miiboo

miiboo robot

Build Your Dream

usage manual

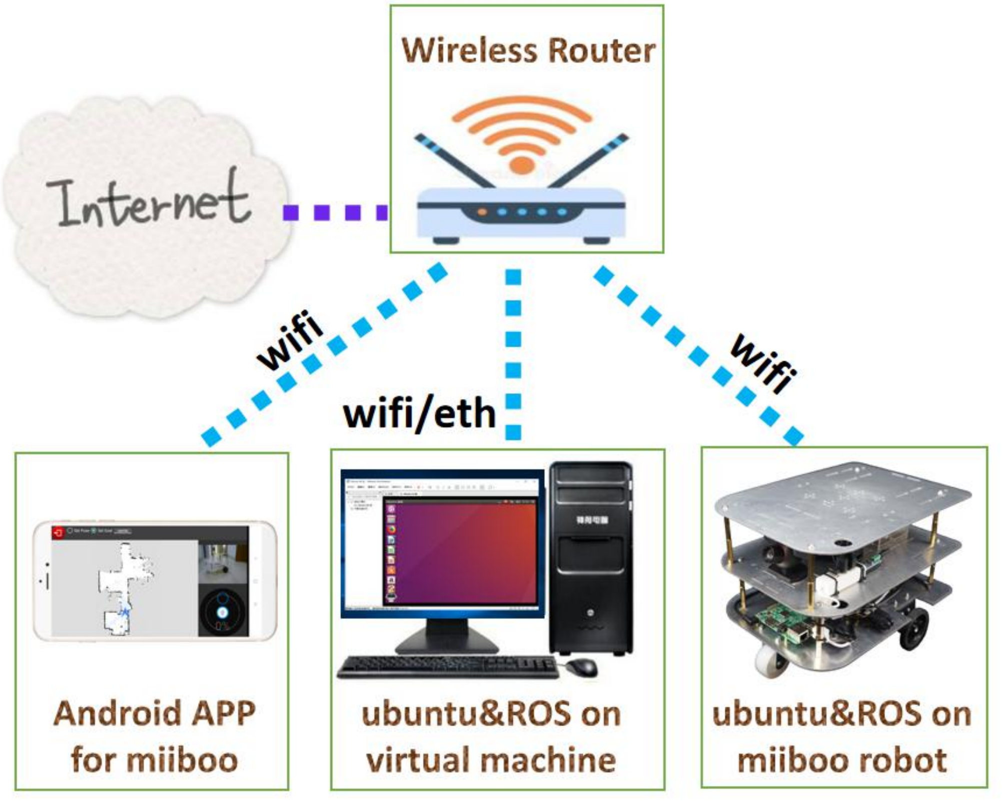
V 3.0

【About us】

website	*www.miiboo.cn*
github	*www.github.com/miiboo/*
amazon	*www.amazon.com/dp/B07X2HQ23D*
email	*robot4miiboo@163.com*

目录

1.Summary



1.1. Joint commissioning and development

The development of miiboo robot can be divided into three parts: robot, remote PC and mobile phone.Robot-side raspberry pi 3B installed Ubuntu and ROS system, mainly running sensor driver, SLAM program, navigation, etc. The remote PC is usually our desktop computer or notebook computer. By running Ubuntu and ROS system on the virtual machine, we use rviz GUI tools and SSH remote login tools to debug the program and algorithm functions on the robot.The mobile phone is equipped with APP, which is mainly responsible for displaying the status information of maps, images and positions returned by the robot, managing scheduling navigation and interactive tasks on the robot.

It should be noted that the remote PC and mobile phone are not necessary, they are only responsible for debugging and managing robots.If the SLAM mapping on the robot side has been completed and the executable task list program of the robot has been set up in advance, only the sensor driver, navigation program and executable task list program of the robot need to be loaded into the boot-up process. Once the robot starts, it can start its work on its own.

1.2. ROS Network Communication

Robot, remote PC and mobile phone communicate data through ROS network. ROS network communication is based on LAN networking, so it is necessary to use a wireless router for LAN networking.The environment variables of ROS network communication are configured on each host. Environment variables are used to specify MASTER and HOST hosts used in ROS network communications.

