ROBOWORKS

Codebot & Ackerman User Manual

we build human friendly robots

at work, at home, in school & in the field

Education Robots

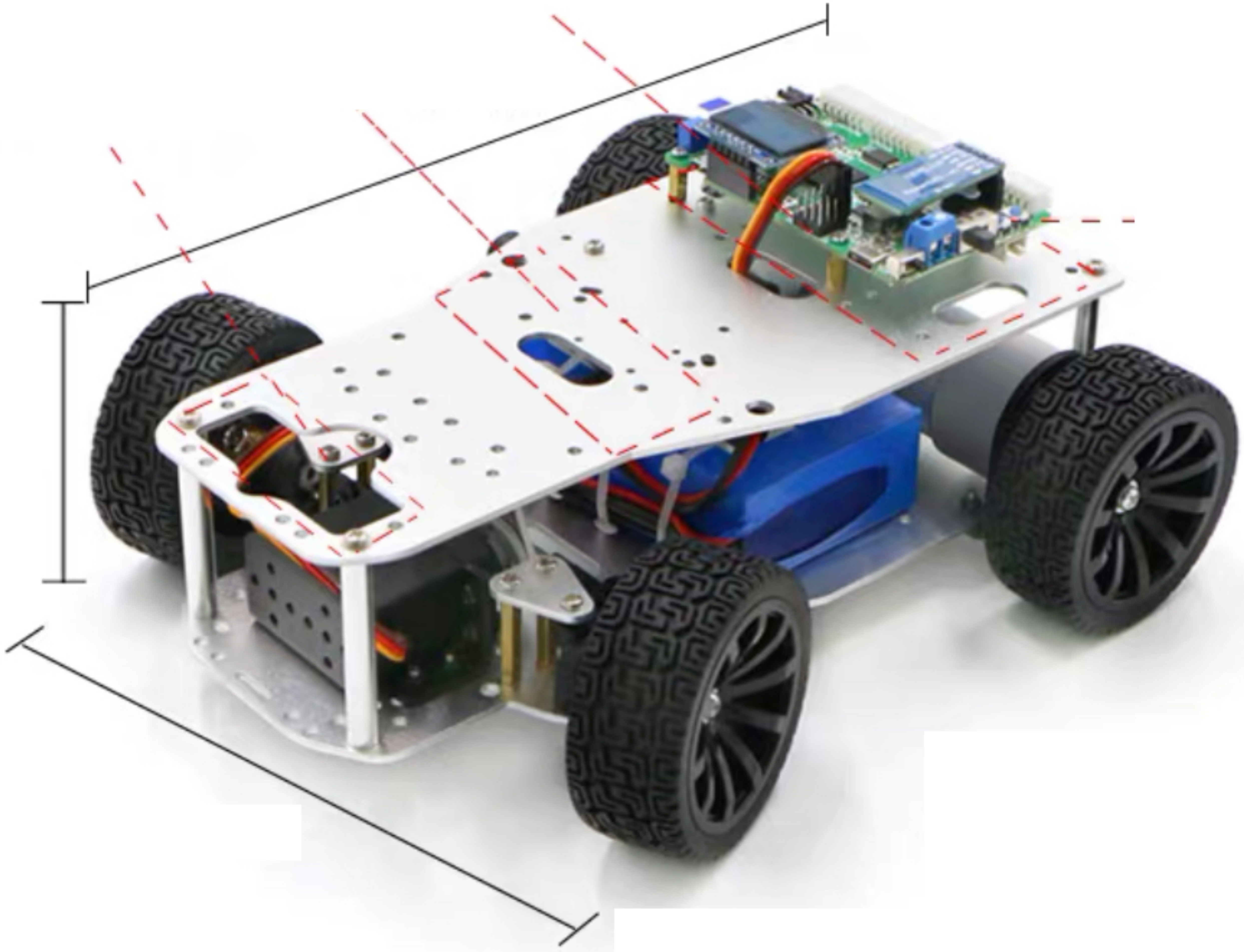
Codebot
for education

- Educational robots based on ROS.
- Ideal for educators and students.
- Affordable, compact and functional.

Codebot Chassis

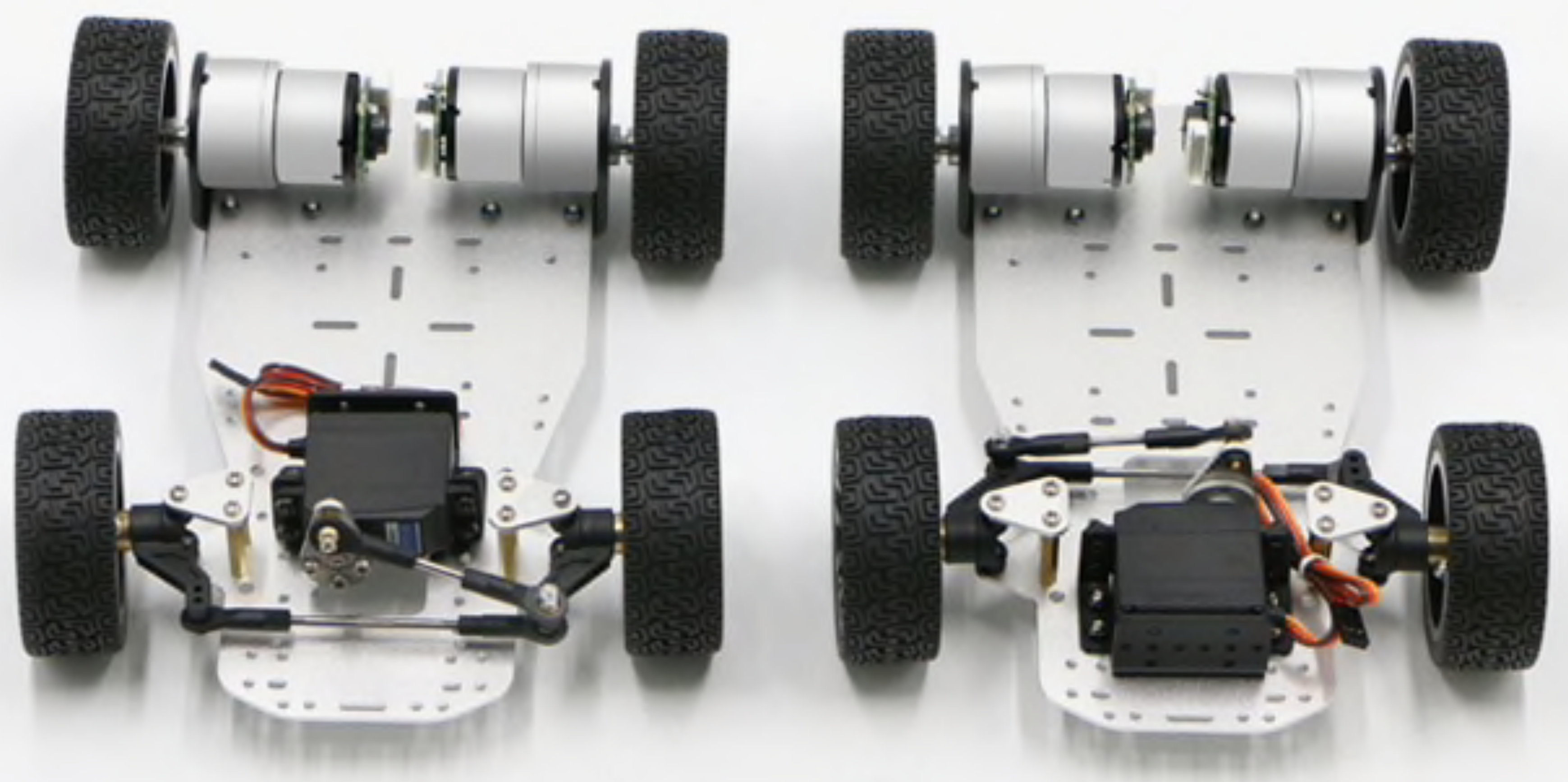
Ackerman

- Robotic chassis ready for ROS development.
- Ready to plug slots for ROS controller, LiDAR and Camera.
- Ackerman wheel system.
- Remote controlled by mobile app.



Variable structure

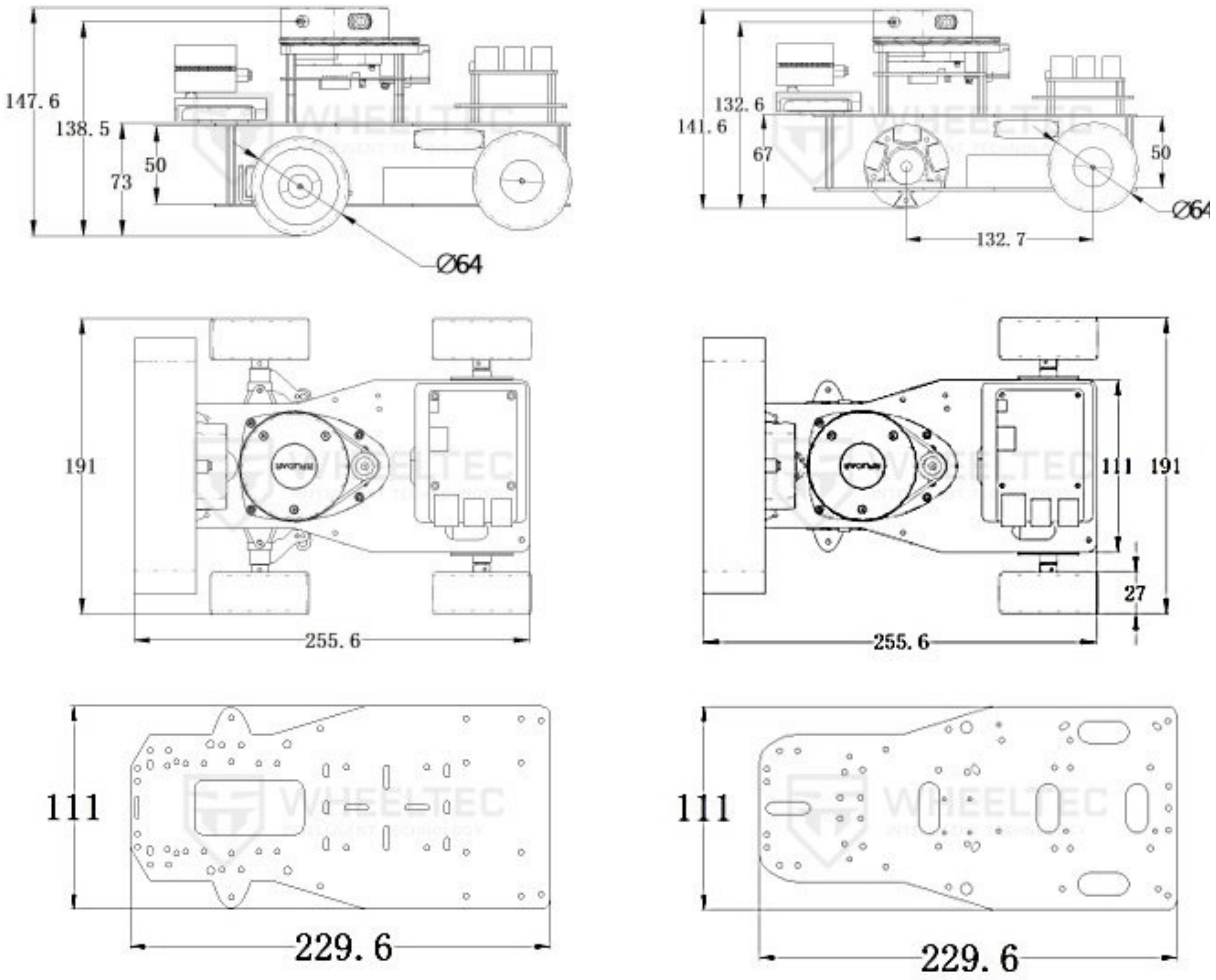
Under normal circumstances, if you need to debug the positive Ackerman structure, the anti-Ackerman structure, and the differential car,It may be necessary to purchase 3 trolleys at the same time. Our trolleys are compatible and replaced through structure and code.It can realize the functions of the above 3 small cars at the same time, which is equivalent to spending the money of one small car and purchasing 3 Trolley



