

# TurtleBot3 Waffle\_TX2 ROS 平台 用户使用手册

版 本：0.1.0

修订日期：2020/06/30

作 者：北京智能佳科技有限公司

## 目录

重要信息	5
一. 笔记本上安装虚拟机、 Ubuntu 系统	6
1.1 虚拟机的安装	6
1.1.1. windows7 系统建议安装 14.1 版本	6
1.1.2. windows10 系统建议安装 15.5 版本	13
1.2. Ubuntu 的安装	18
二. (TX2 版华夫) 系统安装	20
<i>Tx2 系统默认是安装好的, 由于镜像文件大于 20G, 无法上传百度网盘, 所以如有需</i> <i>要请联系我们客服; 下面主要是操作步骤:</i>	20
2.1. 准备工作	20
2.2. 备份镜像文件	20
2.3. 恢复镜像文件	20
三. opencr 系统安装	22
3.1. 安装 Arduino IDE	22
3.2. opencr 固件包下载	24
3.3. 端口设置	27
3.4. 固件编译与下载	28

四. 笔记本与 TX2 的通信	30
4.1. 使用 vnc 控制华夫 TurtleBot3-Tx2 开发板	30
4.2. IP 的配置	31
4.2.1. 获取 Remote PC 和 TurtleBot3 的 IP	31
4.2.2. 修改 Remote PC 和 TurtleBot3 的 IP	33
4.2.3. 保存 Remote PC 和 TurtleBot3 端的修改	35
4.3. 通信测试	35
五. 激活你的雷达	37
5.1. 操作步骤	37
5.2. 实际效果	37
六. kinect 测试	38
七. 底盘测试	40
7.1. 前进测试	40
7.2. 转圈测试	40
7.3. 启动测试	40
八. 键盘控制	42
九. 建图-gmapping 建图 (A2 雷达)	44
9.1. 说明	44

9.2. 通过远程操作创建地图	44
十. 自主导航 (A2 激光雷达)	47
十一. 建图-karto 建图	49
十二. 建图-hector 建图	50
十三. RC100 遥控杆控制	51
13.1. 遥控器说明	51
13.2. 遥控器的安装	51
13.3. 遥控器的操作说明	52



## 重要信息

**注意：在安装机器人之前必须先看电子文档，如果由于不看文档造成接线出现问题一律不再报修范围内；智能佳不对用户因疏忽或误用产品而导致的任何损失或损害负责。**

请在开始之前请仔细阅读电子使用说明文档。

1.严格按照说明手册以及官方网站的网页教程操作；

教程网址 1: <http://www.rosrobot.cn/>

教程网址 2: <http://emanual.robotis.com/docs/en/platform/turtlebot3/overview/>

2.在接线的时候要断电操作，要反复确认，确保接线（电源线）正确，电源正负正确，因接线问题造成机器人损害不在报修范围内。

3. 只能使用套件中包含的电池和充电器。电池一个月最少要充电一次，长时间搁置导致电池过放而不能正常使用也不再报修范围内。

4.当机器人运行时，理清线路，谨防无意间拖拽机器人造成机器人损坏。

5.请勿将机器人放置在靠近水源，热源或火源的地方。

6.冬天容易产生静电，在接触控制板时一定先去除静电，如果由于静电作用造成损坏也不在报修范围之内。

7.文档中或官方网站教程中未提及的做法，要提前询问相关技术人员，谨慎操作。

**重要提示：**[Remote PC]代表 PC 端、[TurtleBot3]代表 TX2 端；

## 一. 笔记本上安装虚拟机、 Ubuntu 系统

### 1.1 虚拟机的安装

#### 1.1.1. windows7 系统建议安装 14.1 版本

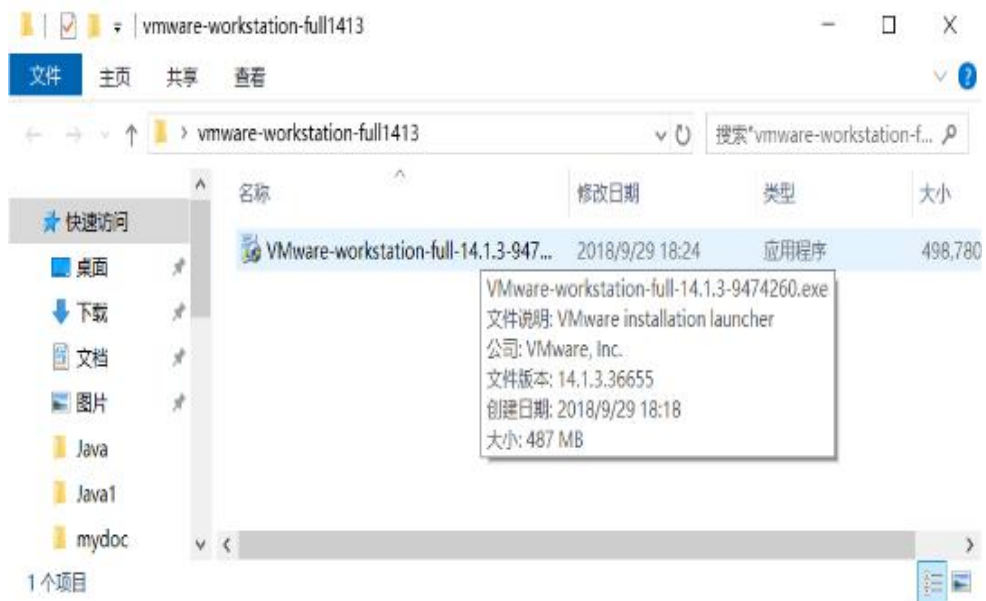
VMware workstation 百度云链接:

链接: <https://pan.baidu.com/s/1q6Lh9fMuXcZENSow0oPNvQ>

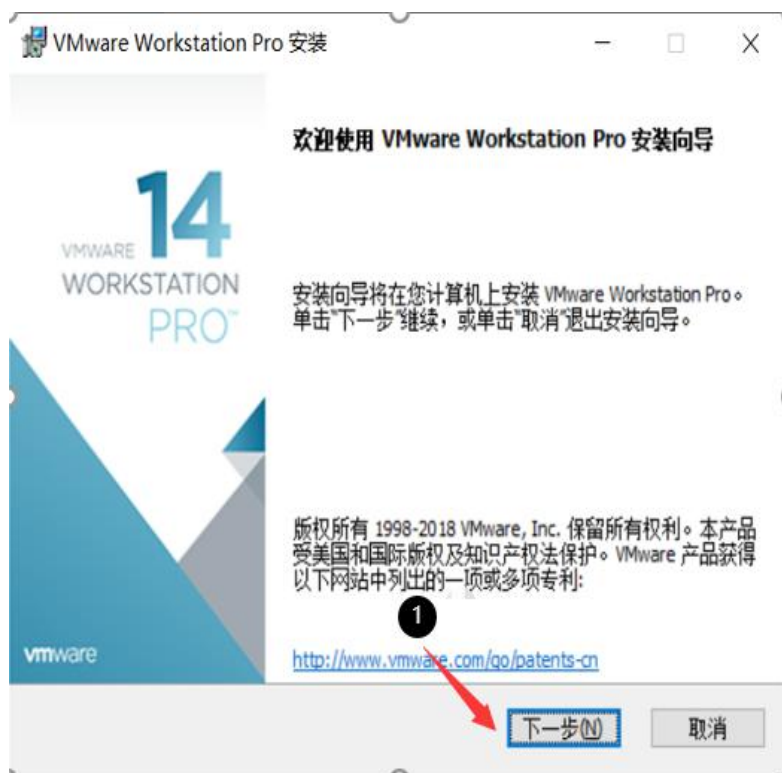
提取码: onja

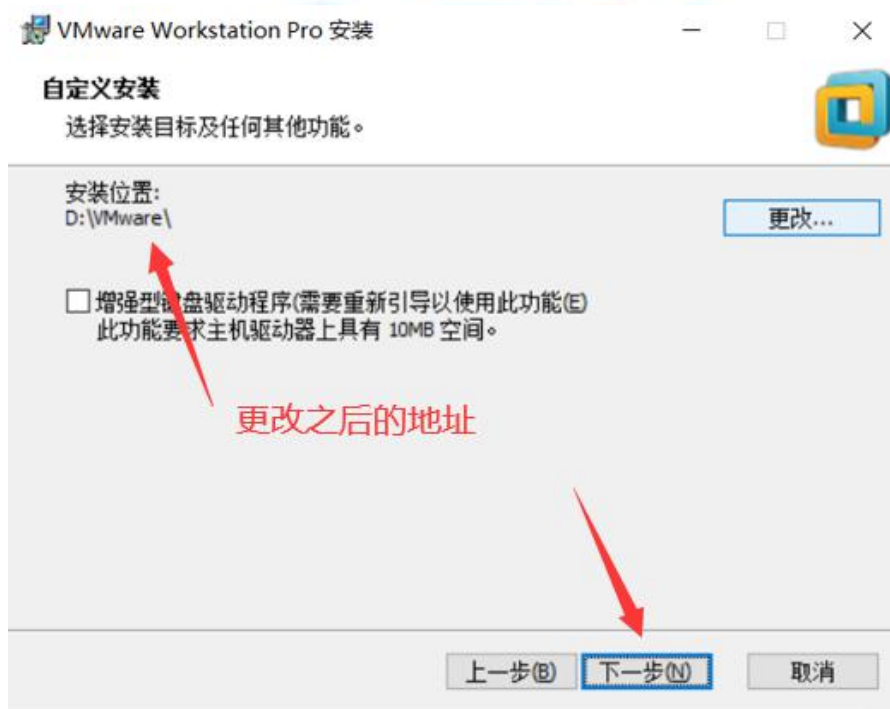
1. 下载好之后, 进行解压缩。双击解压缩文件即可进行安装。此后按照图片的顺序点击。



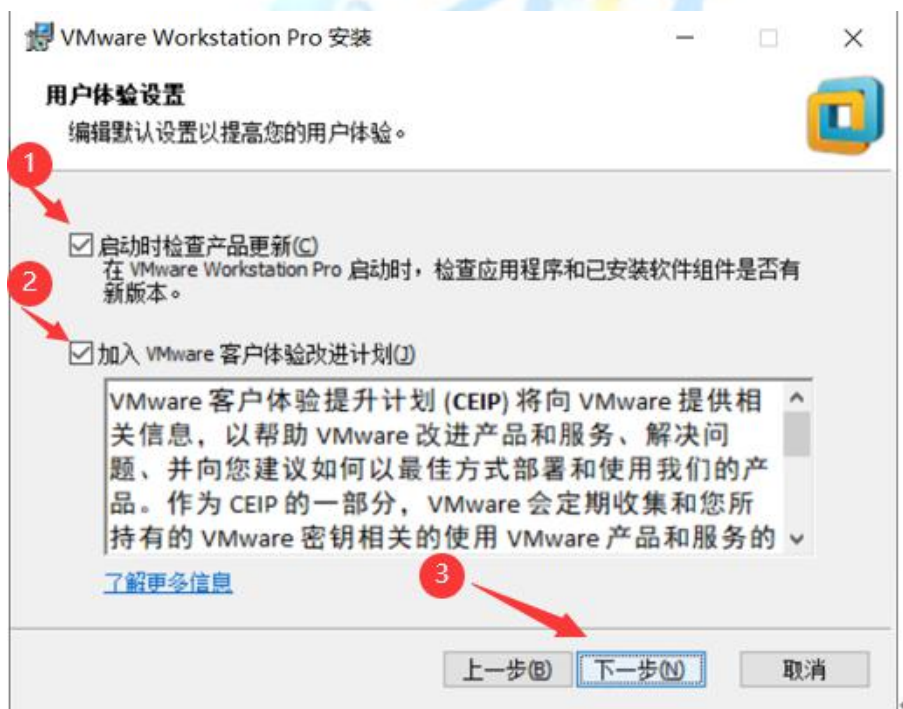
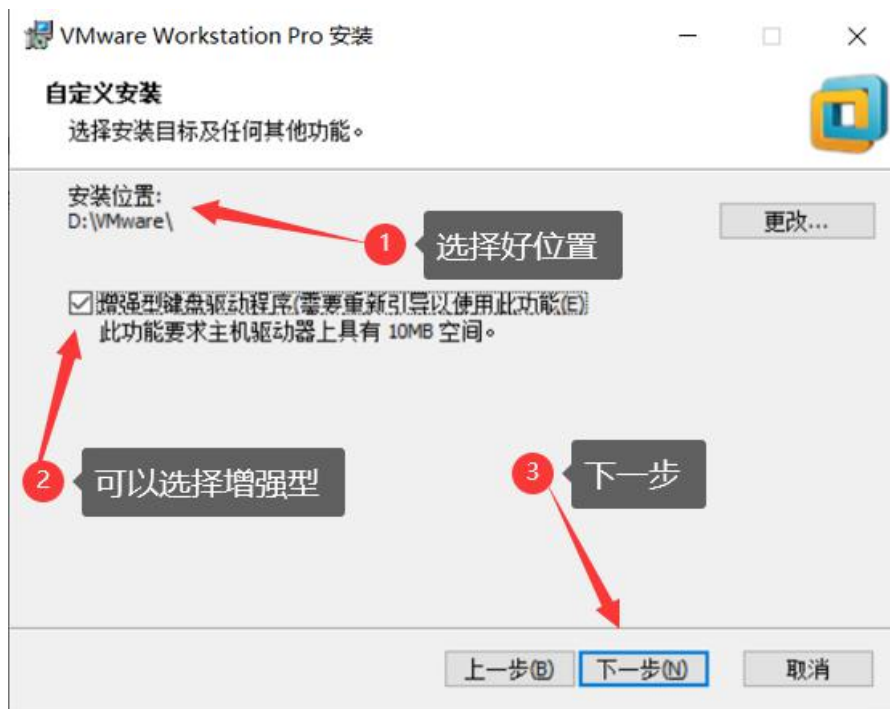