1.3. About Wireless Router

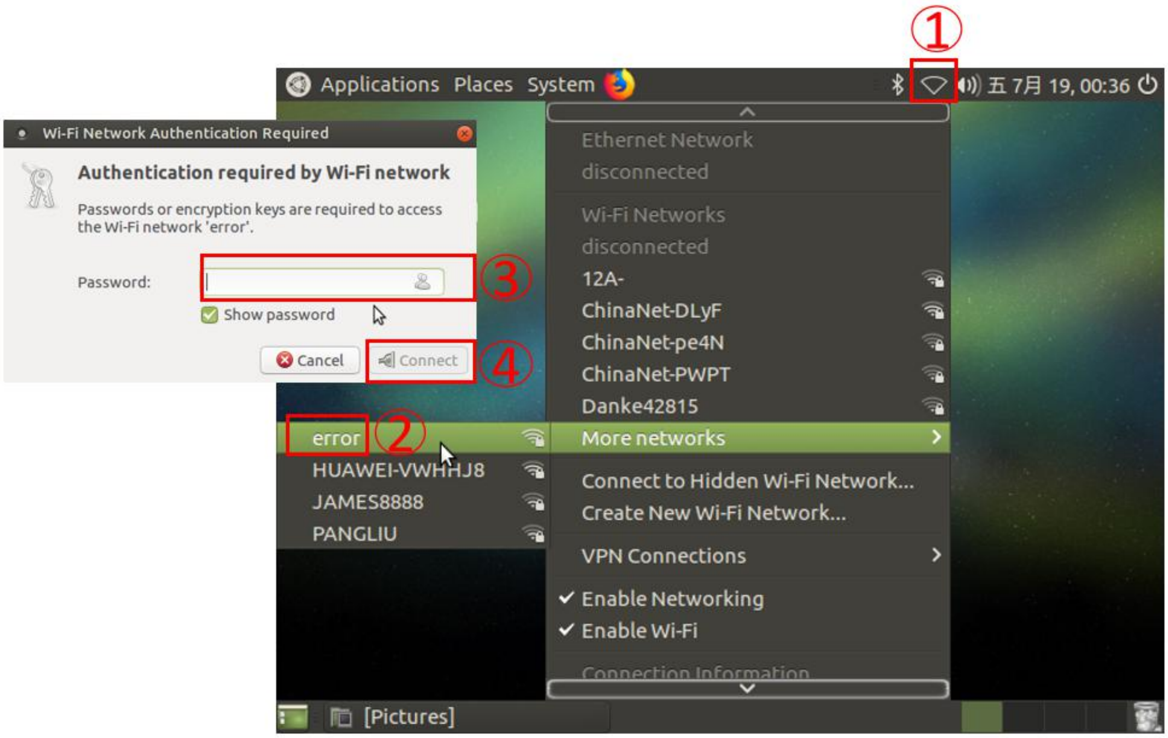
The wireless network card of raspberry pi 3B only supports 2.4 GHz 802.11.b/g/n band WiFi signal, but does not support 5 GHz 802.11.ac band WiFi signal. So we need to choose a wireless router that supports the WiFi band of raspberry pi 3B. Otherwise, the raspberry pi 3B system will not recognize the WiFi hotspot. Wireless routers can access the Internet through WAN network ports or other WiFi hotspots that can access the Internet through AP relay mode. At the same time, it ensures that the firewall, MAC filtering and IP address restriction rules on the wireless router are closed, so that the wireless router can assign legitimate LAN IP addresses to the access host devices.

2.Connect WiFi to raspberry pi

2.1. Miiboo robot starts for the first time

Take out the miiboo robot from the box and connect the prepared wireless keymouse and HDMI display to the robot. Then you can turn on the power switch to turn the robot on.
It should be noted here that the HDMI display is inserted first and then turned on. If the HDMI display is turned on and then inserted, it will not be displayed because it is unrecognized.