Anti-Ackerman structure | Positive Ackerman structure

Differential structure

Trolley drawings



SLAMTEC RPLIDAR A1 Series

Equipped with new genuine Lidar

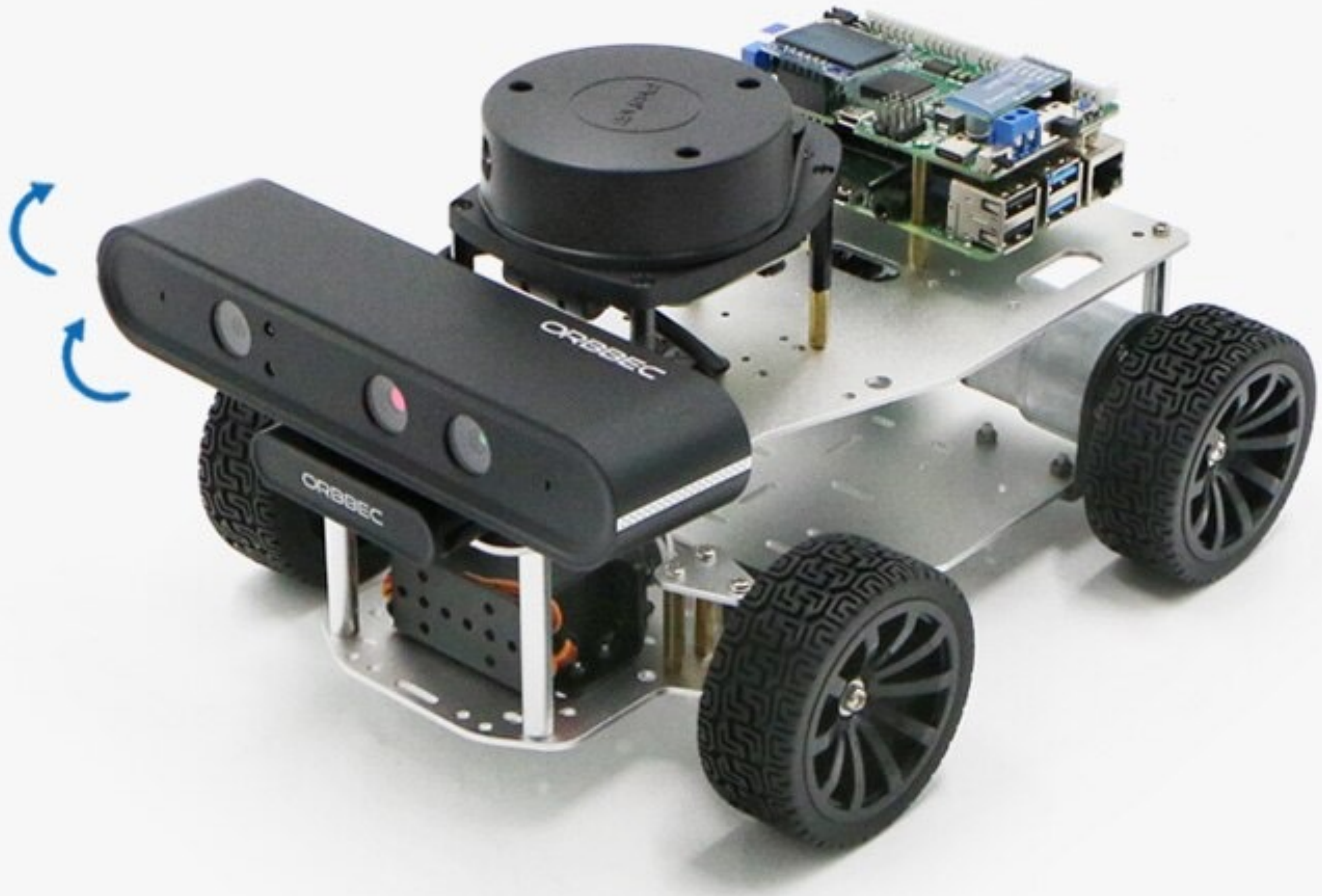
Official standard version *5.5* HZ

12 *Meter radius Measuring range* *360* *degree Scanning and ranging* *8000* *Times/sec Measuring frequency* *OPTMAG Optical and magnetic fusion*

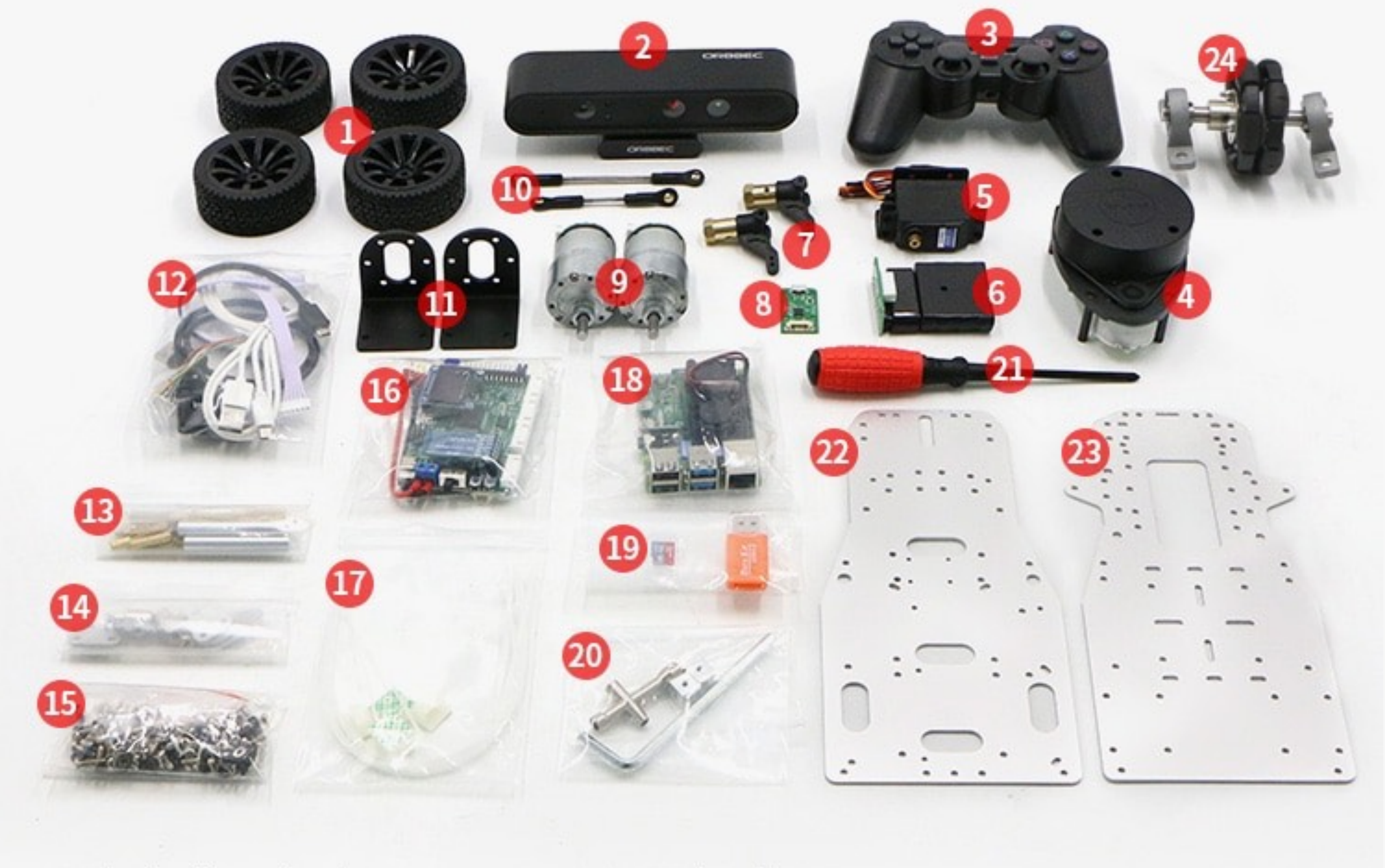
Based on our excellent and concise mechanical design,
the lidar can be 360° unobstructed Make the robot have a better effect
when following and navigating

The camera angle can be adjusted greatly

Whether it is patrolling, following, visual slam, etc., it can be adjusted to a suitable angle, and the camera is placed on the front of the car without obstruction