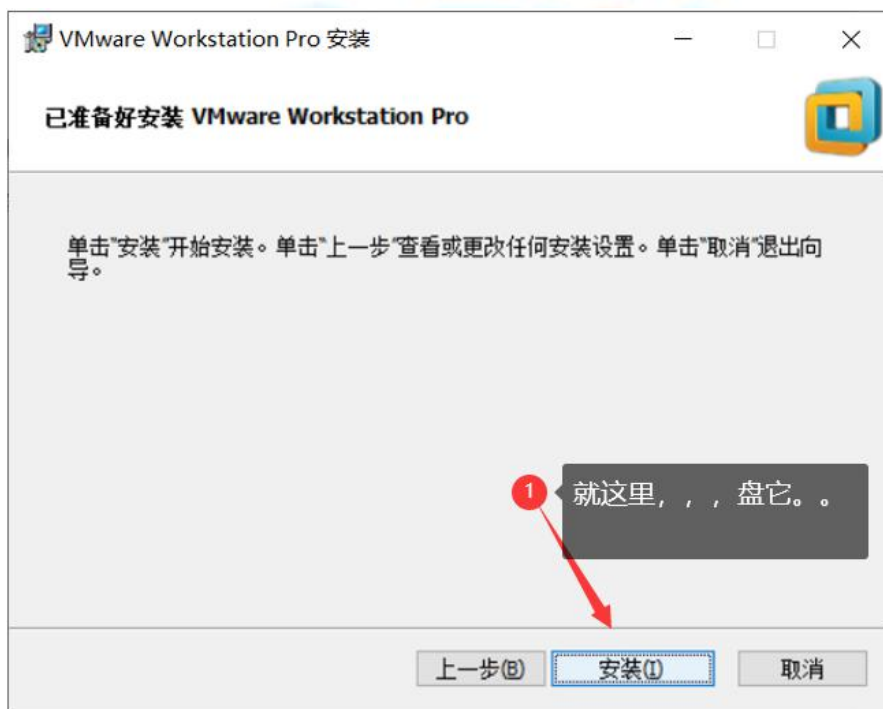
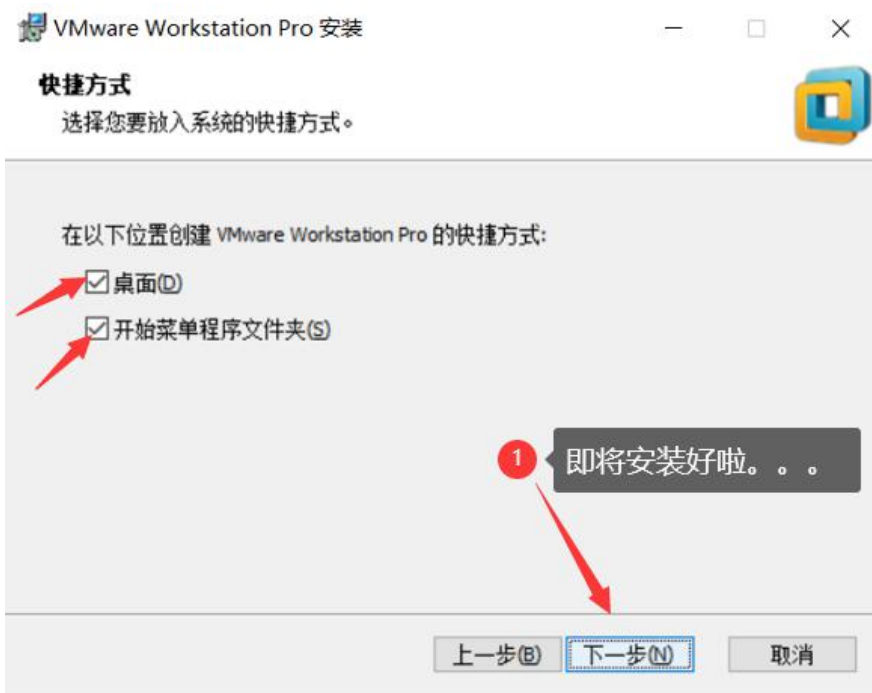
2. 点击下一步和我接受许可协议:

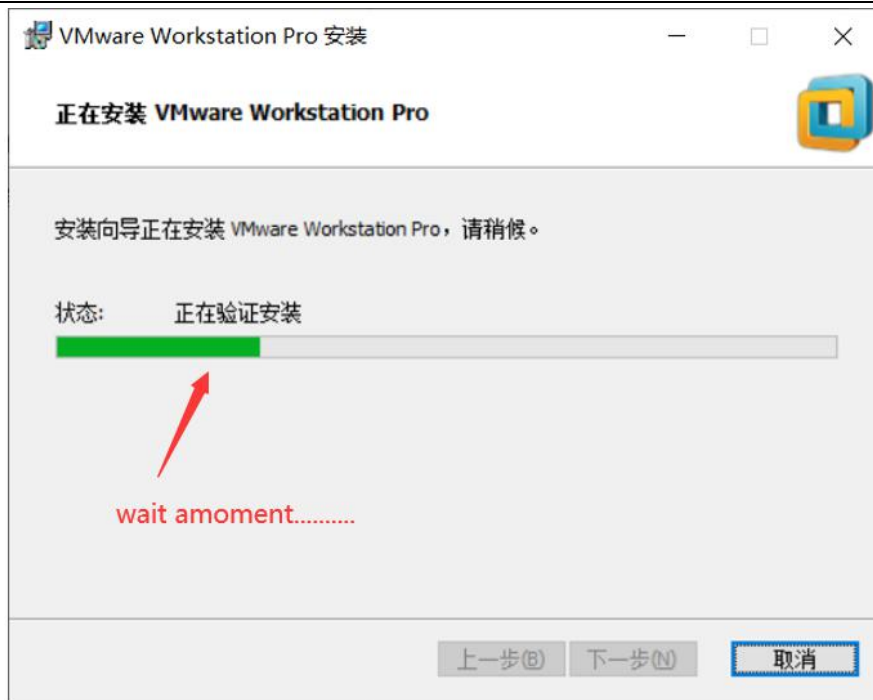


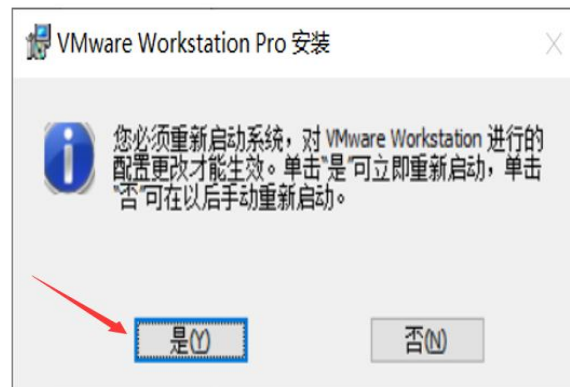




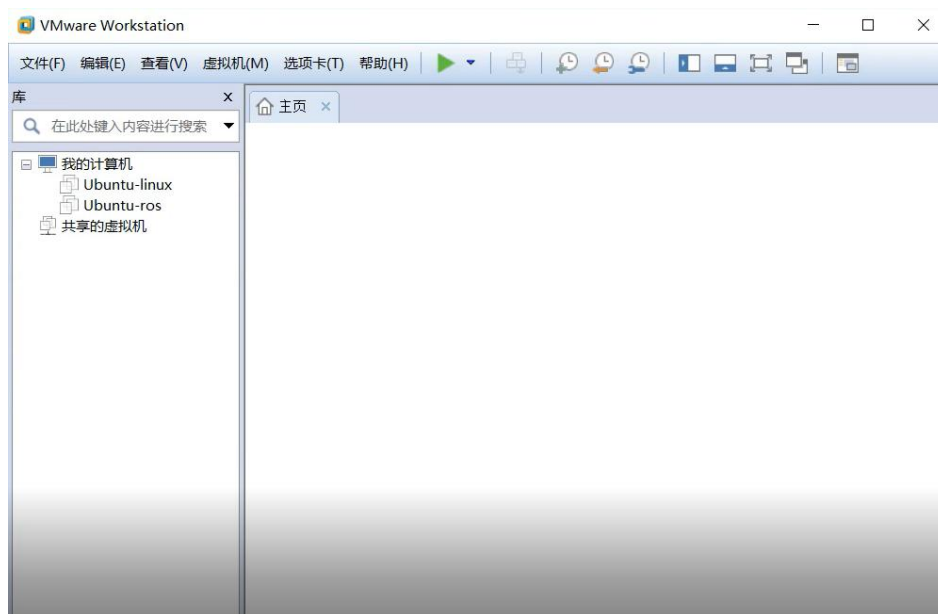








安装好打开之后的样子：



### 1.1.2. windows10 系统建议安装 15.5 版本

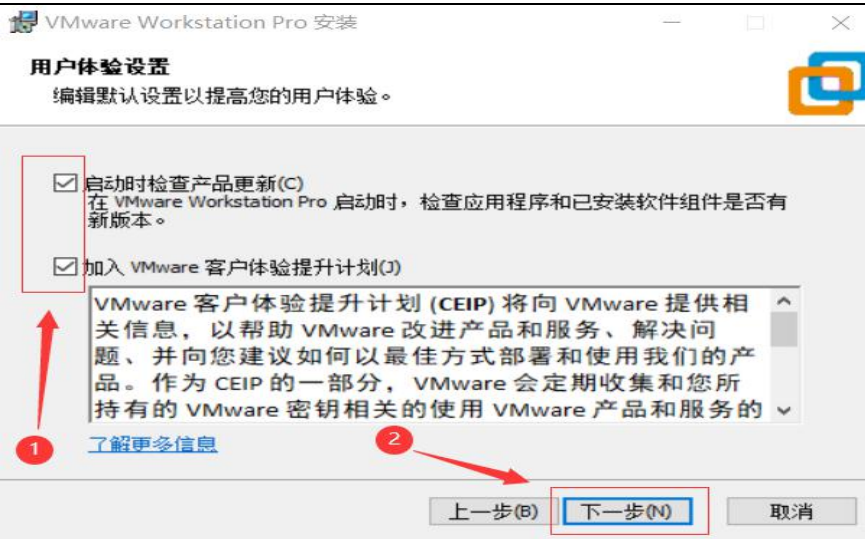
1. 下载提供的 15.5.2 的安装包，双击打开。



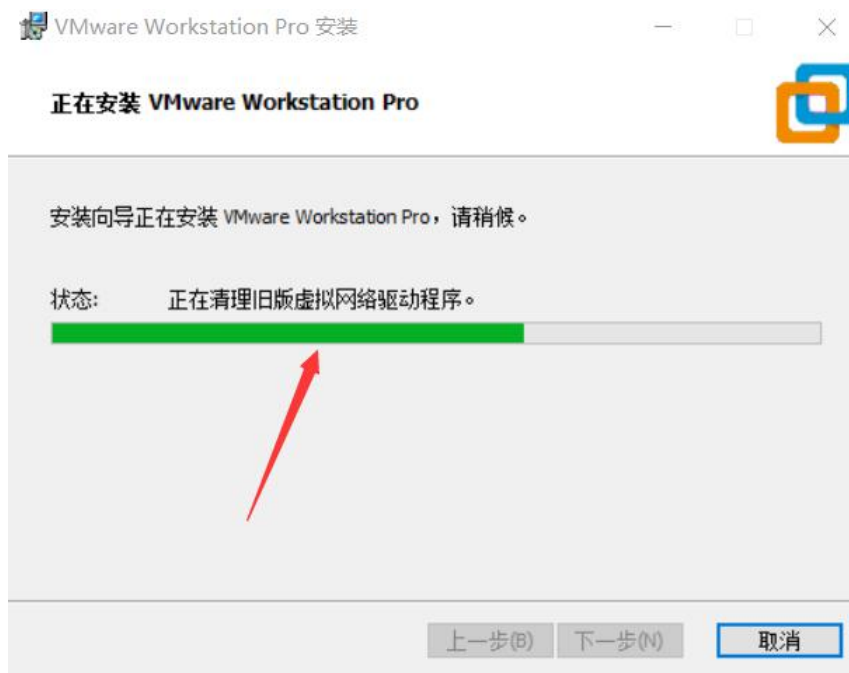
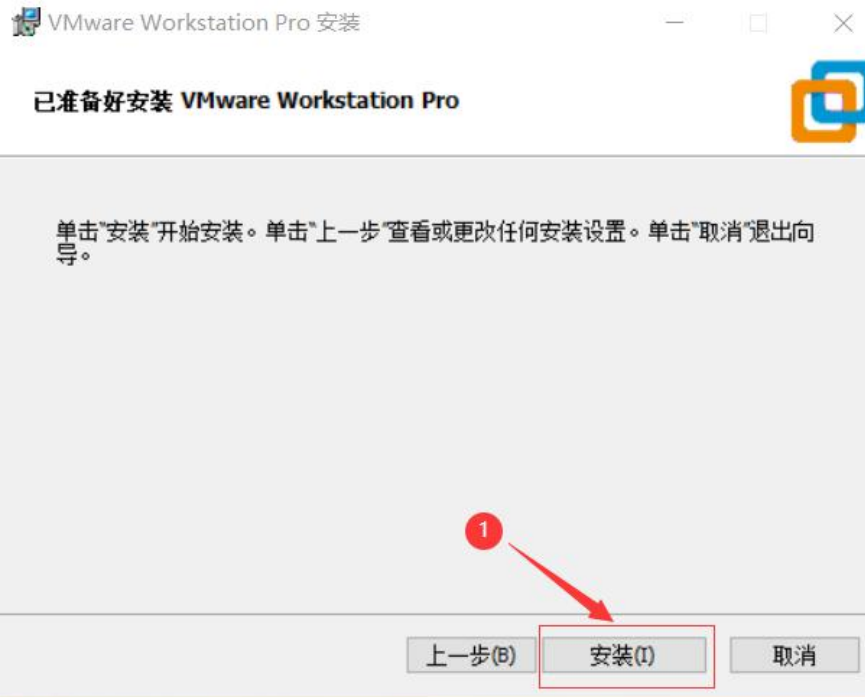
2. 按照图片的提示进行操作。