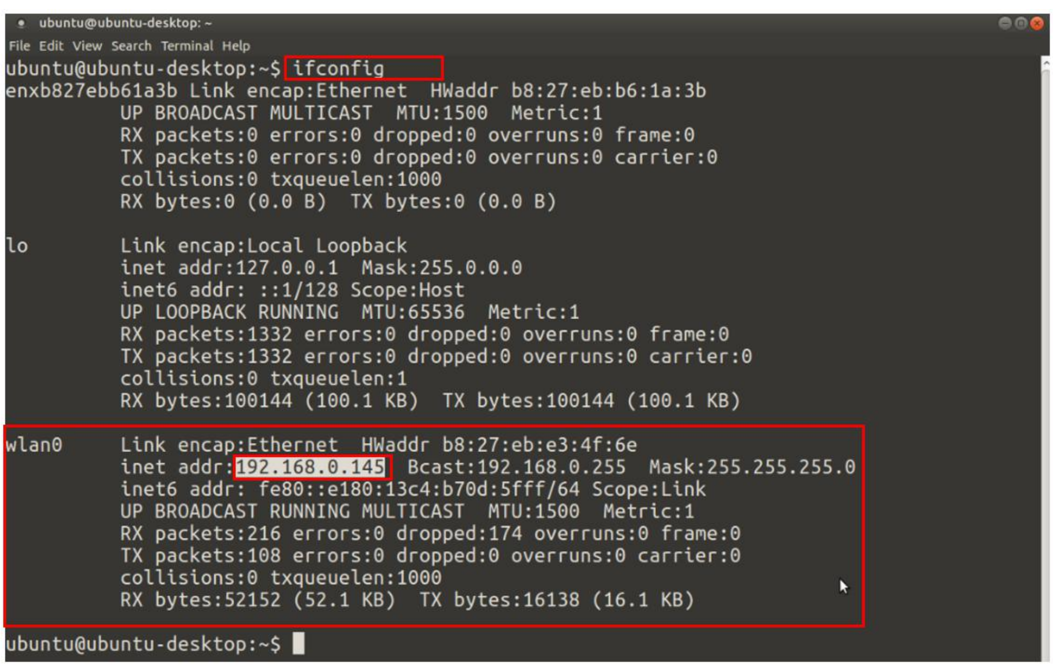
2.2. Setting up WiFi connection



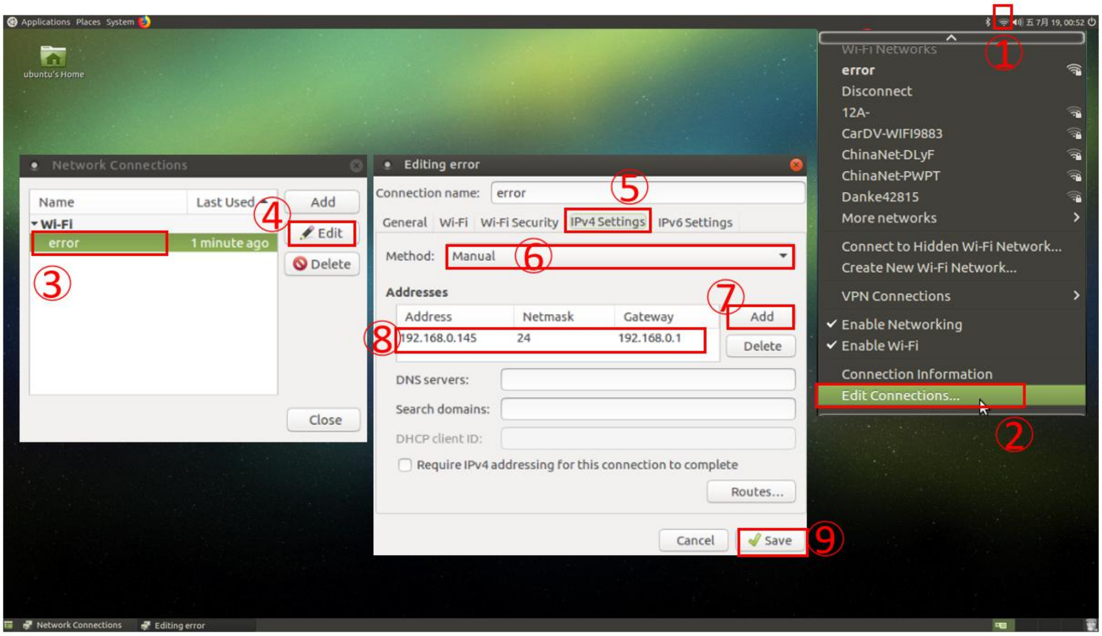
Click on the WiFi style icon in the upper right corner of the desktop to find the name of the WiFi hotspot emitted by your wireless router in the pop-up drop-down list. If you can't find your WiFi hotspot name, try continuing to find it in the more networks drop-down list. Then click on the WiFi hotspot name, Enter the connection password.

2.3. Setting up static IP address

We start the miiboo robot for the first time, connect the HDMI monitor to set the WiFi network connection, and setting up static IP address. In the future, robot will not need to connect to HDMI monitors, you can remote login robot from network.



First check the IP address of the current WiFi connection with ifconfig at the command terminal.

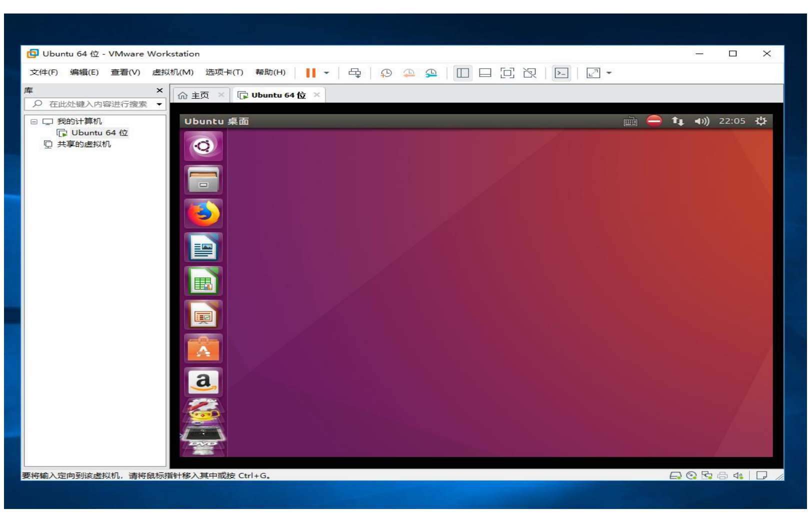


Click on the WiFi icon in the upper right corner of the desktop and select Edit Connections in the drop-down list to set up the network connection. Select your connection, Click the Edit button to set its parameters.Only IPv4 Settings need to be set.First, set IP allocation method to Manual,Then add IP address, subnet mask and gateway parameters manually with Add button in the address bar below,Finally, Save will do it.

Once this step is completed, the wireless keymouse and HDMI display on the robot can be removed. In the future, using robots, you only need to power up, and then log on to the robot remotely with the network.

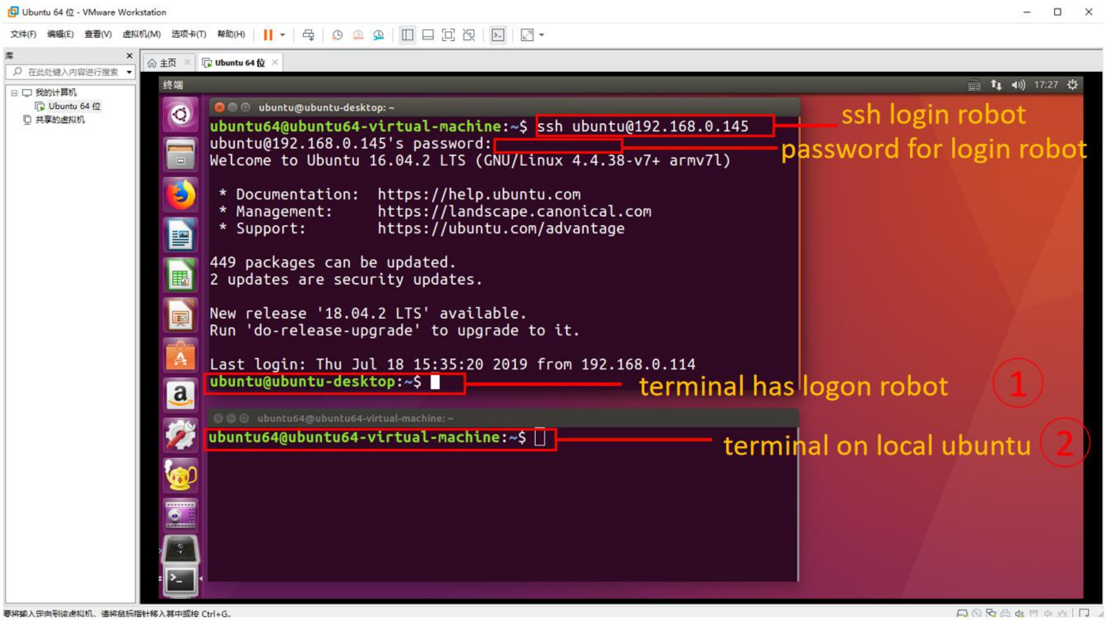
3.Log on to the robot through SSH on the remote PC

3. 1. Remote PC



Remote PC need to be installed ubuntu and ROS. Recommend running Ubuntu and ROS in virtual machine.

