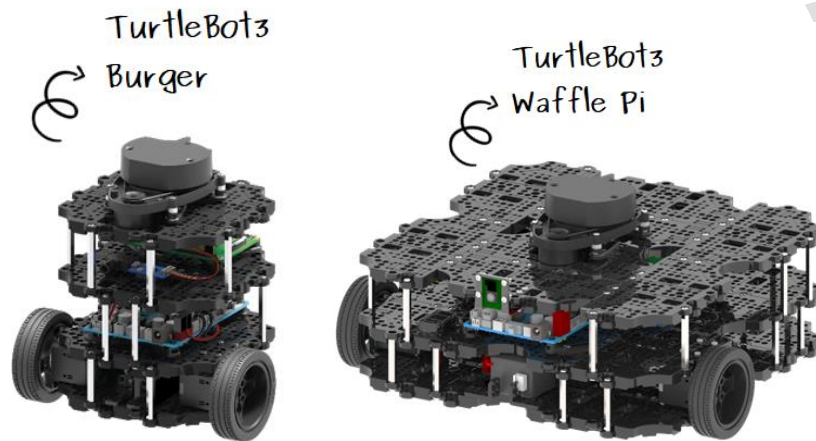


Turtlebot3 京天用户手册 V2.2

TURTLEBOT3 USER MANUAL V2.2



主编：陈睿 邓野

武汉京天电器有限公司

WUHAN JINGTIAN ELECTRICAL CO.,LIMITED

2020 年 11 月 武汉

本用户手册适用于 ROBOTIS 公司的华夫派和汉堡机器人。

安装、使用产品前，请阅读本手册。

用户手册会定期进行检查和修正，更新后的内容将出现在新版本中。本手册中的内容或信息如有变更，恕不另行通知。

对本手册中可能出现的任何错误或遗漏，或因使用本手册及其中所述产品而引起的意外或间接伤害，武汉京天电器有限公司概不负责。

请保管好本手册，以便可以随时阅读和参考。

本手册中所有图片仅供示意参考，请以收到的实物为准。

本手册为武汉京天电器有限公司专有财产，非经武汉京天电器有限公司书面许可，不得复印、全部或部分复制或转变为任何其他形式使用。

Copyright © 2010-2020 武汉京天电器保留所有权利。

注意:

1、这些指令是在Ubuntu 16.04 和 ROS Kinetic测试运行的。

2、如果您想了解移动抓取或是机械臂相关信息，请参考武汉京天电器手持机械臂用户手册以及韭菜盒子用户手册。

我们很高兴宣布一本新书《ROS机器人编程》，这本书是由Turtlebot3开发人员编写的。这本书以韩文、英文、中文和日本出版。它包含以下内容：

- ROS动力学KAM:基本概念、指令和工具
- 如何在ROS上使用传感器和执行器包
- 嵌入式ROS板：OpenCR
- 用Turtlebot3进行SLAM和导航
- 如何使用ROS Java编程实现递送机器人
- *韭菜盒子（LABOX-V1）移动机械臂使用MOVEIT的仿真Gazebo

如果需要详细了解，请参阅了解更多关于ROS、SLAM和导航的信息。

如果需要了解有关ROS2相关操作请参考下方链接：

<http://emanual.robotis.com/docs/en/platform/turtlebot3/bringup/>

3、在使用Turtlebot3之前，请先了解下Ubuntu系统及ros基础

目录

1. 结构特点.....	4
2. 【远程 PC】安装 Ubuntu 和 ROS 配置.....	5
2.1 安装 Ubuntu.....	5
2.2 安装 ROS 版本 kinetic 和相关包.....	6
2.3 【远程 PC】安装 TurtleBot3 及依赖包.....	6
3. 安装树莓派系统.....	8
3.1 系统说明:	8
3.2 下载镜像:	8
3.2 安装镜像:	8
3.4 开机并连接 wifi.....	9
4. OpenCR 固件配置.....	9
5. 远程连接配置.....	10
6. ROS 操作机器人.....	11
6.1 启动 TurtleBot3.....	11
6.2 在 RVIZ 中加载 Turtlebot3.....	12
6.3 控制.....	13
6.4 ROS 基础示例.....	13
7. 地图构建 SLAM.....	15
8. 自主导航和路径规划.....	18
9. Turtlebot3 ROS 仿真.....	21
9.1 安装测试.....	21
9.2 Gazebo 仿真.....	22
10. 安装 Arduino 配置 OpenCR 库.....	29
10.1 安装 Arduino IDE.....	30
10.2 添加 OpenCR 的依赖库.....	30
10.3 端口设置.....	33
10.4 刷固件【Bootloader】.....	33
10.5 上传程序.....	35
11. 树莓派相机的使用.....	36
11.2 树莓派启用摄像头.....	37
11.3 ROS 运行树莓派相机.....	38
11.4 相机标定.....	38
12. 常见问题解决办法.....	40
12.1 Turtlebot3 软件更新.....	40
12.2 时间同步.....	40
12.3 树莓派上 SSH 远程连接.....	41
12.4 远程无法控制小车.....	41
12.5 工作空间编译错误.....	41