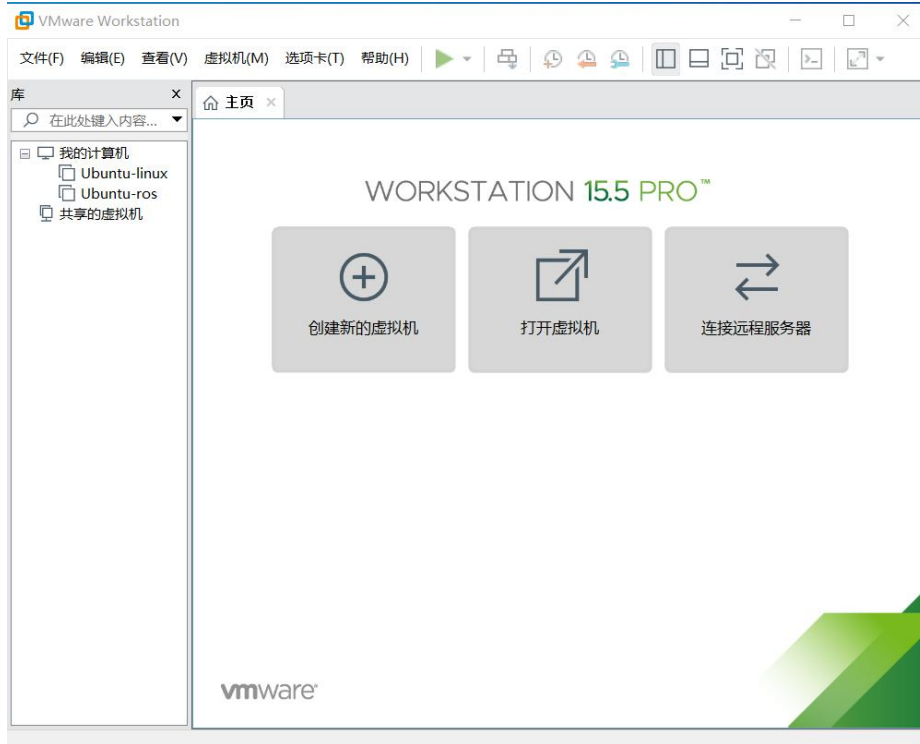






安装好之后的样子





## 1.2. Ubuntu 的安装

本公司提供两种 Ubuntu，一种是纯净版本，需要自己配置文件，一种是已经配置好的。

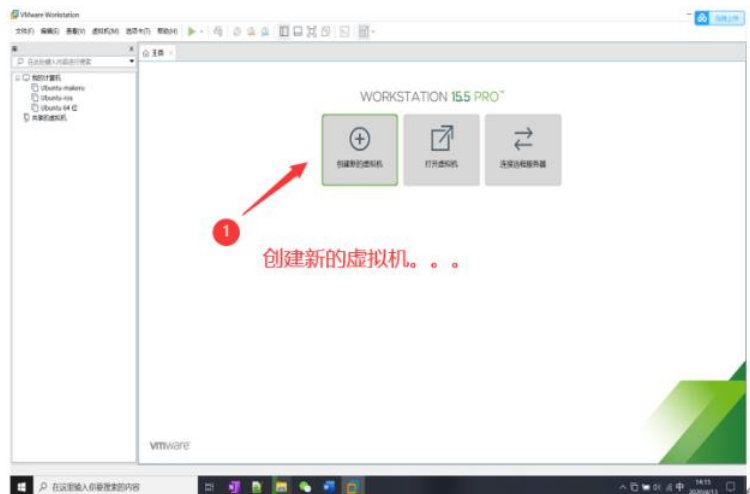
我们先来演示已经配置好的文件。

Ubuntu 16.04 百度云链接

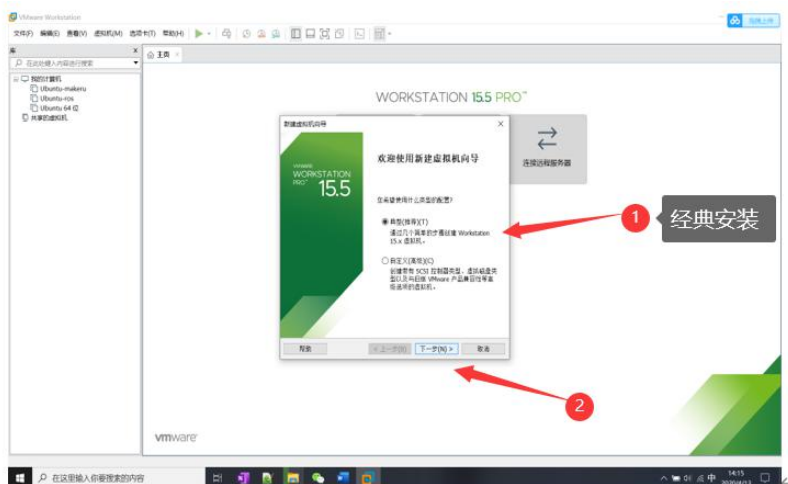
链接: <https://pan.baidu.com/s/1DrhHm9yer3G6qMb3CWse4w>

提取码: svqk

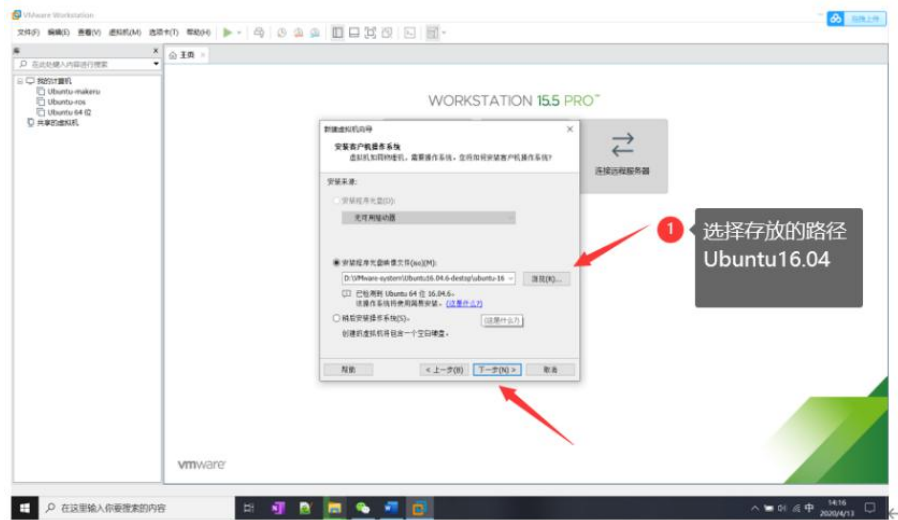
1. 打开已经安装好的虚拟机，点击创建新的虚拟机。
2. 选择经典安装即可，点击下一步。



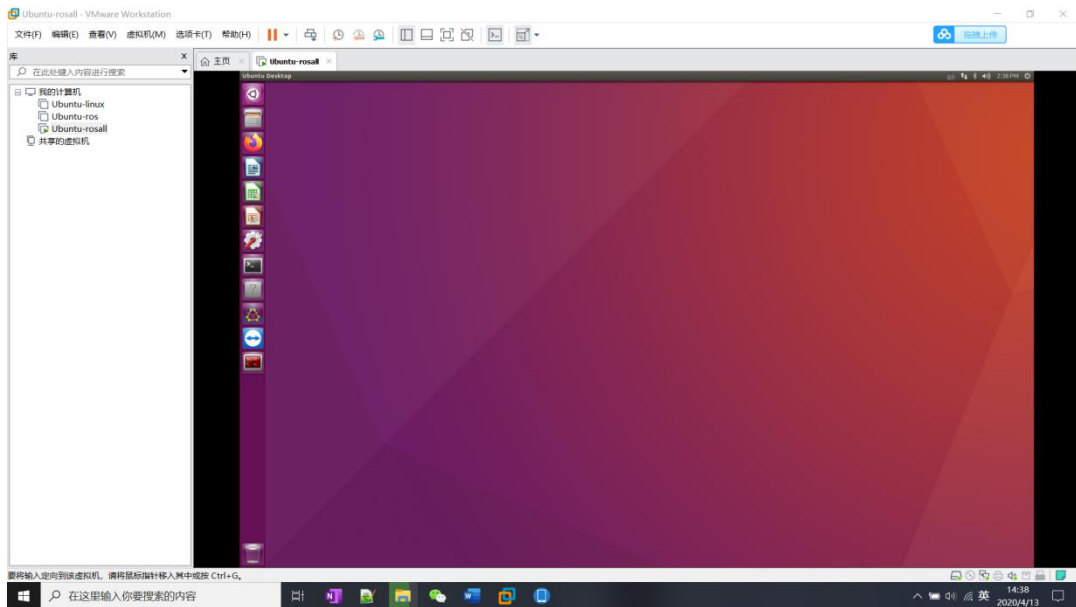
点击下一步。



3. 在浏览窗口内找到下载好的镜像的目录选择即可。之后点击下一步。



#### 4. 完成安装之后的样子:



**特别提示：如果您安装了不同版本的虚拟机，有可能会造成镜像打不开。**

## 二. (TX2 版华夫) 系统安装

**Tx2 系统默认是安装好的，由于镜像文件大于 20G，无法上传百度网盘，所以如有需要请联系我们客服；下面主要是操作步骤：**

### 2.1. 准备工作

- a. 准备好利用 Jetpack 刷过机的 Ubuntu 的主机 (HOST PC)
- b. 在利用 Jetpack 刷机时候，建立目录 TX2，并保存有 JetPack-L4T-\*-linux-x64.run 文件
- c. 再进行第一次刷机之后会增加其他目录，并生成子目录 64\_TX2；

### 2.2. 备份镜像文件

- a. 进入 HOST PC 的 JetPack 安装目录下 BootLoader 目录：

```
cd ~/TX2/64_TX2/Linux_for_Tegra_64_tx2/bootloader
```

- b. 连上 TX2，并进入 recovery 模式（通电，先按开机键 2 秒后松开，再按住 recovery 键 3 秒，同时按一下 reset 键），通过 lsusb 命令检查是否出现 **0955:7140 Nvidia Corp**，出现即代表进入 recovery 模式成功；
- c. 从 TX2 下载镜像：

```
sudo ./flashNew.sh -r -k APP -G my_backup.img jetson-tx2 mmcblk0p1
```

大概需要 30 分钟左右完成镜像下载，文件备份在 ~/TX2/64\_TX2/Linux\_for\_Tegra\_64\_tx2/目录下；

### 2.3. 恢复镜像文件

- a. 进入 HOST PC 的 JetPack 安装目录下 BootLoader 目录：

```
cd ~/TX2/64_TX2/Linux_for_Tegra_64_tx2/bootloader
```

- b. 复制 my\_backup.img.raw 为 system.img

```
sudo cp my_backup.img.raw system.img
```

- c. 连上 tx2，并进入 recovery 模式（通电，先按开机键 2 秒后松开，再按

住 recovery 键 3 秒，同时按一下 reset 键)，通过 lsusb 命令检查是否出现 **0955:7140 Nvidia Corp**，出现即代表进入 recovery 模式成功；

d. 退出到上一层目录，使用 flash.sh 开始烧录：

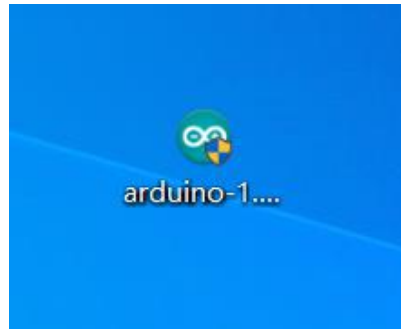
```
cd ../  
sudo ./flash.sh -r jetson-tx2 mmcblk0p1
```

