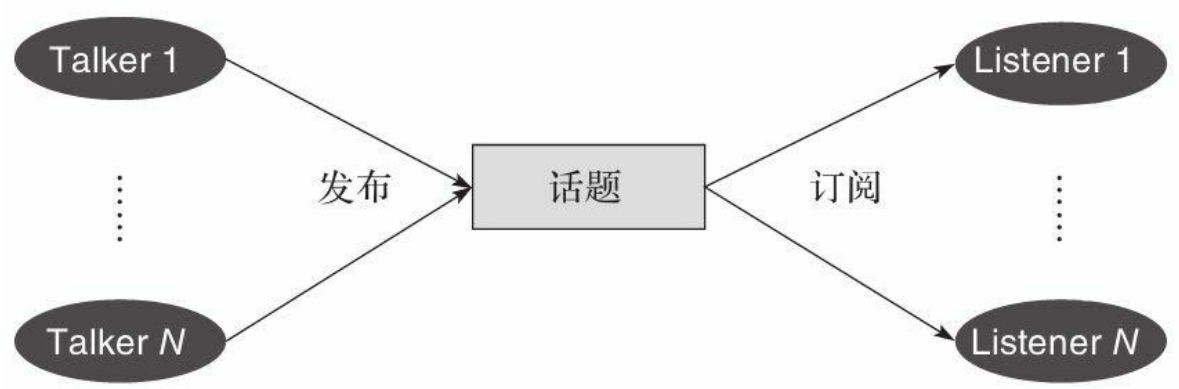
节点之间最重要的通信机制就是基于发布/订阅模型的消息(Message)通信。每一个消息都是一种严格的 数据结构，支持标准数据类型(整型、浮点型、布尔型等)，也支持嵌套结构和数组(类似于C语言的结构体struct)，还可以根据需求由开发者自主定义。

The most important communication mechanism between nodes is the message communication based on the publish/subscribe model. Each message is a strict data structure, which supports standard data types (integer, floating point, Boolean, etc.), as well as nested structures and arrays (similar to the structure of C language), and can also be independently defined by the developer according to requirements.

3.2.3 话题

3.2.3 Topic

消息以一种发布/订阅(Publish/Subscribe)的方式传递(见下图)。一个节点可以针对一个给定的话题(Topic)发布消息(称为发布者/Talker)，也可以关注某个话题并订阅特定类型的数据(称为订阅者/Listener)。发布者和订阅者并不了解彼此的存在，系统中可能同时有多个节点发布或者订阅同一个话 题的消息。



Messages are delivered in a Publish/Subscribe manner (see the figure below). A node can publish a message for a given Topic (called Talker), or it can follow a topic and subscribe to a specific type of data (called Listener). Talkers and Listeners do not know each other's existence. There may be multiple nodes in the system simultaneously publishing or subscribing to the same topic.

原文	译文
发布	Publish
话题	Topic
订阅	Subscribe

3.2.4 服务

3.2.4 Service

虽然基于话题的发布/订阅模型是一种很灵活的通信模式，但是对于双向的同步传输模式并不适合。在ROS中，我们称这种同步传输模式为服务(Service)，其基于客户端/服务器(Client/Server)模型，包含两 个部分的通信数据类型：一个用于请求，另一个用于应答，类似于Web服务器。与话题不同的是，ROS 中只允许有一个节点提供指定命名的服务。

Although the topic-based publish/subscribe model is a very flexible communication mode, it is not suitable for the two-way synchronous transmission mode. In ROS, we call this synchronous transmission mode Service, which is based on the Client/Server model and contains two parts of communication data types: one for request and the other for response, similar to a Web server. Different from the topic, only one node is allowed to provide the specified naming service in ROS.

3.2.5 节点管理器

3.2.5 Master

为了统筹管理以上概念，系统当中需要有一个控制器使得所有节点有条不紊地执行，这就是ROS节点管理器(ROS Master)。ROS Master通过远程过程调用(RPC)提供登记列表和对其他计算图表的查找功能， 帮助ROS节点之间相互查找、建立连接，同时还为系统提供参数服务器，管理全局参数。ROS Master就是一个管理者，没有它的话，节点将无法找到彼此，也无法交换消息或调用服务，整个系统将会瘫痪， 由此可见其在ROS系统中的重要性。

In order to manage the above concepts as a whole, there needs to be a controller in the system to make all nodes execute in an orderly manner. This is the ROS Master. ROS Master provides registration lists and search functions for other computation graphs through remote procedure call (RPC), helps ROS nodes to find and establish connections with each other, and also provides a parameter server for the system to manage global parameters. ROS Master is a manager. Without it, nodes will not be able to find each other, exchange messages or call services, and the entire system will be paralyzed. This shows its importance in the ROS system.

3.3 文件系统

3.3 File system

类似于操作系统，ROS将所有文件按照一定的规则进行组织，不同功能的文件被放置在不同的文件夹下，如图2-5所示。

Similar to the operating system, ROS organizes all files according to certain rules, and files with different functions are placed in different folders, as shown in Figure 2-5.

功能包(Package)：功能包是ROS软件中的基本单元，包含ROS节点、库、配置文件等。

Package: The package is the basic unit of ROS software, including ROS nodes, libraries, configuration files, etc.

功能包清单(PackageManifest)：每个功能包都包含一个名为package.xml的功能包清单，用于记录功能 包的基本信息，包含作者信息、许可信息、依赖选项、编译标志等。

Package Manifest: Each package contains a package manifest named package.xml, which is used to record the basic information of the package, including author information, license information, dependent options, compilation flags, etc.

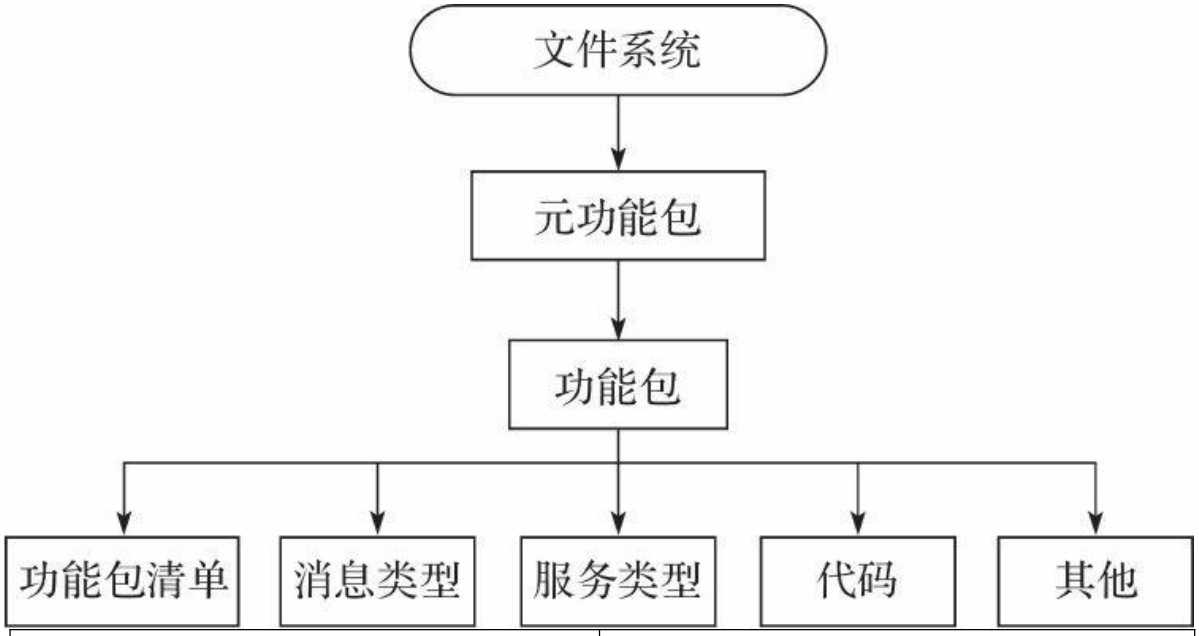
元功能包(Meta Package)：在新版本的ROS中，将原有功能包集(Stack)的概念升级为“元功能包”，主要作用都是组织多个用于同一目的的功能包。例如一个ROS导航的元功能包中会包含建模、定位、导航等 多个功能包。

Meta Package: In the new version of ROS, the concept of the original Stack is upgraded to "Meta Package", whose main function is to organize multiple packages for the same purpose. For example, a ROS navigation meta package will contain multiple packages such

as modeling, positioning, and navigation.

元功能包清单：在下图中并未显示，类似于功能包清单，不同之处在于元功能包清单中可能会包含运行时需要依赖的功能包或者声明一些引用的标签。

Meta Package Manifest: Not shown in the figure below. It is similar to the package manifest. The difference is that the meta package manifest may contain packages that need to be relied upon at runtime or tags for declaring some references.



原文	译文
文件系统	File system
元功能包	Meta Package
功能包	Package
功能包清单	Package Manifest
消息类型	Message type
服务类型	Service type
代码	Code
其他	Other

消息(Message)类型：消息是ROS节点之间发布/订阅的通信信息，可以使用ROS提供的消息类型，也可 以使用.msg文件在功能包的msg文件夹下自定义所需要的消息类型。

Message type: Message is the communication information published/subscribed between ROS nodes. You can use the message type provided by ROS, or you can use the .msg file to customize the required message type in the msg folder of the package.

服务(Service)类型：服务类型定义了ROS客户端/服务器通信模型下的请求与应答数据类型，可以使用ROS系统提供的服务类型；也可以使用.srv文件在功能包的srv文件夹中进行定义。

Service type: The service type defines the request and response data type under the ROS client/server communication model. You can use the service types provided by the ROS system; you can also use the .srv file to define in the srv folder of the package.

代码(Code)：用来放置功能包节点源代码的文件夹。

Code: The folder used to place the source code of the package node.

3.4 开源社区

3.4 Open source community

ROS开源社区中的资源非常丰富，而且可以通过网络共享以下软件和知识(见图2-9)。

The resources in the ROS open source community are very rich, and the following software and knowledge can be shared through the network (see Figure 2-9).

发行版(Distribution)：类似于Linux发行版，ROS发行版包括一系列带有版本号、可以直接安装的功能包，这使得ROS的软件管理和安装更加容易，而且可以通过软件集合来维持统一的版本号。

Distribution: Similar to the Linux distribution, the ROS distribution includes a series of packages with version numbers that can be installed directly. This makes ROS’ s software management and installation easier, and it can maintain a unified version number through software collections.

软件源(Repository)：ROS依赖于共享网络上的开源代码，不同的组织机构可以开发或者共享自己的机器 人软件。

Repository: ROS relies on the open source code on the shared network, and different organizations can develop or share their own robotic software.

ROS wiki：记录ROS信息文档的主要论坛。所有人都可以注册、登录该论坛，并且上传自己的开发文档、进行更新、编写教程。

ROS wiki: The main forum for documenting ROS information. Everyone can register, log in to the forum, upload their own development documents, update, and write tutorials.

邮件列表(Mailing List)：ROS邮件列表是交流ROS更新的主要渠道，同时也可以交流ROS开发的各种疑问。

Mailing List: The ROS mailing list is the main channel for communicating ROS updates, as well as various questions about ROS development.

ROS Answers：ROS Answers是一个咨询ROS相关问题的网站，用户可以在该网站提交自己的问题并得到其他开发者的回答。

ROS Answers: ROS Answers is a website for inquiring about ROS related questions. Users can submit their own questions on this website and get answers from other developers.

博客(Blog)：发布ROS社区中的新闻、图片、视频 (http：//www.ros.org/news)。

Blog: Publish news, pictures, and videos in the ROS community (http：//www.ros.org/news).

3.5 通信机制

3.5 Communication mechanism

ROS是一个分布式框架，为用户提供多节点(进程)之间的通信服务，所有软件功能和工具都建立在这种分布式通信机制上，所以ROS的通信机制是最底层也是最核心的技术。在大多数应用场景下，尽管我们不需要关注底层通信的实现机制，但是了解其相关原理一定会帮助我们在开发过程中更好地使用ROS。以下就ROS最核心的三种通信机制进行介绍。

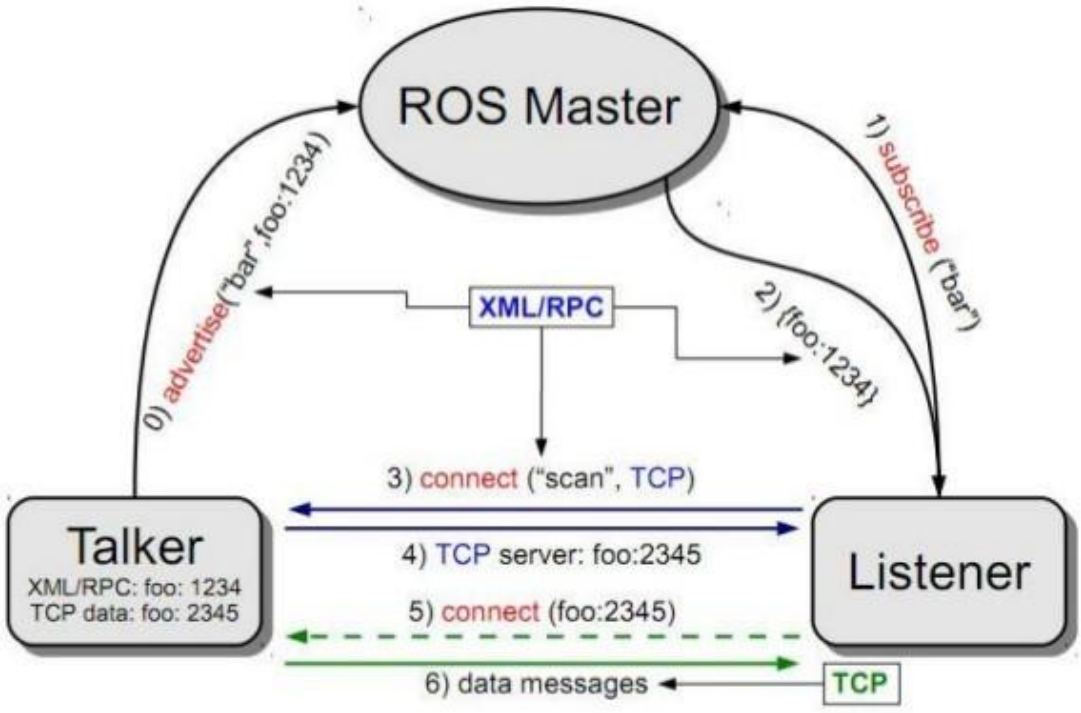
ROS is a distributed framework that provides users with communication services between multiple nodes (processes). All software functions and tools are based on this distributed communication mechanism, so the ROS communication mechanism is the lowest and most core technology. In most application scenes, although we don't need to pay attention to the implementation mechanism of the underlying communication, understanding its related principles will definitely help us make better use of ROS in the development process. The three core communication mechanisms of ROS are introduced below.

3.5.1 话题通信机制

3.5.1 Topic communication mechanism

话题在ROS中使用最为频繁，其通信模型也较为复杂。如下图所示，在ROS中有两个节点：一个是发布者Talker，另一个是订阅者Listener。两个节点分别发布、订阅同一个话题，启动顺序没有强制要求，此 处假设Talker首先启动，可分成图中所示的七步来分析建立通信的详细过程。

Topic is used most frequently in ROS, and its communication model is also more complicated. As shown in the figure below, there are two nodes in ROS: one is Talker, and the other is Listener. Two nodes publish and subscribe to the same topic separately. There is no mandatory requirement for the startup sequence. Here, assuming that Talker starts first, it can be divided into the seven steps shown in the figure to analyze the detailed process of establishing communication.