1. 结构特点

TurtleBot3 是一款完全基于 ROS 开发设计的移动机器人，它的设计初衷是为教育、研发和机器人爱好者的提供一个功能完善的实验平台。

它外形小巧、成本低廉、整体开源、完全可编程的特点，使他成为移动机器人教学领域的标杆，深受全世界各地的师生和科研人员的欢迎。

低成本：TurtleBot 是为了从教育和原型研究和发展的成本意识的需求而建立的。

TurtleBot3 是配备了通用 360 度 LiDAR 的 SLAM 移动机器人中入门级的机器人。

小尺寸：TurtleBot3 Basic 的尺寸为 140mm x 140mm x 150mm（长 x 宽 x 高）。

ROS 标准：TurtleBot 品牌由 Open Source Robotics Foundation, Inc. (OSRF) 管理，OSRF 开发和管理 ROS。

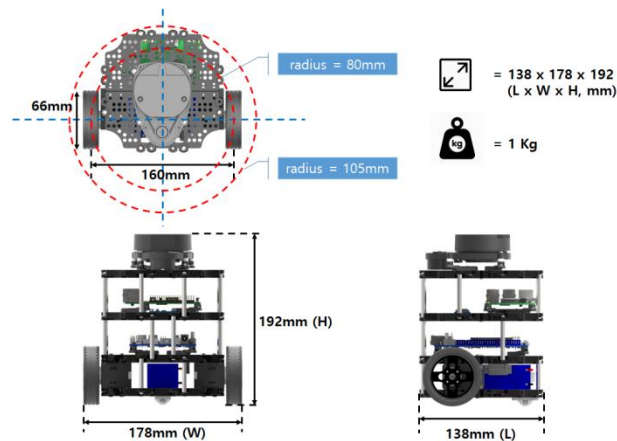
结构可扩展性：TurtleBot3 整体结构由可 3D 打印的模块拼装而成的，支持开发者自己设计、更改结构，并且官方提供了很多的扩展改装案例。

TurtleBot3 汉堡和华夫派的基本参数如下表：

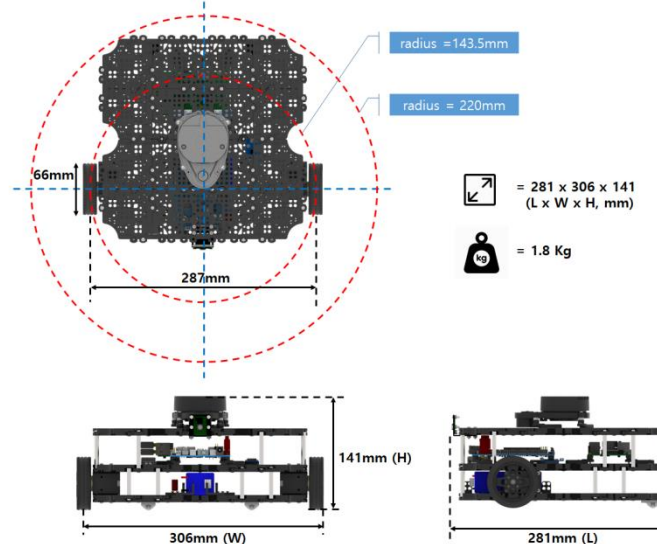
	汉堡	华夫派
最大平移速度	0.22m / s	0.26m / s
最大转速	2.84rad / s	1.82rad / s
最大有效载荷	15kg	30kg
尺寸（长 x 宽 x 高）	176 x 138 x 188 mm	306 x 283 x 143 mm
自身重量	1kg	1.8kg
越障能力	10mm	
连续工作时间	2h 30m	2h
惯性里程计（IMU）	3 轴陀螺仪、3 轴加速度计、3 轴磁力计	
插脚	GPIO 18 引脚	
音频	可编程蜂鸣声序列	
可编程 LED	4 个	
电池	锂聚合物 11.1V 1800mAh / 19.98Wh 5C	
充电适配器	输入：220V，AC 50 / 60Hz，1.5A 输出：12V DC，5A	

TurtleBot3 汉堡和华夫派的外形尺寸：

TurtleBots Burger



TurtleBot3 Waffle Pi



硬件组装手册在产品装箱中可以找到，如有遗失可在下面网址中自行下载。

汉堡 <http://www.robotis.com/service/download.php?no=748>

华夫派 <http://www.robotis.com/service/download.php?no=750>

组装视频参考地址 <https://youtu.be/rvm-m2ogrLA>

2. 【远程 PC】安装 Ubuntu 和 ROS 配置

2.1 安装 Ubuntu

注意：建议不要安装在虚拟机上，网络配置比较难，系统也会很不流畅。最好单独安装 Ubuntu，或者和 windows 并装成双系统。

1. 单独安装 Ubuntu：在远程 PC（笔记本电脑）中安装 Ubuntu 16.04，参考：

<https://www.ubuntu.com/download/desktop/install-ubuntu-desktop>

2. 双系统安装参考：

<https://blog.csdn.net/flyyufenfei/article/details/79187656>

2.2 安装 ROS 版本 kinetic 和相关包

```
$ wget https://raw.githubusercontent.com/oroca/oroca-ros-pkg/kinetic/ros_install.sh && chmod 755 ./ros_install.sh && bash ./ros_install.sh catkin_ws kinetic
```