大概需要花费 15 分钟左右完成恢复镜像工作；

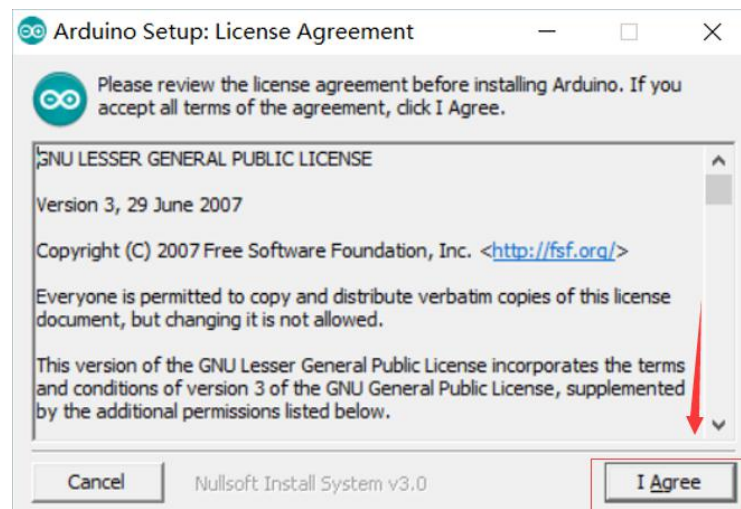


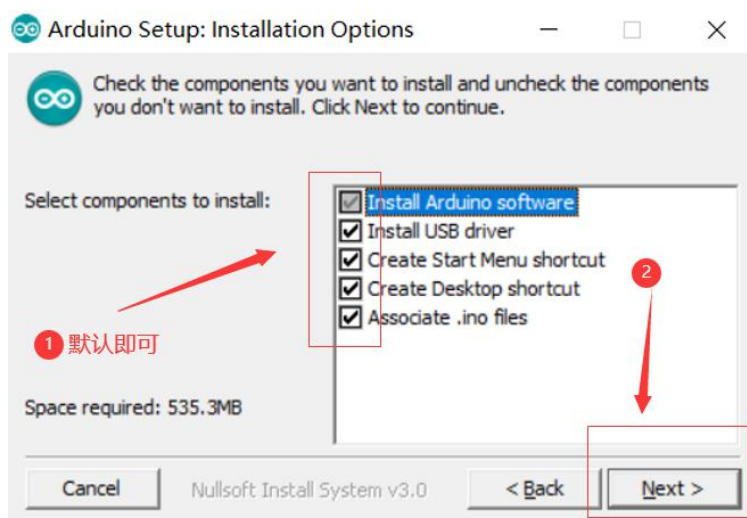
## 三. openCR 固件下载

### 3. 1. Arduino IDE 安装

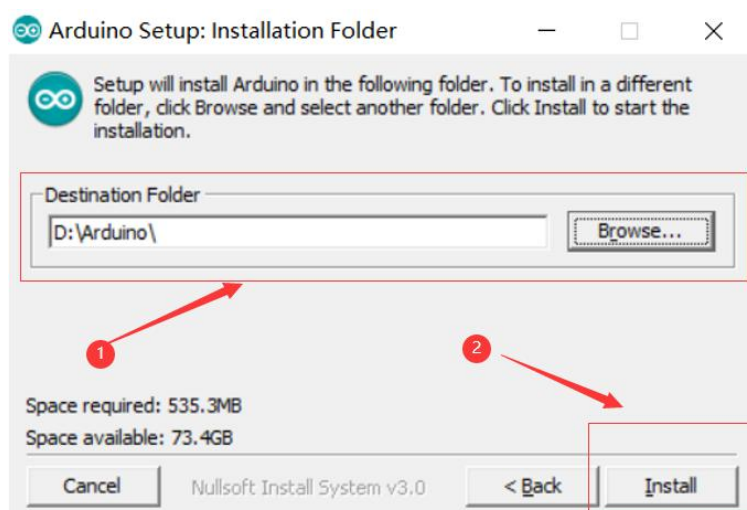


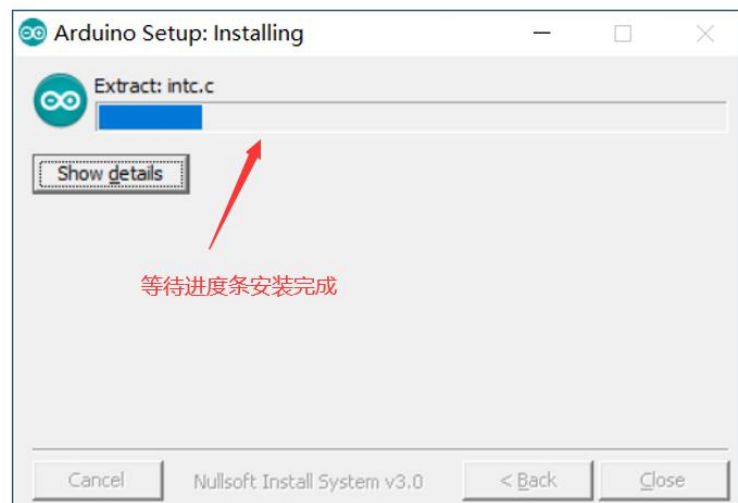
OpenCR 的安装环境的安装包，双击打开即可。进入安装的过程。



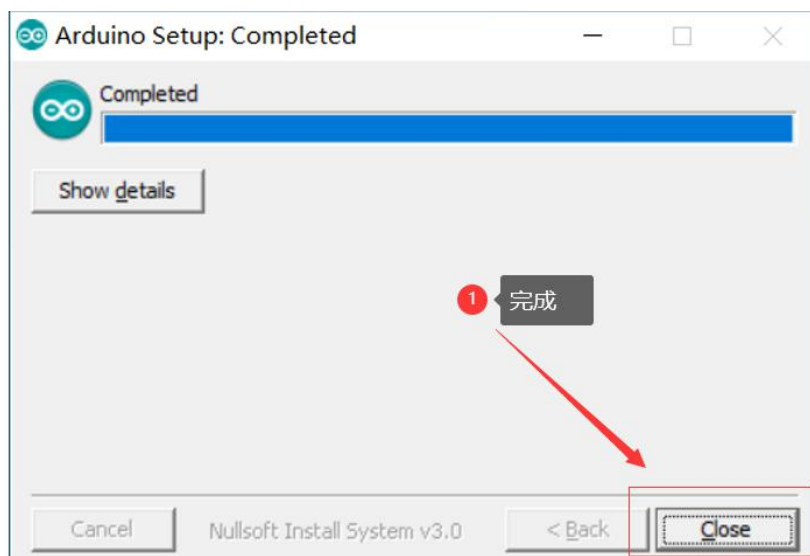


在这里你可以选择自己的安装位置。





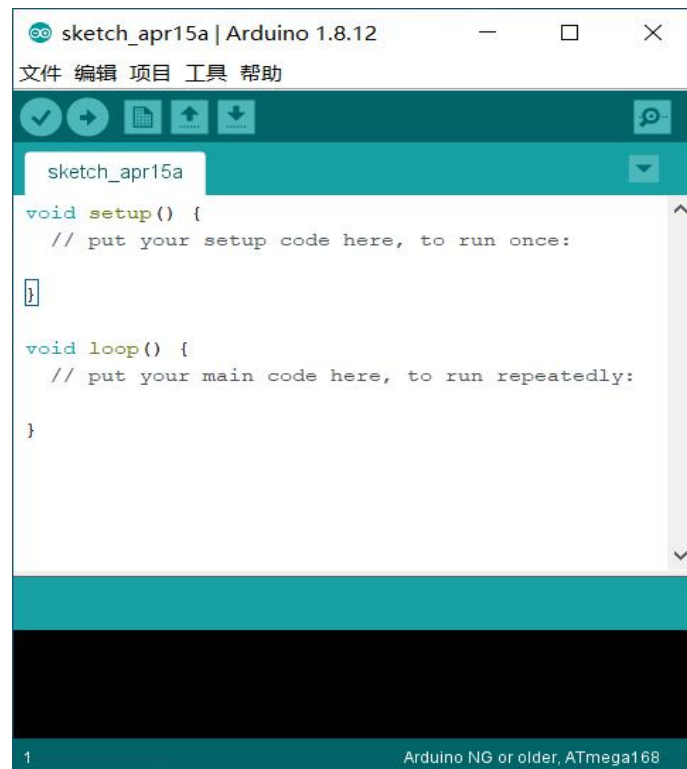
安装已经完成，点击 Close 即可。



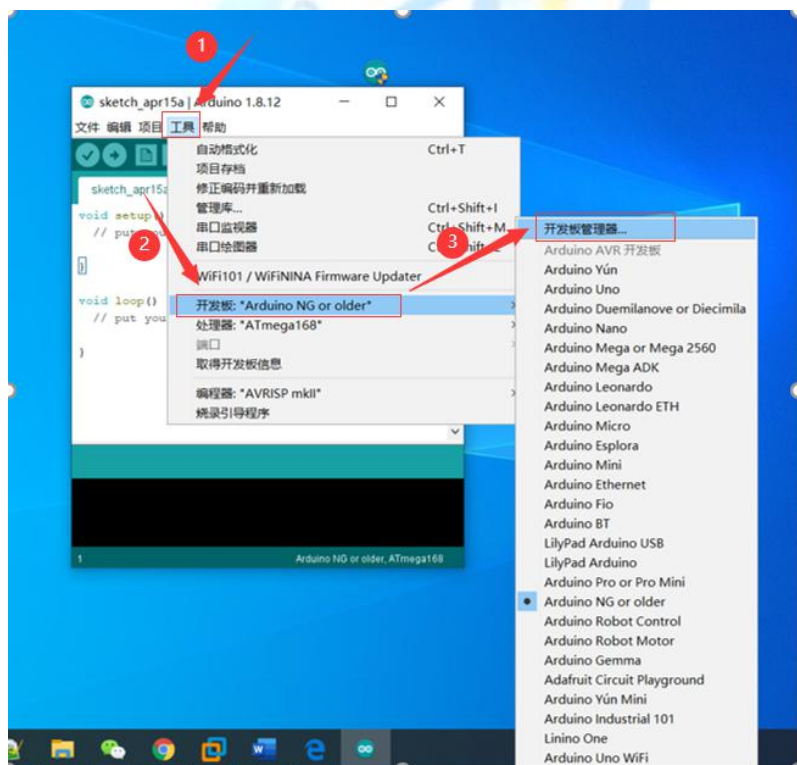
### 3.2. opencr 固件包下载

在安装完成之后，双击即可打开。

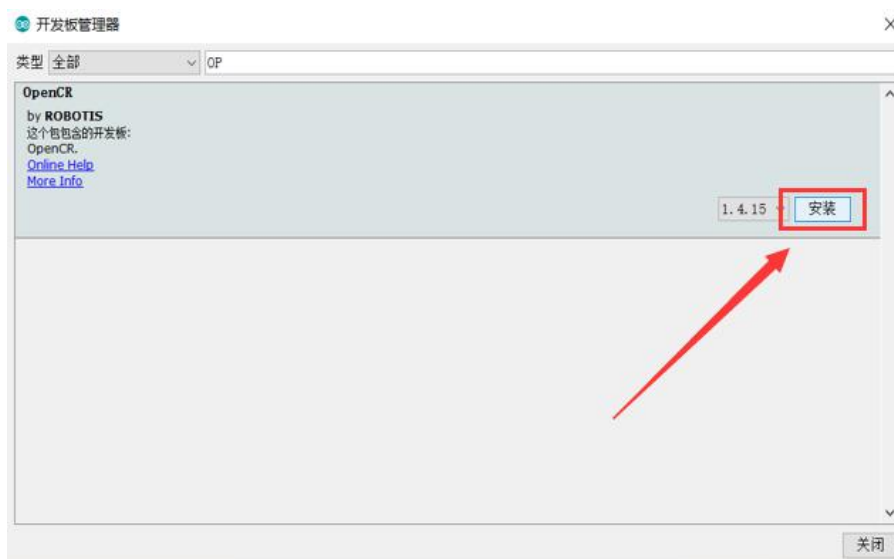




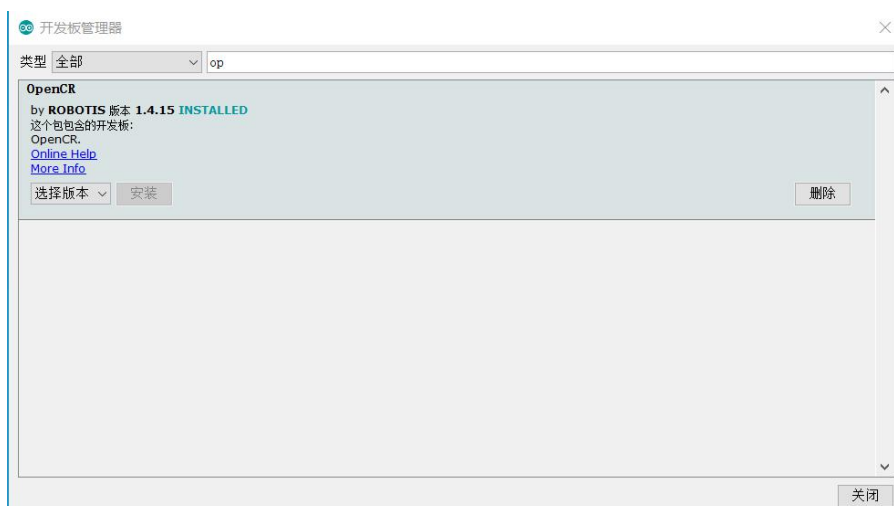
点击 工具→开发板→开发板管理→进入选型下载：



在文本框中键入 OpenCR 以查找包。 找到 OpenCR 后， 点击 安装。

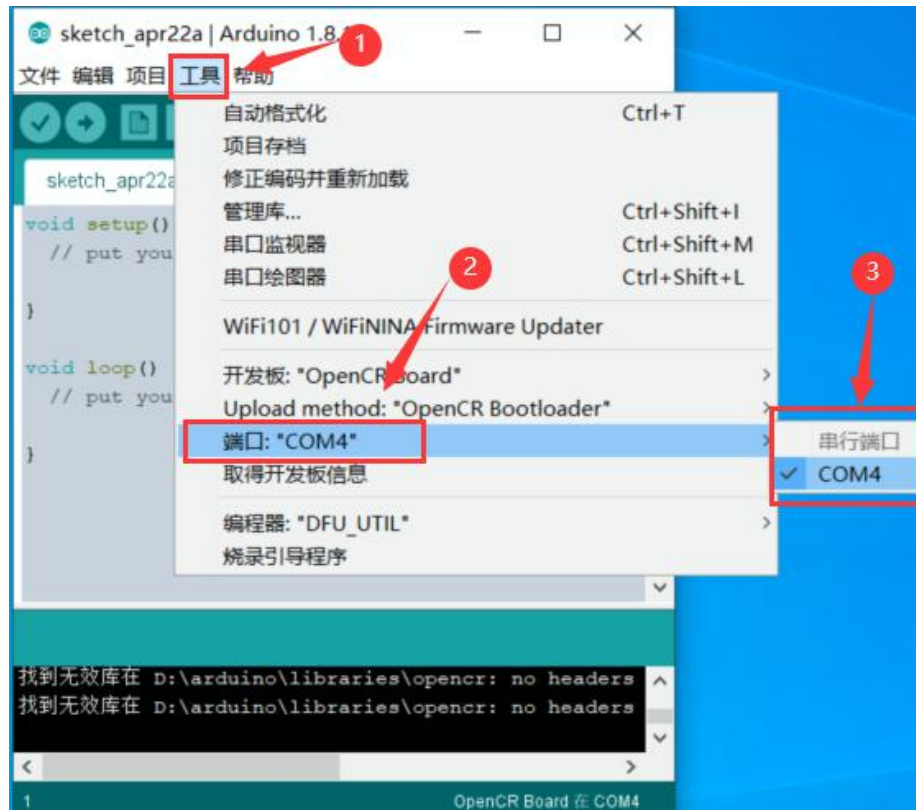