3.2. SSH usage



After opening the Ubuntu in the virtual machine, check that the virtual machine network has been connected to the physical machine through the bridge mode, so as to ensure that the virtual machine and the robot are in the same router's local area network. Then open a terminal, input SSH command:

$ ssh ubuntu@192.168.0.145

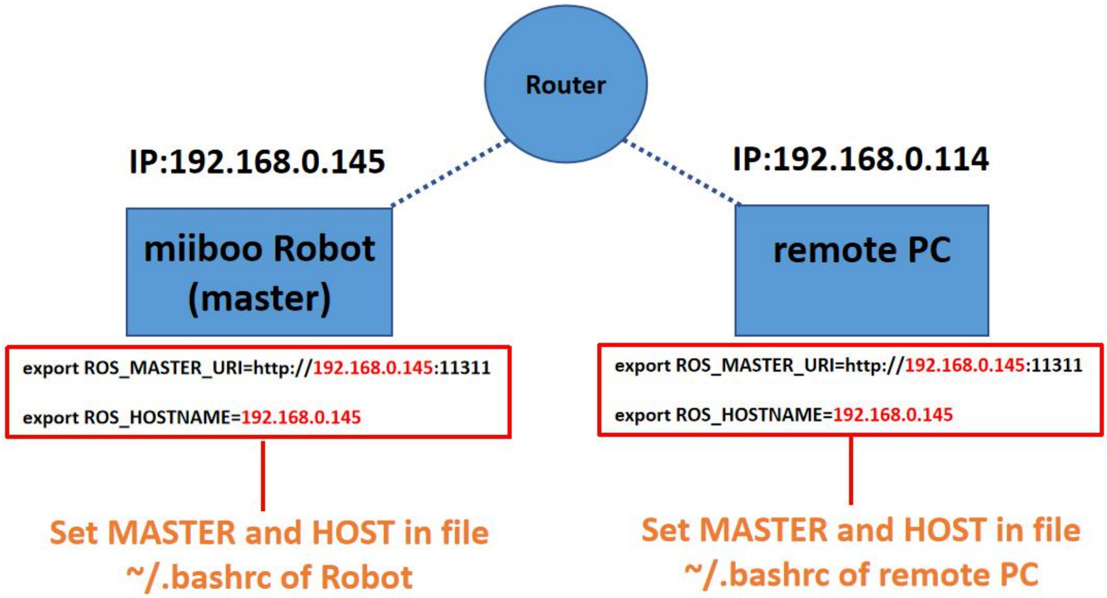
$ ubuntu@192.168.0.145's password: ubuntu

The user name is ubuntu, the login password is ubuntu, and the robot's IP address is 192.168.0.145 (which needs to be replaced with your actual value). Pay attention to distinguishing between local terminals and after-SSH terminals

4.ROS Network Communication between Remote PC and Robot

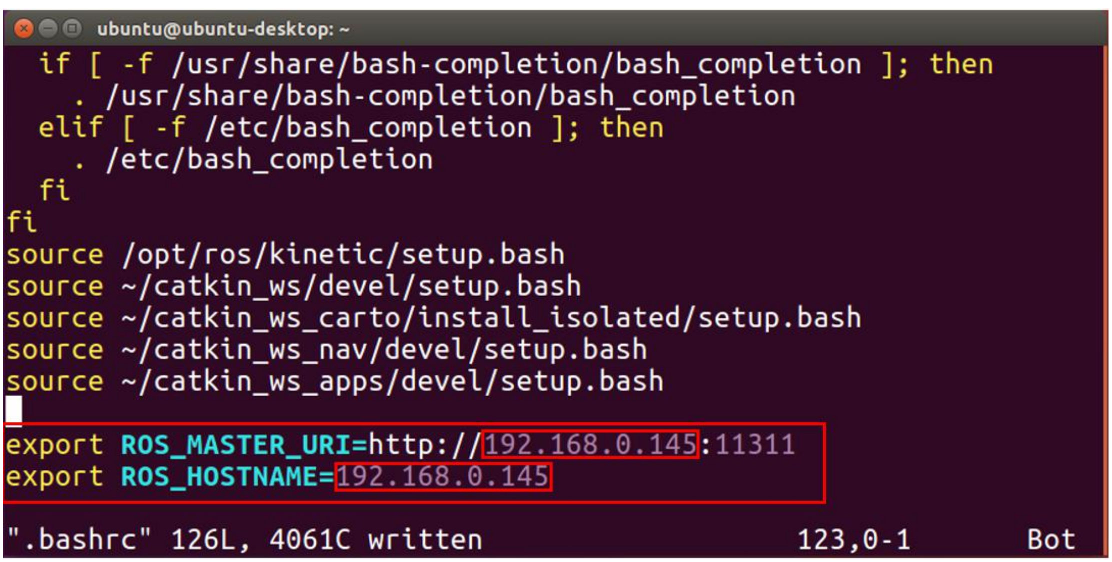
4.1. ROS Network Communication

ROS network communication is a way of distributed computing cluster, which supports the communication between multiple robots and program nodes running on desktop computers with ROS system installed respectively. Robot-side raspberry pi 3B installed Ubuntu and ROS system, mainly running sensor driver, SLAM, navigation, etc. Remote PC runs Ubuntu and ROS system on virtual machine, and uses rviz GUI tool and SSH remote login tool to debug the robot. Android APP mainly displays the status information of maps, images and positions returned by robots, manages navigation and interactive tasks of robot.



