FOXTECH VG-450 UGV Lidar Mapping Robot User Manual

Contents

FOXTECH VG-450 UGV Lidar Mapping Robot

Introduction

Overview
VG-450 is a fast, agile, compact robotic open-source research platform specially designed for UGV developers and scientific researchers, providing abundant demo routines, such as outdoor waypoint planning, 2D LiDAR mapping and obstacle avoidance, 3D mapping, etc.
This development platform is based on ROS open-source system and APM auto navigation system and also equipped with multiple sensors, such as LiDAR, binocular camera, depth camera, RTK, which is suitable for applied research on autonomous delivery vehicles, service robots, and add-on function development.
Specification

Scout MINI

- **Size** 627x549x502 mm
- **Wheelbase** 452mm
- **Frontrear wheel base** 450mm
- **Weight** 26kg
- **Load capacity** 7kg
- **Battery type** 24V 15Ah lithium battery
- **Motor** brushless DC motor 4*150W
- **Drive type** independent four-wheel drive
- **Suspension** independent suspension with rocker arm
- **Steering** four-wheel differential steering
- **Safety equipment** Servo brake/anti-collision tube
- **No-load highest speed** 10.8km/h
- **Minimum turning radius** Om (n-situ rotation)

- **Maximum elimbing capacity** S0
- **Minimum ground clearance** 107mm
- **Max travel** 10km
- **Control mode** remote control/command mode
- **Kemote controller** 2.4Ghz/1km extreme distance
- **Communication interface** CAN

Onboard Computer

- **Model** AS6
- **CPu** Intel 17 8565U, quad-core and eight-thread
- **GPU** Intel UHD Graphics 620
- **Hard drive** 128GB
- **RAM** 8GB DDR4

Binocular Camera

- **Model**
  Intel Realsense T265
- **Chip** Movidius Myraid2
- **Field of view** Two fisheye lenses, close to hemispherical 163+5°FOV
- **IMU** BMIOS5, allows for accurate measurement of rotation and acceleration of the device

Depth Camera

- **Model** ntel Realsense D4351\
- **Depth technology** active IR stereo
- **Depth output resolution** up to 1280°720
- **Depth output frame rate** up to 90fps
- **Min. depth distance** 0.lm

LiDAR

- **Laser ranging technoloEy** TOF
- **Kanging radnus** 0.15m-
- **Sample rate** 9200 times/sec Kanging accuracy 2-10cm (typical Scm)
- **Ranging resouton** Cm
- **SCanning ng** 360
- **Scanning Irequency** 7-1SHz (typical 10Hz)

RTK Positioning Module

- **Frequency** BDS/GPS/GLONASS/QZSs
- **Positioning accuracy** 10cm (typical)

- **Initialization time** < 10 s(typical)
- **Time to first fix** cold start: 40s (typical); hot start: 5s (typical) serial port. TF card. USB 2.00G. CAN. PPS
- **Interface** EVENT
- **Data format** N EA-0183. BINEX. Femtomes ASCII. Binary
- **GNSS data rate** IHz/SHz/ 10Hz/ 20Hz(optional) WIFI Transmission System
- **Weight** 146.8g
- **Size** 88x66x19mm
- **Transmission distance** 800m(without obstacles)
- **Frequency** 5.1GHz-5.9GHz
- **Power** 6W
- **Delay** 200ms
- **Bandwidth** 40MHz or 20MHz
- **Transmitting power** 20mW
- **Work temperature** -10°C~45°Cc