下载好之后，选择关闭这个页面，进行之后 openCR 固件的烧录。

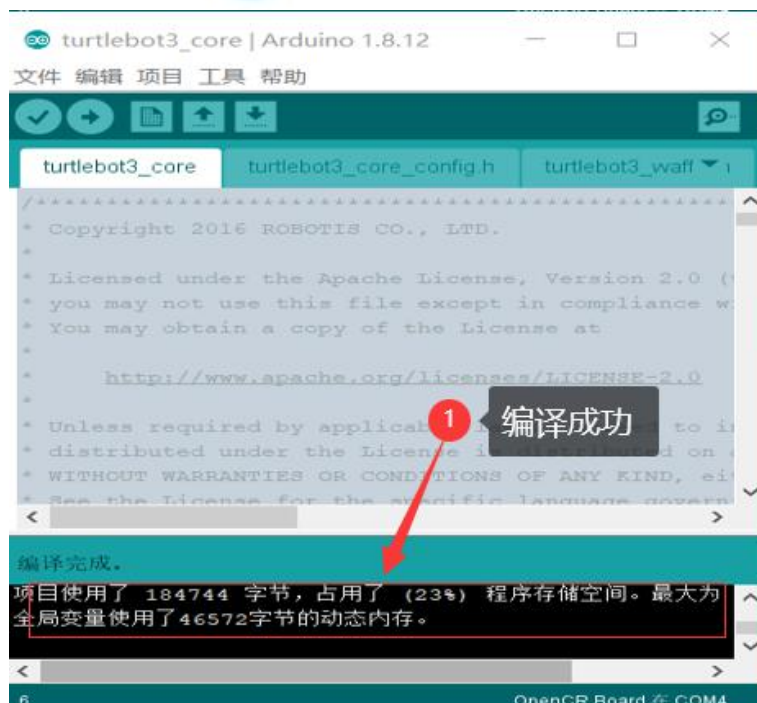
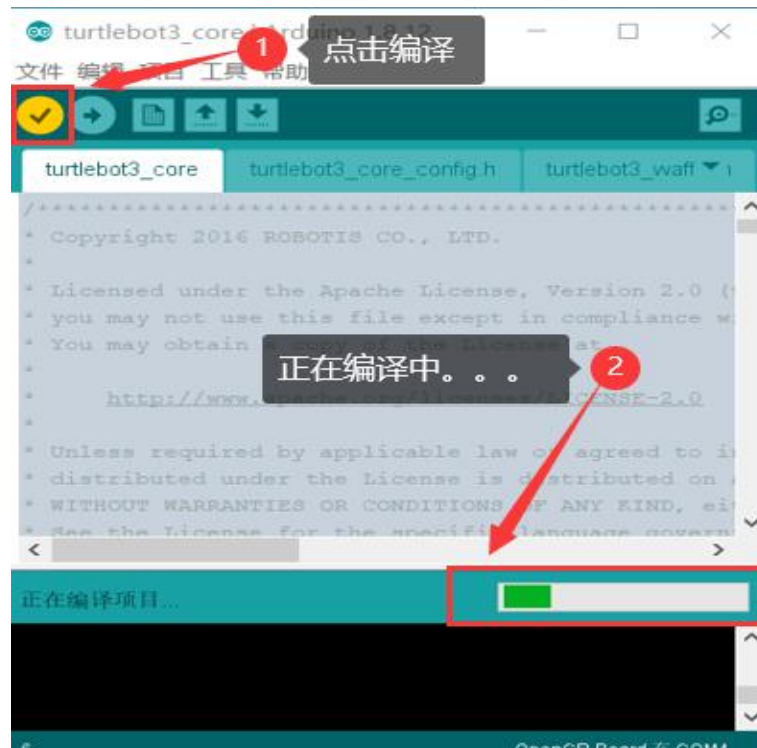


### 3.3. 端口设置

确定下载线已经连接→点击工具→端口→选择出现的端口。



## 3.4. 固件编译与下载





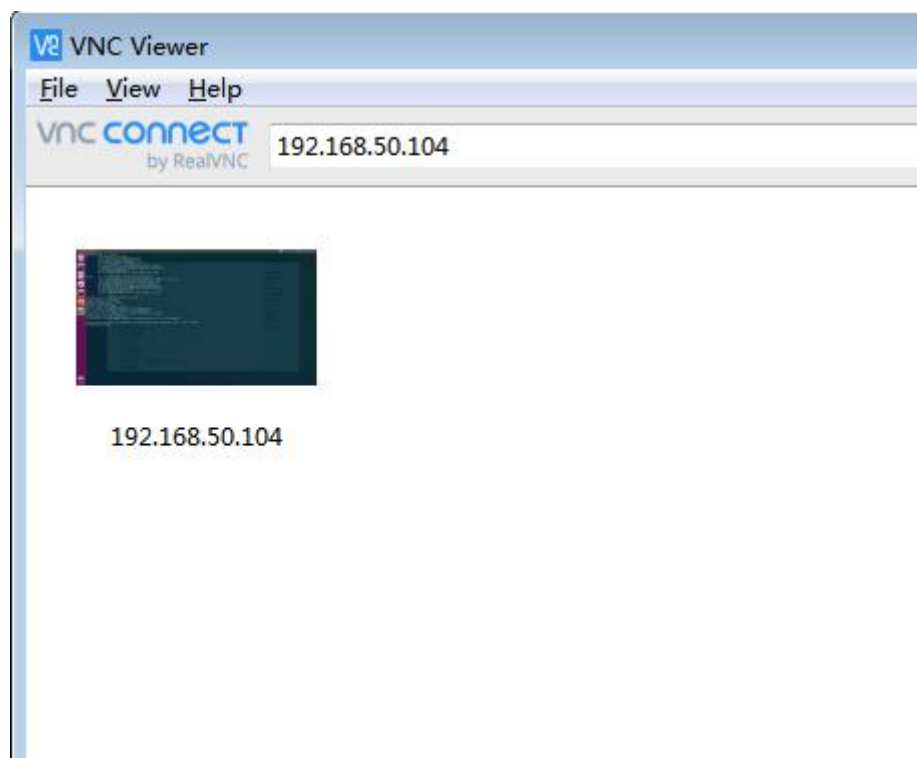
## 四. 笔记本与 TX2 的通信

### 4.1. 使用 vnc 控制华夫 TurtleBot3-Tx2 开发板

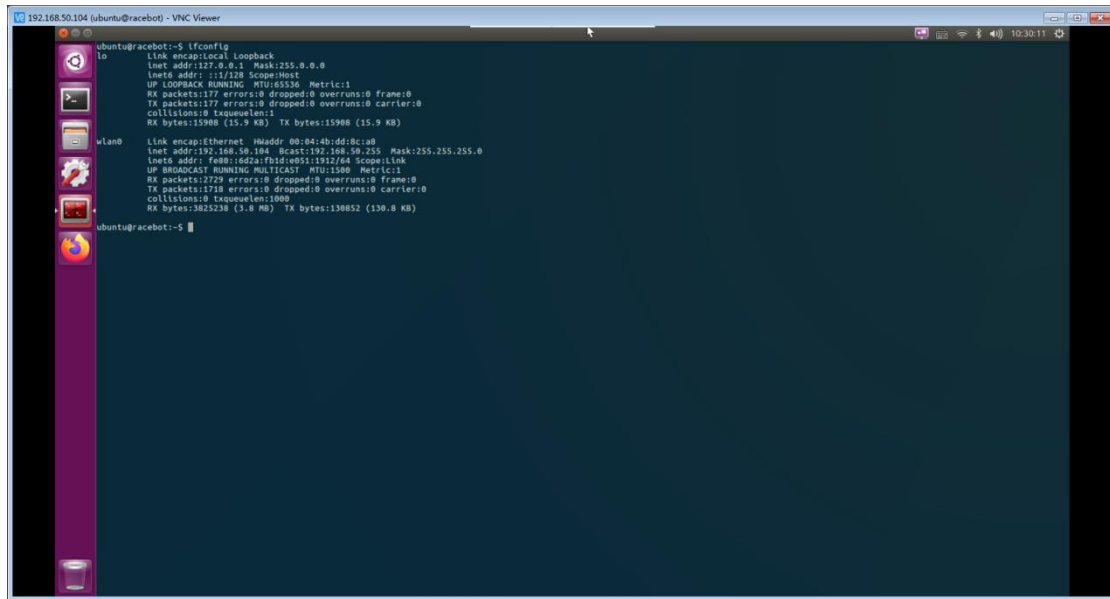
1) 电脑端安装 vnc viewer, 您可以选择应用商城下载安装即可



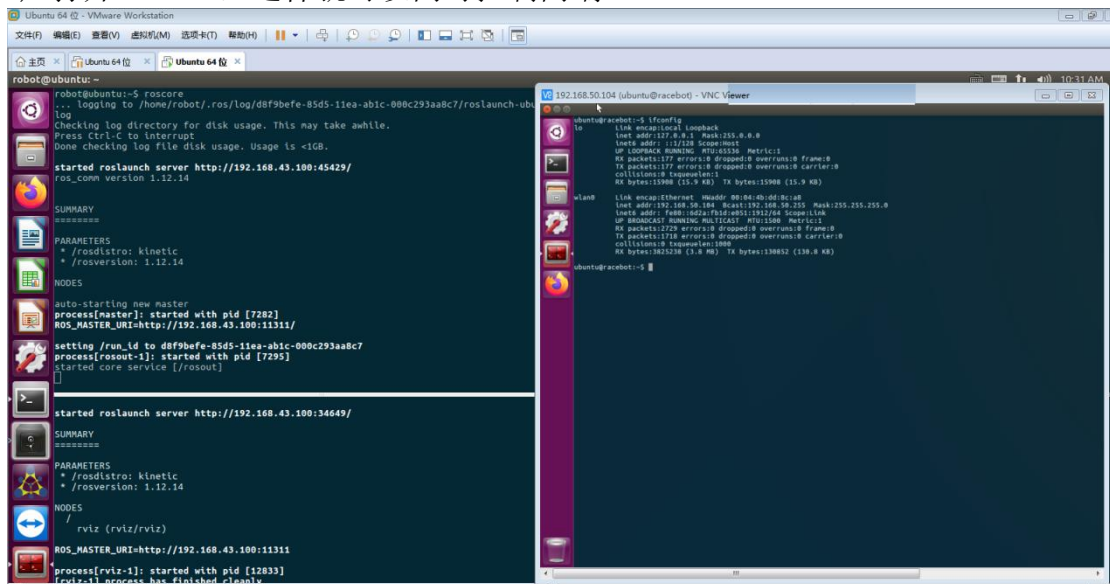
2) 下载后打开, 键入 TurtleBot3 的 ip →回车→选择连接



3) 登录后, 电脑端显示的控制画面, 即可进行操作。



4) 打开 Ubuntu，这样就可以同时控制两端：



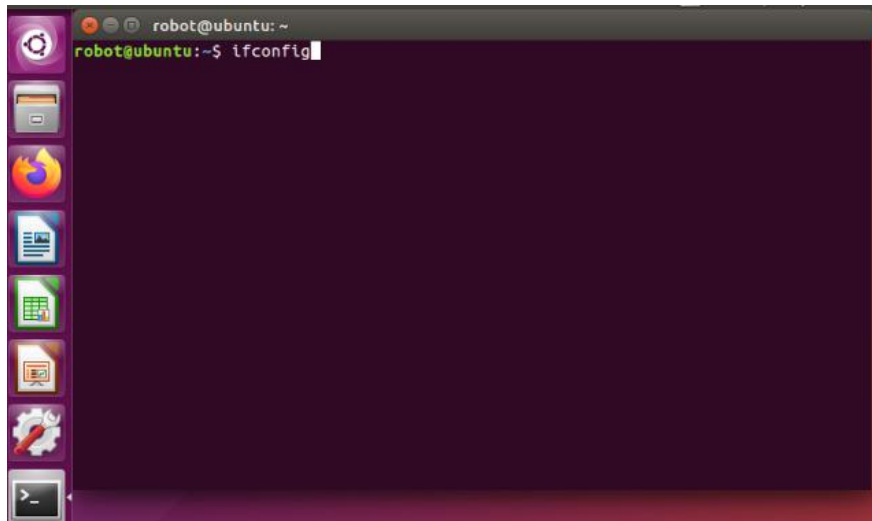
## 4.2. IP 的配置

### 4.2.1. 获取 Remote PC 和 TurtleBot3 的 IP

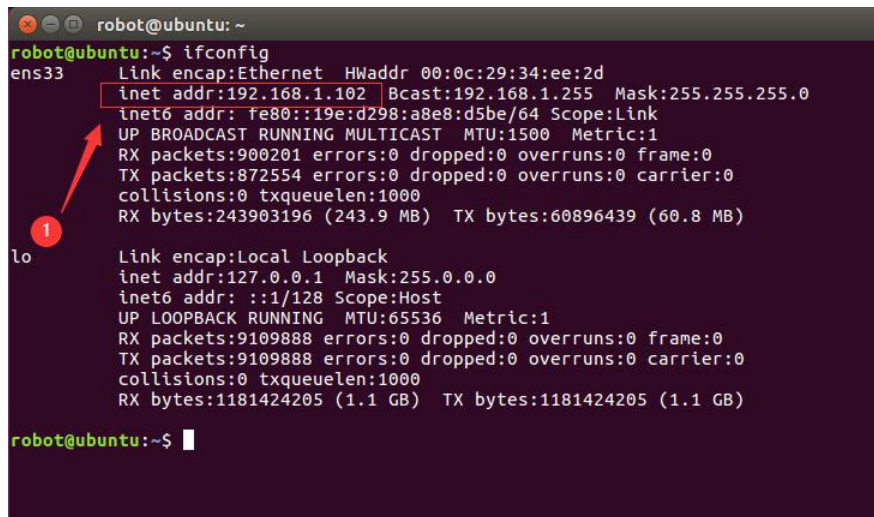
首先获取 Remote PC 端和 TurtleBot3 端的 IP

```
ifconfig
```