1. Talker注册

1. Talker registration

Talker启动，通过1234端口使用RPC向ROS Master注册发布者的信息，包含所发布消息的话题名；ROS Master会将节点的注册信息加入注册列表中。

Talker starts and uses RPC to register Talker's information with ROS Master through port 1234, including the topic name of the published message; ROS Master will add the registration information of the node to the registration list.

2. Listener注册

2. Listener registration

Listener启动，同样通过RPC向ROS Master注册订阅者的信息，包含需要订阅的话题名。

The Listener starts and also registers Listener’ s information with ROS Master through RPC, including the topic name that needs to be subscribed.

3. ROS Master进行信息匹配

3. ROS Master performs information matching

Master根据Listener的订阅信息从注册列表中进行查找，如果没有找到匹配的发布者，则等待发布者的 加入；如果找到匹配的发布者信息，则通过RPC向Listener发送Talker的RPC地址信息。
The Master searches the registration list based on the subscription information of the Listener. If no matching Talker is found, it waits for the Talker to join; if it finds the matching Talker information, it sends the Talker's RPC address information to the Listener via RPC.

4. Listener发送连接请求

4. Listener sends a connection request

Listener接收到Master发回的Talker地址信息，尝试通过RPC向Talker发送连接请求，传输订阅的话题 名、消息类型以及通信协议(TCP/UDP)。

The Listener receives the Talker address information sent back by the Master, and tries to send a connection request to the Talker via RPC, and transmits the topic name, message type and communication protocol (TCP/UDP) subscribed to.

5. Talker确认连接请求

5. Talker confirms the connection request

Talker接收到Listener发送的连接请求后，继续通过RPC向Listener确认连接信息，其中包含自身的TCP 地址信息。
After the Talker receives the connection request sent by the Listener, it continues to confirm the connection information to the Listener via RPC, which contains its own TCP address information.

6. Listener尝试与Talker建立网络连接

6. Listener tries to establish a network connection with Talker

Listener接收到确认信息后，使用TCP尝试与Talker建立网络连接。

After the Listener receives the confirmation message, it uses TCP to try to establish a network connection with the Talker.

7. Talker向Listener发布数据

7. Talker publishes data to Listener

成功建立连接后，Talker开始向Listener发送话题消息数据。

After successfully establishing a connection, Talker starts to send topic message data to Listener.

从上面的分析中可以发现，前五个步骤使用的通信协议都是RPC，最后发布数据的过程才使用到TCP。

From the above analysis, it can be found that the communication protocol used in the first five steps is RPC, and TCP is used only in the process of publishing data.
ROS Master在节点建立连接的过程中起到了重要作用，但是并不参与节点之间最终的数据传输。
ROS Master plays an important role in the process of establishing connections between nodes, but does not participate in the final data transmission between nodes.

节点建立连接后，可以关掉ROS Master，节点之间的数据传输并不会受到影响，但是其他节点也无法加入

这两个节点之间的网络。

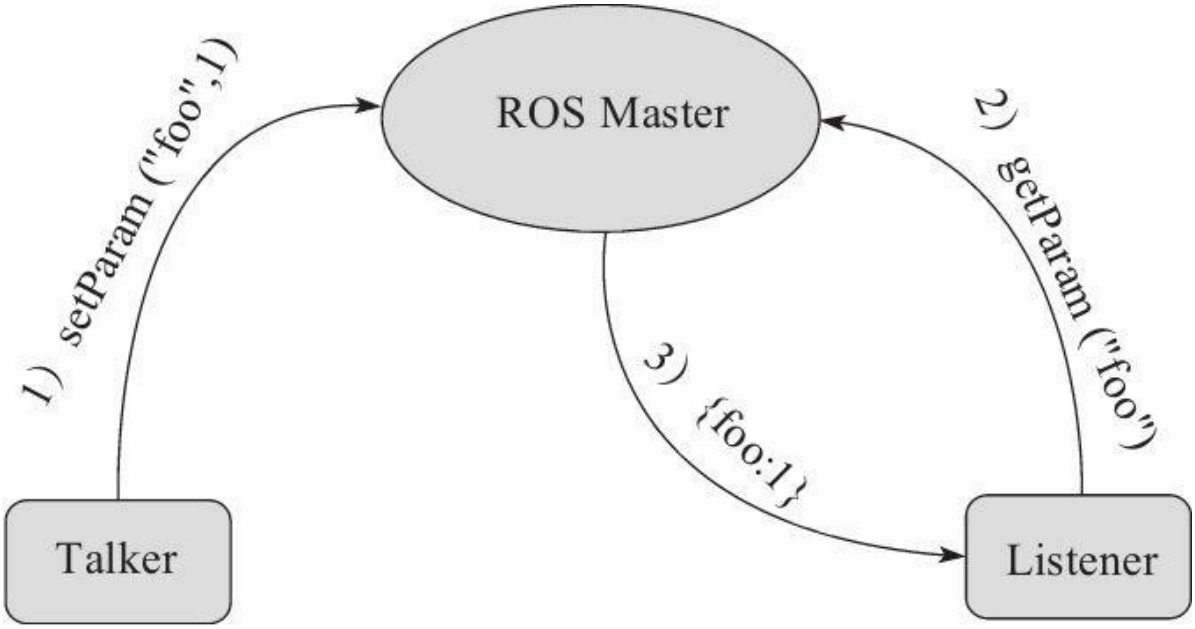
After the nodes have established a connection, you can turn off the ROS Master, and the data transmission between the nodes will not be affected, but other nodes cannot join the network between the two nodes.

3.5.2 参数管理机制

3.5.2 Parameter management mechanism

参数类似于ROS中的全局变量，由ROS Master进行管理，其通信机制较为简单，不涉及TCP/UDP的通信，如下图所示。

The parameters are similar to global variables in ROS and are managed by ROS Master. Their communication mechanism is relatively simple and does not involve TCP/UDP communication, as shown in the figure below.



1. Talker设置变量

1. Talker sets variables

Talker使用RPC向ROS Master发送参数设置数据，包含参数名和参数值；ROS Master会将参数名和参数值保存到参数列表中。

The Talker uses RPC to send parameter setting data to ROS Master, including parameter names and parameter values; ROS Master saves the parameter names and parameter values in the parameter list.

2. Listener查询参数值

2. Listener searches parameter values

Listener通过RPC向ROS Master发送参数查找请求，包含所要查找的参数名。

The Listener sends a parameter search request to the ROS Master via RPC, including the parameter name to be searched.

3.ROS Master向Listener发送参数值

3. ROS Master sends parameter values to Listener

Master根据Listener的查找请求从参数列表中进行查找，查找到参数后，使用RPC将参数值发送给Listener。

The Master searches the parameter list according to the Listener's search request. After finding the parameters, it uses RPC to send the parameter values to the Listener.

这里需要注意的是，如果Talker向Master更新参数值，Listener在不重新查询参数值的情况下是无法知 晓参数值已经被更新的。所以在很多应用场景中，需要一种动态参数更新的机制，第12章会具体讲解ROS中动态参数配置功能的实现。

It should be noted here that if the Talker updates the parameter values to the Master, the Listener cannot know that the parameter values have been updated without searching the parameter values again. Therefore, in many application scenes, a dynamic parameter update mechanism is required. Chapter 12 will specifically explain the implementation of the dynamic parameter configuration function in ROS.

附录4、ROS基础

Appendix 4. ROS Basics

4.1 工作空间

4.1 Workspace

使用ROS实现机器人开发的主要手段当然是写代码，那么这些代码文件就需要放置到一个固定的空间内，也就是工作空间。

The main method of using ROS to realize robot development is of course to write code, so these code files need to be placed in a fixed space, that is, the workspace.

4.1.2 什么是工作空间

4.1.2 What is a workspace

工作空间(workspace)是一个存放工程开发相关文件的文件夹。Fuerte版本之后的ROS默认使用的是Catkin编译系统，一个典型Catkin编译系统下的工作空间结构如下。

Workspace is a folder for storing project development related files. ROS after the Fuerte version uses the Catkin compilation system by default. The workspace structure under a typical Catkin compilation system is as follows.

典型的工作空间中一般包括以下四个目录空间。

A typical workspace generally includes the following four directory spaces.

（1） src：代码空间，开发过程中最常用的文件夹，用来存储所有ROS功能包的源码文件。

(1) src: code space, the most commonly used folder in the development process, used to store the source code files of all ROS packages.

（2） build：编译空间，用来存储工作空间编译过程中产生的缓存信息和中间文件。

(2) build: compilation space, used to store the cache information and intermediate files generated during the compilation of the workspace.

（3） devel：开发空间，用来放置编译生成的可执行文件。

(3) devel: development space, used to place executable files generated by compilation.

（4） lib：安装库空间，编译成功后，可以使用make install命令将可执行文件安装到该空间中，运行该空间中的环境变量脚本，即可在终端中运行这些可执行文件。安装库空间并不是必需的，很多工作空间 中可能并没有该文件夹。

(4) lib: installation library space. After the compilation is successful, you can use the make install command to install the executable files into the space, and run the environment variable scripts in the space to run these executable files in the terminal. The installation library space is not necessary, and this folder may not be available in many workspaces.

4.2.2 创建工作空间

4.2.2 Create a workspace

创建工作空间的命令比较简单，首先使用系统命令创建工作空间目录，然后运行ROS的工作空间初始化命令即可完成创建过程：

The command to create a workspace is relatively simple. First, use the system command to create the workspace directory, and then run the ROS workspace initialization command to complete the creation process:

创建一个名为catkin_ws的文件夹并在catkin_ws下继续创建一个名为src的文件夹

Create a folder named catkin_ws and continue to create a folder named src under catkin_ws

```
mkdir -p ~/catkin_ws/src
```

切换到catkin_ws文件夹下的src文件夹

Switch to the src folder under the catkin_ws folder

```
cd ~/catkin_ws/src
```

初始化文件夹

Initialize the folder

```
catkin_init_workspace
```

创建完成后，可以在工作空间的根目录下使用catkin_make命令编译整个工作空间：

After the creation is complete, you can use the catkin_make command to compile the entire workspace in the root directory of the workspace:

切换到catkin_ws文件夹下

Switch to the catkin_ws folder

```
cd ~/catkin_ws/
```

运行编译命令

Run the compile command

```
catkin_make
```

编译过程中，在工作空间的根目录里会自动产生build和devel两个文件夹及其中的文件。编译完成后， 在devel文件夹中已经产生几个setup.*sh形式的环境变量设置脚本。使用source命令运行这些脚本文 件，则工作空间中的环境变量可以生效。

During the compilation process, two folders, build and devel, and their files will be automatically generated in the root directory of the workspace. After the compilation is complete, several environment variable setting scripts in the form of setup.*sh have been generated in the devel folder. Use the source command to run these script files, then the environment variables in the workspace can take effect.

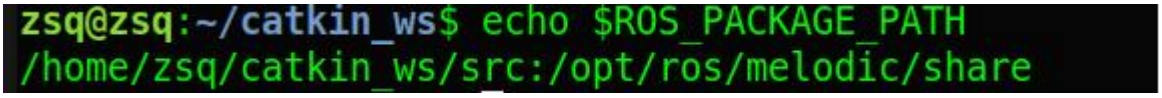
```
source devel/setup.bash
```

为了确保环境变量已经生效，可以使用如下命令进行检查：

In order to ensure that the environment variables have taken effect, you can use the following command to check:

```
echo $ROS_PACKAGE_PATH
```

如果打印的路径中已经包含当前工作空间的路径，则说明环境变量设置成功(见下图)。



If the printed path already contains the path of the current workspace, it means that the environment variable is set successfully (see the figure below).

在终端中使用source命令设置的环境变量只能在当前终端中生效，如果希望环境变量在所有终端中有效，则需要在终端的配置文件中加入环境变量的设置：

The environment variables set by the source command in the terminal can only be effective in the current terminal. If you want the environment variables to be effective in all terminals, you need to add the environment variable settings in the terminal configuration file:

```
echo"source/WORKSPACE/devel/setup.bash">>~/.bashrc
```

请使用工作空间路径代替WORKSPACE。

Please use the workspace path instead of WORKSPACE.

4.2.3 创建功能包

4.2.3 Create a package

ROS中功能包的形式如下：

The form of the package in ROS is as follows:

```
my_package/
    CMakeLists.txt
    package.xml

    ......
```

package.xml文件提供了功能包的元信息，也就是描述功能包属性的信息。CMakeLists.txt文件记录了功 能包的编译规则。

The package.xml file provides meta-information of the package, that is, information describing the attributes of the package. The CMakeLists.txt file records the compilation rules of the package.

ROS不允许在某个功能包中嵌套其他功能包，多个功能包必须平行放置在代码空间中。

ROS does not allow other packages to be nested in a certain package, and multiple packages must be placed in parallel in the code space.

ROS提供直接创建功能包的命令catkin_create_pkg，该命令的使用方法如下：

ROS provides the command catkin_create_pkg to directly create a package. The usage of this command is as follows:

```
catkin_create_pkg <package_name> [depend1] [depend2] [depend3]...
```

在运行catkin_create_pkg命令时，用户需要输入功能包的名称(package_name)和所依赖的其他功能包 名称(depend1、depend2、depend3)。例如，我们需要创建一个learning_limo功能包，该功能包依赖 于std_msgs、roscpp、rospy等功能包。

When running the catkin_create_pkg command, the user needs to enter the package name and the names of other packages that it depends on (depend1, depend2, depend3). For example, we need to create a learning_limo package, which depends on std_msgs, roscpp, rospy and other packages.

首先进入代码空间，使用catkin_create_pkg命令创建功能包：

First enter the code space and use the catkin_create_pkg command to create a package:

切换到catkin_ws文件夹下的src文件夹

Switch to the src folder under the catkin_ws folder

```
cd ~/catkin_ws/src
```

运行创建功能包的命令

Run the command to create the package

```
catkin_create_pkg learning_limo std_msgs rospy roscpp
```

创建完成后，代码空间src中会生成一个learning_limo功能包，其中已经包含package.xml和CMakeLists.txt文件。

After the creation is complete, a learning_limo package will be generated in the code space src, which already contains the package.xml and CMakeLists.txt files.

然后回到工作空间的根目录下进行编译，并且设置环境变量：

Then go back to the root directory of the workspace to compile and set the environment variables:

```
cd ~/catkin_ws
```

切换到catkin_ws文件夹下

Switch to the catkin_ws folder

运行编译命令

Run the compile command

```
 catkin_make
```

设置环境变量

Set environment variables

```
source ~/catkin_ws/devel/setup.bash
```

以上便是创建一个功能包的基本流程。

The above is the basic process of creating a package.

注意：在同一个工作空间下，不允许存在同名功能包，否则在编译时会报错。

Note: In the same workspace, no package with the same name is allowed, otherwise an error will be reported during compilation.

4.3 编写功能包控制limo

4.3 Write package to control limo

使用话题通信机制控制limo运动，需要编写Publisher和Subscriber两个节点，Publisher负责向limo发 布速度控制指令，Subscriber负责监听limo接收到的速度大小。

To use topic communication mechanism to control the movement of limo, you need to write two nodes, Publisher and Subscriber. Publisher is responsible for issuing speed control instructions to limo, and Subscriber is responsible for monitoring the speed received by limo.