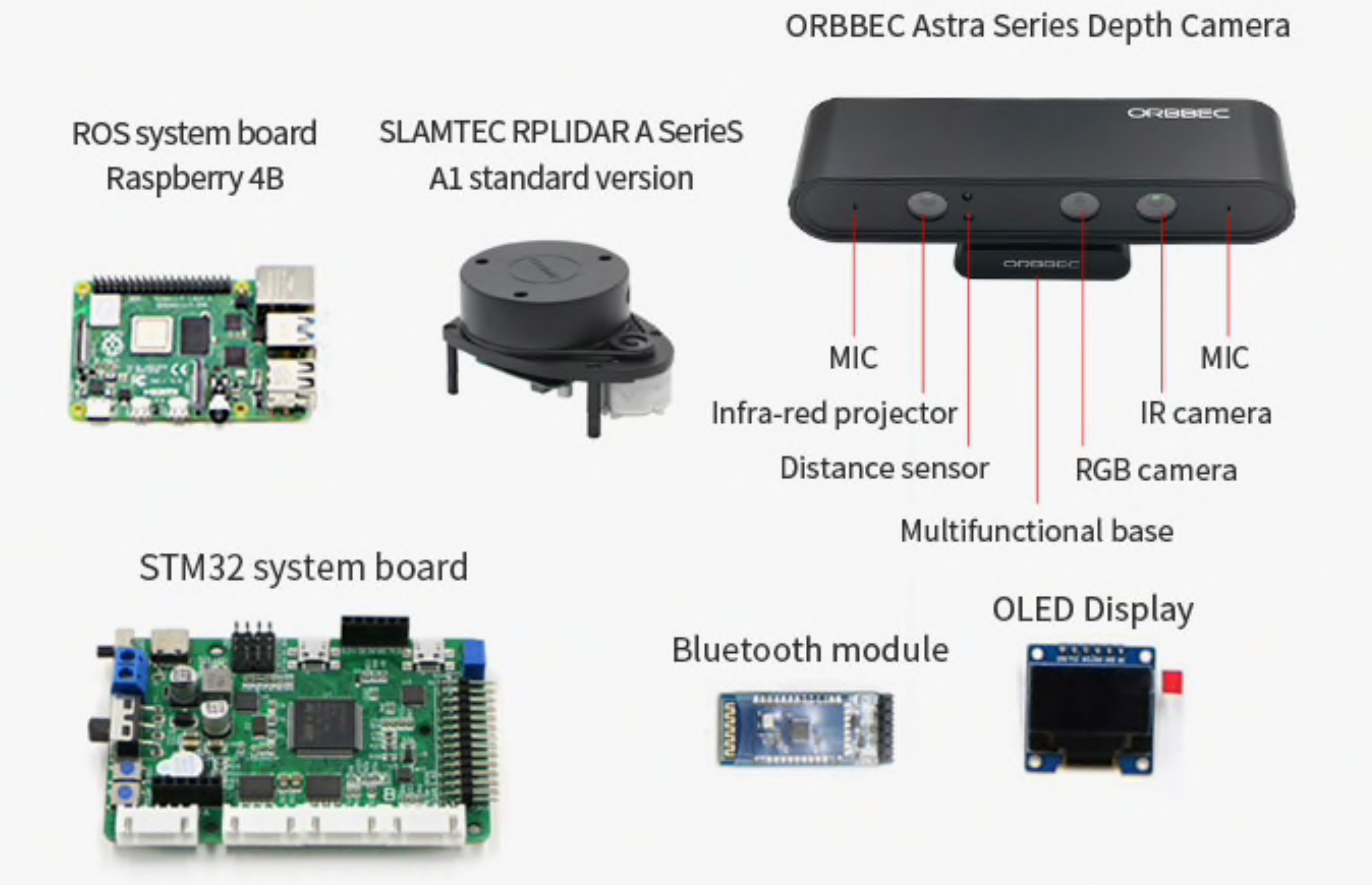


Listing show



1. Black rubber wheels
2. ORBBEC Astra Series Depth Camera
3. PS2 wireless controller
4. RPLIDAR A1
5. Steering gear + multi-function support
6.Remote control receiver
7.Turn to the horn floor
8.Radar adapter board
9.12V30F MG513 motor
10.Ball head pull rod is long/short
11.37 Motor bracket
12.Certain wire rods
13.The pillars
14.Steering plate + hex coupling + Angle press plate
15.Screw and nut package
16.STM32F407VET6 integrated master control board
17.Strap bag
18.Raspberry pie
19.32G memory card+card reader
20.Assemble the kit
21.Cross screwdriver
22. Aluminum alloy top plate
23. Aluminum alloy base plate
24.Omnidirectional wheel set module

Hardware inventory



✔ ORBBEC Astra series depth camera parameter table

Depth resolution	Up to 640×480
Depth frame rate	Up to 640×480 at 30fps
RGB resolution	Up to 640×480
RGB frame rate	Up to 640×480 at 30fps
RGB sensor field of view (H×V)	66.1°×40.2°
Depth sensor field of view (H×V)	58.4°×45.5°
Depth range	0.6m to 4m
Dimensions (diameter×H)	165×40×30mm
Data transmission interface	USB2.0 and above
Whether single/binocular structured light	Monocular structured light + monocular RGB

Hall encoder

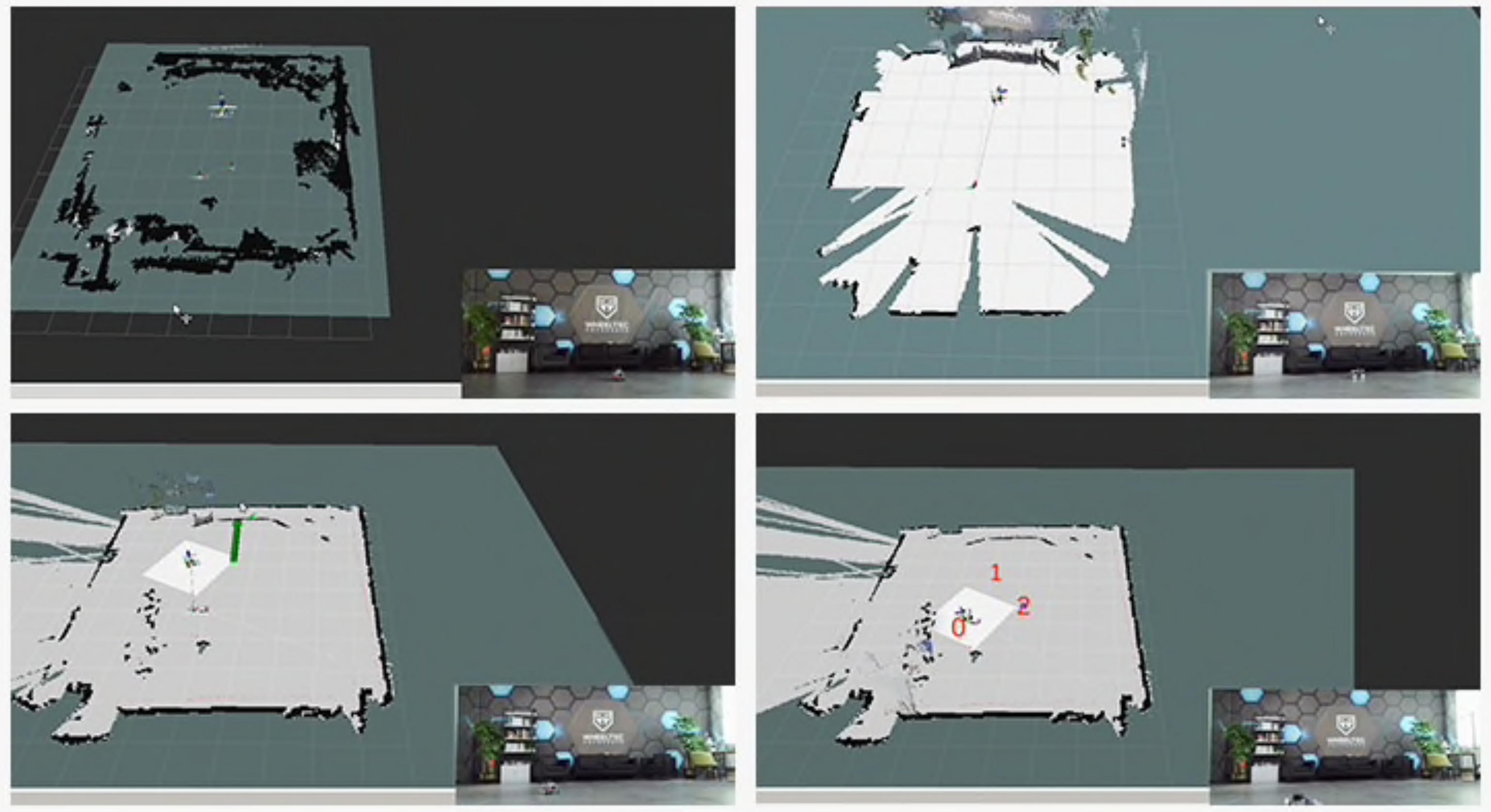
The encoder has a pull-up output, which is pulled up to the power supply VCC pin by default, which can be directly collected by the single-chip microcomputer

Types of	Magnetic induction
Number of lines	13ppr
Supply voltage	5V
Encoder protection	Bare drain (relatively stable without back cover)
Adapt to MCU	Almost all microcontrollers

Key function introduction

☑ RTABMAP VISION AND LIDAR MAPPING NAVIGATION

Support visual SLAM, gmapping, hector, karto, Google Cartographer and other algorithms to build maps, support fixed-point navigation, multi-point navigation

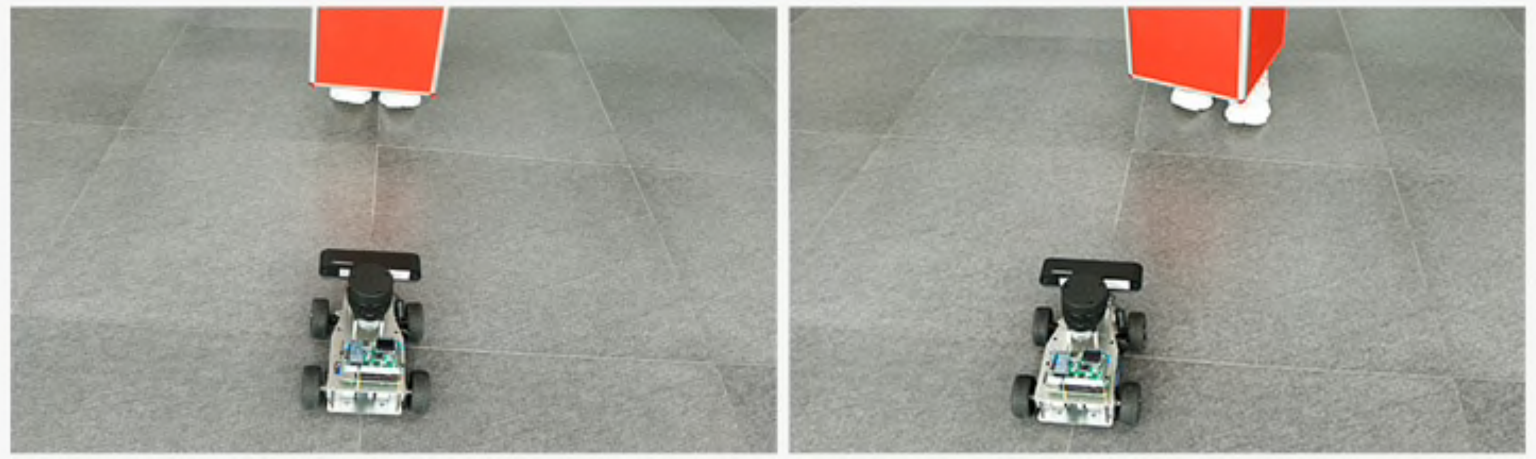


☑ LIDAR FOLLOW

Lidar can follow any object including people in all directions

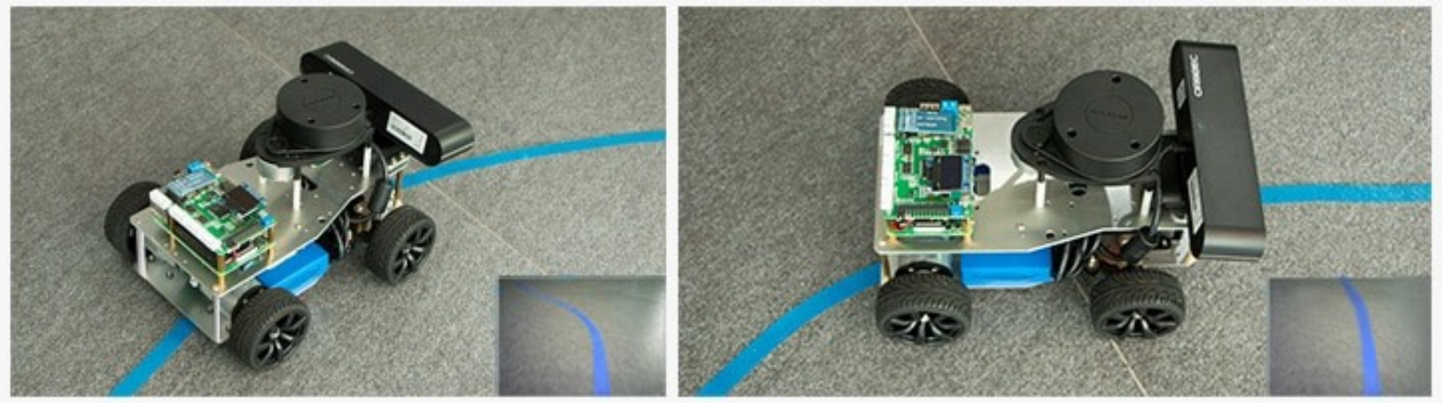
☑ DEPTH CAMERA FOLLOW

Through the RGBD depth camera, you can measure the distance to the front object and follow