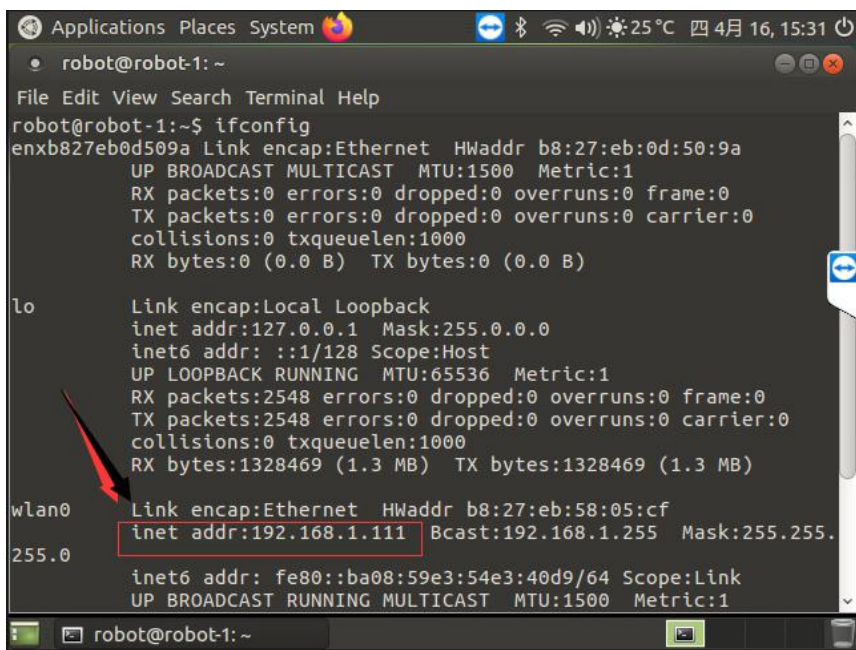




键入命令后回车→获取 IP







```
robot@robot-1: ~
File Edit View Search Terminal Help
robot@robot-1:~$ ifconfig
enxb827eb0d509a Link encap:Ethernet HWaddr b8:27:eb:0d:50:9a
  UP BROADCAST MULTICAST MTU:1500 Metric:1
    RX packets:0 errors:0 dropped:0 overruns:0 frame:0
    TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
    collisions:0 txqueuelen:1000
    RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)

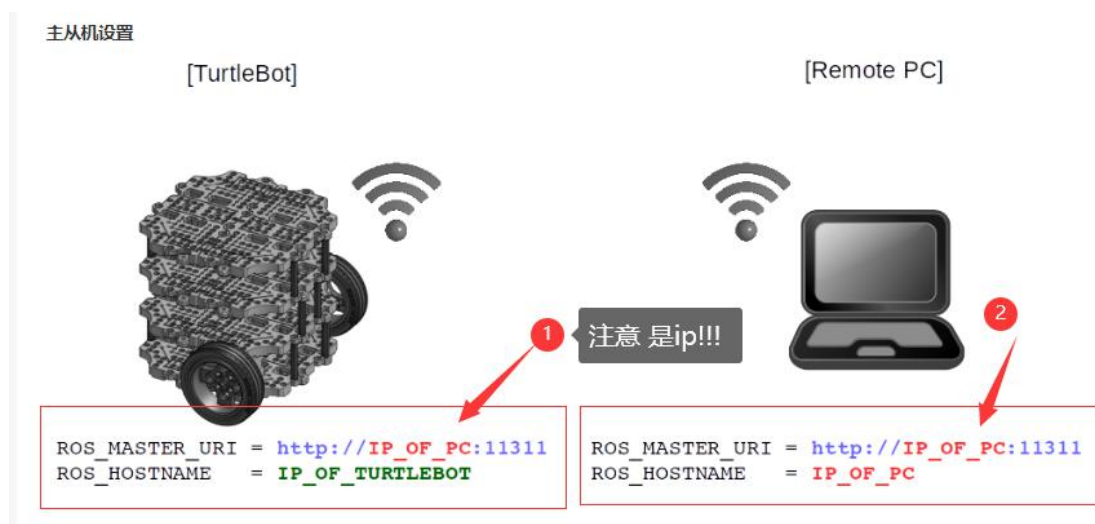
lo        Link encap:Local Loopback
  inet addr:127.0.0.1 Mask:255.0.0.0
  inet6 addr: ::1/128 Scope:Host
    UP LOOPBACK RUNNING MTU:65536 Metric:1
    RX packets:2548 errors:0 dropped:0 overruns:0 frame:0
    TX packets:2548 errors:0 dropped:0 overruns:0 carrier:0
    collisions:0 txqueuelen:1000
    RX bytes:1328469 (1.3 MB) TX bytes:1328469 (1.3 MB)

wlan0     Link encap:Ethernet HWaddr b8:27:eb:58:05:cf
  inet addr:192.168.1.111 Bcast:192.168.1.255 Mask:255.255.255.0
  inet6 addr: fe80::ba08:59e3:54e3:40d9/64 Scope:Link
    UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
```

如上图获取的 Remote PC 的 IP 为: 192.168.1.102

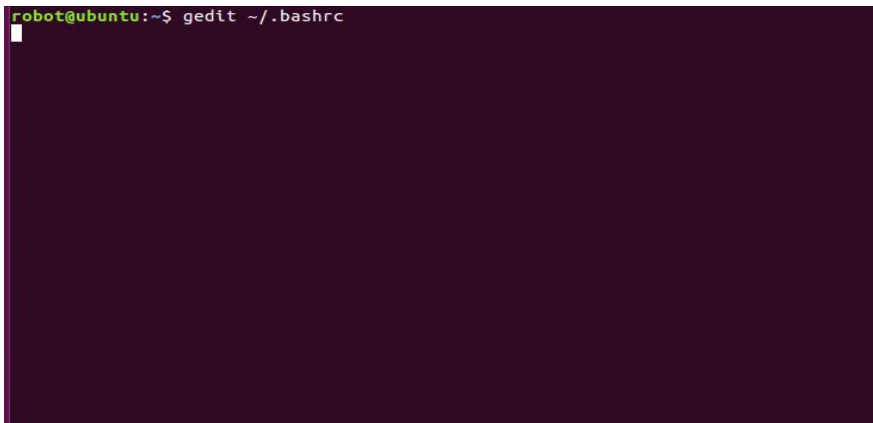
获取的 TurtleBot3 的 IP 为: 192.168.1.111

#### 4.2.2. 修改 Remote PC 和 TurtleBot3 的 IP



修改 IP 的命令为:

```
gedit ~/.bashrc
```



```
# some more ls aliases
alias ll='ls -alF'
alias la='ls -A'
alias l='ls -CF'

# Add an "alert" alias for long running commands. Use like so:
# sleep 10; alert
alias alert='notify-send --urgency=low -i "${[ $? = 0 ]} && echo terminal || echo error" "${history|tail -n1|sed -e '\''s/^[0-9]\+\s*//;s/[\;|]\s*alert$/\''}'"

# Alias definitions.
# You may want to put all your additions into a separate file like
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi

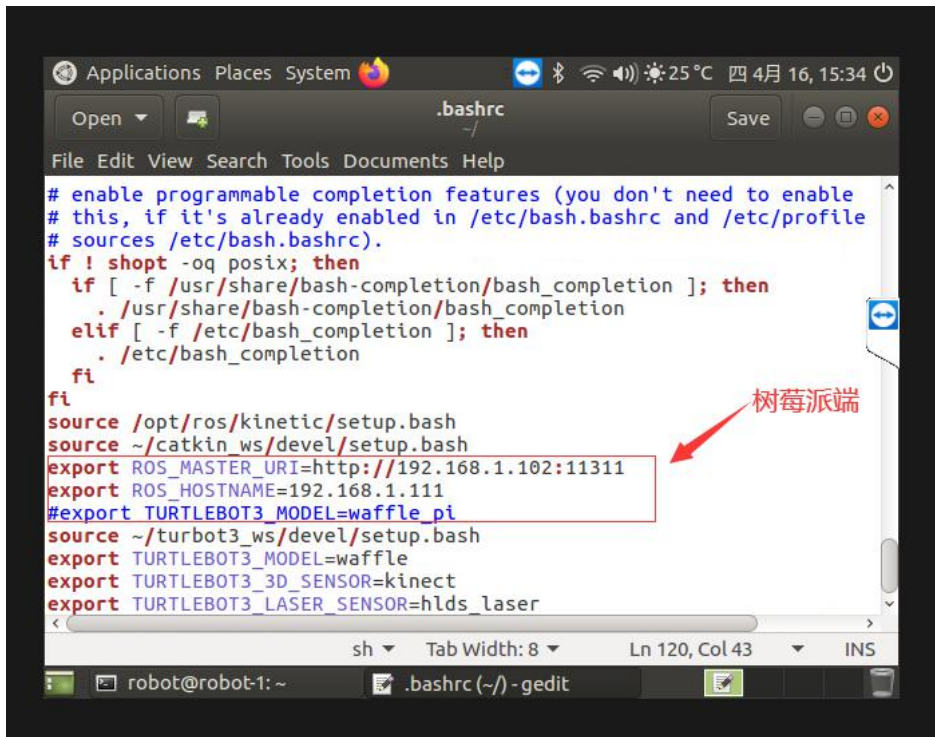
# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
    if [ -f /usr/share/bash-completion/bash_completion ]; then
        . /usr/share/bash-completion/bash_completion
    elif [ -f /etc/bash_completion ]; then
        . /etc/bash_completion
    fi
fi

source /opt/ros/kinetic/setup.bash

#source /home/robot/catkin_ws/devel/setup.bash

export ROS_MASTER_URI=http://192.168.1.102:11311
export ROS_HOSTNAME=192.168.1.102
#export TURTLEBOT3_MODEL=waffle_pi
source ~/turtlebot3_ws/devel/setup.bash
export TURTLEBOT3_MODEL=waffle
export TURTLEBOT3_3D_SENSOR=kinect
export TURTLEBOT3_LASER_SENSOR=rplidar
source ~/google_ws/devel_isolated/setup.bash
export ROS_PACKAGE_PATH=${ROS_PACKAGE_PATH}:/ORB_SLAM2/Examples/ROS
```

设置ip



```
# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
  if [ -f /usr/share/bash-completion/bash_completion ]; then
    . /usr/share/bash-completion/bash_completion
  elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
  fi
fi
source /opt/ros/kinetic/setup.bash
source ~/catkin_ws/devel/setup.bash
export ROS_MASTER_URI=http://192.168.1.102:11311
export ROS_HOSTNAME=192.168.1.111
#export TURTLEBOT3_MODEL=waffle_pi
source ~/turbot3_ws/devel/setup.bash
export TURTLEBOT3_MODEL=waffle
export TURTLEBOT3_3D_SENSOR=kinect
export TURTLEBOT3_LASER_SENSOR=hlds_laser
```

#### 4.2.3. 保存 Remote PC 和 TurtleBot3 端的修改

[TurtleBot3 && Remote PC] 让环境生效，配置的命令为：

```
source ~/.bashrc
```

#### 4.3. 通信测试

[Remote PC]端键入命令，启动 roscore

```
roscore
```

```
roscore http://192.168.1.102:11311/
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://192.168.1.102:45303/
ros_comm version 1.12.14

SUMMARY
=====

PARAMETERS
* /roscdistro: kinetic
* /rosversion: 1.12.14

NODES

auto-starting new master
process[master]: started with pid [19907]
ROS_MASTER_URI=http://192.168.1.102:11311/

setting /run_id to 2b771d1a-7fb5-11ea-8bcf-000c2934ee2d
process[rosout-1]: started with pid [19920]
started core service [/rosout]
```

成功!

[TurtleBot3]端键入通信命令(查看是否订阅到主机发布的话题):

```
rostopic list
```

```
robot@robot-1:~$ rostopic list
/rosout
/rosout_agg
robot@robot-1:~$
```

出现上图，及说明通信成功。

## 五. 激活你的雷达

重要提示：请在配网通信成功后进行操作，配网后再次开机需要重新验证通信；

重要提示：[Remote PC]代表 PC 端、[TurtleBot3]代表树莓派端；

### 5.1. 操作步骤

1) [Remote PC] 启动 roscore;

```
roscore
```

2) [TurtleBot3] 启动雷达;

```
roslaunch turbot3_tools rplidar.launch
```

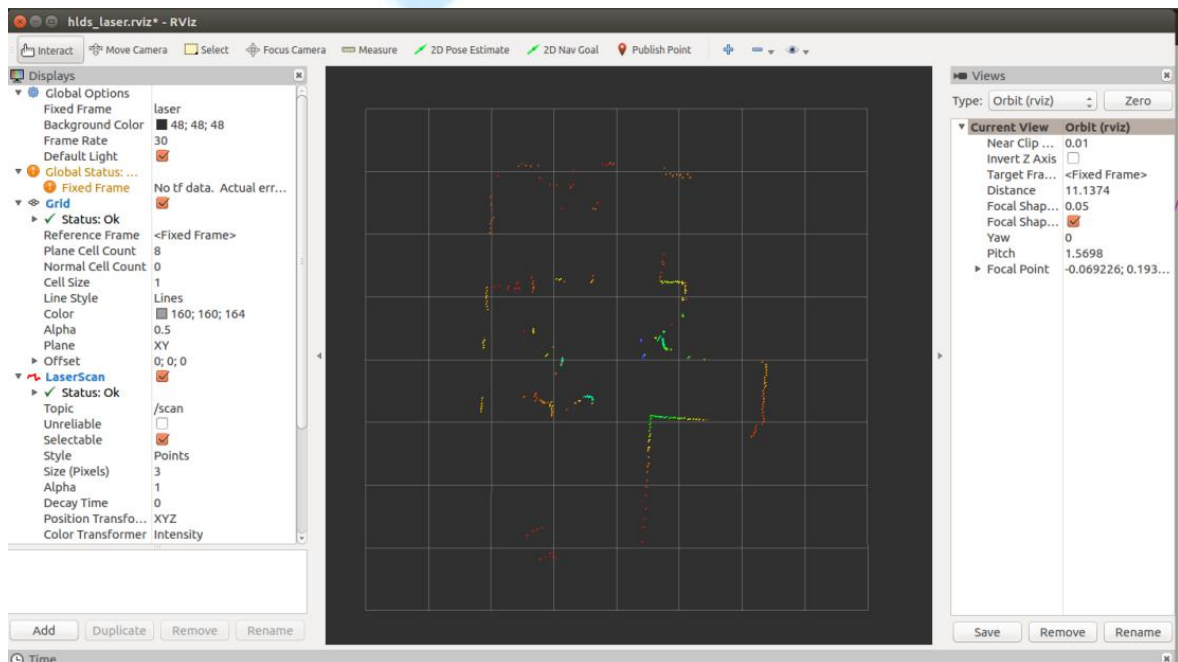
3) [Remote PC] 启动 rviz;

```
roslaunch turbot3_tools test_rplidar.launch
```



### 5.2. 实际效果

1) 命令执行之后，就可以看到当前的雷达扫描图了；





## 六. kinect 测试

重要提示：请在配网通信成功后进行操作，配网后再次开机需要重新验证通信；  
重要提示：[Remote PC]代表 PC 端、[TurtleBot3]代表 TX2 端；