4.3.1 如何创建Publisher

4.3.1 How to create Publisher

Publisher的主要作用是针对指定话题发布特定数据类型的消息。我们尝试使用代码实现一个节点，节点 中创建一个Publisher并控制limo运动。源码文件位置为

~/agilex_ws/src/limo_ros/learning_limo/src/talker.cpp，其内容如下：
The main role of Publisher is to publish messages of specific data types for specific topics. We try to use code to implement a node, create a Publisher in the node and control the movement of limo. The source code file’ s location is ~/agilex_ws/src/limo_ros/learning_limo/src/talker.cpp, and its contents are as follows:

```
#include <sstream>
#include "ros/ros.h"
#include "std_msgs/String.h"
#include <geometry_msgs/Twist.h>

int main(int argc, char **argv)
{
    // ROS节点初始化//
    ROS node initialization

    // 创建节点句柄
    // Create node handle


    // 创建一个Publisher，发布名为chatter的topic，消息类型为geometry_msgs::Twist
    // Create a Publisher, publish a topic named chatter, and the message
    type is geometry_msgs::Twist
```

```
    // 设置循环的频率
    // Set the frequency of the

    for ( int count = 0; count<10;count++)
    {
    //设置需要发布的速度大小
    //Set the speed to be published
     geometry_msgs::Twist twist;
     geometry_msgs::Vector3
     linear; linear.x=0.1;
    linear.y=0;
    linear.z=0;
    geometry_msgs::Vector3 angular;
    angular.x=0;
    angular.y=0;

    //将设置好的速度赋值给twist
    //Assign the set speed
    to twist

     //将设置好的速度发布出去
     //Publish the set speed


     // 循环等待回调函数
     // loop waiting for

     // 按照循环频率延时
     // Delay according to

    }

return 0;

}
```

4.3.2 如何创建Subscriber

4.3.2 How to create a Subscriber

接下来，我们尝试创建一个Subscriber以订阅Publisher节点发布的控制指令。源码文件位置为~/agilex_ws/src/limo_ros/learning_limo/src/listener.cpp，其内容如下：

Next, we try to create a Subscriber to subscribe to the control command published by the Publisher node. The source code file’ s location is ~/agilex_ws/src/limo_ros/learning_limo/src/listener.cpp, and its contents are as follows:

```
#include "ros/ros.h"
#include "std_msgs/String.h"
#include <geometry_msgs/Twist.h>

// 接收到订阅的消息后，会进入消息回调函数
// After receiving the subscribed message, it will enter the message
callback function
void chatterCallback(const geometry_msgs::TwistConstPtr& msg )
{
    //提取msg中的数据并赋值
    //Extract the data in msg and
       .   .

    // 将接收到的消息打印出来
    // Print out the received message
    ROS_INFO("I get x: [%f]", x);
    ROS_INFO("I get y: [%f]", y);
```

```
}

int main(int argc, char **argv)
{
    // 初始化ROS节点
    // Initialize the ROS node

// 创建节点句柄
// Create node handle

//   创建一个Subscriber，订阅名为chatter的topic，注册回调函数chatterCallback
// Create a Subscriber, subscribe to the topic named chatter, register

// 循环等待回调函数
// loop waiting for

return 0;

}
```

4.3.3 编译功能包

4.3.3 Compile package

节点的代码已经完成，C++是一种编译语言，在运行之前需要将代码编译成可执行文件，如果使用Python等解析语言编写代码，则不需要进行编译，可以省去此步骤。
The code of the node has been completed. C++ is a compiled language. You need to compile the code into an executable file before running. If you use a parsing language like Python to write the code, you do not need to compile and you can skip this step.

ROS中的编译器使用的是CMake，编译规则通过功能包中的CMakeLists.txt文件设置，使用catkin命令 创建的功能包中会自动生成该文件，已经配置多数编译选项，并且包含详细的注释，我们几乎不用查看 相关的说明手册，稍作修改就可以编译自己的代码。

The compiler in ROS uses CMake. The compilation rules are set through the CMakeLists.txt file in the package. This file will be automatically generated in the package created by the catkin command. Most compilation options have been configured and detailed comments are included. We hardly need to check the relevant manuals, and we can compile our own code with a little modification.

打开功能包中的CMakeLists.txt文件，找到以下配置项，去掉注释并稍作修改：

Open the CMakeLists.txt file in the package, find the following configuration items, remove the comments and make slight modifications:

```
include_directories(include ${catkin_INCLUDE_DIRS})

add_executable(talker src/talker.cpp)
target_link_libraries(talker ${catkin_LIBRARIES})
add_dependencies(talker   ${PROJECT_NAME}_generate_messages_cpp)

add_executable(listener src/listener.cpp)
target_link_libraries(listener ${catkin_LIBRARIES})
add_dependencies(talker   ${PROJECT_NAME}_generate_messages_cpp)
```

4.3.4 运行Publisher与Subscriber

4.3.4 Run Publisher and Subscriber

编译完成后，我们终于可以运行Publisher和Subscriber节点了。在运行节点之前，需要在终端中设置环 境变

量，否则无法找到功能包最终编译生成的可执行文件：

After the compilation is complete, we can finally run the Publisher and Subscriber nodes. Before running the nodes, you need to set the environment variables in the terminal, otherwise the executable file generated by the final compilation of the package cannot be found:

切换至catkin_ws目录下

Switch to the catkin_ws directory

```
cd ~/catkin_ws
```

设置环境变量

Set environment variables

```
source ./devel/setup.bash
```

也可以将环境变量的配置脚本添加到终端的配置文件中：

You can also add the configuration script of environment variables to the configuration file of the terminal:

```
echo "source ~/catkin_ws/devel/setup.bash" >> ~/.bashrc
source ~/.bashrc
```

在limo中，环境变量已经设置好了，可以按照以下步骤启动例程：

In limo, the environment variables have been set, and you can start the routine according to the following steps:

1.启动底盘节点

1. Launch the chassis node

注：在运行命令之前，请确保其他终端中的程序已经终止，终止命令为：Ctrl+c

Note: Before running the command, please make sure that the programs in other terminals have been terminated. The termination command is: Ctrl+c

```
roslaunch limo_base limo_base.launch
```

在运行节点之前，首先需要确保底盘节点已经成功启动：

Before running the node, you first need to ensure that the chassis node has been successfully launched:

2.启动Subscriber

2. Launch Subscriber

先使用rosrun命令启动Subscriber节点，订阅Publisher即将发布的消息：

First use the rosrun command to launch the Subscriber node and subscribe to the news about to be published by Publisher:

```
rosrun learning_limo listener
```

如果消息订阅成功，会在终端中显示接收到的消息内容。

If the message is successfully subscribed, the content of the received message will be displayed in the terminal.

3.启动Publisher

3. Launch Publisher

接下来使用rosrun命令启动Publisher：

Next use the rosrun command to launch Publisher:

```
rosrun learning_limo talker
```

如果Publisher节点运行正常，小车会动起来，Subscriber节点也会接收到消息。

If the Publisher node is running normally, the vehicle will move and the Subscriber node will also receive the message.

4.4 ROS常用组件

4.4 ROS common components

4.4.1 launch启动文件

4.4.1Launch File

启动文件(LaunchFile)便是ROS中一种同时启动多个节点的途径，它还可以自动启动ROS Master节点管理器，并且可以实现每个节点的各种配置，为多个节点的操作提供很大便利。

The Launch File is a way to launch multiple nodes at the same time in ROS. It can also automatically start the ROS Master, and can realize various configurations of each node, which provides great convenience for the operation of multiple nodes.

（1） launch 标签

(1) Launch tag

launch标签就像一个有容乃大的括号，规定了一片区域，所有的launch文件都由< launch >开头，由< /launch>结尾，所有的描述标签都要写在< launch>< /launch>之间
The launch tag is like a generous bracket that defines an area. All launch files start with <launch> and end with </launch>. All description tags must be written between <launch> and < /launch>

```
<launch>
.....
.....
</launch>
```

（2） node 标签

(2) Node tag

node标签可以说是launch文件里最常见的标签了，每个node标签里包括了ROS图中节点的名称属性name、该节点所在的包名pkg以及节点的类型type，常见的使用方式如下：
The node tag can be said to be the most common tag in the launch file. Each node tag includes the name attribute of the node in the ROS graph (name), the package name where the node is located (pkg), and the type of the node (type). The common usage is as follows:

```
<node pkg="package-name" type="executable-name" name="node-name" />
```

标签属性 Tag attribute	属性作用 Attribute function
name="NODE_NAME"	为节点指派名称，这将会覆盖掉ros::init()定义的node_name Assign a name to the node, which will override the node_name defined by ros::init()
pkg="PACKAGE_NAME"	节点所在的功能包名称 The name of the package where the node is located
type="FILE_NAME"	定义节点的可执行文件名称 Define the executable file name of the node
output="screen"	将节点的标准输出打印到终端屏幕，默认输出为日志文档。 Print the standard output of the node to the terminal screen, and the default output is a log file.
respawn="true"	复位属性，该节点停止时，会自动重启，默认为false。 Reset the properties. When the node stops, it will automatically restart. The default is false.
ns = "NAME_SPACE"	命名空间，为节点内的相对名称添加命名空间前缀。 Namespace, which adds a namespace prefix to the relative name in the node.
args="arguments"	节点需要的输入参数。 Input parameters required by the node.

（3） include 标签

(3) Include tag

该标签可以导入另一个roslaunch XML文件到当前文件。

This tag can import another roslaunch XML file to the current file.

标签属性 Tag attribute	属性作用 Attribute function
file ="$(find pkg-name)/path/filename.xml"	指明想要包含进来的文件 Specify the files you want to include

使用起来就像下面这样

Use it like this

```
<include file="$(find demo)/launch/demo.launch" />
```

（4） remap 标签

(4) Remap tag

remap标签顾名思义重映射，ROS支持topic的重映射，remap标签里包含一个original-name和一个new-name，及原名称和新名称。

As the name implies, the remap tag is remapping. ROS supports topic remapping. The remap tag contains an original-name and a new-name.

比如现在你拿到一个节点，这个节点订阅了"/chatter"topic，然而你自己写的节点只能发布

到"/demo/chatter"topic，由于这两个topic的消息类型是一致的，你想让这两个节点进行通讯，那么可 以在launch文件中这样写：
For example, now you get a node that subscribes to the "/chatter" topic, but the node you write can only be published to the "/demo/chatter" topic. Since the message types of these two topics are the same, if you want these two nodes to communicate, you can write this in the launch file:

```
<remap from="chatter" to="demo/chatter"/>
```

这样就可以直接把/chattertopic重映射到/demo/chatter，这样子不用修改任何代码，就可以让两个节 点进行通讯。
In this way, you can directly remap /chattertopic to /demo/chatter, so that the two nodes can communicate without modifying any code.

（5） param标签

(5) Param tag

param标签的作用相当于命令行中的rosparam set。

The role of the param tag is equivalent to the rosparam set in the command line.

比如现在在参数服务器中添加一个名为demo_param，值为1.0的参数

For example, now add a parameter named demo_param with a value of 1.0 in the parameter server

```
<param name="demo_param" type="int" value="1.0"/>
```

（6） rosparam 标 签

(6) Rosparam tag

rosparam标签允许从YAML文件中一次性导入大量参数。

The rosparam tag allows a large number of parameters to be imported from the YAML file at once.

```
<rosparam command="load" file="$(find pkg-name)/path/name.yaml"/>
```

使用方式如下：

The usage is as follows:

（7） arg 标签

(7) Arg tag

argument是另外一个概念，类似于launch文件内部的局部变量，仅限于launch文件使用，便于launch文件的重构，与ROS节点内部的实现没有关系。

Argument is another concept, similar to the local variables in the launch file. It is limited to the launch file and facilitates the reconstruction of the launch file. It has nothing to do with the internal implementation of the ROS node.

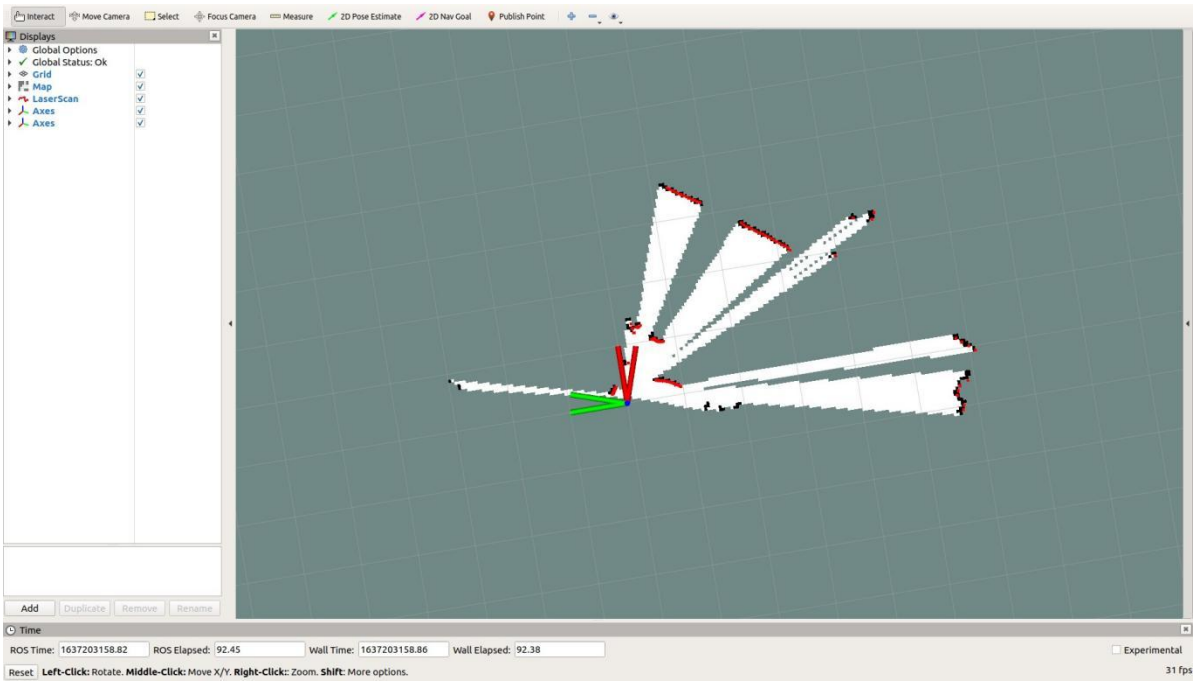
设置argument使用arg标签元素，语法如下：

Set argument to use the arg tag element, and the syntax is as follows:

```
<arg name="arg-name" default= "arg-value"/>
```

4.4.2 Rviz

rviz是一款三维可视化工具,很好地兼容了各种基于ROS软件框架的机器人平台。在rviz中,可以使用XML对机器人、周围物体等任何实物进行尺寸、质量、位置、材质、关节等属性的描述,并且在界面中呈现出来。同时,rviz还可以通过图形化方式,实时显示机器人传感器的信息、机器人的运动状态、周围环境的变化等。总而言之,rviz可以帮助开发者实现所有可监测信息的图形化显示,开发者也可以在rviz的控制界面下,通过按钮、滑动条、数值等方式控制机器人的行为。下图为构建地图过程中，rviz图形化显示的栅格 地图和激光数据。



Rviz is a 3D visualization tool, which is well compatible with various robot platforms based on ROS software framework. In rviz, you can use XML to describe the size, quality, position, material, joints and other attributes of any physical objects such as robots and surrounding objects, and present them in the interface. At the same time, rviz can also graphically display the information of the robot's sensors, the robot's motion status, and the changes in the surrounding environment in real time. All in all, rviz can help developers realize the graphical display of all monitorable information. Developers can also control the behavior of the robot through buttons, sliders, and values under the control interface of rviz. The following figure shows the raster map and laser data graphically displayed by rviz during the map building.