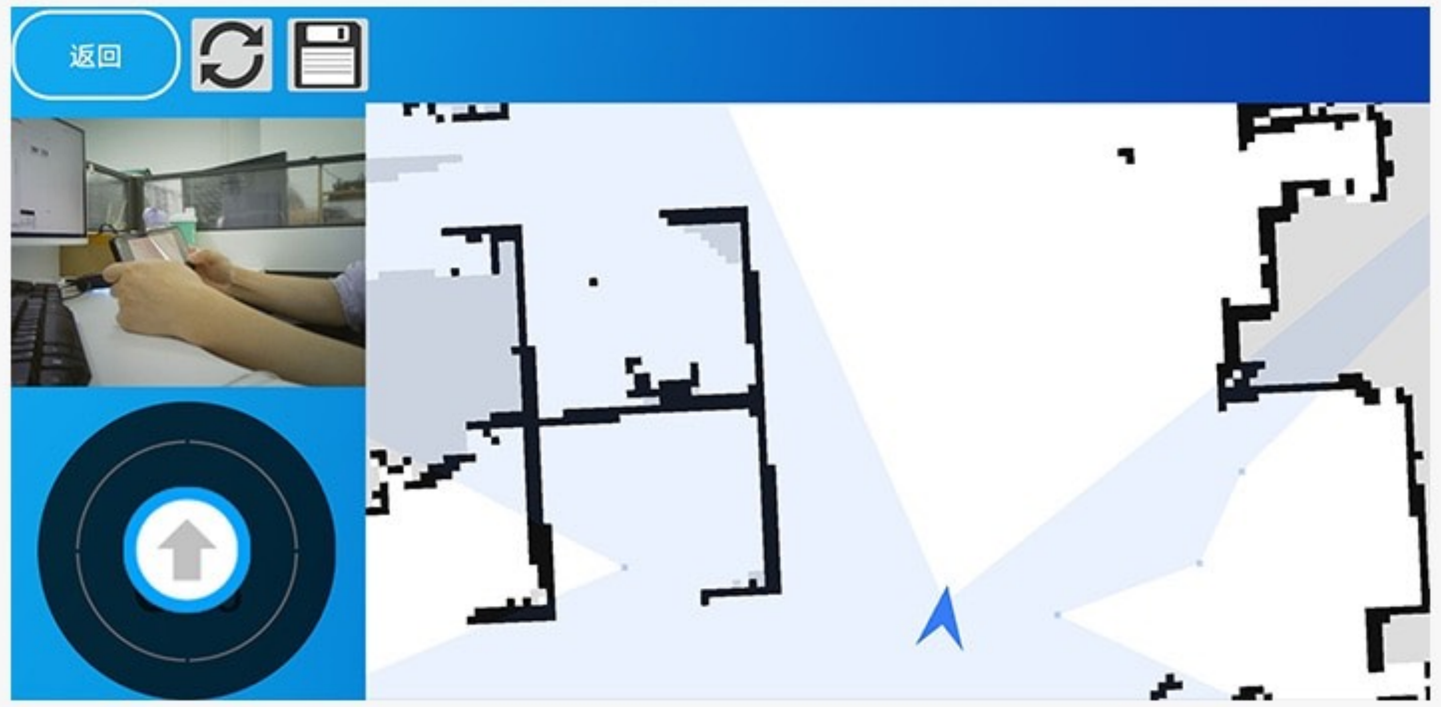
☑ VISUAL PATROL

The camera can be navigated by sticking lines, and the general electrical glue can be used. The color of the line patrol is blue, black, red, green, yellow, etc. adjustable



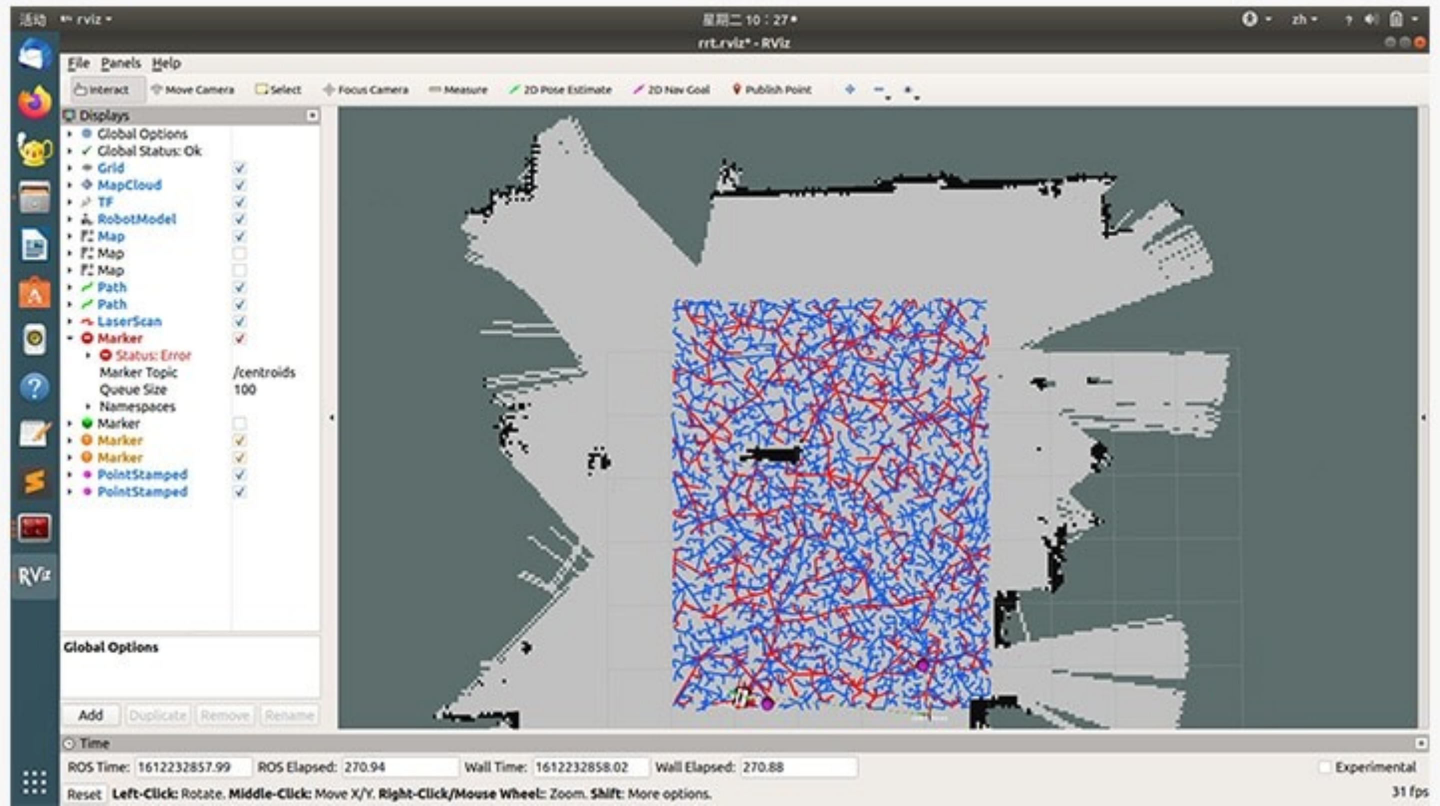
☑ SUPPORT APP CONTROL, VIEW IMAGES, MAP CREATION, NAVIGATION

Realize car mapping and 2D navigation functions through Android ROSAPP



☑ RRT AUTONOMOUS EXPLORATION AND MAPPING

No need to manually control the car, use the RRT algorithm to autonomously complete the exploration map, save the map, and return to the starting point



Marker Topic /centroids
Queue Size 100

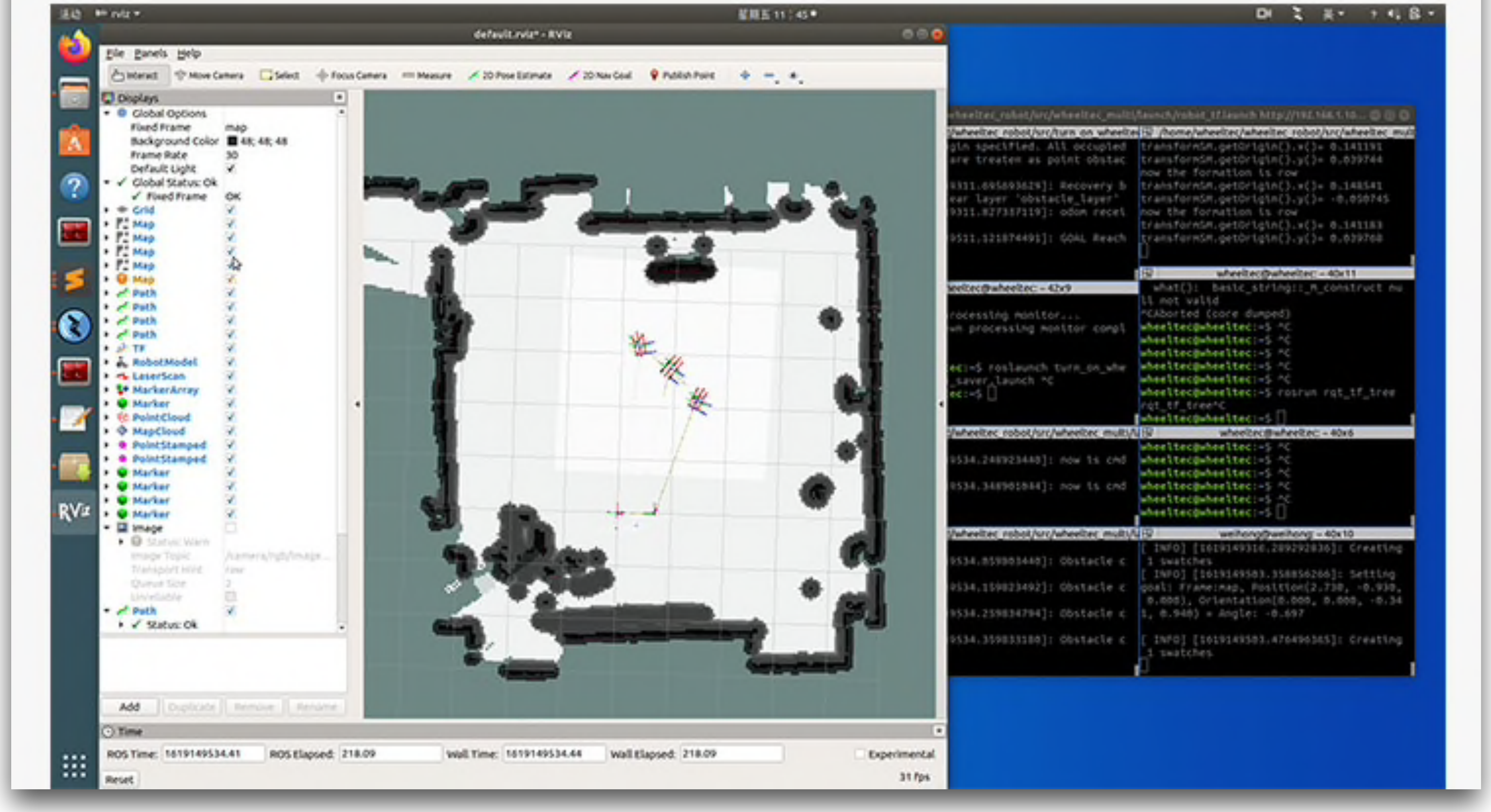
ROS Time: 1612232857.99 ROS Elapsed: 270.94 Wall Time: 1612232858.02 Wall Elapsed: 270.88

Left-Click: Rotate. Middle-Click: Move X/Y. Right-Click/Mouse Wheel:: Zoom. Shift: More options.