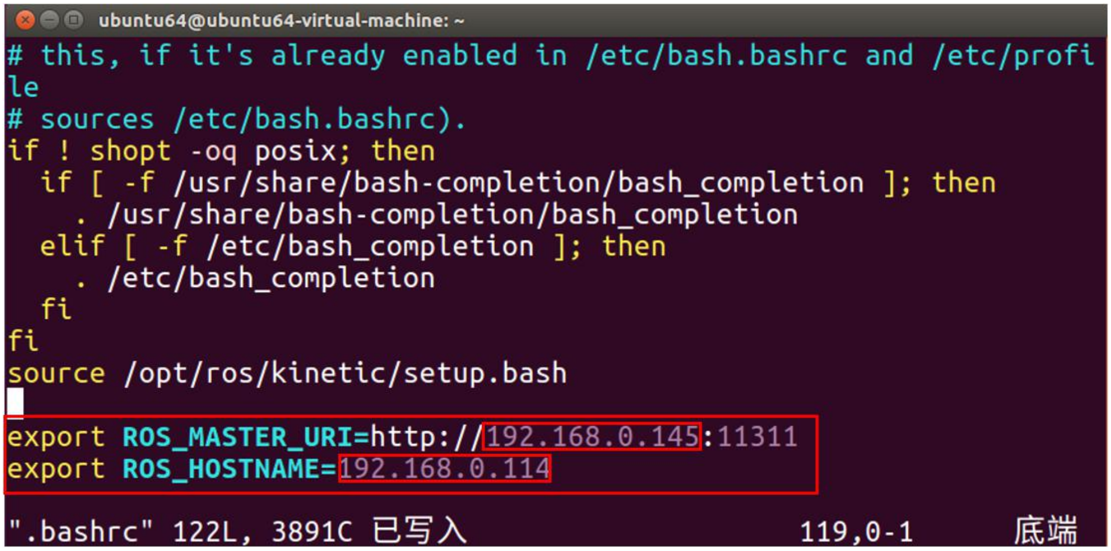
ROS network is a centralized structure,Each host must specify the same host as MASTER and declare its own host as HOST.

4.2. Setting up ROS Network Environment Variables on Robot



Open a terminal in the remote PC,then login to the robot with SSH.After successful login,open the ~/.bashrc file with vim, find the last line of the file, modify the IP address values of MASTER and HOST environment variables, then save and exit SSH login. Please replace the IP address value in the example with your actual value.

4.3. Setting up ROS Network Environment Variables on remote PC



Open a terminal in the remote PC, then open the ~/.bashrc file with vim, find the last line of the file, modify the IP address values of MASTER and HOST environment variables, then save. Please replace the IP address value in the example with your actual value.

4.4. Distinction between SSH Login and ROS Network Communication

After SSH remote login, all operations are done on the robot. In the process of ROS network communication, the program runs on their respective hosts.

5.SLAM Mapping

5.1. Start all sensors on the robot

Open a terminal in the remote PC, then login to the robot with SSH. After successful login, input command:

$ roslaunch miiboo_bringup miiboo_all_sensor.launch

5.2. Start cartographer mapping

Open a terminal in the remote PC, then login to the robot with SSH. After successful login, input command:

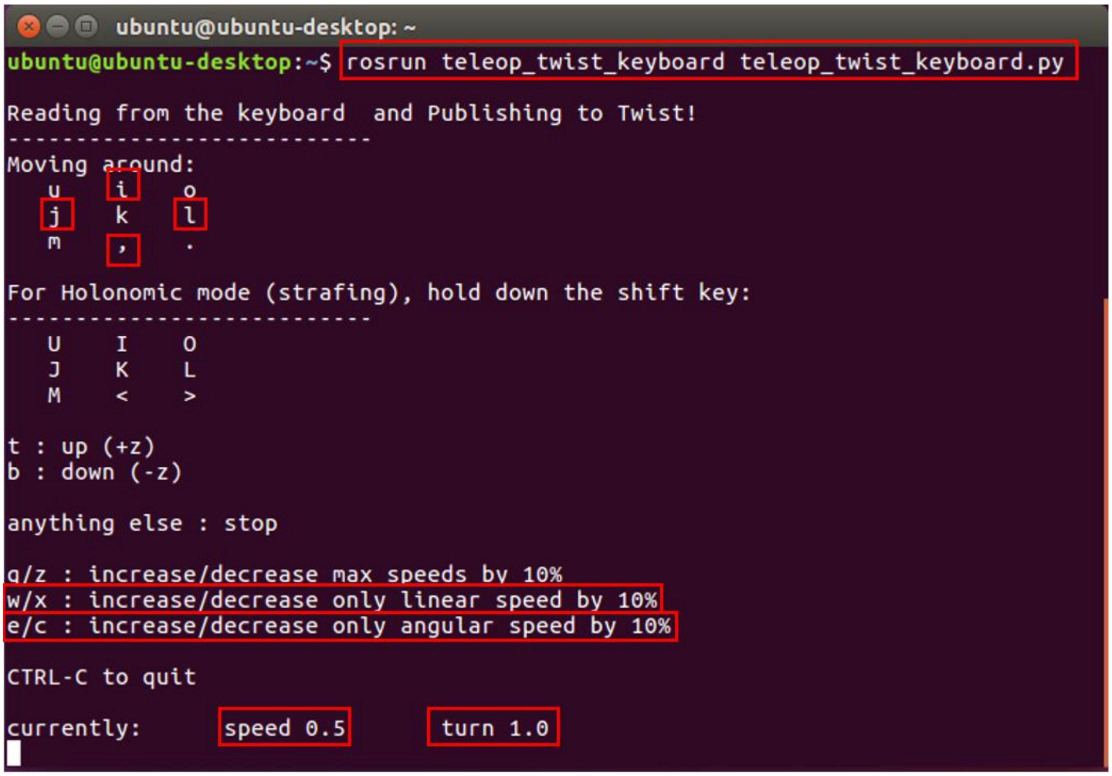
$ roslaunch cartographer_ros miiboo_mapbuild.launch