4.4.3 Qt工具箱

4.4.3 Qt toolbox

计算图可视化工具(rqt_graph)

Computation graph visualization tool (rqt_graph)

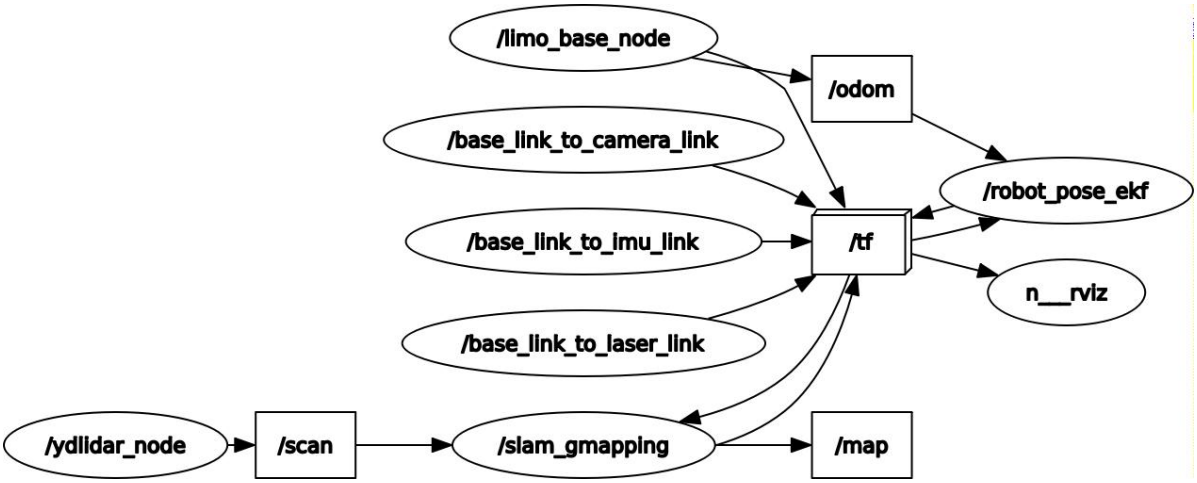
rqt_graph工具可以图形化显示当前ROS系统中的计算图。在运行建图功能时,使用如下命令即可启动该 工具:

The rqt_graph tool can graphically display the computation graph in the current ROS system. When running the mapping function, use the following command to launch the tool:

```
rqt_graph
```

启动成功后的计算图显示如下图所示。

The computation graph after successful launch is shown in the figure below.



TF关系可视化工具（rqt_tf_tree）

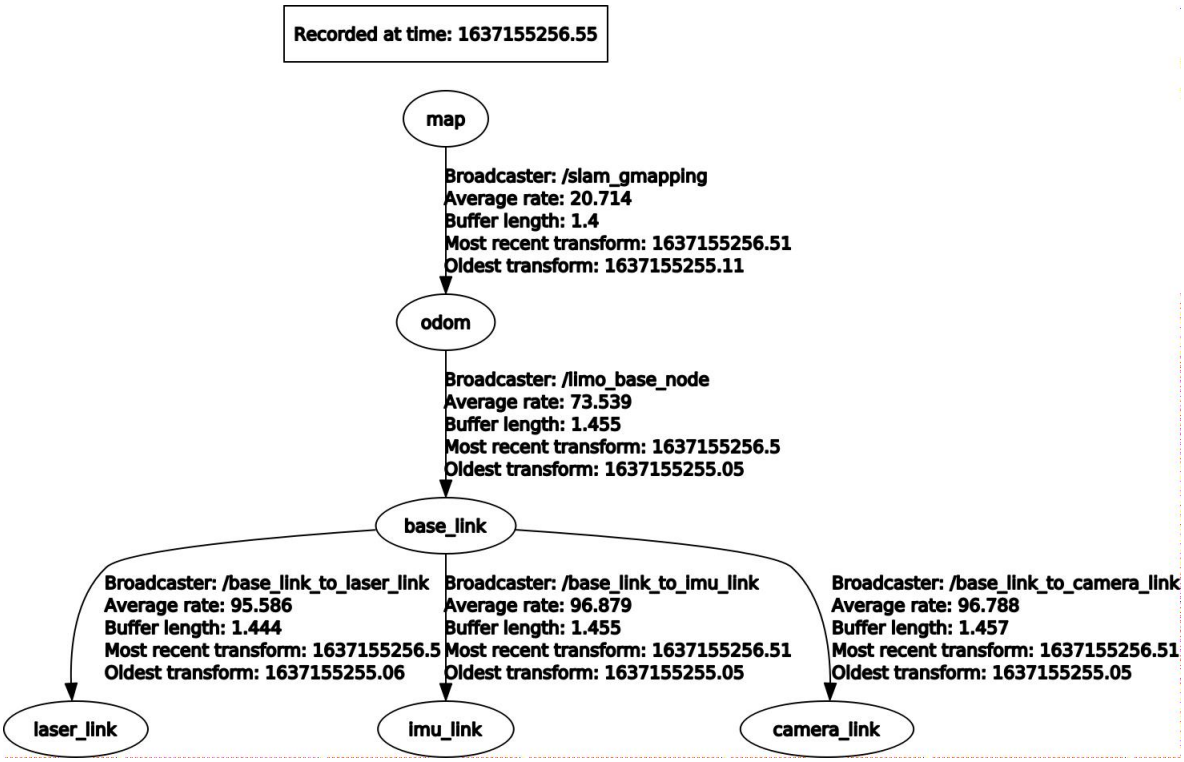
TF relationship visualization tool (rqt_tf_tree)

rqt_tf_tree工具可以图形化显示当前运行节点之间的TF关系，在运行建图功能时，使用如下命令即可启 动该工具：

The rqt_tf_tree tool can graphically display the TF relationship between the currently running nodes. When running the mapping function, use the following command to launch the tool:

```
rosrun rqt_tf_tree rqt_tf_tree
```

启动成功后的TF关系图显示如下图所示



The TF relationship diagram after successful launch is shown in the figure below

附录5、系统烧录

Appendix 5. System Burning

5.1 下载安装balenaetcher

5.1 Download and install balenaetcher

在个人PC电脑上下载安装balenaetcher，下载链接：https://www.balena.io/etcher/，根据个人电脑的 操作系统和架构下载不同的版本。

Download and install balenaetcher on a PC; download link: https://www.balena.io/etcher/; download different versions according to the operating system and architecture of the PC.

5.2 下载需要烧录的镜像

5.2 Download the image to be burned

这里以我司提供的官方镜像为例，使用百度云盘下载镜像，下载链接为：

Here is an example of the official image provided by our company. Use Baidu cloud disk to download the image, and the download link is:

5.3 软件使用说明

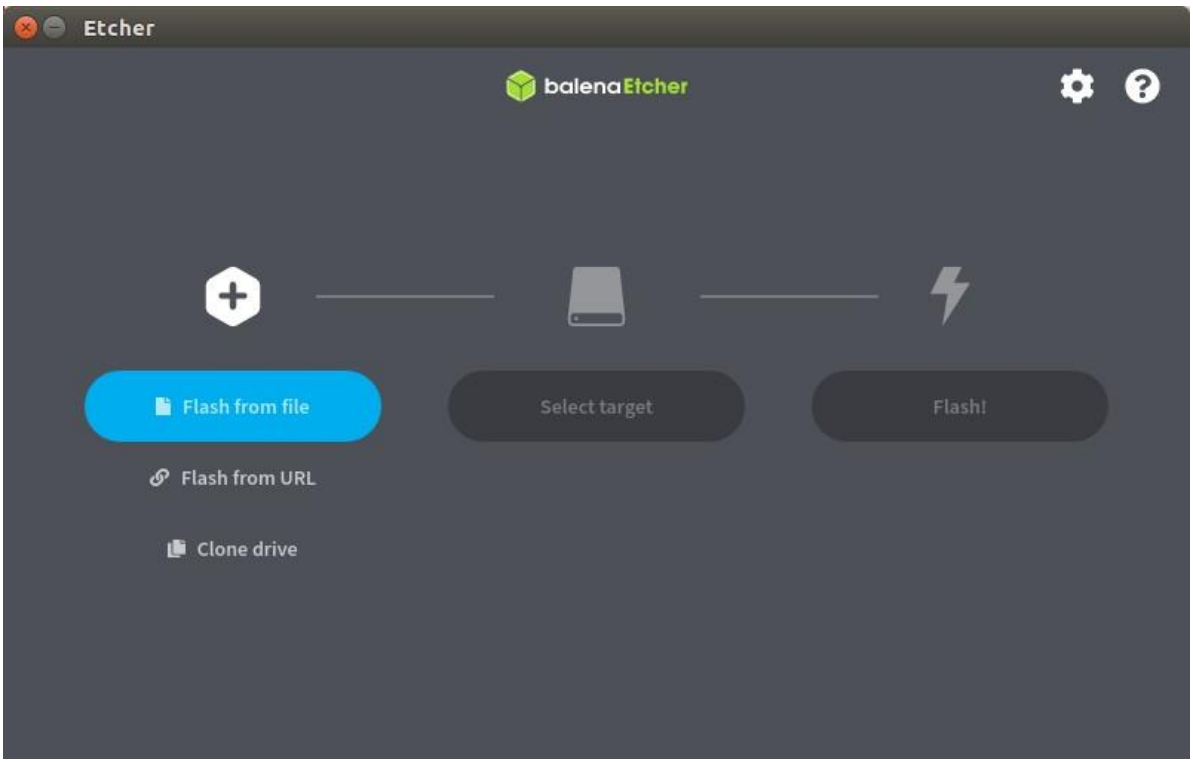
5.3 Instructions on software usage

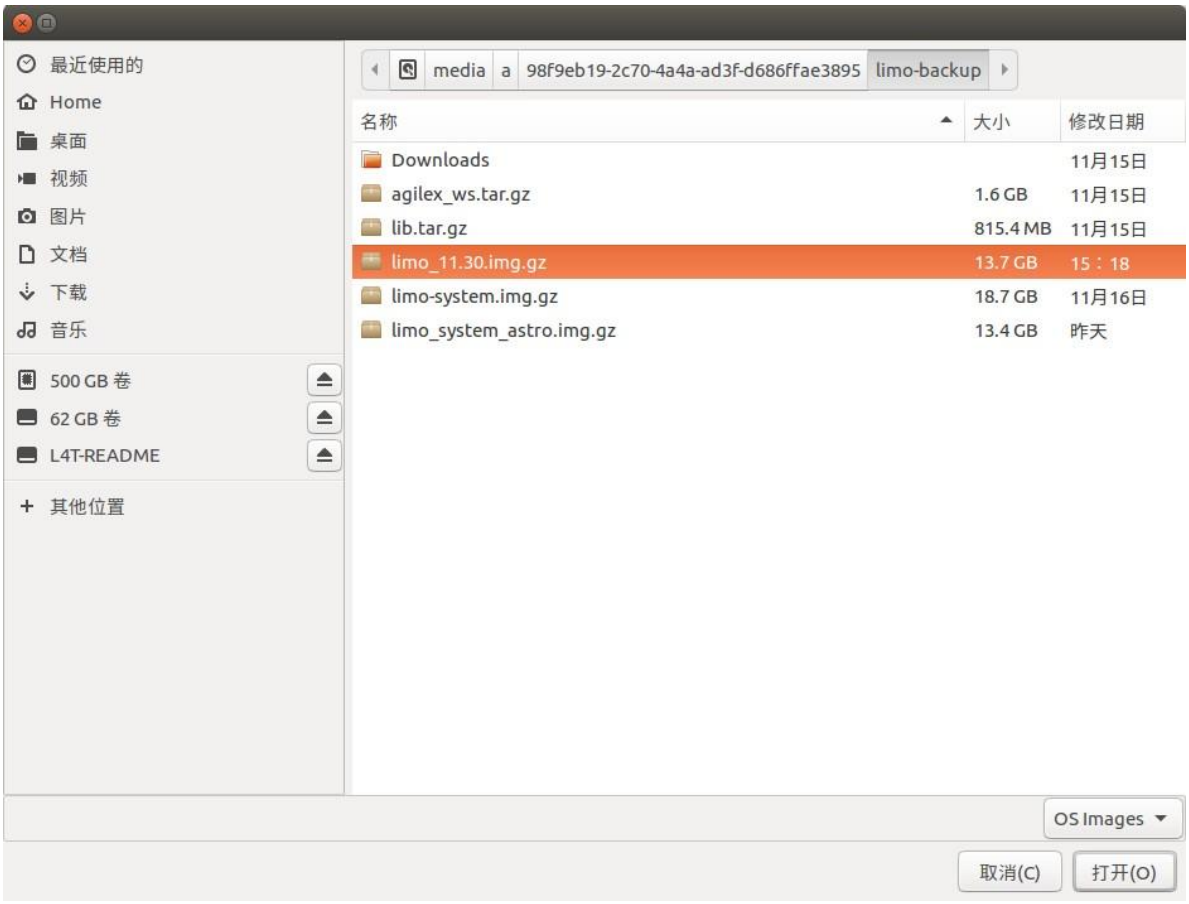
软件使用说明以在Linux系统下为例子。

The instructions on software usage in the Linux system are taken as an example.