1. Turn on the remote controller and turn the SWB shift lever to the middle position to switch the UGV to the remote control mode, so that the UGV can move to the test site (*Because the point when the navigation control panel is activated is considered to be the home point, it is recommended to restart the UGV after arriving at the test site)
2. Connect to the UGV's WiFi, and use the Mission Planner ground station and NoMachine to connect to the UGV
3. Click the flight plan in the upper left corner in the ground station to enter the waypoint setting interface. And click any point on the map with the left mouse button to set the waypoint. The attributes and settings of the waypoint will be displayed in the upper left corner and the lower corner, which can be modified according to the situation. After setting the waypoint, click the write waypoint on the right. (*After completing the above steps, restart the onboard computer to obtain a new waypoint.)
4. Turn on the onboard computer power, connect to the UGV X86 computer through NoMachine, and open the sh script folder on the desktop
5. Click the right mouse button in the folder, select Open in Terminal option to open a terminal, and enter the following command to start waypoint planning and VFH obstacle avoidance function
6. In the Mission Planner ground station, select Action -> Mode (AUTO or GUIDED) -> Set Mode to set the UGV to AUTO or GUIDED mode (* In AUTO mode, the UGV will move according to the waypoint plan. In GUIDED mode, the UGV will move according to the waypoint plan and has the VFH obstacle avoidance function)

2D Mapping

1. Press the power button of the UGV
2. Turn on the remote controller
3. Connect to the WiFi of the UGV, and start NoMachine to connect to the UGV's onboard computer
4. Find the sh script folder of the UGV's onboard computer desktop in the NoMachine interface
5. Click the right mouse button in the folder, select the Open in Terminal option to open a terminal, and enter the following command to start 2D mapping function r300_cartographer_slam.sh
6. Under normal circumstances, each node starts normally, and you can see the map displayed in rviz
7. Use the remote controller to control the UGV to move. After the map of the area is built, enter the following

command to save the map rosrun map_server map_saver -f map_name

8. tip (map_name in the command is the name of the saved map-related file, and a pgm and yaml format file will be generated. The file will be saved in the folder path of the terminal where the command is entered)

3D Mapping

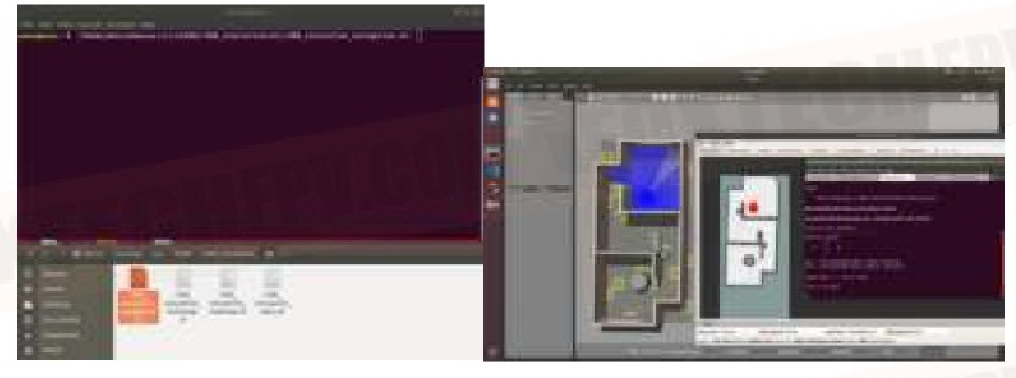
1. Press the power button of the UGV
2. Turn on the remote controller
3. Connect to the WiFi of the UGV, and start NoMachine to connect to the UGV's onboard computer
4. Find the sh script folder of the UGV's onboard computer desktop in the NoMachine interface
5. Click the right mouse button in the folder, select Open in Terminal option to open a terminal, and enter the following command to start 2D mapping function
   r300_rtabmap.sh
6. Use the remote controller to control the movement of the UGV to establish a 3D map of the area
   - tip (If the node starts abnormally, please use the rs-sensor-control command to check whether the T265 and D435i cameras appear. If they do not appear, it means that the device is not connected to the onboard computer normally. Please try to change the USB port or restart the UGV)