31 fps

☑ MULTI-AIRCRAFT FORMATION

Multi-machine coordinated operation, distributed formation control, support single-point and multi-point navigation functions



```
Global Options
  Fixed Frame          map
  Background Color     48; 48; 48
  Frame Rate           30
  Default Light        ✓
✓ Global Status: Ok
  ✓ Fixed Frame        OK
```

ROS Time: 1619149534.41 ROS Elapsed: 218.09 Wall Time: 1619149534.44 Wall Elapsed: 218.09 □ Experimental

31 fps

```
wheeltec_robot/src/wheeltec_multi/launch/robot_tf.launch http://192.168.1.10...
```

```
gin specified. All occupied
are treaten as point obstac

9311.695893629]: Recovery b
ear layer 'obstacle_layer'
9311.827337119]: odom recei

9511.121874491]: GOAL Reach
```

```
transformSM.getOrigin().x()= 0.141191
transformSM.getOrigin().y()= 0.039744
now the formation is row
transformSM.getOrigin().x()= 0.148541
transformSM.getOrigin().y()= -0.050745
now the formation is row
transformSM.getOrigin().x()= 0.141183
transformSM.getOrigin().y()= 0.039768
```

```
rocessing monitor...
wn processing monitor compl

ec:~$ roslaunch turn_on_whe
_saver.launch ^C
ec:~$
```

```
  what():  basic_string::_M_construct nu
ll not valid
^CAborted (core dumped)
wheeltec@wheeltec:~$ ^C
wheeltec@wheeltec:~$ ^C
wheeltec@wheeltec:~$ ^C
wheeltec@wheeltec:~$ ^C
wheeltec@wheeltec:~$ ^C
wheeltec@wheeltec:~$ rosrun rqt_tf_tree
rqt_tf_tree^C
wheeltec@wheeltec:~$
```

```
9534.248923440]: now is cmd

9534.348901844]: now is cmd
```

```
wheeltec@wheeltec:~$ ^C
wheeltec@wheeltec:~$ ^C
wheeltec@wheeltec:~$ ^C
wheeltec@wheeltec:~$ ^C
wheeltec@wheeltec:~$ ^C
wheeltec@wheeltec:~$
```

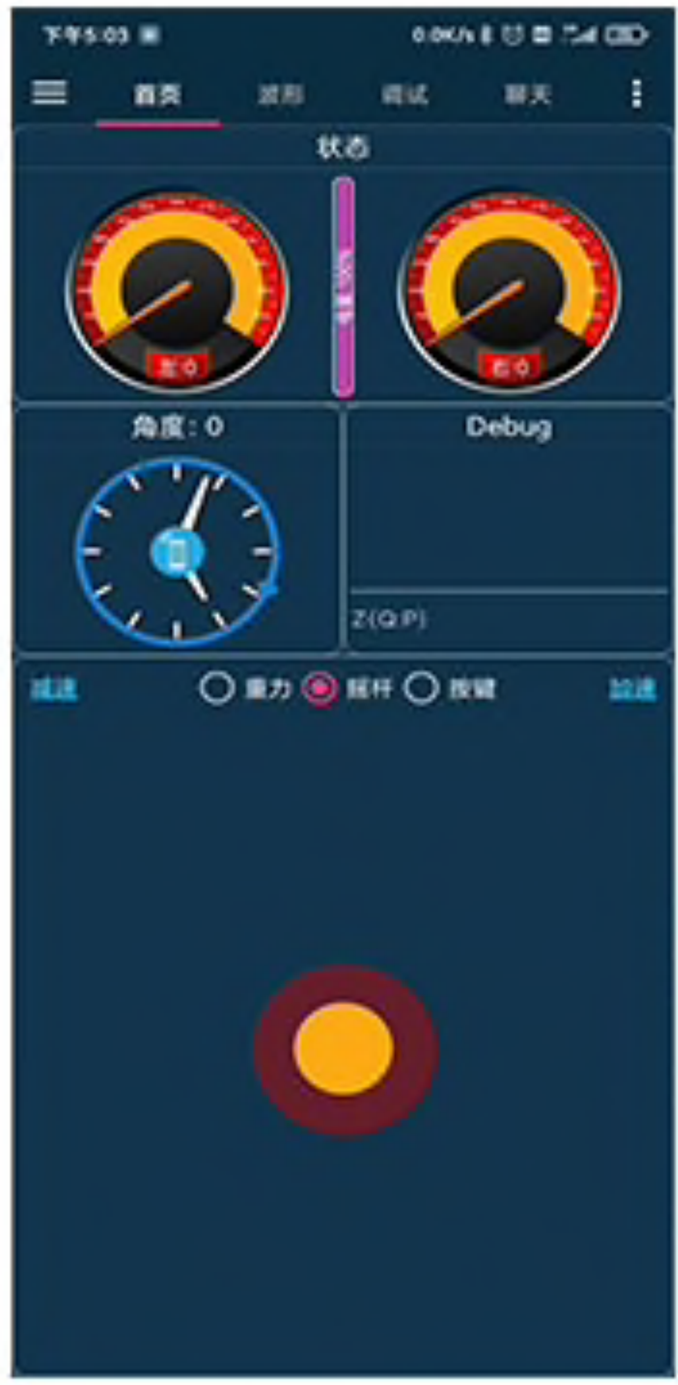
```
9534.059903440]: Obstacle c

9534.159823492]: Obstacle c

9534.259834794]: Obstacle c

9534.359833180]: Obstacle c
```

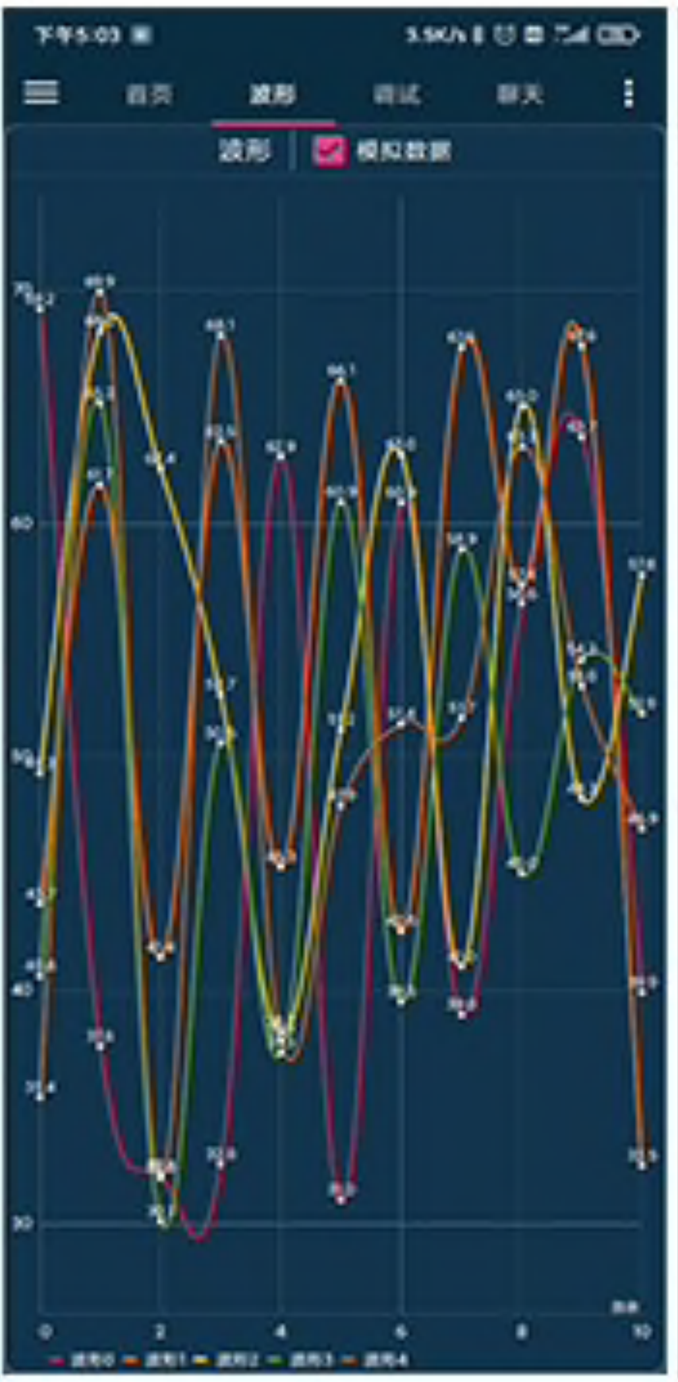
```
[ INFO] [1619149316.289292836]: Creating
 1 swatches
[ INFO] [1619149503.358856266]: Setting
goal: Frame:map, Position(2.730, -0.930,
 0.000), Orientation(0.000, 0.000, -0.34
1, 0.940) = Angle: -0.697

[ INFO] [1619149503.476496365]: Creating
 1 swatches
```

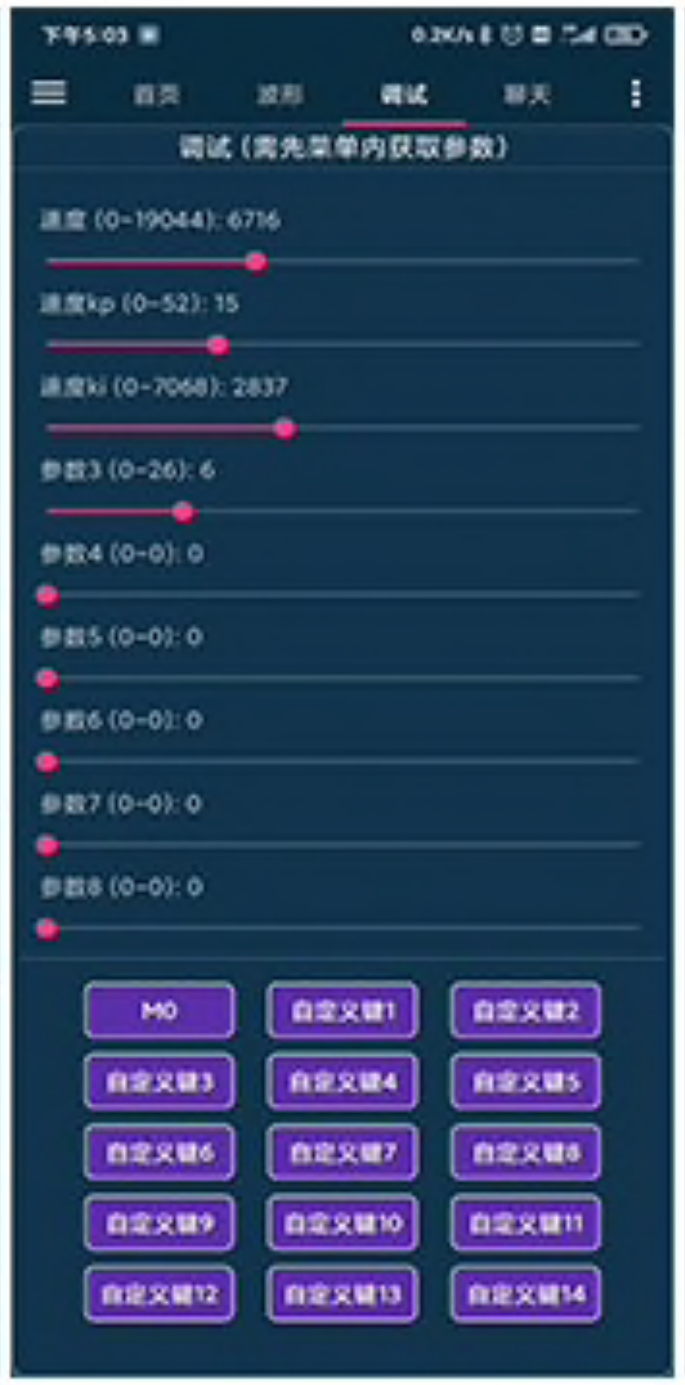
Provide Bluetooth & WIFI version APP

Support Android and IOS

1. Support gravity sensor remote control and two-hand button remote control mode
2. Supports 5-channel waveform display interface, you can view the waveform at any time without a data line
3. Support 9-channel parameter adjustment interface and online adjustment of PID parameters
4. Optimize the battery alarm mechanism, APP accurately pushes low-voltage alarm notifications
5. Support for BLE Bluetooth 4.0 module