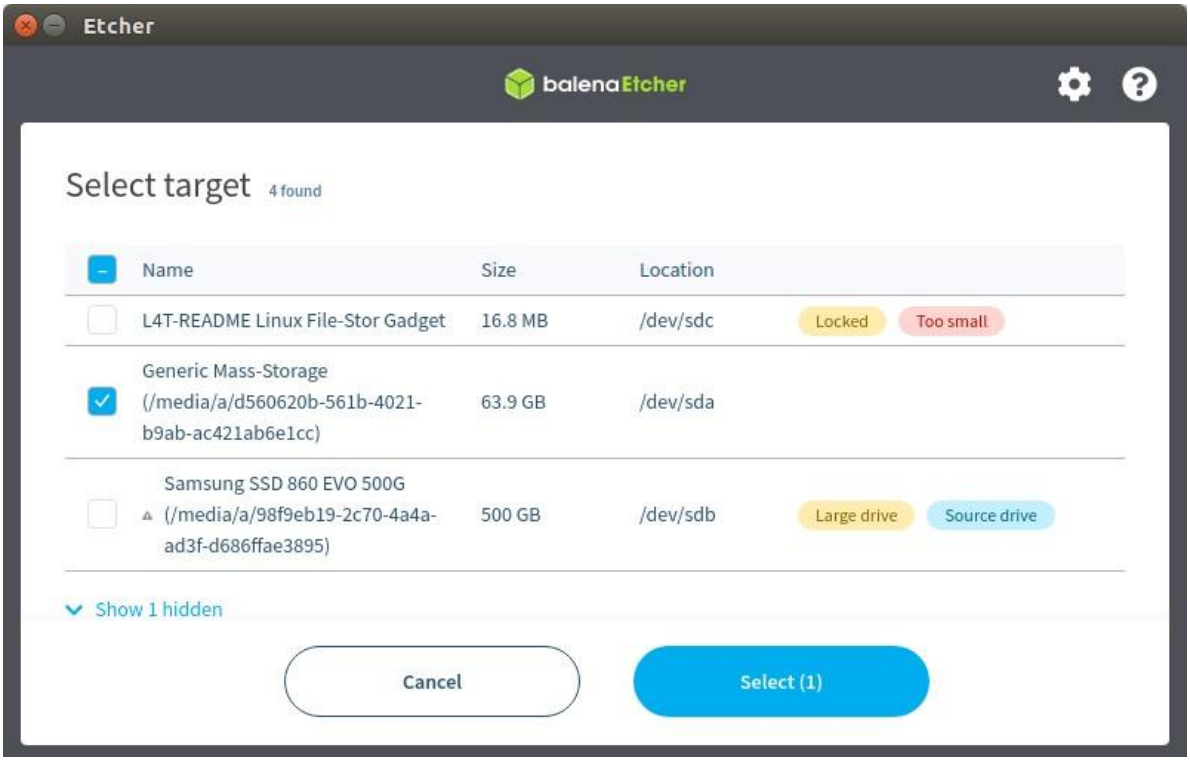
1、打开软件，选择需要烧录的镜像

1. Open the software and select the image to be burned





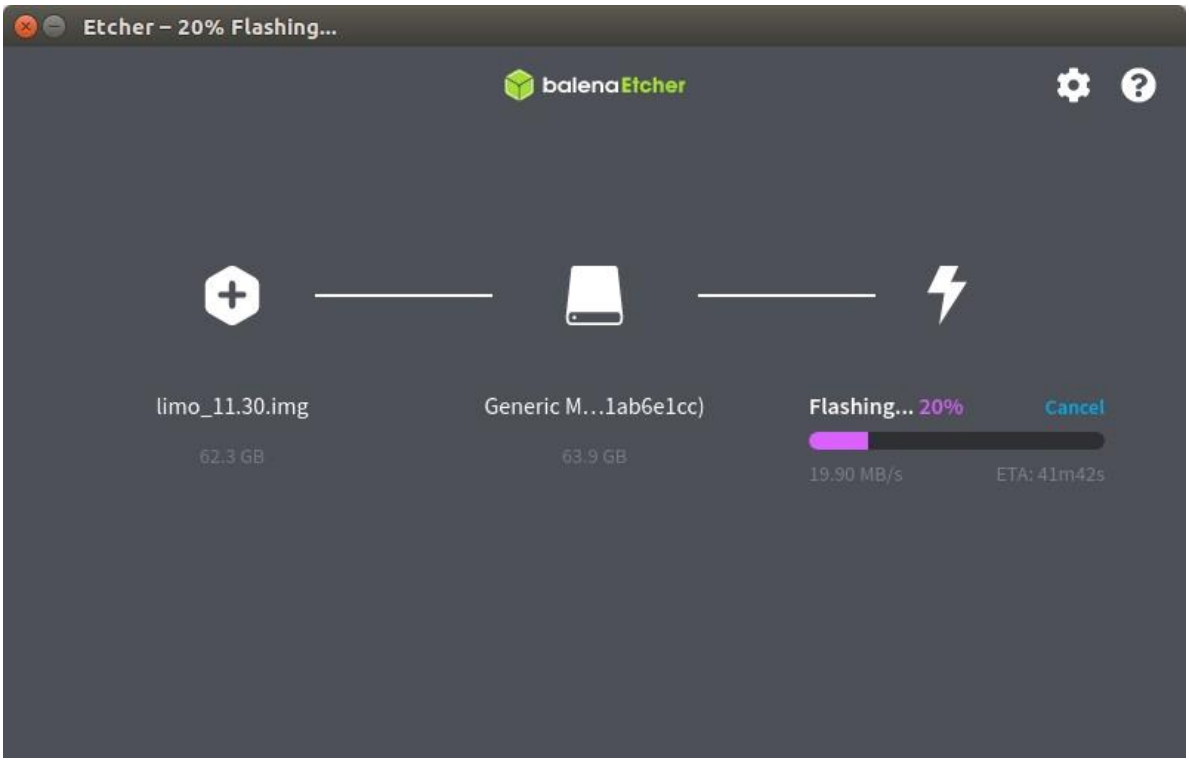
2、选择需要进行烧录的SD卡



2. Select the SD card that needs to be burned

3、点击Flash，开始烧录

3. Click Flash to start burning



附录 6、固件升级

Appendix 6. Firmware Upgrade

固件升级的软件在limo的主目录下，文件名称为LimonTest_Nano，升级所需要的固件在我司官方GitHub中，链接为：https://github.com/agilexrobotics/limo-doc
The firmware upgrade software is in the main directory of limo, and the file name is LimonTest_Nano. The firmware required for the upgrade is in our official GitHub, and the link is: https://github.com/agilexrobotics/limo-doc

注：如limo中的LimonTest_Nano不能正常打开，请到我司官方GitHub中下载最新的软件，链接为：https://github.com/agilexrobotics/limo-doc

Note: If LimonTest_Nano in limo cannot be opened normally, please download the latest software from our official GitHub, and the link is: https://github.com/agilexrobotics/limo-doc

下载完所需要升级的固件，接下来开始操作步骤：

After downloading the firmware that needs to be upgraded, then start the operation steps:

1、进入固件升级模式

1. Enter the firmware upgrade mode

关机状态下连续按两下开机键进入固件升级模式，当开机键闪烁时，成功进入固件升级模式，等待几秒钟之后，nano会正常启动。

In the shutdown state, press the power button twice to enter the firmware upgrade mode. When the power button flashes, it enters the firmware upgrade mode successfully. After a few seconds, nano will start normally.

2、赋予LimonTest_Nano软件运行权限

2. Grant LimonTest_Nano software running permissions

打开终端，在终端中输入命令：

Open the terminal and enter the command in the terminal:

```
chmod +x LimonTest_Nano
```

3、启动软件，开始升级固件

3. Launch the software and start to upgrade the firmware

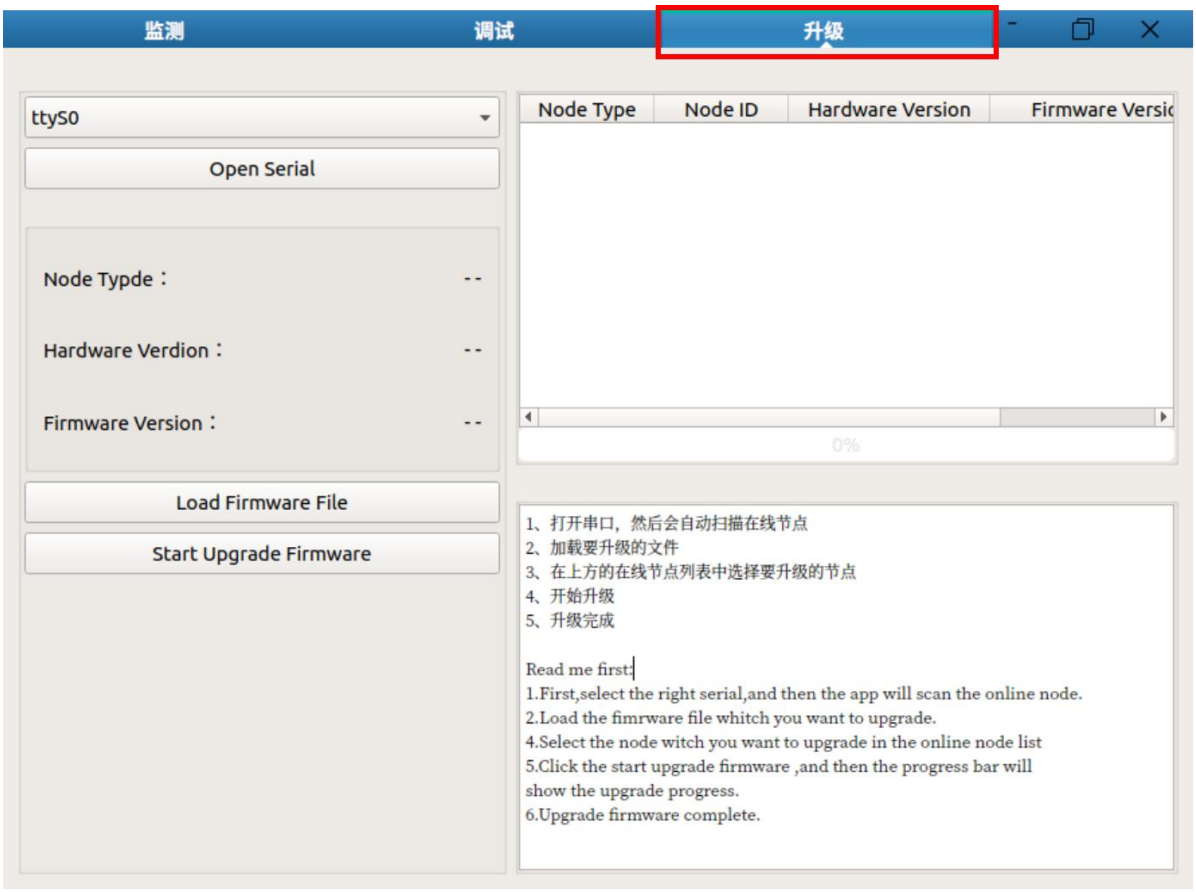
在终端中输入命令：

Enter the command in the terminal:

```
./LimonTest_Nano
```

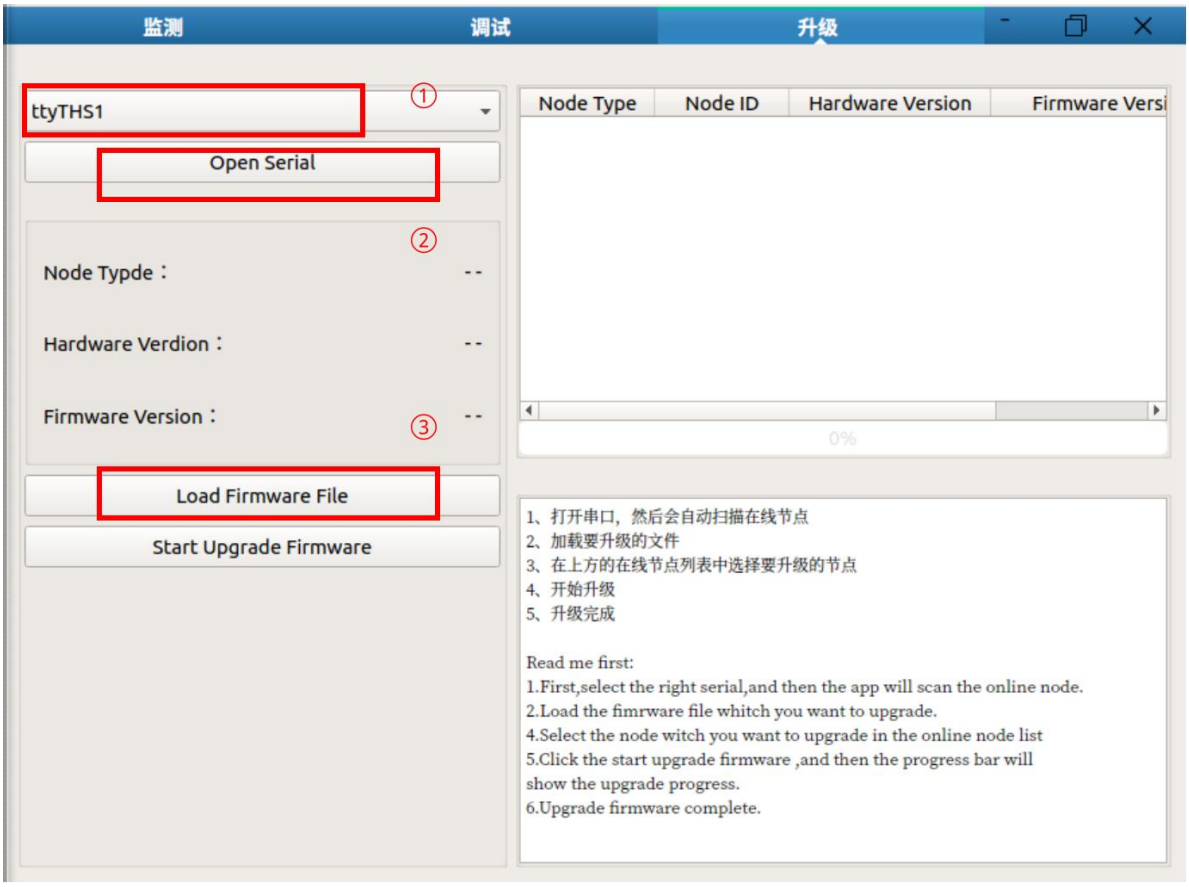
软件成功打开之后，点击升级按钮，显示的画面如下图所示：

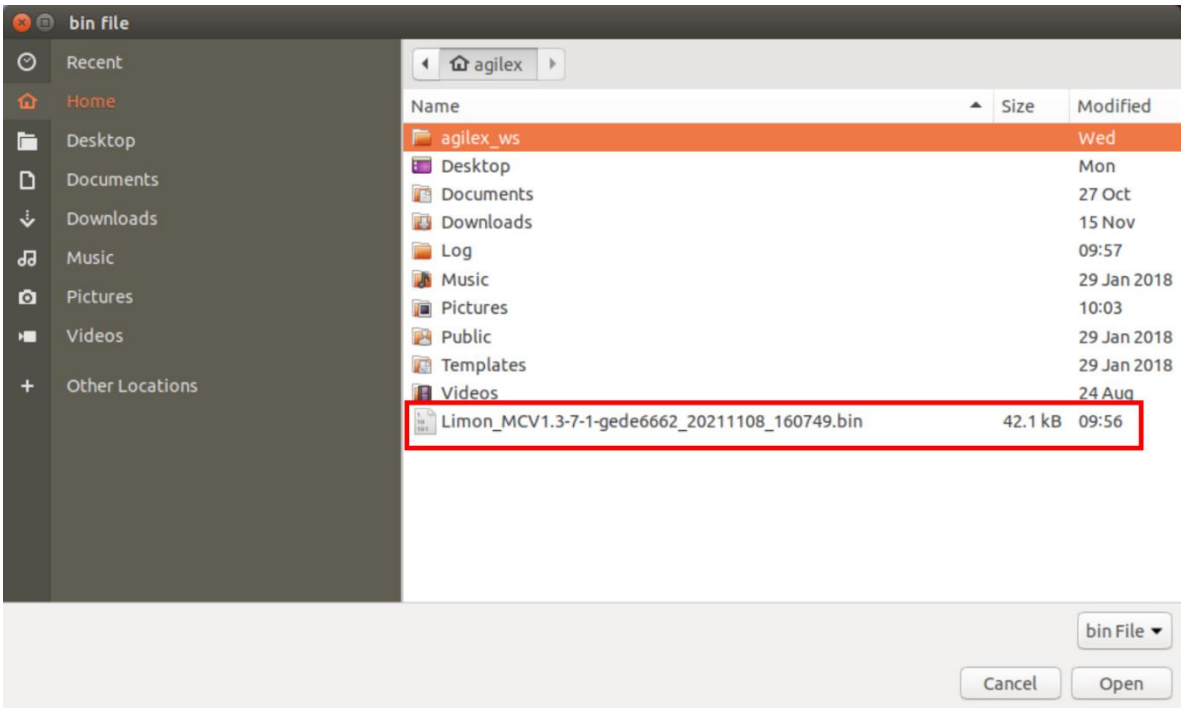
After the software is successfully opened, click the upgrade button, and the displayed screen is as shown in the figure below:



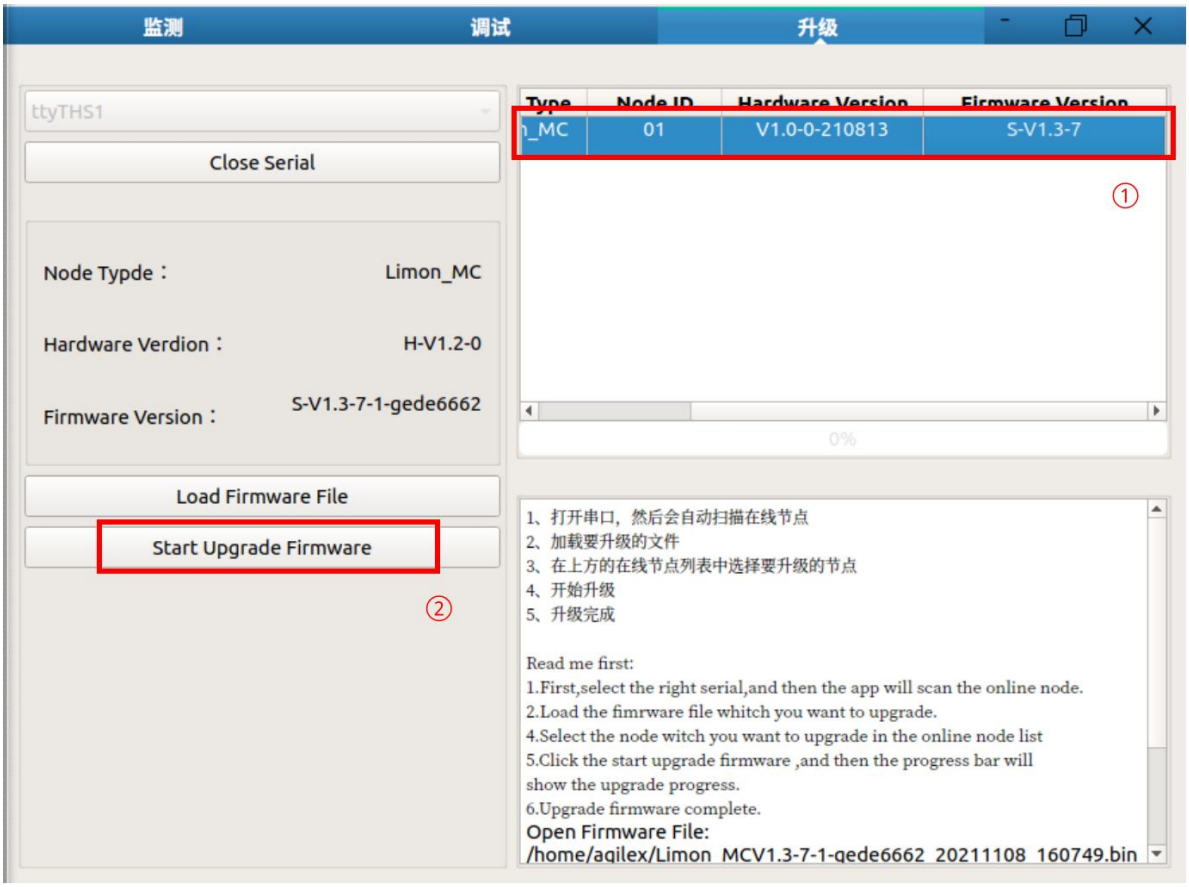
选择对应的串口，一般情况下，选择串口ttyTHS1，点击Open Serial，打开串口，然后点击Load Firmware File 选择需要升级的固件

Select the corresponding serial port; under normal circumstances, select the serial port ttyTHS1; click Open Serial to open the serial port, and then click Load Firmware File to select the firmware to be upgraded





选择固件列表中的固件信息，然后点击Start Upgrade按钮开始固件升级。



Select the firmware information in the firmware list, and then click the Start Upgrade button to start the firmware upgrade.

升级成功，点击 Close Serial按钮，关闭串口

After the upgrade is successful, click the Close Serial button to close the serial port

附录7、导航功能包参数配置

Appendix 7. Parameter Configuration of Navigation Package

如需要自己尝试调试功能包中的参数，可参考以下列表。

If you need to try to debug the parameters in the package yourself, you can refer to the following list.

7.1 gmapping功能包中可供配置的参数

7.1 Configurable parameters in the gmapping package

注：gmapping功能包的参数的配置文件为：

Note: The parameter configuration file of the gmapping package is:
~/agilex_ws/src/limo_ros/limo_bringup/launch/limo_gmapping.launch

参数 Parameter	类型 Type	默认值 Default	描述 Description
~throttle_scans	int	1	处理的扫描数据门限，默认每次处理1个扫描数据（可以设置更大跳过一些扫描数据） The scan data threshold to be processed; the default is to process 1 scan data at a time (it can be set larger to skip some scan data)
~base_frame	string	base_link	机器人基坐标系 Robot base coordinate system
~map_frame	string	map	地图坐标系 Map coordinate system
~odom_frame	string	odom	里程计坐标系 Odometer coordinate system
~map_update_interval	float	5.0	地图更新频率 Map update frequency
~maxUrange	float	80	探测最大可用范围，即光束能到达的范围 Detect the maximum available range, that is, the range that the beam can reach
~sigma	float	0.05	端点匹配的标准差 Standard deviation of endpoint matching
~kernelSize	int	1	用于查找对应的kernel size Used to find the corresponding kernel size
~lstep	float	0.05	平移优化步长 Translation optimization step
~astep	float	0.05	旋转优化步长 Rotation optimization step
~iterations	int	5	扫描匹配迭代次数 Scan matching iterations
~lsigma	float	0.075	似然计算的激光标准差 Laser standard deviation for likelihood calculation
~ogain	float	3.0	似然计算时用于平滑重采样效果 Used for smooth resampling effect during likelihood calculation
~lskip	int	0	每次扫描跳过的光束数. The number of beams skipped in each scan.
~minimumScore	float	0.0	扫描匹配结果的最低值 The lowest value of the scan

			matching result
~srr	float	0.1	平移时里程误差作为平移函数(rho/rho) The mileage error during translation as a translation function (rho/rho)
~srt	float	0.2	平移时的里程误差作为旋转函数 (rho/theta) The mileage error during translation as a rotation function (rho/theta)
~str	float	0.1	旋转时的里程误差作为平移函数 (theta/rho) The mileage error during rotation as a translation function (theta/rho)
~stt	float	0.2	旋转时的里程误差作为旋转函数 (theta/theta) The mileage error during rotation as a rotation function (theta/theta)
~linearUpdate	float	1.0	机器人平移一定距离后处理一次激光数据 The robot translates a certain distance and processes the laser data once
~angularUpdate	float	0.5	机器人旋转一定距离后处理一次激光数据 The robot rotates a certain distance and processes the laser data once
~temporalUpdate	float	-1.0	如果最新扫描处理比更新慢，则处理一次扫描。该值为负数时关闭基于时间的更新 If the latest scan processing is slower than the update, one scan is processed. Turn off time-based updates when the value is negative
~resampleThreshold	float	0.5	基于Neff的重采样阀值 Resampling threshold based on Neff
~particles	int	30	滤波器中粒子数目 Number of particles in the filter

参数 Parameter	类型 Type	默认值 Default	描述 Description
~xmin	float	-100.0	地图x方向初始最小尺寸 The initial minimum size of the map in the x direction
~ymin	float	-100.0	地图y方向初始最小尺寸 The initial minimum size of the map in the y direction
~xmax	float	100.0	地图x方向初始最大尺寸 The initial maximum size of the map in the x direction
~ymax	float	100.0	地图y方向初始最大尺寸 The initial maximum size of the map in the y direction
~delta	float	0.05	地图分辨率 Map resolution
~llsamplerange	float	0.01	似然计算的平移采样距离 The translation sampling distance of likelihood calculation
~llsamplestep	float	0.01	似然计算的平移采样步长 The translation sampling step of likelihood calculation
~lasamplerange	float	0.005	似然计算的角度采样距离 The angle sampling distance of likelihood calculation
~lasamplestep	float	0.005	似然计算的角度采样步长 The angle sampling step of likelihood calculation
~transform_publish_period	float	0.05	TF变换发布时间间隔 TF transform publishing period
~occ_thresh	float	0.25	栅格地图占用率的阀值 The threshold of raster map occupancy rate
~maxRange	float	——	传感器的最大范围 The maximum range of sensor

7.2 cartographer功能包中可供配置的参数

7.2 Configurable parameters in the cartographer package

注：cartographer功能包的参数文件为

Note: The parameter file of the cartographer package is ~/agilex_ws/src/limo_ros/limo_bringup/param/build_map_2d.lua

参数 Parameter	默认值 Default	解析 Analysis
map_frame	map	用于发布submaps的ROS坐标系ID，位姿的父坐标系，通常使用“map” The ID of the ROS coordinate system used to publish submaps, the parent coordinate system of the pose, usually "map"
tracking_frame	base_footprint	由SLAM算法追踪的ROS坐标系ID，如果使用IMU，应该使用其坐标系，通常选择是“imu_link” The ID of the ROS coordinate system tracked by the SLAM algorithm. If IMU is used, its coordinate system should be used, usually "imu_link"
published_frame	odom	用于发布位姿子坐标系的ROS坐标系ID，例如“odom”坐标系，如果一个“odom”坐标系由系统的不同部分提供，在这种情况下，map_frame中的“odom”姿势将被发布。 否则，将其设置为“base_link”可能是合适的 The ID of the ROS coordinate system used to publish the pose sub-coordinate system, like the "odom" coordinate system. If an "odom" coordinate system is provided by different parts of the system, in this case, the "odom" pose in the map_frame will be published. Otherwise, it may be appropriate to set it to "base_link"
odom_frame	odom	在provide_odom_frame为真才启用，坐标系在published_frame和map_frame之间用于 发布局部SLAM结果，通常是“odom” It is enabled when provide_odom_frame is true. The coordinate system is used to publish local SLAM results between published_frame and map_frame, usually "odom"
provide_odom_frame	true	如果启用，局部，非闭环，持续位姿会作为odom_frame发布在map_frame中发布。 If enabled, local, non-closed-loop, and continuous poses will be published as odom_frame in map_frame.
use_odometry	false	如果启用，订阅关于“odom”话题的nav_msgs/Odometry消息。里程信息会提供，这些信息包含在SLAM里 If enabled, subscribe to nav_msgs/Odometry messages on the "odom" topic. The mileage information will be provided, which is included in SLAM
num_laser_scans	1	订阅的激光扫描话题数量。在一个激光扫描仪的“scan”话题上订阅sensor_msgs/LaserScan 或在多个激光扫描仪上订阅话题“scan_1”，“scan_2”等 The number of laser scanning topics subscribed. Subscribe to sensor_msgs/LaserScan on the "scan" topic of one laser scanner or subscribe to the topics "scan_1", "scan_2", etc. on multiple

		laser scanners
num_multi_echo_laser_scans	0	订阅的多回波激光扫描主题的数量。在一个激光扫描仪的“echoes”话题上订阅sensor_msgs/MultiEchoLaserScan或者为多个激光扫描仪订阅话题“echoes_1”，“echoes_2”等。 The number of subscribed multi-echo laser scanning topics. Subscribe to sensor_msgs/MultiEchoLaserScan on the "echoes" topic of a laser scanner or subscribe to the topics "echoes_1", "echoes_2", etc. for multiple laser scanners.
num_subdivisions_per_laser_scan	1	将每个接收到的（多回波）激光扫描分成的点云数。细分扫描可以在扫描仪移动时取消扫描获取的扫描。有一个相应的轨迹构建器选项可将细分扫描累积到将用于扫描匹配的点云中。 The number of point clouds that divide each received (multi-echo) laser scan. The subdivision scan can cancel the scan acquired by the scan when the scanner is moving. There is a corresponding trajectory builder option to accumulate subdivision scans into the point cloud that will be used for scan matching.
num_point_clouds	0	要订阅的点云话题的数量。在一个测距仪的“points2”话题上订阅sensor_msgs/PointCloud2或者为多个测距仪订阅话题“points2_1”，“points2_2”等。 The number of point cloud topics to be subscribed to. Subscribe to sensor_msgs/PointCloud2 on the "points2" topic of a range finder or subscribe topics "points2_1", "points2_2", etc. for multiple range finders.
lookup_transform_timeout_sec	0.2	使用tf2查找变换的超时秒数 The timeout seconds of looking up and transforming with tf2
submap_publish_period_sec	0.3	发布submap的间隔（以秒为单位），例如，0.3秒 The period (in seconds) for publishing submaps, eg.0.3 seconds
pose_publish_period_sec	5e-3	发布姿势的间隔（以秒为单位），例如 5e-3，频率为200 Hz。 The period (in seconds) for publishing poses, eg. 5e-3, with a frequency of 200 Hz.
trajectory_publish_period_sec	30e-3	以秒为单位发布轨迹标记的时间间隔，例如，30e-3持续30毫秒 The period for publishing trajectory tag in seconds, eg. 30e-3, lasting 30 milliseconds

7.3 amcl功能包中可供配置的参数

7.3 Configurable parameters in the amcl package

注：amcl功能包参数配置文件为：amcl_param_diff.yaml（该文件为四轮差速、全向轮、履带运动模式 所用的amcl参数文件）、amcl_param.yaml（该文件为阿克曼运动模式所用的amcl参数文件）

Note: The parameter configuration files of the amcl package are: amcl_param_diff.yaml (the file is the amcl parameter file used in the four-wheel differential, omnidirectional wheel, and track motion modes), and amcl_param.yaml (the file is the amcl parameter file used in the Ackermann motion mode)