Simulation Introduction

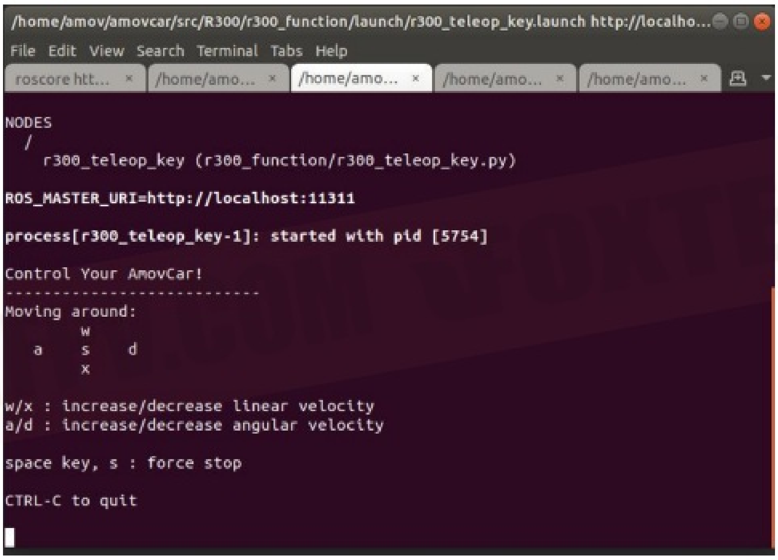
Simulation System
The R300 simulation system is based on ROS and Gazebo simulation system. It provides UGV body models, and sensor simulations such as 2D lidar, 3D lidar and depth camera. It is currently equipped with navigation function, RtabMap 3D mapping function, OctoMap 3D mapping function, and SLAM mapping function.

Navigation Function
The started sh script is /src/R300/r300_simulation/sh/r300_simulation_navigation.sh. The sh file contains the following parts:

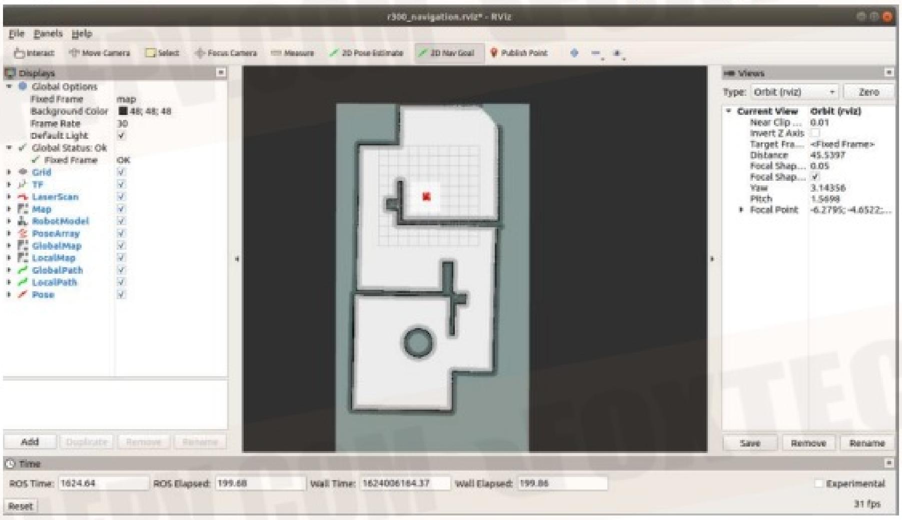
1. Start the ros master node
2. Start R300 simulation, including simulation environment, UGV model, sensor simulation, TF, etc.
3. Keyboard control node to control the movement of UGV
4. Navigation function
5. The rviz visual interface, which is set for the navigation function, displays routes planning, maps, positioning, lidar data, and UGV models.
   Open a terminal arbitrarily, drag the r300_simulation_navigation.sh file into the terminal window, and a command to start the sh script will appear. Press Enter to start it.