速度 (0~19044): 6716

速度kp (0~52): 15

速度ki (0~7068): 2837

参数3 (0~26): 6

参数4 (0~0): 0

参数5 (0~0): 0

参数6 (0~0): 0

参数7 (0~0): 0

参数8 (0~0): 0

Product Information Introduction

version	Ackerman/Differential Edition
Drive structure	Can freely switch positive Ackerman Anti-Ackerman, differential structure
wheel	Driving wheel: 67mm rubber wheel Driven wheel:Ackerman Same as driving wheel， 60mm metal omnidirectional wheel
Steering gear	HWZ020 20KG high torque digital steering gear
Size(mm)	255.6*191*147.6
Trolley weight	1.8kg
load capacity	3kg
Maximum speed	1.2m/s
Light load life Speed 0.45m/s	5.5h
Load 1kg battery life Speed 0.45m/s	4h
Motor	MG513 motor
Encoder	Hall encoder
way to control	APP, PS2 wireless controller, CAN, serial port, etc.
STM32 master	STM32F407VET6
Lidar	SLAMTEC A1 standard version
ROS master	Raspberry 4B 4GB
Depth camera	Astra series RGBD depth camera
IMU sensor	ICM20948 (Three-axis gyroscope + three-axis accelerometer + three-axis magnetometer)
operating system	STM32 is equipped with freeRTOS ROS is equipped with ubuntu18.04+melodic
data	Full series of development manuals, video tutorials, ROS source code, STM32 source code, ROS mirror

ROS MASTER PARAMETERS

Raspberry is generally available for initial entry

ROS master	Raspberry 4B
CPU	ARM Cortex-A72 64-bit@1.5GHz (quad core)
GPU	Broadcom VideaCore VI(32-bit)
RAM	4GB
USB interface	2*USB3.0+2*USB2.0
Video input	MIPI CSI
Video output	Micro-HDMI（two）Resolution up to 4Kp60
Video encoding	H.264(1080p30)
Video decoding	H.264(1080p60) H.265(4Kp60)
Onboard storage	32G Micro SD card
Network Interface	Gigabit Ethernet/Wifi802.11.ac
GPIO pin number	40
rated power	15W(5V/3A)
power input	5V
Overall size	85.60*53.98(mm)

SERVO PARAMETERS

Steering gear	HWZ020	reaction speed	0.14sec/60°
angle	180°	Servo type	Digital Servo
Voltage	4.8-7.4V	gear	Metal gear

Torque	15.3kg.cm(5V)
	20kg.cm(6.5V)
		weight	59.6g

		Working dead zone	3us

☑ BATTERY DESCRIPTION

Due to overseas shipment issues, no batteries are prepared
Please prepare 11.1V lithium battery by yourself

Capacity range	2600~12000mah
Power supply range	10~12.6V
Rated voltage	11.1V

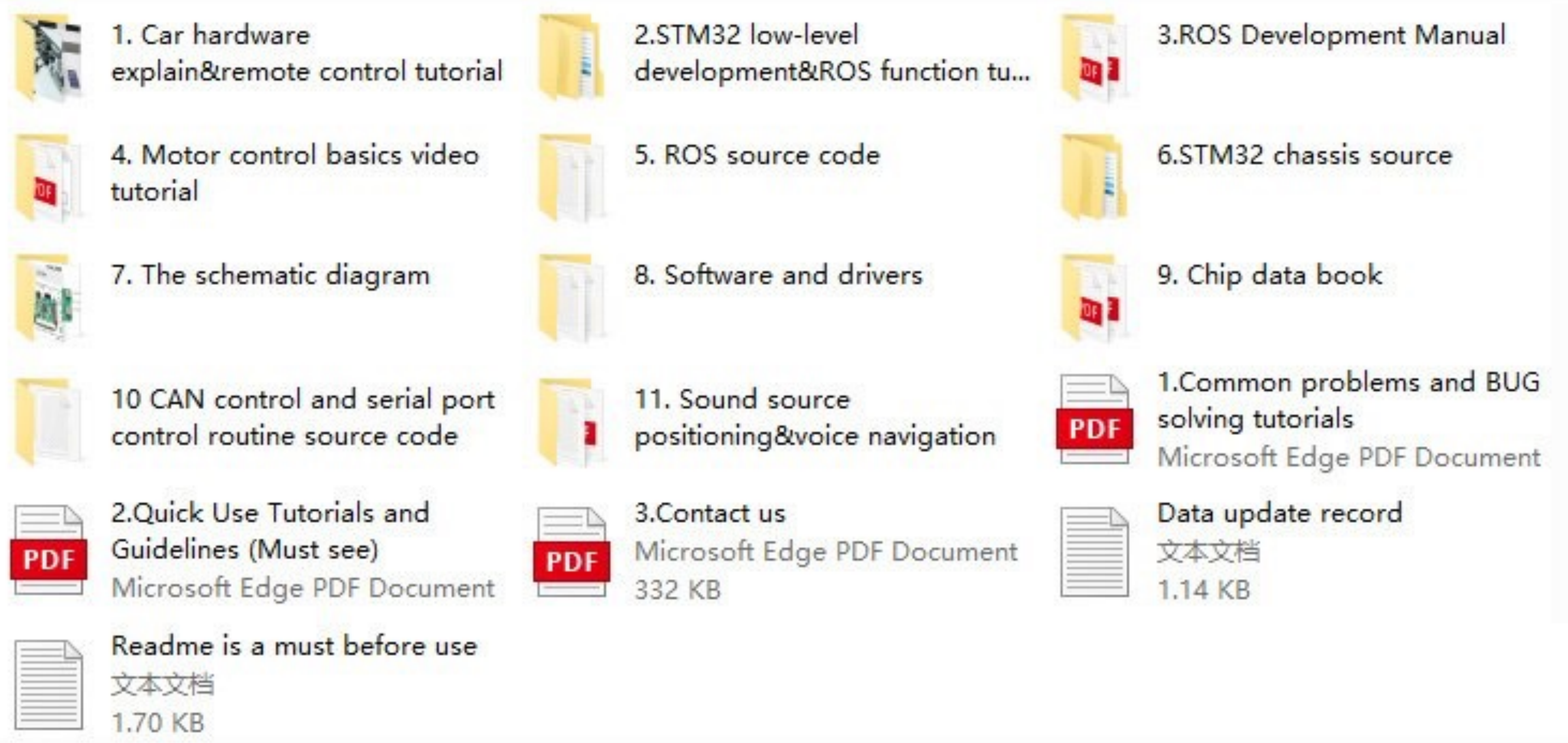
☑ INTERFACE DESCRIPTION

CAN: The mobile platform can receive commands from the CAN port and send its own data (odometer and IMU) through the CAN port.
Serial port: The mobile platform can receive commands from the serial port, or send its own data (odometer and IMU) through the serial port.
USB interface: used to connect to a computer, download the program with one key, receive command control sent by the computer, and send its own data (odo meter and IMU) to the serial port.
Bluetooth (or wifi): can send its own information to APP, can receive APP remote control commands, and can adjust PID parameters.
PS2 interface: Provide PS2 handle socket, provide code plug and play.
SWD interface: SWD interface is provided for online debugging.

Information Description

注：In addition to information related to Raspberry and jetson nano, we also present information such as Xavier NX, industrial computer, etc., so that you can continue to update the functions and information of this product from scientific research to application. The updated information will be provided to users free of charge for life

☑ THE DATA IS COMPLETELY OPEN SOURCE, SUPPORTING SECONDARY DEVELOPMENT



1. Car hardware explain&remote control tutorial

2.STM32 low-level development&ROS function tu...

3.ROS Development Manual

4. Motor control basics video tutorial

5. ROS source code

6.STM32 chassis source

7. The schematic diagram

8. Software and drivers

9. Chip data book

10 CAN control and serial port control routine source code

11. Sound source positioning&voice navigation

1.Common problems and BUG solving tutorials
Microsoft Edge PDF Document

2.Quick Use Tutorials and Guidelines (Must see)
Microsoft Edge PDF Document

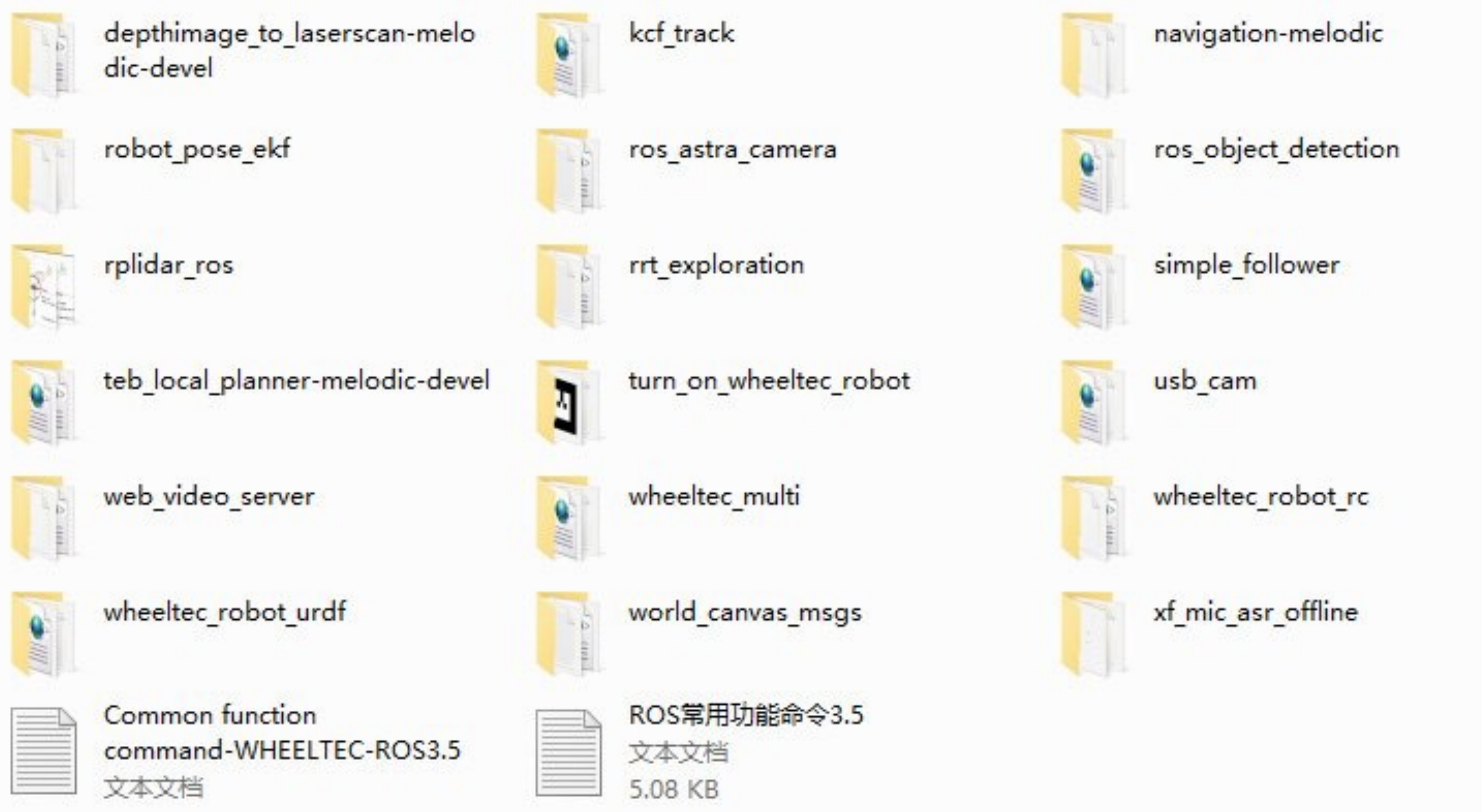
3.Contact us
Microsoft Edge PDF Document
332 KB