参数 Parameter	类型 Type	默认值 Default	描述 Description
min_particles	int	100	最小允许的颗粒数。 The minimum number of particles allowed.
max_particles	int	5000	允许的最大粒子数。 The maximum number of particles allowed.
kld_err	double	0.01	真实分布和估计分布之间的最大误差。 The maximum error between the true distribution and the estimated distribution.
kld_z	double	0.99	（1-p）的上标准正常分位数，其中p是估计的失谐上的误差将小于kld_err的概率。 The upper normal quantile of (1-p), where p is the probability that the error on the estimated detuning will be less than kld_err.
update_min_d	double	0.2米 0.2m	执行过滤器更新之前需要执行平移运动。 A translation movement needs to be performed before performing the filter update.
update_min_a	double	π/ 6.0 radians	执行过滤器更新之前需要执行旋转运动。 A rotation movement needs to be performed before performing the filter update.
resample_interval	int	2	重新采样之前所需的过滤器更新数。 The number of filter updates required before resampling.
transform_tolerance	double	0	将发布的变换后期化的时间，以指示此变换在未来有效。 The time at which the published transformation will be post-processed to indicate that the transformation will be effective in the future.
recovery_alpha_slow	double	0	慢平均权重滤波器的指数衰减率，用于决定何时通过添加随机姿态来恢复。良好的值可能为0.001。 The exponential decay rate of the slow average weight filter is used to decide when to recover by adding random poses. A good value may be 0.001.
recovery_alpha_fast	double	0.0米 0.0m	快速平均权重滤波器的指数衰减率，用于决定何时通过添加随机姿态来恢复。好的值可能为0.1。 The exponential decay rate of the fast average weight filter is used to decide when to recover by adding random poses. A

			good value may be 0.1.
initial_pose_x	double	0.0米 0.0m	初始姿态均（x），用于初始化具有高斯分布的滤波器。 The initial pose average (x), used to initialize the filter with Gaussian distribution.
initial_pose_y	double	0.0弧度 0.0 radian	初始姿态平均值（y），用于初始化具有高斯分布的滤波器。 The initial pose average (y), used to initialize the filter with Gaussian distribution.
initial_pose_a	double	0.5 * 0.5米 0.5 * 0.5 m	初始姿态平均（偏航），用于初始化具有高斯分布的滤波器。 The initial pose average (yaw), used to initialize the filter with Gaussian distribution.
initial_cov_xx	double	0.5 * 0.5米 0.5 * 0.5 m	初始姿态协方差（x * x），用于初始化具有高斯分布的滤波器。 The initial pose covariance (x * x), used to initialize the filter with Gaussian distribution.
initial_cov_yy	double	-1.0 Hz	初始姿态协方差（y * y），用于初始化具有高斯分布的滤波器。 The initial pose covariance (y * y), used to initialize the filter with Gaussian distribution.
initial_cov_aa	double	0.5 Hz	初始姿态协方差（yaw * yaw），用于初始化具有高斯分布的滤波器。 The initial pose covariance (yaw * yaw), used to initialize the filter with Gaussian distribution.
gui_publish_rate	double	FALSE	发布可视化扫描和路径的最大速率（Hz），禁用-1.0。 The maximum rate (Hz) of publishing visual scans and paths. -1.0 is disabled.
save_pose_rate	double	FALSE	在变量~initial_pose_ 和~initial_cov_ 中存储参数服务器的最后估计姿态和协方差的最大速率（Hz）。此保存的姿势将用于后续运行以初始化过滤器。-1.0禁用。 Store the maximum rate (Hz) of the last estimated pose and covariance of the parameter server in the variables ~initial_pose_ and ~initial_cov_. This saved pose will be used in subsequent runs to initialize the filter. -1.0 is disabled.

use_map_topic	bool	-1	当设置为true时，AMCL将订阅地图主题，而不是进行服务调用以接收其地图。 When set to be true, AMCL will subscribe to the map topic instead of making a service call to receive its map.

参数 Parameter	类型 Type	默认值 Default	描述 Description
first_map_only	bool	-1	当设置为true时，AMCL将只使用它订阅的第一个映射，而不是每次接收到新的映射时更新。 When set to be true, AMCL will only use the first mapping it subscribes to instead of updating each time a new mapping is received.

7.4 DWA中可供配置的参数

7.4 Configurable parameters in DWA

注：DWA配置参数文件为~/agilex_ws/src/limo_ros/limo_bringup/param/diff/planner.yaml

Note: The configuration parameter file of DWA is ~/agilex_ws/src/limo_ros/limo_bringup/param/diff/planner.yaml

参数 Parameter	类型 Type	默认值 Default	描述 Description
acc_lim_x	double	2.5	机器人的 x 加速度限制（m/s2） Robot’ s x acceleration limit (m/s2)
acc_lim_y	double	2.5	机器人的 y 加速度限制（m/s2） Robot’ s y acceleration limit (m/s2)
acc_lim_th	double	3.2	机器人的旋转加速度限制（m/s2） Robot’s rotational acceleration limit (m/s2)
max_vel_trans	double	0.55	机器人最大平移速度的绝对值 （m/s） The absolute value of the maximum translational velocity of the robot (m/s)
min_vel_trans	double	0.1	机器人最小平移速度的绝对值（m/s） The absolute value of the minimum translational velocity of the robot (m/s)
max_vel_x	double	0.55	机器人的最大 x 速度（m/s） Robot’ s maximum x velocity (m/s)
min_vel_x	double	0.0	机器人的最小 x 速度（m/s），反向运动时为负 Robot’ s minimum x velocity (m/s), negative when moving in reverse
max_vel_y	double	0.1	机器人的最大 y 速度（m/s） Robot’ s maximum y velocity (m/s)
min_vel_y	double	-0.1	机器人的最小 y 速度（m/s） Robot’ s minimum y velocity (m/s)
max_rot_vel	double	1.0	机器人最大旋转速度的绝对值（rad/s） The absolute value of the maximum rotation velocity of the robot (rad/s)
min_rot_vel	double	0.4	机器人最小旋转速度的绝对值（rad/s） The absolute value of the minimum rotation velocity of the robot (rad/s)
yaw_goal_tolerance	double	0.05	控制器在实现其目标时偏航/旋转的弧度公差 The radian tolerance of the yaw/rotation when the controller achieves its goal
xy_goal_tolerance	double	0.10	实现目标时控制器在 x 和 y 距离内的公差（m/s） The tolerance of the controller in the distance between x and y when achieving the goal (m/s)
latch_xy_goal_tolerance	bool	false	如果目标容差被锁定，如果机器人到达目标 xy 位置，它将简单地旋转到位，即使它在这样做时最终超出了目标容差。 If the goal tolerance is locked, when the

			robot reaches the goal xy position, it will simply rotate into position, even if it eventually exceeds the goal tolerance while doing so.
sim_time	double	1.7	以秒为单位向前模拟轨迹的时间 Time to simulate the trajectory forward in seconds
sim_granularity	double	0.025	在给定轨迹上的点之间采取的步长（m/s） Step taken between points on a given trajectory (m/s)
vx_samples	int	3	探索 x 速度空间时使用的样本数 The number of samples used when exploring the x velocity space
vy_samples	int	10	探索 y 速度空间时使用的样本数 The number of samples used when exploring the y velocity space
vth_samples	int	20	探索 theta 速度空间时使用的样本数 The number of samples used when exploring the theta velocity space
controller_frequency	double	20.0	调用此控制器的频率。如果未在控制器的命名空间中设置，则使用 searchParam 从父命名空间读取参数。与move_base一起使用，这意味着您只需要设置它的“controller_frequency”参数并且可以安全地 不设置这个参数。 Call the controller’ s frequency. If it is not set in the controller's namespace, use searchParam to read the parameters from the parent namespace. Use together with move_base, which means you only need to set its "controller_frequency" parameter and you can safely not set this parameter.
path_distance_bias	double	32.0	控制器应该在多大程度上靠近给定的路径的权重 The weight that how close the controller should be to the given path
goal_distance_bias	double	24.0	控制器应该尝试达到其本地目标的权重，也控制速度 The weight that the controller should try to reach its local goal and it should also control the velocity
occdist_scale	double	0.01	控制器应尝试避开障碍物的权重 The weight that the controller should try to avoid obstacles

参数 Parameter	类型 Type	默认值 Default	描述 Description
forward_point_distance	double	0.325	机器人中心点到放置附加记分点的距离，单位为米 The distance from the center of the robot to the additional scoring point, in meters
stop_time_buffer	double	0.2	机器人在碰撞前必须停止以使轨迹以秒为单位有效的时间量 The amount of time the robot must stop before colliding for the trajectory to be valid, in seconds
scaling_speed	double	0.25	开始缩放机器人足迹的速度的绝对值（m/s） The absolute value of the speed at which the robot's footprint is scaled (m/s)
max_scaling_factor	double	0.2	缩放机器人足迹的最大因素 The biggest factor in scaling a robot's footprint
publish_cost_grid	bool	false	是否发布计划员在计划时将使用的成本网格。当为 true 时，sensor_msgs/PointCloud2 将在 ~/cost_cloud 主题上可用。每个点云代表成本网格，并有一个字段用于每个单独的评分函数组件以及每个单元格的总成本，并将评分参数考虑在内。 Whether the cost grid that the planner will use when planning will be published? When it's true, sensor_msgs/PointCloud2 will be available on the ~/cost_cloud topic. Each point cloud represents a cost grid and has a field for each individual scoring function component and the total cost of each cell, taking the scoring parameters into account.
oscillation_reset_dist	double	0.05	在重置振荡标志之前机器人必须以米为单位移动多远 How far the robot must move in meters before resetting the oscillation tag
prune_plan	bool	true	定义机器人沿路径移动时是否吃掉计划。如果设置为 true，一旦机器人移动超过它们 1 米，点就会从计划的末端掉下来。 Define whether the robot will eat the plan when moving along the path. If it's set to be true, the points will fall from the end of the plan as soon as the robots move more than 1 meter.

7.5 TEB可供配置的参数

7.5 Configurable parameters in TEB

注：TEB参数配置文件为：
~/agilex_ws/src/limo_ros/limo_bringup/param/carlike2/teb_local_planner_params.yaml

Note: The parameter configuration file of TEB is:
~/agilex_ws/src/limo_ros/limo_bringup/param/carlike2/teb_local_planner_params.yaml

参数 Parameter	类型 Type	默认值 Default	描述 Description
acc_lim_x	double	0.5	机器人的最大平移加速度（米/秒^2） Robot's maximum translational acceleration (m/s ^2)
acc_lim_theta	double	0.5	机器人的最大角加速度（弧度/秒^2） Robot's maximum angular acceleration (radian/s ^2)
max_vel_x	double	0.4	机器人的最大平移速度（米/秒） Robot's maximum translational velocity (m/s)
max_vel_x_backwards	double	0.2	机器人向后行驶时的最大绝对平移速度（以米/秒为单位）。 The maximum absolute translational velocity (in m/s) when the robot is traveling backwards.
max_vel_theta	double	0.3	机器人的最大角速度（弧度/秒） Robot's maximum angular velocity (radian/s)
min_turning_radius	double	0.0	汽车机器人的最小转弯半径（对于差速驱动机器人设置为零）。 Automotive robot's minimum turning radius (set to be zero for differential drive robots).
wheelbase	double	1.0	后轴与前轴之间的距离。对于后轮机器人，该值可能为负（仅当cmd_angle_instead_rotvel设置为true时才需要）。 The distance between the rear axle and the front axle. For rear-wheel robots, this value may be negative (only required when cmd_angle_instead_rotvel is set to be true).
cmd_angle_instead_rotvel	bool	false	用相应的转向角[-pi/2,pi/2]代替指令速度消息中的旋转速度。请注意，根据应用程序更改偏航率的语义不是可取的。在这里，它只是符合舞台模拟器所需的输入。ackermann_msgs中的数据类型更合适，但 move_base 不支持。本地规划器本身并不打算发送命令。 Replace the rotation velocity in the command velocity message with the corresponding steering angle [-pi/2, pi/2]. Note that it is not advisable to change the semantics of the yaw rate according to the application. Here, it is only the input required by the stage simulator. The data type in ackermann_msgs is more appropriate, but move_base does not support it. The local planner itself does not intend to send commands.
max_vel_y	double	0.0	机器人的最大扫射速度（非完整机器人应该为零！） Robot's maximum sweep velocity (it should be zero for incomplete robots!)
acc_lim_y	double	0.5	机器人最大扫射加速度 Robot's maximum sweep acceleration
footprint_model/type	double	point	指定用于优化的机器人足迹模型类型。不同的类型是"点"、"圆形"、"线"、"two_circles"和"多边形"。模型的类型会显着影响所需的计算时间。 Specify the type of robot footprint model used for optimization. The different types are "point", "circle", "line", "two_circles" and "polygon". The type of model can significantly affect the required calculation time.
footprint_model/radius	double	0.2	此参数仅与"圆形"类型相关。它包含圆的半径。圆心位于机器人的旋转轴上。 This parameter is only related to the "circle" type. It contains the radius of the circle. The center of the circle is on the rotation axis of the robot.
footprint_model/line_start	double	[-0.3, 0.0]	此参数仅与"行"类型相关。它包含线段的起始坐标。 This parameter is only related to the "line" type. It contains the starting coordinates of the line segment.
footprint_model/line_end	double	[0.3, 0.0]	此参数仅与"行"类型相关。它包含线段的结束坐标。 This parameter is only related to the "line" type. It contains the ending coordinates of the line segment.
footprint_model/front_offset	double	0.2	此参数仅与"two_circles"类型相关。它描述了前圆的中心沿机 器人的 x 轴移动了多少。假设机器人的旋转轴位于 [0,0]。 This parameter is only related to the "two_circles" type. It describes how much the center of the front circle has moved along the x-axis of the robot. Assume that the rotation axis of the robot is located at [0,0].

footprint_model/front_radius	double	0.2	此参数仅与“two_circles”类型相关。它包含前圆的半径。 This parameter is only related to the "two_circles" type. It contains the radius of the front circle.
footprint_model/rear_offset	double	0.2	此参数仅与“two_circles”类型相关。它描述了后圆的中心沿机 器人的负 x 轴移动了多少。假设机器人的旋转轴位于 [0,0]。 This parameter is only related to the "two_circles" type. It describes how much the center of the back circle has moved along the negative x-axis of the robot. Assume that the rotation axis of the robot is located at [0,0].
footprint_model/rear_radius	double	0.2	此参数仅与“two_circles”类型相关。它包含后圆的半径。 This parameter is only related to the "two_circles" type. It contains the radius of the back circle.
footprint_model/vertices	double	[0.25,-0.05]	此参数仅与“多边形”类型相关。它包含多边形顶点列表（每个都是二维坐标）。多边形始终是封闭的：不要在末尾重复第一个顶点。 This parameter is only related to the "polygon" type. It contains a list of polygon vertices (each is a two-dimensional coordinate). Polygons are always closed: do not repeat the first vertex at the end.
is_footprint_dynamic	bool	false	如果为 true，则在检查轨迹可行性之前更新足迹 If it’s true, the footprint is updated before checking the trajectory's feasibility
xy_goal_tolerance	double	0.2	允许到目标位置的最终欧几里得距离（以米为单位） Allowable final Euclidean distance to the goal position (in meters)
yaw_goal_tolerance	double	0.2	允许的最终方向误差（以弧度为单位） Allowable final direction error (in radians)
free_goal_vel	bool	false	去除目标速度约束，使机器人能够以最大速度到达目标 Remove the goal velocity constraint, so that the robot can reach the goal at the maximum velocity
dt_ref	double	0.3	轨迹的所需时间分辨率（轨迹不固定为dt_ref，因为时间分辨 率是优化的一部分，但如果违反dt_ref +-dt_hysteresis，将在迭代之间调整轨迹大小。 The required time resolution of the trajectory (the trajectory is not fixed to dt_ref, because the time resolution is part of the optimization, but if dt_ref +-dt_hysteresis is violated, the trajectory size will be adjusted between iterations.
dt_hysteresis	double	0.1	根据当前时间分辨率自动调整大小的滞后，通常约为 建议使用 dt_ref 的10% The lag that is automatically resized according to the current time resolution, usually about 10% of the recommended dt_ref
min_samples	int	3	最小样本数（应始终大于 2） Minimum number of samples (should always be greater than 2)
global_plan_overwrite_orientation	bool	true	覆盖全局规划器提供的局部子目标的方向（因为它们通常只提供二维路径） Override the direction of the local sub-goals provided by the global planner (because they usually only provide a two-dimensional path)
global_plan_viapoint_sep	double	-0.1 (disabled))	如果为正，则从全局计划中提取通孔点（路径跟随模式）。该值决定了参考路径的分辨率（沿全局平面每两个连续通孔点之间的最小间隔，如果为负 If it is positive, the via points are extracted from the global plan (path following mode). This value determines the resolution of the reference path (the minimum period between every two consecutive via points along the global plane, if it is negative
max_global_plan_lookahead_dist	double	3.0	指定考虑优化的全局计划子集的最大长度（累积欧几里德距离）。实际长度由本地代价图大小和这个最大界限的逻辑结合决定。设置为零或负数以取消激活此限制。 Specify the maximum length (cumulative Euclidean distance) of the subset of the global plan considered for optimization. The actual length is determined by the logical combination of the size of the local cost map and this maximum limit. Set to be zero or a negative number to deactivate this limit.
force_reinit_new_goal_dist	double	1.0	如果先前的目标更新间隔大于指定值（以米为单位），则重新初始化轨迹（跳过热启动） If the previous goal update period is greater than the

			specified value (in meters), re-initialize the trajectory (skip hot start)
feasibility_check_no_poses	bool	4	指定每个采样间隔应检查预测计划上的哪个姿势的可行性。 Specify that the feasibility of the pose on the prediction plan should be checked during each sampling period.