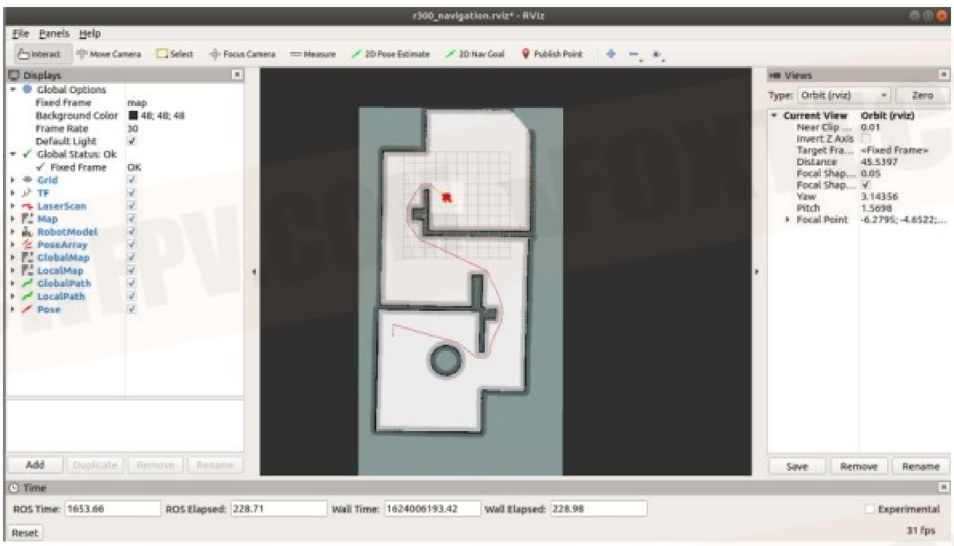


Check the terminal window that pops up and whether the nodes in each terminal are started normally. After confirming that the nodes are running normally, press the A or D key in the third keyboard-controlled terminal to give the UGV an angular velocity (Angular velocity is recommended to control within 0.5). After the UGV's positioning has converged, under normal circumstances, the UGV can rotate for one full circle. Press crtl + c in the keyboard control terminal to close the node.



Select the 2D Nav Goal plug-in in rviz, select any point in the map, click the left mouse button and long press to select the direction, and then release it to send the navigation target point, and the UGV will automatically navigate to the target point.