操作步骤如下：

- 1) [Remote PC] 启动 roscore；

```
roscore
```

- 2) [TurtleBot3] 启动底盘；

```
roslaunch turbot3_bringup minimal.launch
```

- 3) [TurtleBot3] 启动相机；

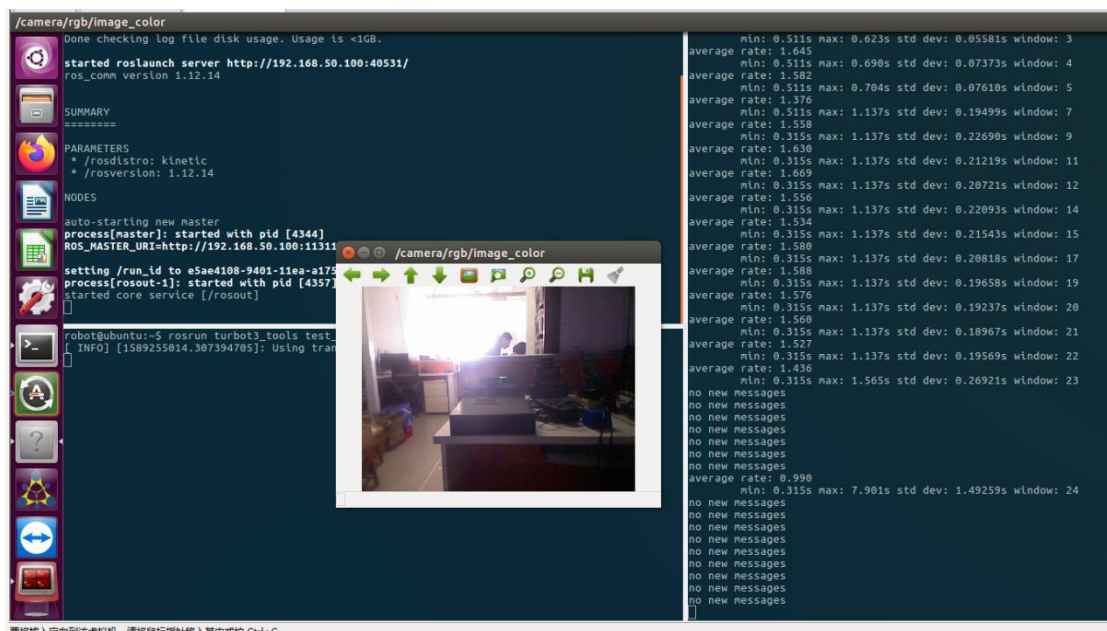
```
roslaunch freenect_launch freenect-registered-xyzrgb.launch
```

- 4) [Remote PC] 监听是否获取到图像数据；

```
rostopic hz /camera/rgb/image_color
```

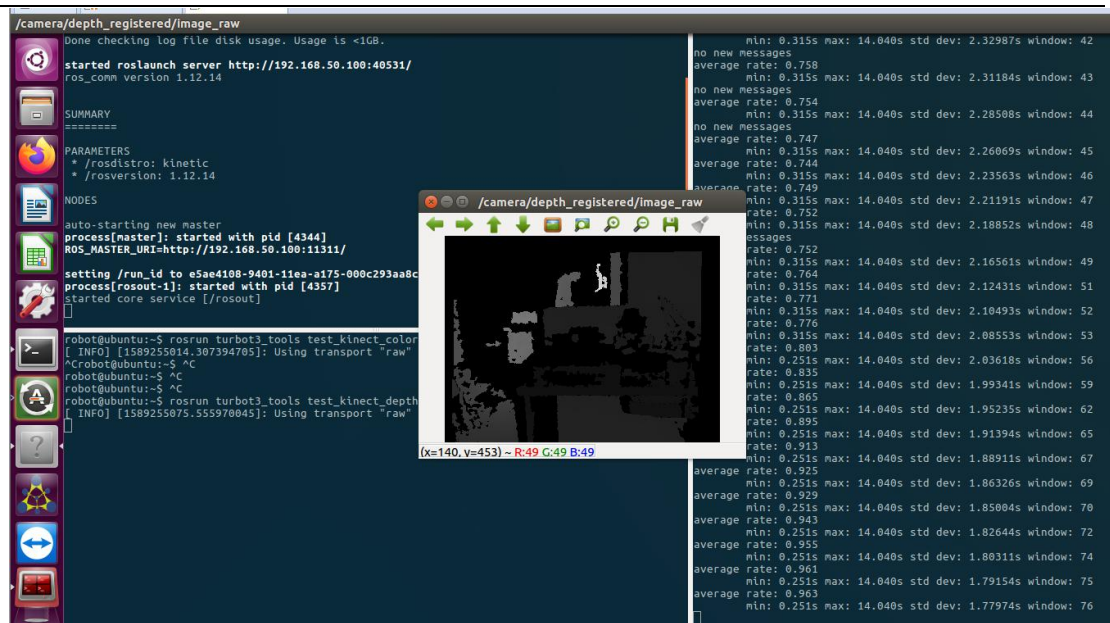
- 5) [Remote PC] 执行显示图像；

```
roslaunch turbot3_tools test_kinect_color
```



- 6) [Remote PC] 执行显示深度图；

```
roslaunch turbot3_tools test_kinect_depth
```



## 七. 底盘测试

说明：opencr 本身带有底盘自测功能，通过按 opencr 的 sw1 和 sw2 来自检底盘是否正确安装和运行；

### 7.1. 前进测试

- 1) 测试前，先把小车架空，轮子不要着地；
- 2) 接好电源后，打开 opencr 的开关，完成启动；
- 1) 按住 sw1 几秒后，两边轮子同时向前转动，实现前进；

### 7.2. 转圈测试

- 1) 测试前，先把小车架空，轮子不要着地；
- 2) 接好电源后，打开 opencr 的开关，完成启动；
- 3) 按住 sw2 几秒后，一个轮子向前转动，一个轮子向后转动，实现转圈；

### 7.3. 启动测试

- 1) [Remote PC] 启动 roscore;

```
roscore
```

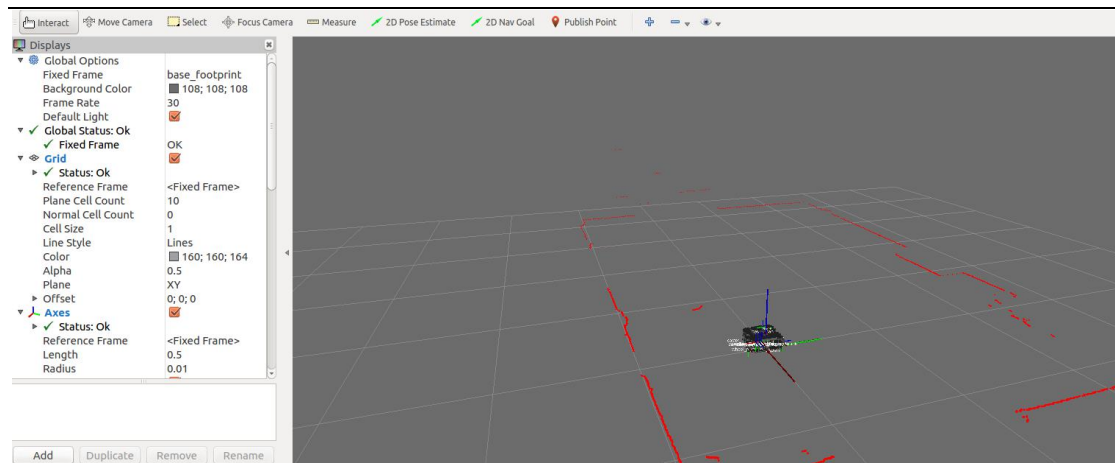
- 2) [TurtleBot3] 启动 TurtleBot3;

```
roslaunch turbot3_bringup minimal.launch
```

- 3) [Remote PC] 启动 rviz;

```
roslaunch turbot3_bringup turtlebot3_model.launch
```





## 八. 键盘控制

- 1) [Remote PC] 启动 roscore

```
roscore
```

- 2) [TurtleBot3] 启动 TurtleBot3

```
roslaunch turbot3_bringup minimal.launch
```

```
ubuntu@racebot:~/catkin_ws$ roslaunch turbot3_bringup minimal.launch
... logging to /home/ubuntu/.ros/log/41fc8ce2-bab0-11ea-9e7f-dd6d3bf47f8b/roslaunch-racebot-6058.log
Checking log directory for disk usage. This may take awhile.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://192.168.50.106:33412/

SUMMARY
=====

PARAMETERS
* /rostdistro: kinetic
* /rosversion: 1.12.14
* /rplidarNode/angle_compensate: True
* /rplidarNode/frame_id: base_scan
* /rplidarNode/inverted: False
* /rplidarNode/scan_mode: Sensitivity
* /rplidarNode/serial_baudrate: 256000
* /rplidarNode/serial_port: /dev/rplidar
* /turtlebot3_core/baud: 115200
* /turtlebot3_core/port: /dev/ttyACM0
* /turtlebot3_core/tf_prefix:
```

- 3) [Remote PC] 启动进行简单远程操作测试

```
roslaunch turbot3_teleop keyboard.launch
```

- 4) [Remote PC] 如果文件成功启动，终端将显示以下内容。

```
robot@ubuntu:~$ roslaunch turbot3_teleop keyboard.launch
... logging to /home/robot/.ros/log/486f67f0-9410-11ea-a175-000c293aa8c7/roslaunch-ubuntu-6294.
log
Checking log directory for disk usage. This may take awhile.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://192.168.50.100:40449/

SUMMARY
=====

PARAMETERS
* /model: waffle
* /rostdistro: kinetic
* /rosversion: 1.12.14

NODES
/
  turtlebot3_teleop_keyboard (turtlebot3_teleop/turtlebot3_teleop_key)

ROS_MASTER_URI=http://192.168.50.100:11311

process[turtlebot3_teleop_keyboard-1]: started with pid [6311]

Control Your TurtleBot3!
-----
Moving around:
    w
    a  s  d
    x

w/x : increase/decrease linear velocity (Burger : ~ 0.22, Waffle and Waffle Pi : ~ 0.26)
a/d : increase/decrease angular velocity (Burger : ~ 2.84, Waffle and Waffle Pi : ~ 1.82)

space key, s : force stop

CTRL-C to quit
```

**注意：**

<b>w</b> 是增加线速度	<b>x</b> 是降低线速度
<b>a</b> 是增加角速度	<b>d</b> 是降低角速度
<b>s</b> 是停止运动	



## 九. 建图-gmapping 建图（A2 雷达）

### 9.1. 说明

这一节我们来讲 Turtlebot3 的 SLAM

**SLAM**(The Simultaneous Localization and Mapping) 同步定位与地图构建：希望机器人从未知环境的未知地点出发，在运动过程中通过重复观测到的地图特征（比如，墙角，柱子等）定位自身位置和姿态，再根据自身位置增量式的构建地图，从而达到同时定位和地图构建的目的。（维基百科）

### 9.2. 通过远程操作创建地图

- 1) [Remote PC] 启动 roscore

```
roscore
```

- 2) [TurtleBot3] 启动 TurtleBot3

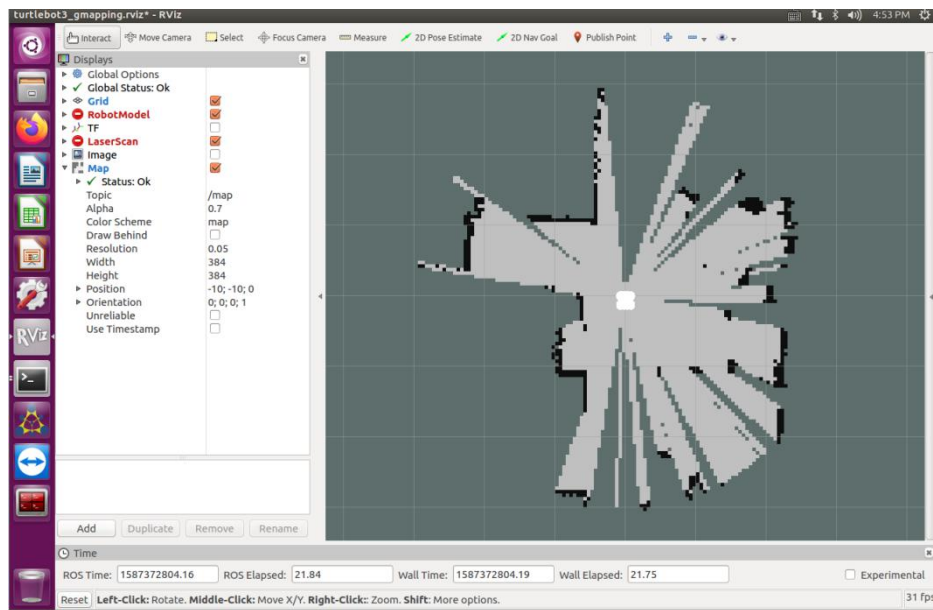
```
roslaunch turbot3_bringup minimal.launch
```

- 3) [TurtleBot3] 运行 SLAM 启动文件

```
roslaunch turbot3_slam gmapping_demo.launch
```

- 4) [Remote PC] 启动 Rviz 可视化模型

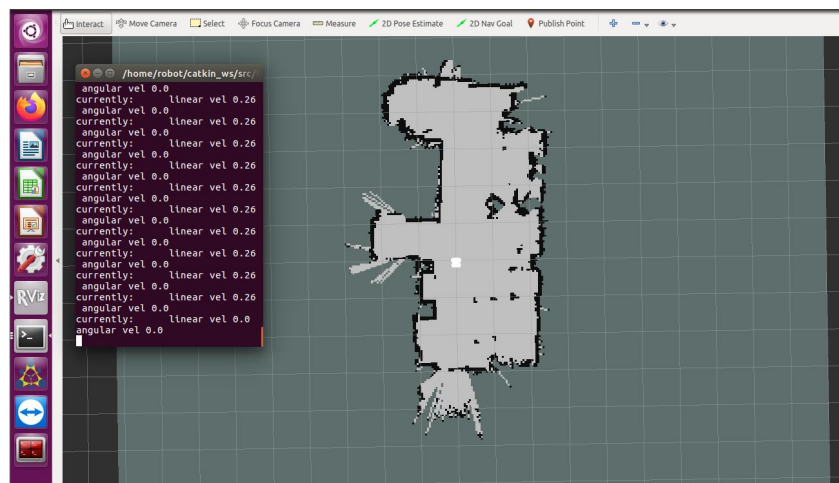
```
roslaunch turbot3_rviz navigation_rviz.launch
```