Data update record
文本文档
1.14 KB

Readme is a must before use
文本文档
1.70 KB

☑ Provide ROS source code package

The ROS source code can quickly help you connect the car to the ROS system, and provide technical support for STM32, ROS, Linux, and SLAM



depthimage_to_laserscan-melodic-devel

kcf_track

navigation-melodic

robot_pose_ekf

ros_astra_camera

ros_object_detection

rplidar_ros

rrt_exploration

simple_follower

teb_local_planner-melodic-devel

turn_on_wheeltec_robot

usb_cam

web_video_server

wheeltec_multi

wheeltec_robot_rc

wheeltec_robot_urdf

world_canvas_msgs

xf_mic_asr_offline

Common function command-WHEELTEC-ROS3.5
文本文档

ROS常用功能命令3.5
文本文档
5.08 KB

☑ Provide a rich ROS development manual

1.ROS development tutorial

The Preface
1. Fix Raspberry Pi peripheral serial port number
2. SLAM car ROS source code analysis
 2.1 File system preview
 2.2 Code composition
 2.3 Serial communication with the lower computer
 2.4 ROS topics and sensor data release
 2.5 Robot node analysis
 2.6 Parameter analysis of robot
 2.7 Analysis of robot TF coordinate transformation
 2.8 Start the robot through the launch file
3. Laser radar mapping
 3.1 Start the mapping node
 3.2 Map preservation
4. Robot navigation
 4.1 Start the navigation node
 4.2 rviz navigation goal setting
 4.3 Multi-point navigation
 4.4 Navigation parameter setting
 4.5 Navigation status monitoring and custom goals
 4.6 Common navigation fault troubleshooting

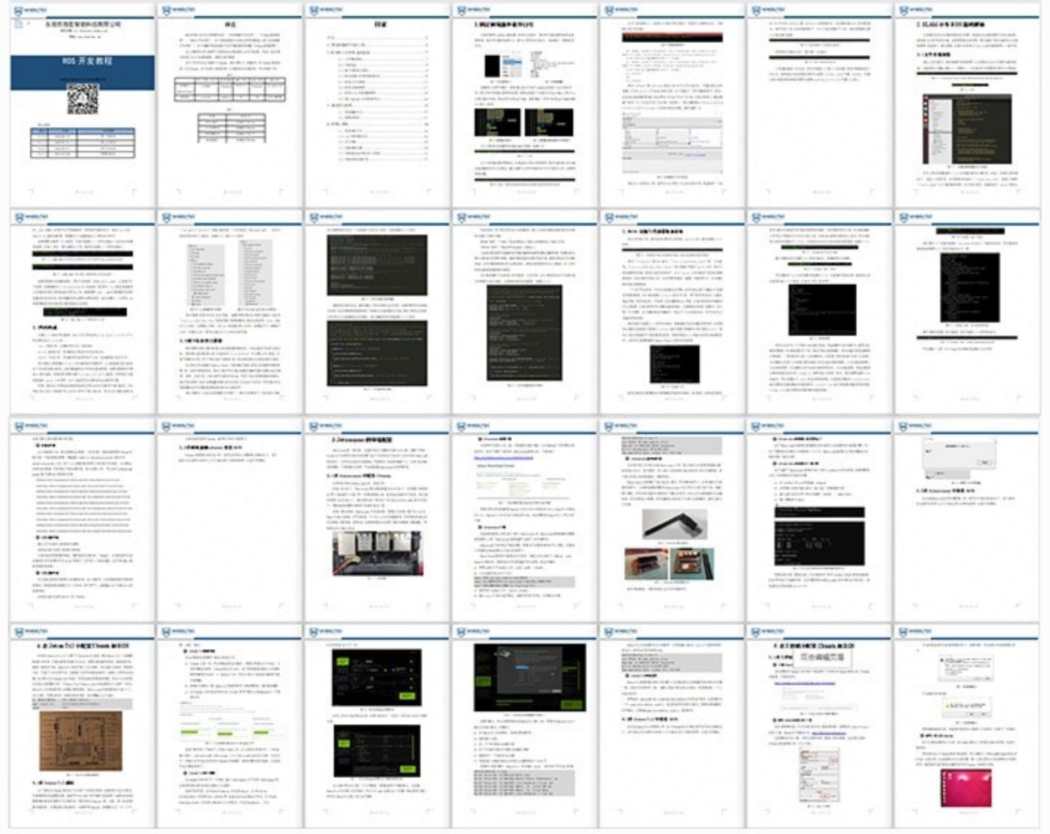
3.STM32 Moving Chassis Development Manual

The Preface
1. Robot control mode
 1.1 Robot movement speed unit
 1.2 ROS (serial port 3) control
 1.3 APP control
 1.4 PS2 control
 1.5 Hot-RC remote control
 1.6 CAN control
 1.7 Serial port 1 control
2. OLED display content
 2.1 OLED specific content
 2.2 OLED universal display content
 2.3 car self-inspection
3. Elimination of gyroscope zero drift
4. Robot kinematics analysis
 4.1 Two-wheel differential (tracked vehicle) car
 4.2 Ackerman car
 4.3 Mecanum wheel carv
 4.4 Omni wheel car
 4.5 Four-wheel drive car
 4.6 PI control program source code
5. Wiring instructions
6. Control flow chart
 6.1 Control flowchart of robot motor
 6.2 Robot STM32 program structure diagram
 6.3 Robot controller connection diagram
7. Matters needing attention
 7.1 About the code
 7.2 About the power interface on the adapter board
 7.3 About the motor
 7.4 About the battery
8. How to download program to STM32 controller
 8.1 Serial download
 8.2 SWD download

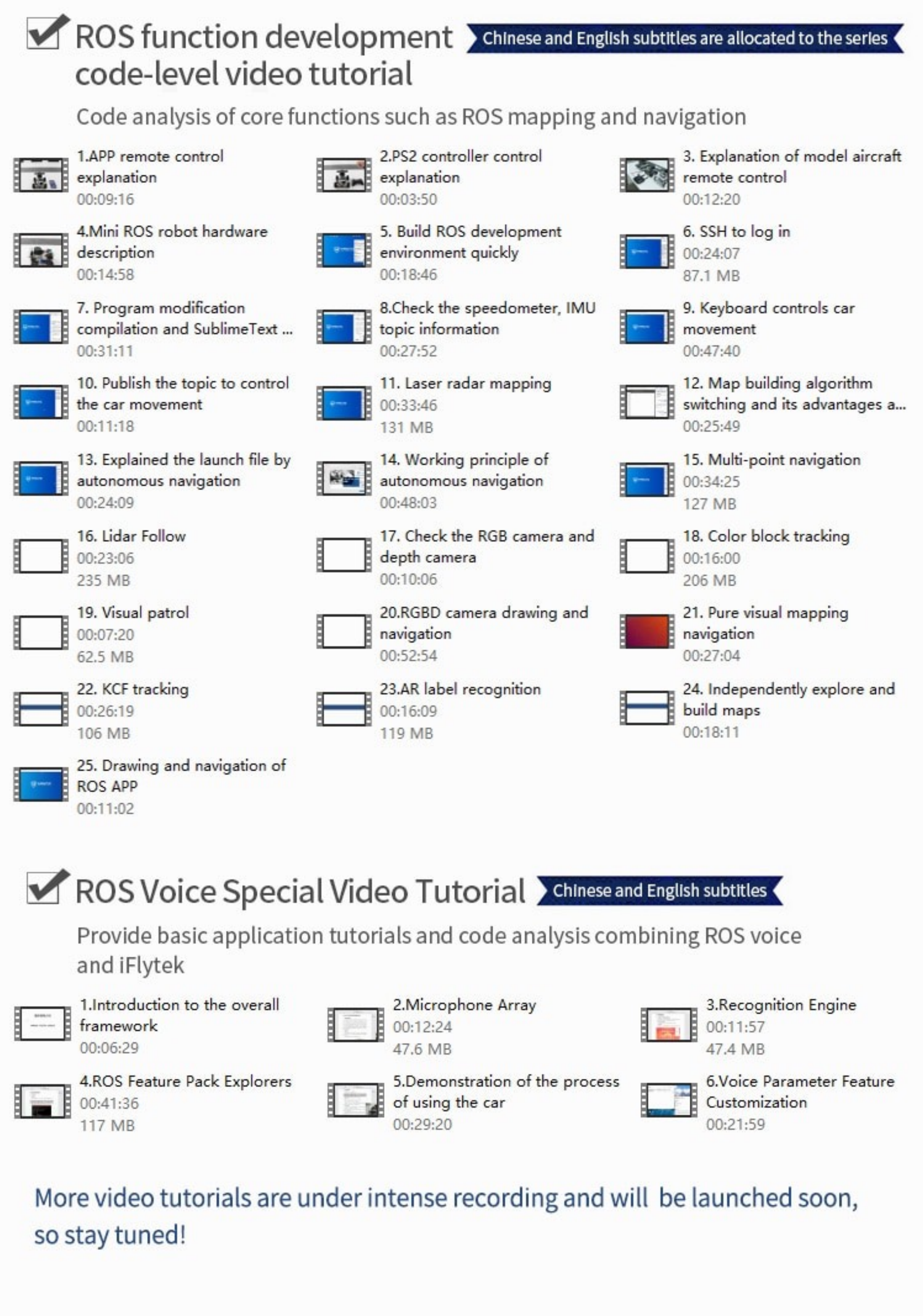
2.Ubuntu configuration tutorial

The Preface
1.Install Ubuntu and ROS on the virtual machine
 1.1 Ubuntu Installation on theVirtual Machine and Utility Plug-in Installation
 1.2 ROS installation with Ubuntu
 1.3 Establish the ROS workspace
 1.4 Configure static IP address with Ubuntu on the Virtual Machine
2. Configure Ubuntu and ROS on Raspberry Pi
 2.1 Configure Ubuntu on Raspberry Pi
 2.2 Install ROS on Ubuntu of Raspberry Pi
3. Environmental configuration of Jetson Nano
 3.1 Configure Ubuntu in Jetson Nano
 3.2 Install ROS in Jetson Nano
4. Configure Ubuntu and ROS in Jetson TX2
 4.1 Flash the Jetson TX2
 4.2 Install ROS on Jetson TX2
5. Configure Ubuntu and ROS on the IPC
 5.1 Install Ubuntu on the IPC
 5.2 Install ROS in IPC
 5.3 Configure wireless WiFi and static IP with Ubuntu on IPC
6. Configure Ubuntu and ROS in Jetson Xavier NX
 6.1 Install Ubuntu in Jetson Xavier NX
 6.2 Install ROS in Jetson Xavier NX
7. Configure wireless WiFi and static IP with Ubuntu
 7.1 Configure wireless WiFi with Ubuntu
 7.2 Ubuntu configures static IP
8. The NFS mount
9. Execute the script at boot time
10. SSH remote login
11. ROS multi-machine communication setup
12. Raspberry Pi image backup and recovery
 12.1 Raspberry Pi image backup
 12.2 Raspberry Pi image recovery
13. Jetson Nano image backup and recovery
 13.1 Jetson Nano image backup
 13.2 Jetson Nano image recovery
14. Jetson TX2 image backup and recovery
15. IPC image backup and recovery
16. Jetson Xavier NX image backup and recovery
17. The basics of Ubuntu

Standard paper format, easy to understand, covering ROS basics, STM32 low-level control, ROS development, UBUNTU tutorials, etc.