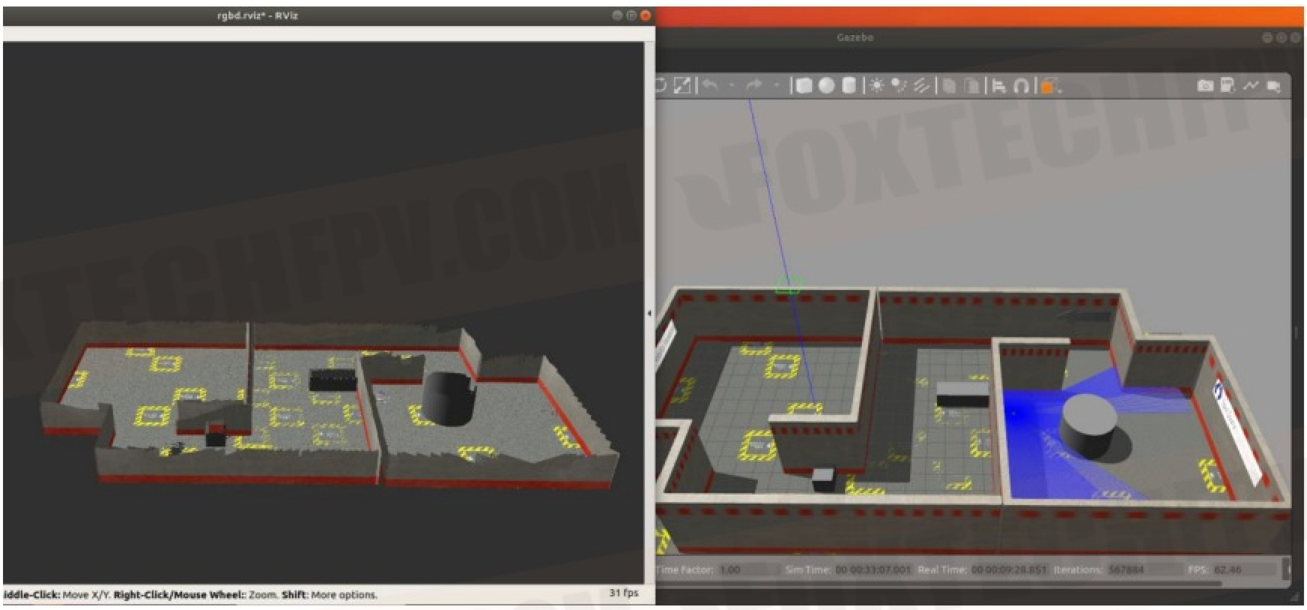




RtabMap 3D Mapping Function

The started sh script is /src/R300/r300_simulation/sh/r300_simulation_rtabmap.sh. The sh file contains the following parts:

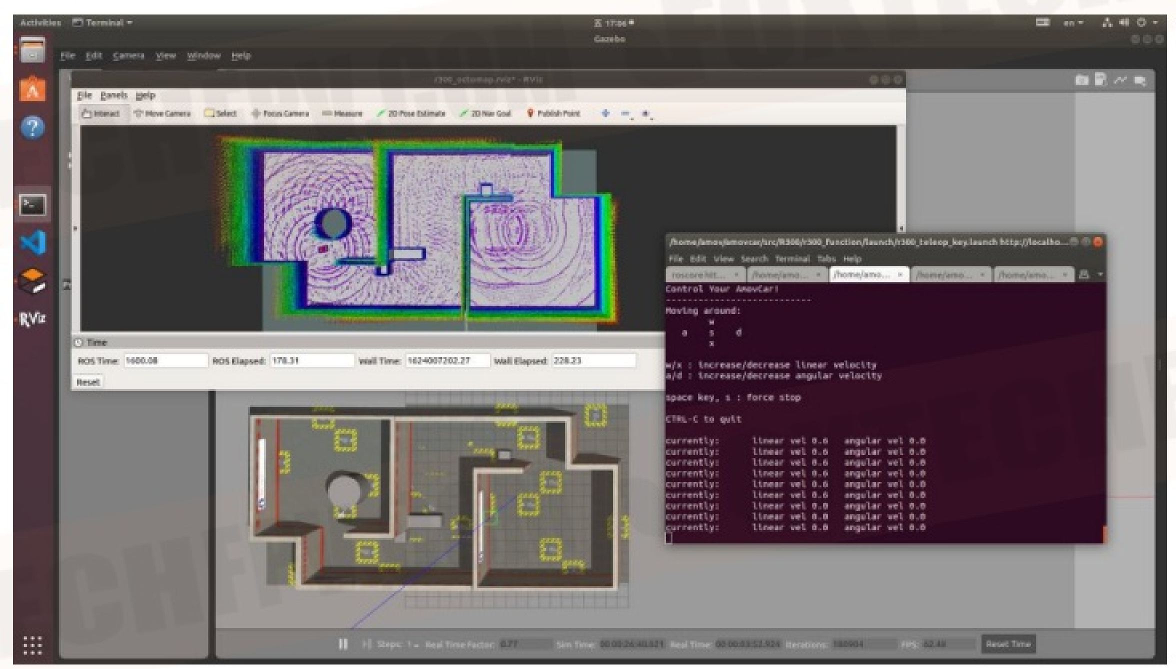
1. Start the ros master node
2. Start R300 simulation, including simulation environment, UGV model, sensor simulation, TF, etc.
3. Keyboard control node to control the movement of UGV
4. Rtabmap mapping function. This function mainly uses the visual image and the depth image of the depth camera to make 3D mapping. Open a terminal arbitrarily, drag the r300_simulation_rtabmap.sh file into the terminal window, and a command to start the sh script will appear. Press Enter to start it. Check the terminal window that pops up and whether the nodes in each terminal are started normally. After confirming that the nodes are running normally, enter the corresponding control instructions in the third keyboard-controlled terminal to control the movement of the UGV for 3D mapping.