参数 Parameter	类型 Type	默认值 Default	描述 Description
publish_feedback	bool	false	发布包含完整轨迹和活动障碍物列表的规划器反馈（应仅在评估或调试时启用）。请参阅上面的出版商列表。 Publish planner feedback with complete trajectory and active obstacle list (it should only be enabled for evaluation or debugging). See the publisher list above.
shrink_horizon_backup	bool	true	允许规划者在自动检测到问题（例如不可行性）的情况下临时缩小范围 (50%)。另见参数 shrink_horizon_min_duration。 Allow the planner to temporarily shrink the scope (50%) in the event that a problem (such as infeasibility) is automatically detected. See also parameter shrink_horizon_min_duration.
allow_init_with_backwards_motion	bool	false	如果为 true，则可能会使用向后运动初始化基础轨迹，以防目标在本地成本图中的起点后面（仅当机器人配备有后部传 感器时才推荐这样做）。 If it's true, the base trajectory may be initialized with backward motion in case the goal is behind the starting point in the local cost map (this is only recommended if the robot is equipped with a rear sensor).
exact_arc_length	double	false	如果为真，规划器在速度、加速度和转弯率计算中使用精确的弧长（-> 增加的 CPU 时间），否则使用欧几里得近似。 If it's true, the planner uses the precise arc length (-> increased CPU time) in the calculation of velocity, acceleration and turn rate, otherwise it uses the Euclidean approximation.
shrink_horizon_min_duration	double	10.0	如果检测到不可行的轨迹，请指定缩小地平线的最短持续时间（请参阅参数shrink_horizon_backup以激活缩小地平线模式）。 If an infeasible trajectory is detected, please specify the shortest duration for shrinking the horizon (see the parameter shrink_horizon_backup to activate the shrink horizon mode).
min_obstacle_dist	double	0.5	与障碍物的最小期望距离（以米为单位） The minimum expected distance to the obstacle (in meters)
include_costmap_obstacles	double	true	指定是否应考虑本地成本图的障碍。每个标记为障碍物的单元格都被视为一个点障碍物。因此不要选择非常小的代价地图分辨率，因为它会增加计算时间。在未来的版本中，这种情况将得到解决，并为动态障碍提供额外的 api。 Specify whether or not the obstacles of the local cost map should be considered. Each cell marked as an obstacle is treated as a point obstacle. Therefore, do not choose a very small cost map resolution, because it will increase the calculation time. In future versions, this situation will be resolved and additional api will be provided for dynamic obstacles.
costmap_obstacles_behind_robot_dist	bool	1.0	限制在机器人后面进行规划时考虑到的占用的本地成本图障碍（以米为单位指定距离）。 Limit the occupied local cost map obstacles that are taken into account when planning behind the robot (specify the distance in meters) .
obstacle_poses_affected	double	30	每个障碍物位置都附加到轨迹上最近的姿势以保持距离。也可以考虑额外的邻居。请注意，此参数可能会在未来版本中删除，因为在 kinetic+ 中已修改了障碍关联策略。参考 legacy_obstacle_association的参数说明。 Each obstacle position is attached to the nearest pose on the trajectory to maintain the distance. You can also consider additional neighbors. Please note that this parameter may be removed in a future version because the obstacle association strategy has been modified in kinetic+. Refer to the parameter description of legacy_obstacle_association.
inflation_dist	double	pre kinetic	具有非零惩罚成本的障碍物周围的缓冲区（应大于 min_obstacle_dist才能生效）。另请参阅权重 weight_inflation。 Buffer around obstacles with non-zero penalty cost (should be greater than min_obstacle_dist to take effect). See also weight

			weight_inflation.
include_dynamic_obstacles	string	false	如果此参数设置为 true，则在优化过程中通过恒速模型预测和考虑具有非零速度的障碍物的运动（通过用户提供的障碍 物~/obstacles 提供或从costmap_converter获得）。 新的 If this parameter is set to be true, the motion of obstacles with non-zero velocity will be predicted and considered through the constant velocity model during the optimization process (provided through user-provided obstacles or obtained from the costmap_converter). New
legacy_obstacle_association	bool	false	连接轨迹姿势与优化障碍的策略已被修改（参见变更日志）。您可以通过将此参数设置为true来切换到旧的/以前的 策略。旧策略 The strategy for connecting trajectory poses and optimizing obstacles has been revised (see change log). You can switch to the old/previous strategy by setting this parameter to be true. Old strategy
obstacle_association_force_inclusion_factor	double	1.5	非遗留障碍关联策略试图在优化过程中仅将相关障碍与离散化轨迹连接起来。但是指定距离内的所有障碍物都被强制包括在内（作为min_obstacle_dist 的倍数）。例如，选择 2.0 以在2.0* min_obstacle_dist的半径内强制考虑障碍。[仅当参数legacy_obstacle_association为false时才使用此参数] The non-legacy obstacle association strategy tries to connect only the relevant obstacles with the discretized trajectory in the optimization process. But all obstacles within the specified distance are forcibly included (as a multiple of min_obstacle_dist). For example, choose 2.0 to force the consideration of obstacles within a radius of 2.0*min_obstacle_dist. [This parameter is used only when the parameter legacy_obstacle_association is false]
obstacle_association_cutoff_factor	int	5	见obstacle_association_force_inclusion_factor，但超出[值]的倍数* min_obstacle_dist一切障碍，在优化过程中被忽略。参数障碍_association_force_inclusion_factor被首先处理。[仅当参数legacy_obstacle_association为false时才使用 此参数] See obstacle_association_force_inclusion_factor, but all obstacles that exceed the multiple of [value] * min_obstacle_dist are ignored in the optimization process. The parameter obstacle _association_force_inclusion_factor is processed first. [This parameter is used only when the parameter legacy_obstacle_association is false]
costmap_converter_plugin	int	""	定义插件名称以将成本图单元格转换为点/线/多边形。设置一 个空字符串以禁用转换，以便将所有单元格视为点障碍。 Define the plug-in name to convert cost map cells to points/lines/polygons. Set an empty string to disable conversion so that all cells are treated as point obstacles.
costmap_converter_spin_thread	double	true	如果设置为 true，则成本图转换器在不同的线程中调用其回调队列。 If set to be true, the cost map converter calls its callback queue in a different thread.
costmap_converter_rate	double	5.0	Rate 定义 costmap_converter 插件处理当前成本图的频率（该值不应高于成本图更新率）[以赫兹为单位]。 Rate defines the frequency that the costmap_converter plugin processes the current cost map (This value should not be higher than the cost map update rate) [in Hertz].
no_inner_iterations	double	5	每次外循环迭代中调用的实际求解器迭代次数。请参阅参数no_outer_iterations。 The actual number of solver iterations called in each outer loop iteration. See parameter no_outer_iterations.
no_outer_iterations	double	4	每个外循环迭代都会根据所需的时间分辨率dt_ref自动调整轨 迹大小并调用内部优化器（执行no_inner_iterations）。因此，每个规划周期中求解器迭代的总数是两个值的乘积。 Each outer loop iteration will automatically adjust the trajectory size and call the internal optimizer (execute no_inner_iterations) according to the required time resolution dt_ref. Therefore, the total number of solver iterations in each planning cycle is the product of the two values.
penalty_epsilon	double	0.1	为硬约束逼近的惩罚函数添加一个小的安全裕度 Add a small safety margin to the penalty function of the hard constraint approximation
weight_max_vel_x	double	2.0	满足最大允许平移速度的优化权重 Optimized weight to meet the maximum allowable translational velocity

weight_max_vel_theta	double	1.0	满足最大允许角速度的优化权重 Optimized weight to meet the maximum allowable angular velocity
weight_acc_lim_x	double	1.0	满足最大允许平移加速度的优化权重 Optimized weight to meet the maximum allowable translational acceleration

参数 Parameter	类型 Type	默认值 Default	描述 Description
weight_acc_lim_theta	double	1.0	满足最大允许角加速度的优化权重 Optimized weight to meet the maximum allowable angular acceleration
weight_kinematics_nh	double	1000.0	用于满足非完整运动学的优化权重（此参数必须很高，因为运动学方程构成了等式约束，由于与其他成本相比，“原始”成本值较小，因此即使值为 1000 也不意味着矩阵条件不好）。 The optimized weight used to meet non-holonomic kinematics (this parameter must be very high, because the kinematics equation constitutes an equality constraint, and the "original" cost value is small compared with other costs, so even a value of 1000 does not mean the matrix condition is not good).
weight_kinematics_forward_drive	double	1.0	强制机器人仅选择向前方向（正平移速度）的优化权重。小重量（例如 1.0）仍然允许向后行驶。1000 左右的值几乎可以防止向后行驶（但不能保证）。 The optimized weight to force the robot to select only the forward direction (positive translational velocity). The small weight (eg. 1.0) still allows driving backwards. A value around 1000 can almost prevent backward driving (but it cannot be guaranteed).
weight_kinematics_turning_radius	double	1.0	强制最小转弯半径的优化权重（仅适用于汽车机器人）。 The optimized weight to force the minimum turning radius (only for automobile robots).
weight_optimaltime	double	1.0	缩短轨迹wrt转换/执行时间的优化权重 Optimized weight to shorten trajectory wrt conversion/execution time
weight_obstacle	bool	50.0	与障碍物保持最小距离的优化权重 Optimized weight to keep the minimum distance from the obstacle
weight_viapoint	bool	1.0	用于最小化到通过点的距离的优化权重（相应的参考路径）。0.4 新版本 The optimized weight (corresponding reference path) used to minimize the distance to the passing point. 0.4 new version
weight_inflation	int	0.1	通货膨胀惩罚的优化权重（应该很小）。 The optimized weight of the inflation penalty (should be small).
weight_adapt_factor	double	2.0	在每次外部 TEB 迭代（weight_new = weight_old*factor）中，一些特殊的权重（当前是weight_obstacle）被这个因子 重复缩放。迭代地增加权重而不是设置一个巨大的先验值会导致基础优化问题的更好的数值条件。 In each outer TEB iteration (weight_new = weight_old*factor), some special weights (currently weight_obstacle) are repeatedly scaled by this factor. Iteratively increasing the weights instead of setting a huge prior value will lead to better numerical conditions for the basic optimization problem.
enable_homotopy_class_planning	double	true	在不同的拓扑中激活并行规划（需要更多的 CPU 资源，因为多个轨迹一次被优化） Activate parallel planning in different topologies (requires more CPU resources because multiple trajectories are optimized at once)
enable_multithreading	double	true	激活多个线程以便在不同线程中规划每个轨迹 Activate multiple threads to plan each trajectory in different threads
max_number_classes	bool	4	指定考虑的不同轨迹的最大数量（限制计算工作量） Specify the maximum number of different trajectories to be considered (limits the computational effort)
selection_cost_hysteresis	int	1.0	指定新候选者必须具有与先前选择的轨迹相比多少轨迹成本才能被选中（如果 new_cost < old_cost*factor 则选择）。 Specify how much trajectory cost the new candidate must have compared to the previously selected trajectory to be selected (select if new_cost <old_cost*factor).
selection_obst_cost_scale	double	100.0	仅用于选择“最佳”候选者的障碍成本项的额外缩放。 Additional scaling of the obstacle cost term used only to select the "best" candidate.

selection_viapoint_cost_scale	double	1.0	仅用于选择“最佳”候选者的额外缩放通过点成本条款。0.4 新版本 The additional scaling used only to select the "best" candidate through the point cost clause. 0.4 new version
selection_alternative_time_cost	double	false	如果为真，则时间成本（时间差的平方和）将替换为总转换时间（时间差的总和）。 If it's true, the time cost (sum of squares of the time difference) will be replaced with the total conversion time (sum of the time difference).
roadmap_graph_no_samples	double	15	指定为创建路线图生成的样本数量 Specify the number of samples generated to create the roadmap
roadmap_graph_area_width	bool	6	在起点和目标之间的矩形区域中对随机关键点/航路点进行采样。以米为单位指定该区域的宽度。 Sample the random keypoints/waypoints in the rectangular area between the starting point and the goal. Specify the width of the area in meters.
h_signature_prescaler	bool	1.0	用于区分同伦类的比例内部参数（H-signature）。警告 The internal parameter (H-signature) of the ratio used to distinguish homotopy classes. Warn
h_signature_threshold	double	0.1	如果实部和复部的差异都低于指定的阈值，则假定两个 H 签名相等。 If the differences between the real part and the complex part are lower than the specified threshold, it is assumed that the two H signatures are equal.
obstacle_heading_threshold	string	1.0	指定障碍航向和目标航向之间的标量积的值，以便在探索时将它们（障碍物）考虑在内。 Specify the value of the scalar product between the obstacle course and the goal course so that they (obstacles) are taken into account when exploring.
visualize_hc_graph	string	false	可视化为探索独特轨迹而创建的图形（在 rviz 中检查标记消息） Visualize the graph created to explore the unique trajectory (check the tag message in rviz)
viapoints_all_candidates	bool	true	如果为真，则不同拓扑的所有轨迹都附加到一组通孔点，否则只有与初始/全局计划共享相同拓扑的轨迹与它们连接（对test_optim_node没有影响）。0.4 新版本 If it’s true, all trajectories of different topologies are attached to a set of via points, otherwise only trajectories that share the same topology with the initial/global plan are connected to them (no effect on test_optim_node). 0.4 new version
switching_blocking_period	double	0.0	指定在允许切换到新等价类之前需要到期的持续时间（以秒为单位）。 Specify the duration (in seconds) that needs to expire before being allowed to switch to the new equivalence class.
odom_topic	double	odom	里程计消息的主题名称，由机器人驱动程序或模拟器提供。 The subject name of the odometer message, provided by the robot driver or simulator.
map_frame	bool	odom	全局规划框架（如果是静态地图，这个参数通常必须改为“/map”。 Global planning framework (if it is a static map, this parameter usually must be changed to "/map".