5.3. Mapping with Remote Controller

The first way: Open a terminal in the remote PC, then login to the robot with SSH. After successful login, use keyboard command to control robot, At the same time, use rviz to display the map in remote PC.

The second way: Open Android APP of miiboo, remote control and display the map.

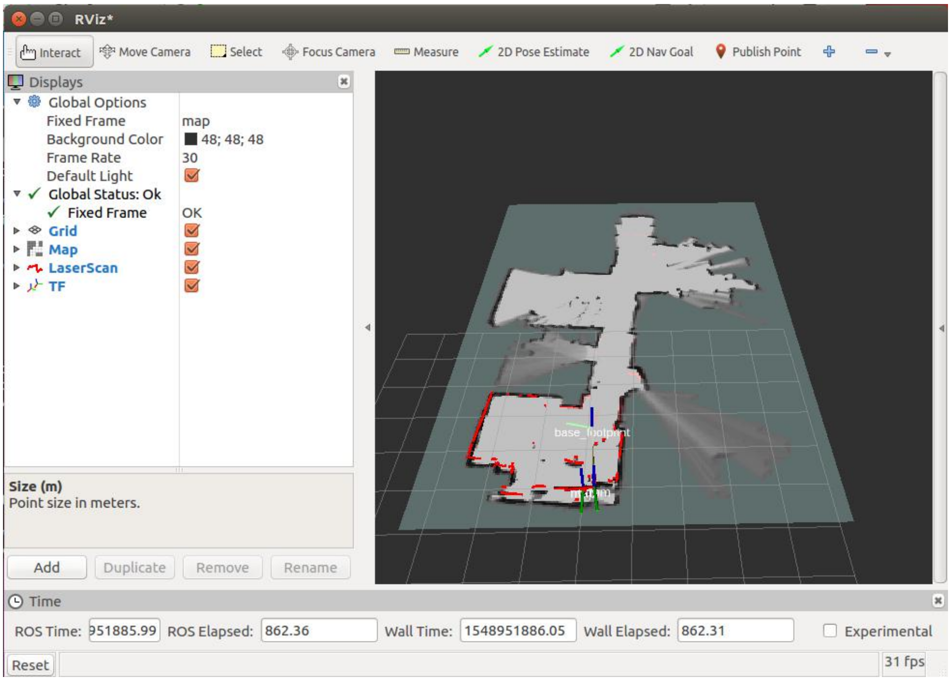
The first way:



Open a terminal in the remote PC, then login to the robot with SSH. After successful login, input command:

$ rosrun teleop_twist_keyboard teleop_twist_keyboard.py

In the program (i/, /j/l), the four buttons are (forward/backward/left/right), The two buttons are (increase/decrease) linear speed, The two buttons (e/c) are (increase/decrease) angular speed. It is suggested that speed should be set at about 0.2 and turn should be set at about 0.4. The drawing effect will be better at a slower speed.

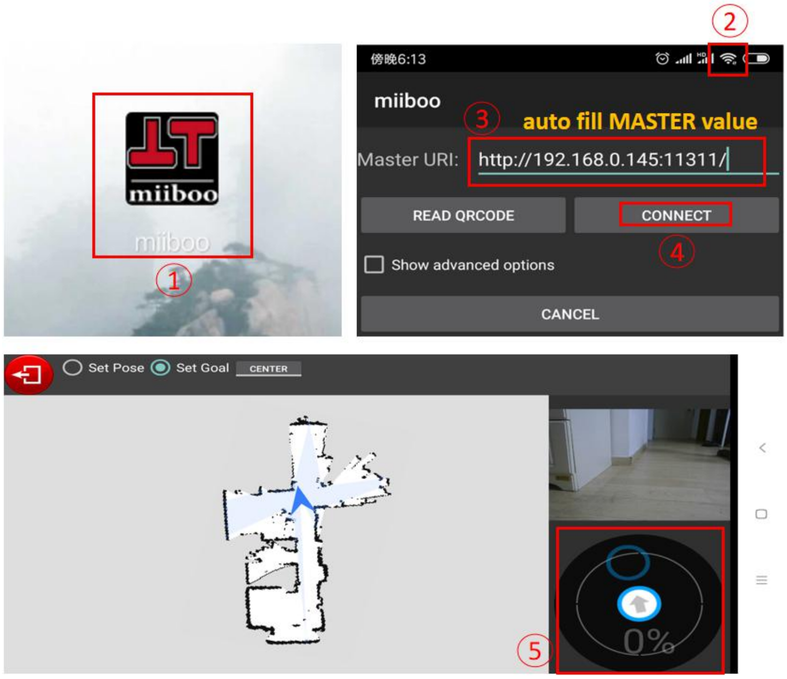


Open a terminal in the remote PC, then open rviz:

$ rosrun rviz rviz

We can subscribe to maps, lidar data, TF, etc. in rviz.