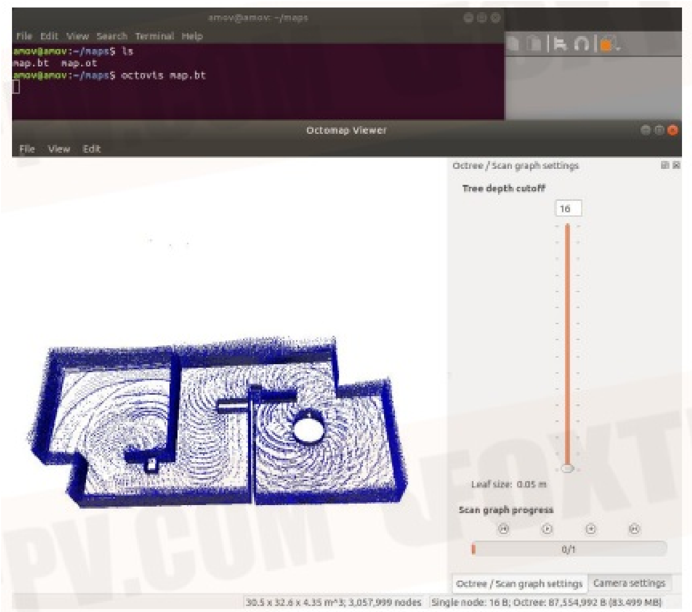
OctoMap 3D Mapping Function

The started sh script is /src/R300/300_simulation/sh/r300_simulation_octomap.sh. The sh file contains the following parts:

1. Start the ros master node
2. Start R300 simulation, including simulation environment, UGV model, sensor simulation, TF, etc.
3. Keyboard control node to control the movement of UGV
4. Octomap mapping function, mainly using 3D lidar point cloud data for mapping
5. The rviz visual interface, which is set for the octomap function, displays maps and UGV models. Open a terminal arbitrarily, drag the r300_simulation_octomap.sh file into the terminal window, and a command to start the sh script will appear. Press Enter to start it. Check the terminal window that pops up and whether the nodes in each terminal are started normally. After confirming that the nodes are running normally, enter the corresponding control instructions in the third keyboard-controlled terminal to control the movement of the UGV for 3D mapping.



After the map is created, you can enter the following command to save the map: rosrun octomap_server octomap_saver -f map_name.ot To view the 3D map, enter the following commands: octovis map_name.ot Install sudo apt-get install octovis