## 5) [Remote PC] 启动键盘操作

```
roslaunch turbot3_teleop keyboard.launch
```

移动 TurtleBot3 进行建图



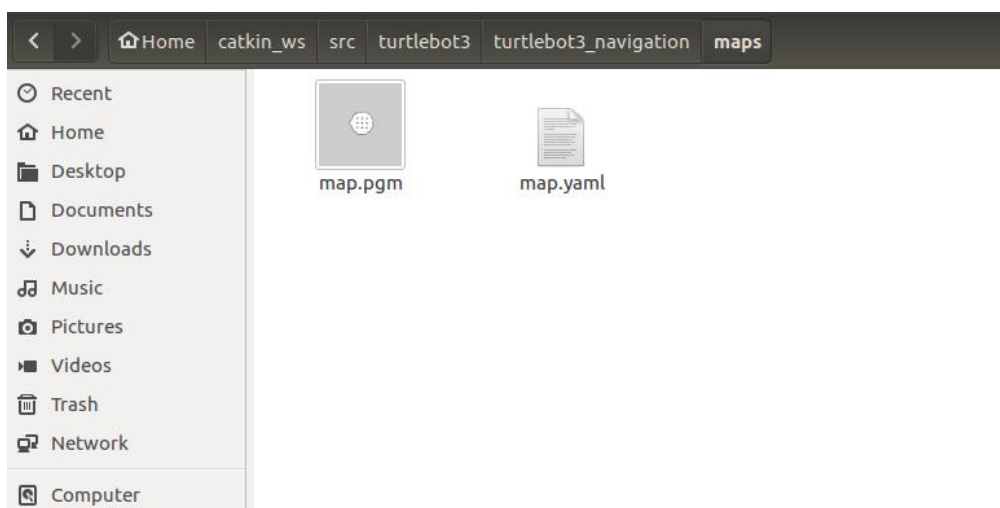
## 6) [TurtleBot3] 完成建图后运行地图保存节点

```
roslaunch map_server map_saver -f ~/map
```

map.pgm 和 map.yaml 在 ~/ 目录里创建

```
robot@ubuntu: ~  
robot@ubuntu:~$ rosrun map_server map_saver -f ~/map  
[ INFO] [1587373042.620293744]: Waiting for the map  
[ INFO] [1587373042.872568381]: Received a 384 X 384 map @ 0.050 m/pix  
[ INFO] [1587373042.872681519]: Writing map occupancy data to /home/robot/map.pg  
[ INFO] [1587373042.875940278]: Writing map occupancy data to /home/robot/map.ya  
[ INFO] [1587373042.876183348]: Done  
robot@ubuntu:~$
```

创建好的地图在文件夹下的位置如图所示：



## 十. 自主导航（A2 激光雷达）

- 1) [Remote PC] 启动 roscore

```
roscore
```

- 2) [TurtleBot3] 启动 TurtleBot3

```
roslaunch turbot3_bringup minimal.launch
```

- 3) [TurtleBot3] 启动导航

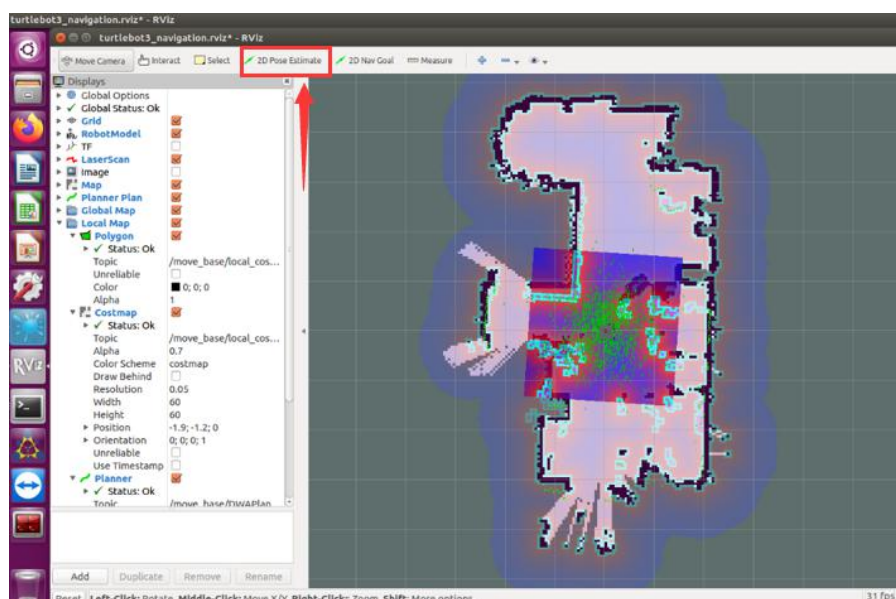
```
roslaunch turbot3_navigation navigation_laser.launch map_file:=$HOME  
/map.yaml
```

- 4) [Remote PC] 启动 Rviz

```
roslaunch turbot3_rviz navigation_rviz.launch
```

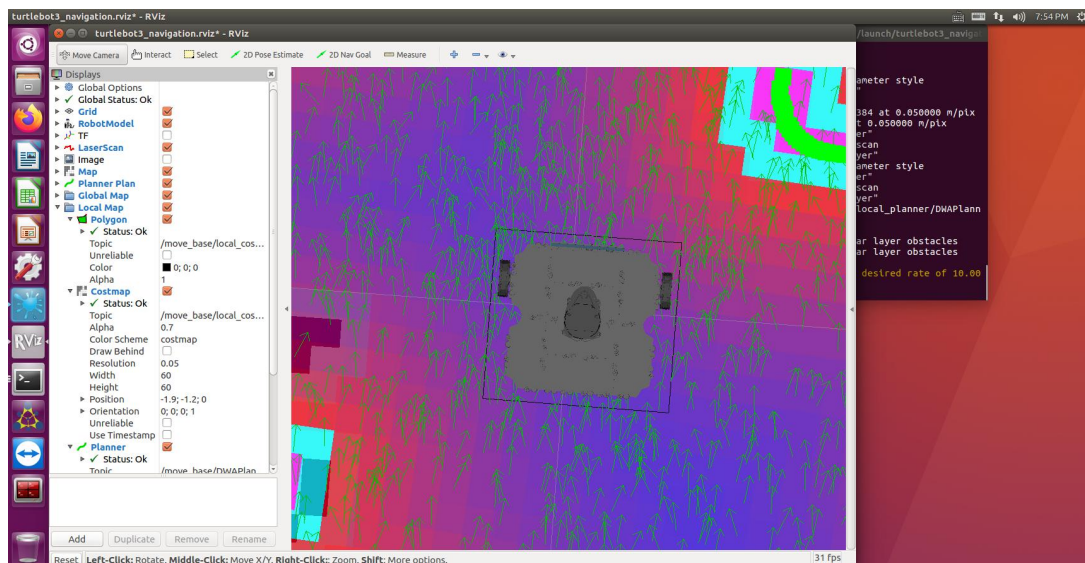
- 5) 在开始导航之前，TurtleBot3 应该知道它的位置和姿势，要给出初始数据，请按照说明进行操作。

点击 2D Pose Estimate 按钮，通过单击并拖动地图上的方向来设置地图上  
的大致位置。





箭头的每个点意味着 TurtleBot3 的预期姿势。激光扫描仪将在近似位置绘制线条，如地图上的墙壁如果图形没有显示线条，请重复上述过程。



当 TurtleBot3 已经定位，它将自动计划路径，要发送目标位置：

点击 2D Nav Goal 按钮，点击地图上你想要的 TurtleBot 驱动和拖动方向 TurtleBot3 应该指向地方，如果目标位置的路径被阻止，这可能会失败。要在机器人到达目标位置之前停止机器人，请发送 TurtleBot3 的当前位置。



## 十一. 建图-karto 建图

- 1) [Remote PC] 启动 roscore

```
roscore
```

- 2) [TurtleBot3] 启动 TurtleBot3

```
roslaunch turbot3_bringup minimal.launch
```

- 3) [TurtleBot3] 启动 karto 算法。

```
roslaunch turbot3_slam karto_demo.launch
```

- 4) [Remote PC] 启动 Rviz

```
roslaunch turbot3_rviz navigation_rviz.launch
```

- 5) [Remote PC] 启动键盘操作。

```
roslaunch turbot3_teleop keyboard.launch
```

移动机器人进行建图。

- 6) [TurtleBot3] 完成建图后运行地图保存节点。

```
roslaunch map_server map_saver -f ~/map
```

map.pgm 和 map.yaml 在 ~/ 目录里创建。

## 十二. 建图-hector 建图

- 1) [Remote PC] 启动 roscore

```
roscore
```

- 2) [TurtleBot3] 启动 TurtleBot3

```
roslaunch turbot3_bringup minimal.launch
```

- 3) [TurtleBot3] 启动 hecotor

```
roslaunch turbot3_slam hector_demo.launch
```

- 4) [Remote PC] 启动 rviz

```
roslaunch turbot3_rviz navigation_rviz.launch
```

- 5) [Remote PC] 启动键盘操作

```
roslaunch turbot3_teleop keyboard.launch
```

移动机器人进行建图

- 6) [TurtleBot3] 完成建图后运行地图保存节点

```
roslaunch map_server map_saver -f ~/map
```

map.pgm 和 map.yaml 在 ~/ 目录里创建。

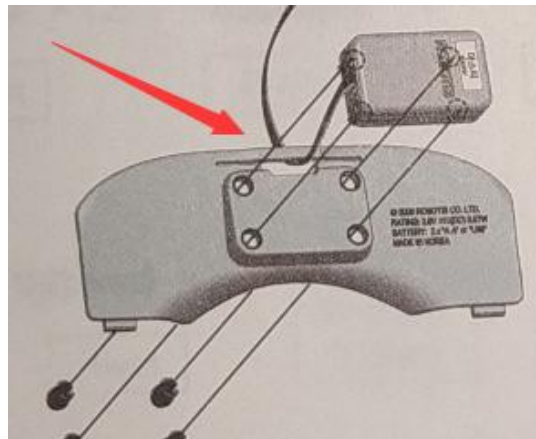
## 十三. RC100 遥控控制

### 13.1. 遥控说明

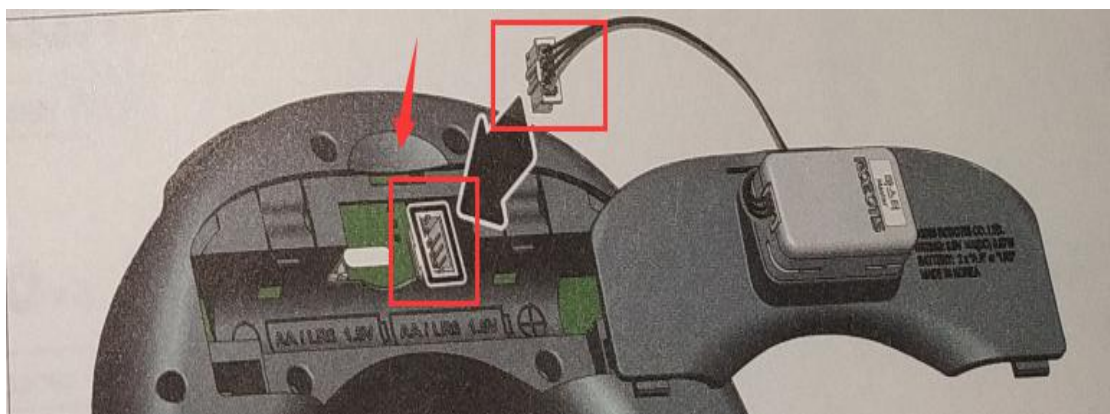
使用 ROBOTIS RC100 的设置已经在 ROS 的 OpenCR 固件中, 因此不需要安装必需软件包, 安装好即可使用。

### 13.2. 遥控的安装

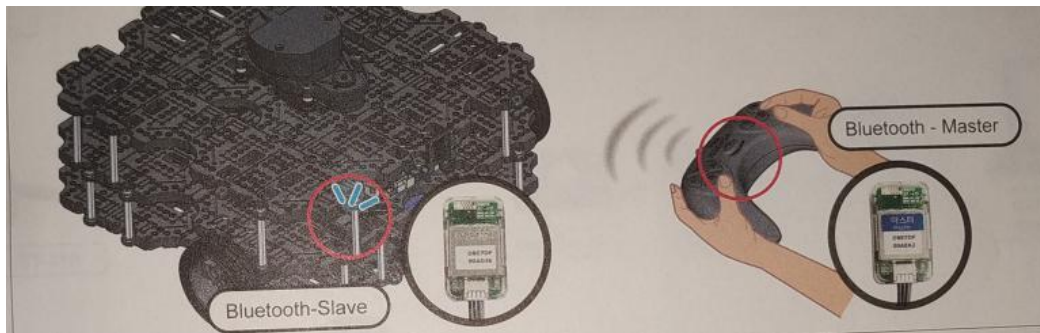
① 接线口穿过后壳的过孔。



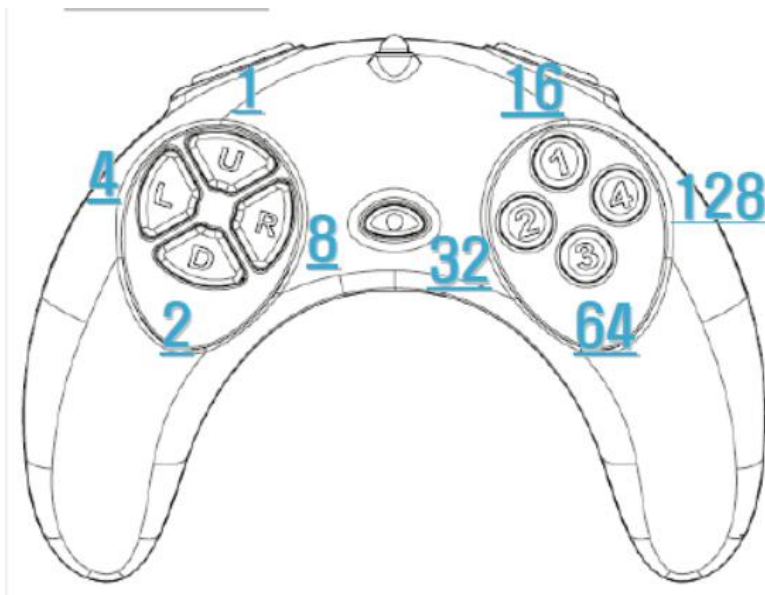
② 接线口连接到另一端的母座, 安装好电池即可。



### 13.3. 遥控的操作说明



当您按下 RC-100 上的按钮时，每个按钮的对应代码值将被发送出去。



按照左侧的 UDLR 分别是前后左右的加速度，右手的 5 按键是停止键。操作类似键盘 wasdx 控制。