Provide code-level video tutorials, senior engineers will teach you how to learn ROS, and refuse to be a "tuner"
ROS related video tutorials are equipped with bilingual subtitles in both Chinese and English. We make the video tutorials according to the standards and investment of making movies.



☑ ROS function development code-level video tutorial

Chinese and English subtitles are allocated to the series

Code analysis of core functions such as ROS mapping and navigation

1.APP remote control explanation
00:09:16

2.PS2 controller control explanation
00:03:50

3. Explanation of model aircraft remote control
00:12:20

4.Mini ROS robot hardware description
00:14:58

5. Build ROS development environment quickly
00:18:46

6. SSH to log in
00:24:07
87.1 MB

7. Program modification compilation and SublimeText ...
00:31:11

8.Check the speedometer, IMU topic information
00:27:52

9. Keyboard controls car movement
00:47:40

10. Publish the topic to control the car movement
00:11:18

11. Laser radar mapping
00:33:46
131 MB

12. Map building algorithm switching and its advantages a...
00:25:49

13. Explained the launch file by autonomous navigation
00:24:09

14. Working principle of autonomous navigation
00:48:03

15. Multi-point navigation
00:34:25
127 MB

16. Lidar Follow
00:23:06
235 MB

17. Check the RGB camera and depth camera
00:10:06

18. Color block tracking
00:16:00
206 MB

19. Visual patrol
00:07:20
62.5 MB

20.RGBD camera drawing and navigation
00:52:54

21. Pure visual mapping navigation
00:27:04

22. KCF tracking
00:26:19
106 MB

23.AR label recognition
00:16:09
119 MB

24. Independently explore and build maps
00:18:11

25. Drawing and navigation of ROS APP
00:11:02

☑ ROS Voice Special Video Tutorial

Chinese and English subtitles

Provide basic application tutorials and code analysis combining ROS voice and iFlytek

1.Introduction to the overall framework
00:06:29

2.Microphone Array
00:12:24
47.6 MB

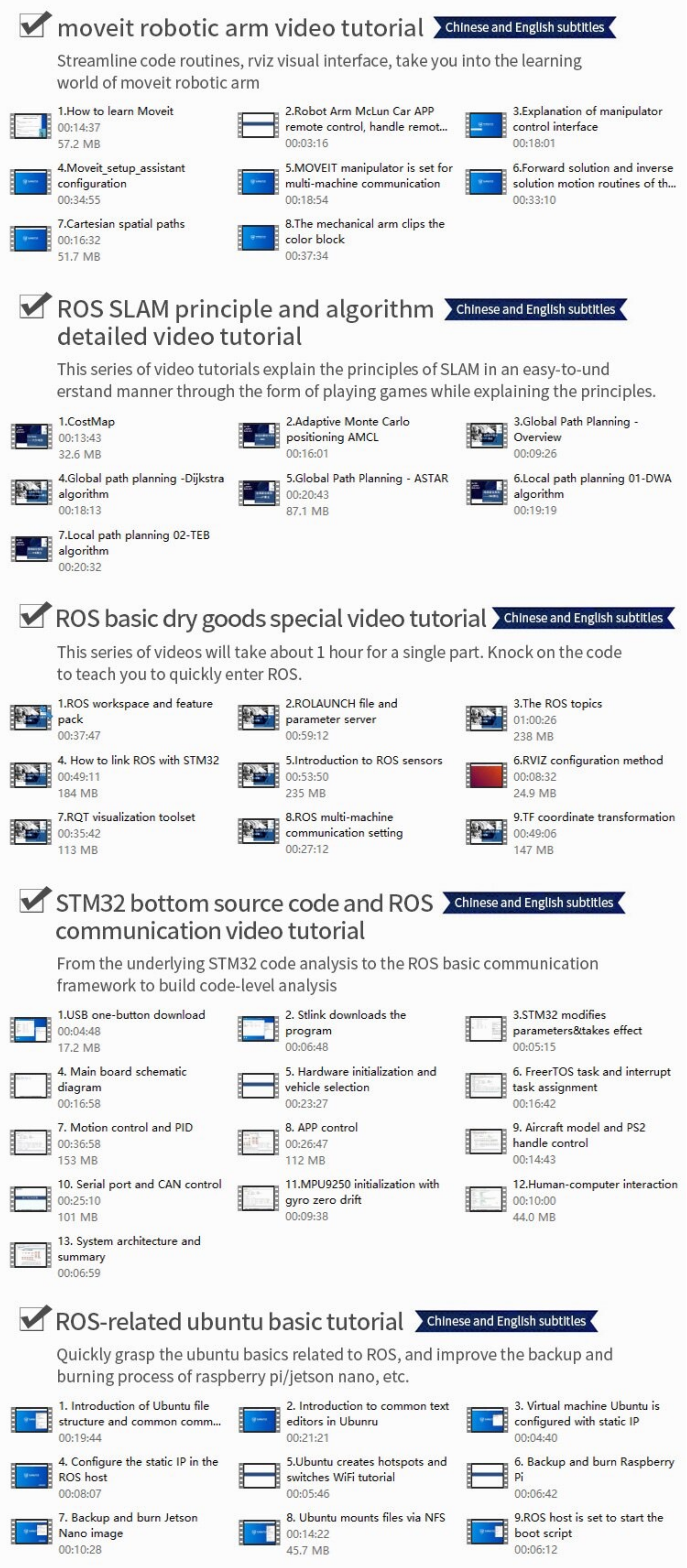
3.Recognition Engine
00:11:57
47.4 MB

4.ROS Feature Pack Explorers
00:41:36
117 MB

5.Demonstration of the process of using the car
00:29:20

6.Voice Parameter Feature Customization
00:21:59

More video tutorials are under intense recording and will be launched soon, so stay tuned!



☑ moveit robotic arm video tutorial

Chinese and English subtitles

Streamline code routines, rviz visual interface, take you into the learning world of moveit robotic arm

1.How to learn Moveit
00:14:37
57.2 MB

2.Robot Arm McLun Car APP remote control, handle remot...
00:03:16

3.Explanation of manipulator control interface
00:18:01

4.Moveit_setup_assistant configuration
00:34:55

5.MOVEIT manipulator is set for multi-machine communication
00:18:54

6.Forward solution and inverse solution motion routines of th...
00:33:10

7.Cartesian spatial paths
00:16:32
51.7 MB

8.The mechanical arm clips the color block
00:37:34

☑ ROS SLAM principle and algorithm detailed video tutorial

Chinese and English subtitles

This series of video tutorials explain the principles of SLAM in an easy-to-und erstand manner through the form of playing games while explaining the principles.

1.CostMap
00:13:43
32.6 MB

2.Adaptive Monte Carlo positioning AMCL
00:16:01

3.Global Path Planning - Overview
00:09:26

4.Global path planning -Dijkstra algorithm
00:18:13

5.Global Path Planning - ASTAR
00:20:43
87.1 MB

6.Local path planning 01-DWA algorithm
00:19:19

7.Local path planning 02-TEB algorithm
00:20:32

☑ ROS basic dry goods special video tutorial

Chinese and English subtitles

This series of videos will take about 1 hour for a single part. Knock on the code to teach you to quickly enter ROS.

1.ROS workspace and feature pack
00:37:47

2.ROLAUNCH file and parameter server
00:59:12

3.The ROS topics
01:00:26
238 MB

4. How to link ROS with STM32
00:49:11
184 MB

5.Introduction to ROS sensors
00:53:50
235 MB

6.RVIZ configuration method
00:08:32
24.9 MB

7.RQT visualization toolset
00:35:42
113 MB

8.ROS multi-machine communication setting
00:27:12

9.TF coordinate transformation
00:49:06
147 MB

☑ STM32 bottom source code and ROS communication video tutorial

Chinese and English subtitles

From the underlying STM32 code analysis to the ROS basic communication framework to build code-level analysis

1.USB one-button download
00:04:48
17.2 MB

2. Stlink downloads the program
00:06:48

3.STM32 modifies parameters&takes effect
00:05:15

4. Main board schematic diagram
00:16:58

5. Hardware initialization and vehicle selection
00:23:27

6. FreerTOS task and interrupt task assignment
00:16:42

7. Motion control and PID
00:36:58
153 MB

8. APP control
00:26:47
112 MB

9. Aircraft model and PS2 handle control
00:14:43

10. Serial port and CAN control
00:25:10
101 MB

11.MPU9250 initialization with gyro zero drift
00:09:38

12.Human-computer interaction
00:10:00
44.0 MB

13. System architecture and summary
00:06:59

☑ ROS-related ubuntu basic tutorial

Chinese and English subtitles

Quickly grasp the ubuntu basics related to ROS, and improve the backup and burning process of raspberry pi/jetson nano, etc.

1. Introduction of Ubuntu file structure and common comm...
00:19:44

2. Introduction to common text editors in Ubunru
00:21:21

3. Virtual machine Ubuntu is configured with static IP
00:04:40

4. Configure the static IP in the ROS host
00:08:07

5.Ubuntu creates hotspots and switches WiFi tutorial
00:05:46

6. Backup and burn Raspberry Pi
00:06:42

7. Backup and burn Jetson Nano image
00:10:28

8. Ubuntu mounts files via NFS
00:14:22
45.7 MB

9.ROS host is set to start the boot script
00:06:12

Shipping list

☑ CHASSIS PART

12V30F MG513 motor X2
Hexagonal coupling-6mm X2
Trolley aluminum alloy floor X1pcs
Omni-directional wheel module X1pcs
Black rubber wheels X4
Steering Claw Pressing Plate X2
Servo multi-function bracket X1pcs
Trolley aluminum alloy upper plate X1pcs
Several standard parts and their connecting parts

Servo X1
Tie rod short X1
37 Motor bracket X2
Tie rod length X1
Rudder wheel X1
Steering claw assembly X2
Steering gear rocker arm X1

☑ ELECTRONIC CONTROL AND ROS PART

Electronic control part:

STM32F407VET6 integrated main control board
Bluetooth module X1
OLED display X1
Data download line X1

ROS part:

Raspberry 4B X1
Lidar X1
32G high speed memory card and card reader X1
PS2 wireless controller X1
Dual fan heat sink X1
Several wires
Depth camera and its angle adjustment mechanism

☑ PEARL COTTON PACKAGING

The following is the quality and volume of the packaged product :

Volume：370*300*170mm
Weight：3kg

ROBOWORKS

we build human friendly robots