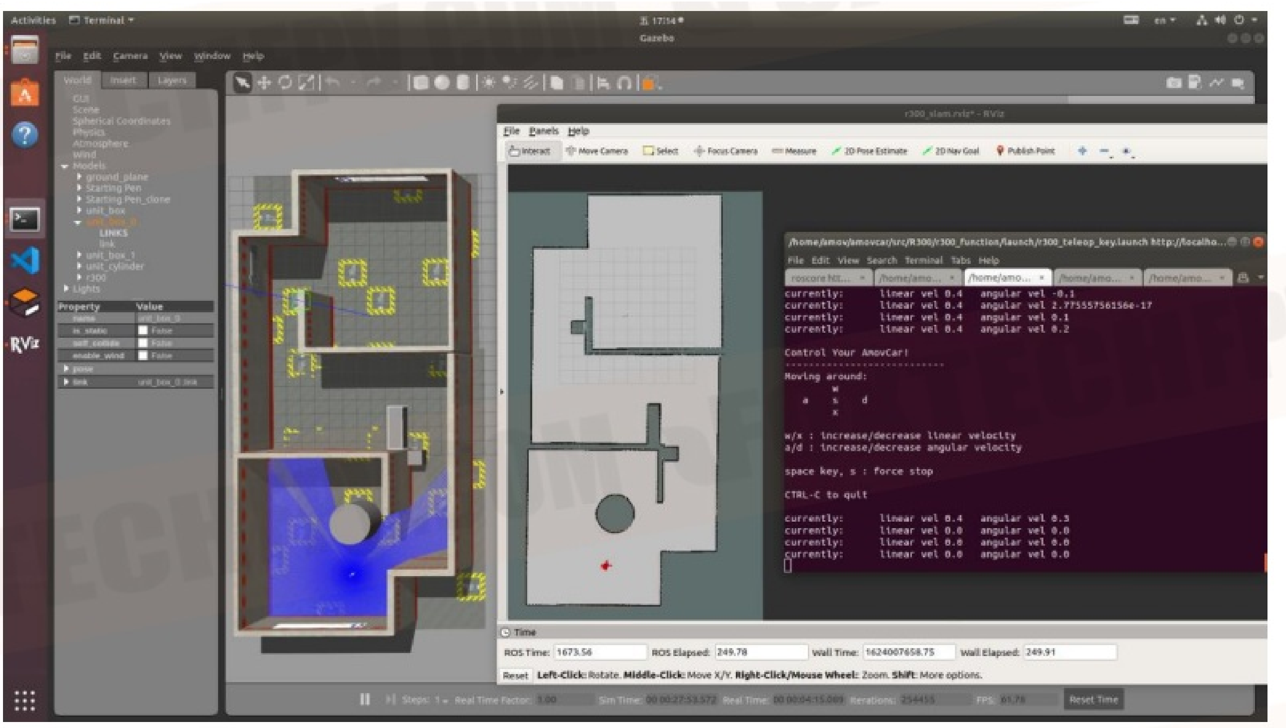


SLAM 2D Mapping Function

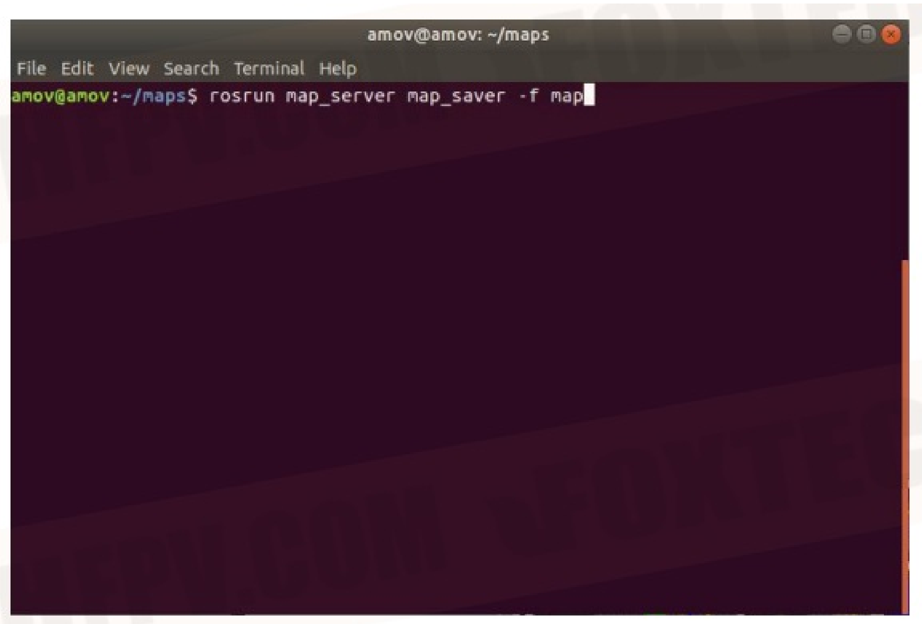
The started sh script is amovcar/src/R300/r300_simulation/sh/r300_simulation_slam.sh. The sh file contains the following parts:

1. Start the ros master node
2. Start R300 simulation, including simulation environment, UGV model, sensor simulation, TF, etc.
3. Keyboard control node to control the movement of UGV
4. 2D mapping function, mainly using 2D lidar data for mapping
5. The rviz visual interface, which is set for the SLAM 2D mapping function, displays maps and UGV models. Open a terminal arbitrarily, drag the r300_simulation_slam.sh file into the terminal window, and a command to start the sh script will appear. Press Enter to start it. Check the terminal window that pops up and whether the nodes in each terminal are started normally. After confirming that the nodes are running normally, enter the

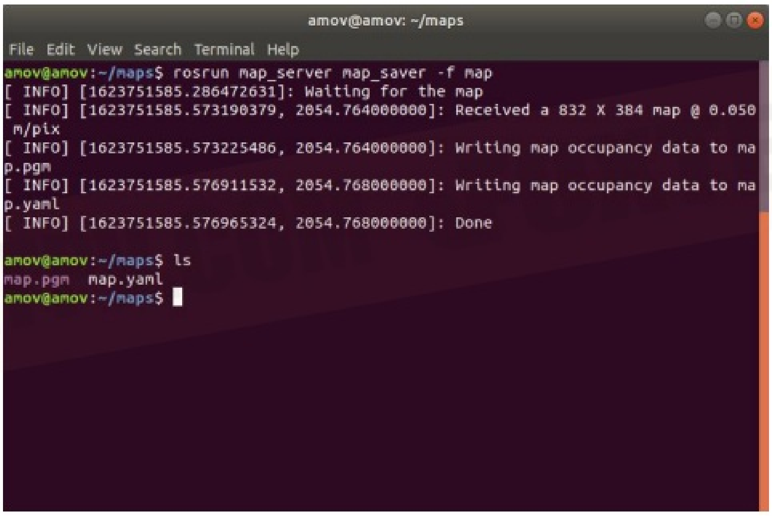
corresponding control instructions in the third keyboard-controlled terminal to control the movement of the UGV for 2D mapping.

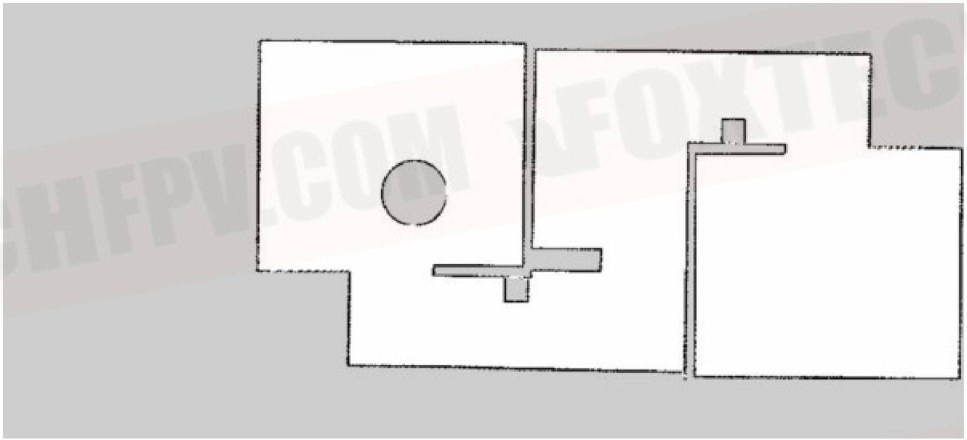


After the map is created, you can enter the following command to save the map: rosrun map_server map_saver -f map_name



After entering this command, files in .pgm and .yaml formats will be generated in the current folder, and the saved map file map.yaml and map.pgm can be viewed through the ls command.





Documents / Resources

	FOXTECH VG-450 UGV Lidar Mapping Robot [pdf] User Manual VG-450 UGV Lidar Mapping Robot, VG-450 UGV, Lidar Mapping Robot