The second way:



Contact us to get the APP installation package, Connect the mobile phone and the robot to the same WiFi, Open Android APP of miiboo, then you can remote control and display the map.

5.4. Save map

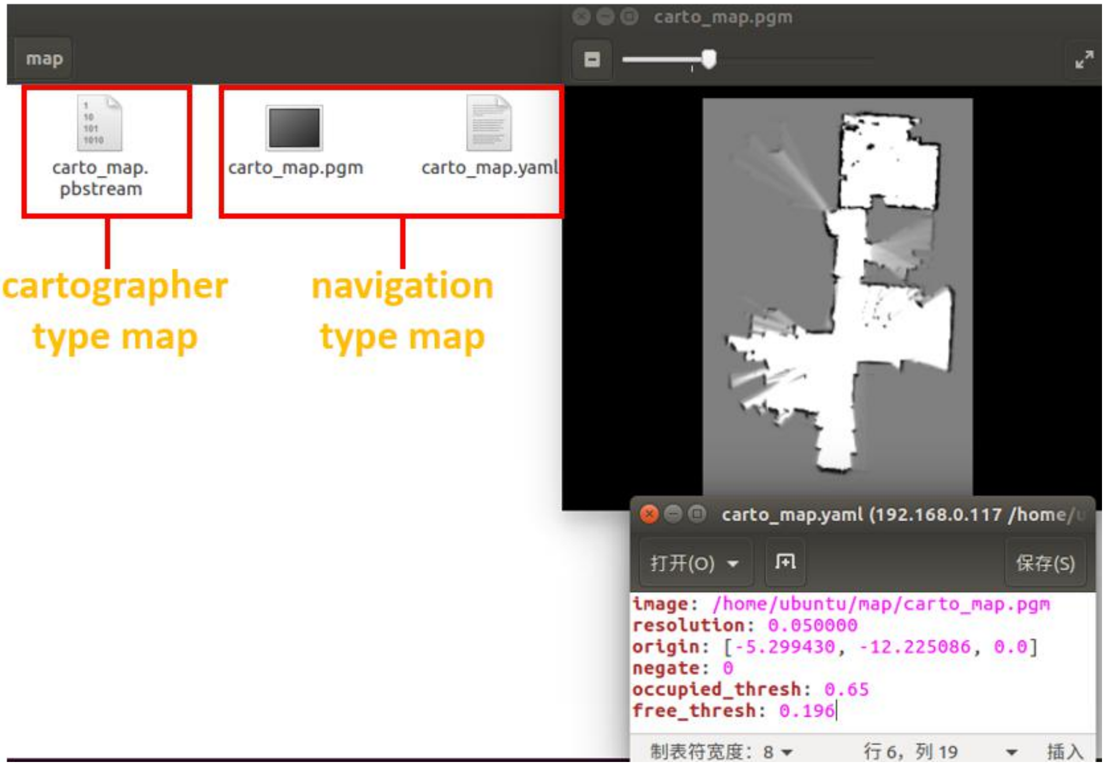
```
ubuntu@ubuntu-desktop:~$ rosservice call /write_state  /home/ubuntu/map/carto_map.pbstream
status:
  code: 0
  message: "State written to '/home/ubuntu/map/carto_map.pbstream'."
ubuntu@ubuntu-desktop:~$
```

When the environment scan is completed and the path loops back to the starting point, you can save the map. Open a terminal in the remote PC, then login to the robot with SSH. After successful login, input command:

$ rosservice call /write_state /home/ubuntu/map/carto_map.pbstream

When saved successfully, the corresponding status information is returned.

5.5. Convert format of map



Because the map built by cartographer is in pbstream format, the map used in navigation is in GridMap format. So we need to convert pbstream format to GridMap format. Open a terminal in the remote PC, then login to the robot with SSH. After successful login, input command:

```
$ roslaunch cartographer_ros miiboo_pbstream2rosmap.launch \
  pbstream_filename:=/home/ubuntu/map/carto_map.pbstream \
  map_filestem:=/home/ubuntu/map/carto_map
```

After the map format conversion is successful, SLAM mapping is completed, and all command-line terminal programs can be closed and exited.

6.Autonomous navigation

6.1. Start all sensors on the robot

Open a terminal in the remote PC, then login to the robot with SSH. After successful login, input command:

```
$ roslaunch miiboo_bringup miiboo_all_sensor.launch
```

6.2. Start autonomous navigation

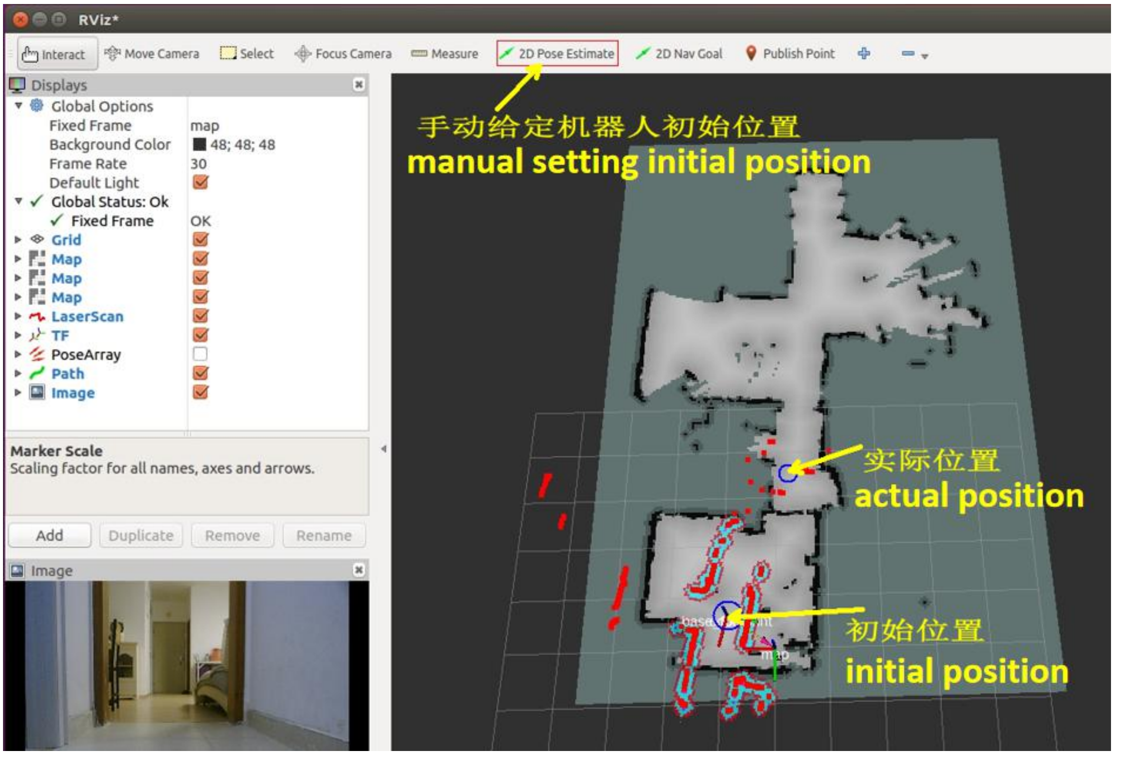
Open a terminal in the remote PC, then login to the robot with SSH. After successful login, input command:

```
$ roslaunch miiboo_nav miiboo_nav.launch
```

6.3. Send Navigation Target Point

After the navigation program on the robot is started, the robot enters the navigation standby state. It only needs to send the navigation target point to the robot, and the navigation task can begin to execute.

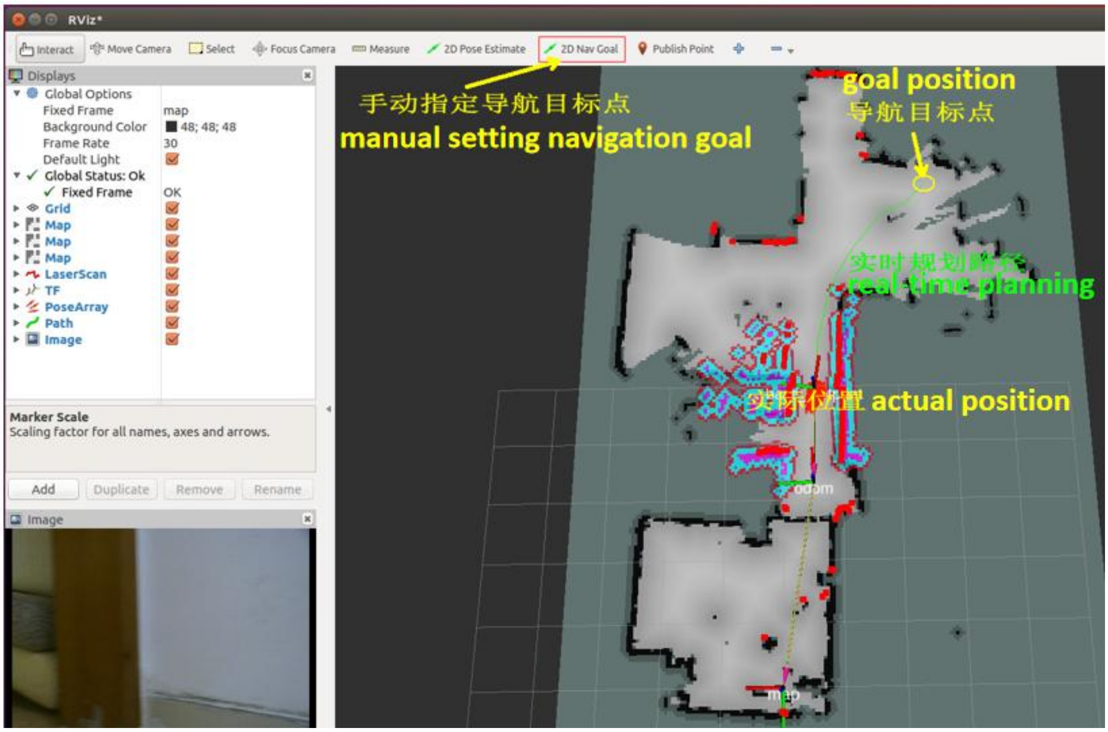
The first way:



Open a terminal in the remote PC, then open rviz, input command:

```
$ rosrun  rviz  rviz
```

If the initial position of the robot is incorrect, a correct initial position needs to be given manually with the [2D Pose Estimate] button.



Once the initial position is set correctly, you can manually specify the navigation target point with the [2D Nav Goal] button.

The second way:

Similar to the use of rviz above, The [set pose] button is used to initialize the pose of the robot, The [set goal] button is used to send the navigation target point.

7.Voice Interaction

Open a terminal in the remote PC,then login to the robot with SSH.After successful login,input command：

```
$ roslaunch  miiboo_asr  xf.launch
```

When the program starts, we hear a welcome note, and then we can talk to the robot and start our robot chat tour.