- 详细安装过程: <http://wiki.ros.org/kinetic/Installation/Ubuntu>

2.3 【远程 PC】安装 TurtleBot3 及依赖包

(1) 安装依赖库

```
$ sudo apt-get install ros-kinetic-joy ros-kinetic-teleop-twist-joy
ros-kinetic-teleop-twist-keyboard ros-kinetic-laser-proc ros-kinetic-rgbd-launch
ros-kinetic-depthimage-to-laserscan ros-kinetic-rosserial-arduino ros-kinetic-rosserial-python
ros-kinetic-rosserial-server ros-kinetic-rosserial-client ros-kinetic-rosserial-msgs
ros-kinetic-amcl ros-kinetic-map-server ros-kinetic-move-base ros-kinetic-urdf
ros-kinetic-xacro ros-kinetic-compressed-image-transport ros-kinetic-rqt-image-view
ros-kinetic-gmapping ros-kinetic-navigation ros-kinetic-interactive-markers
```

(3) 编译 turtlebot3

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

```
$ git clone https://github.com/ROBOTIS-GIT/turtlebot3_msgs.git
```

```
$ git clone -b kinetic-devel https://github.com/ROBOTIS-GIT/turtlebot3.git
```

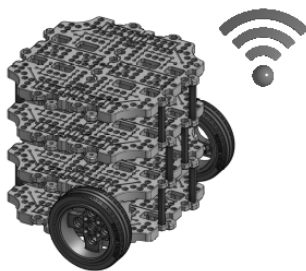
```
$ cd ~/catkin_ws && catkin_make
```

如果 `catkin_make` 出现错误 运行:

```
$ cd ~/catkin_ws && rosdep install -r --from-path . # (后面有一个点, 别漏掉)
```

(4) 网络配置

[TurtleBot]



```
ROS_MASTER_URI = http://IP_OF_PC:11311
ROS_HOSTNAME   = IP_OF_TURTLEBOT
```

[Remote PC]



```
ROS_MASTER_URI = http://IP_OF_PC:11311
ROS_HOSTNAME   = IP_OF_PC
```

* ROS Master is running in Remote PC.

ROS 需要 IP 地址在 turtlebot 和远程 PC 之间进行通信

分别在 turtlebot 和 PC, 执行如下命令获得对应的 IP 地址:

```
$ ifconfig
```

```
enp3s0  Link encap:Ethernet  HWaddr d8:cb:8a:f3:d3:00
        inet addr:192.168.0.165  Bcast:192.168.0.255  Mask:255.255.255.0
        inet6 addr: fe80::b5ed:414a:b396:f212/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:118368 errors:0 dropped:0 overruns:0 frame:0
        TX packets:62573 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:114480539 (114.4 MB)  TX bytes:8118317 (8.1 MB)
        Interrupt:19

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:8912 errors:0 dropped:0 overruns:0 frame:0
        TX packets:8912 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1
        RX bytes:1713294 (1.7 MB)  TX bytes:1713294 (1.7 MB)

wlp2s0  Link encap:Ethernet  HWaddr ac:2b:6e:6d:08:ee
        inet addr:192.168.0.100  Bcast:192.168.1.255  Mask:255.255.255.0
        inet6 addr: fe80::77a:7d5c:9ca8:bd9c/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:468 errors:0 dropped:0 overruns:0 frame:0
        TX packets:630 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:107986 (107.9 KB)  TX bytes:89522 (89.5 KB)
```

修改.bashrc

\$ gedit ~/.bashrc 或者 nano ~/.bashrc

```
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

if [ -x /usr/bin/mint-fortune ]; then
    /usr/bin/mint-fortune
fi

alias eb='nano ~/.bashrc'
alias sb='source ~/.bashrc'
alias gs='git status'
alias gp='git pull'
alias cw='cd ~/catkin_ws'
alias cs='cd ~/catkin_ws/src'
alias cm='cd ~/catkin_ws && catkin_make'

source /opt/ros/kinetic/setup.bash
source ~/catkin_ws/devel/setup.bash

export ROS_MASTER_URI=http://192.168.0.100:11311
export ROS_HOSTNAME=192.168.0.100
```

Get Help Write Out Where Is Cut Text Justify Cur Pos
Exit Read File Replace Uncut Text To Spell Go To Line

在最下面添加 turtlebot3 的 IP 地址:

```
$ ROS_MASTER_URI = http://【PC 的 IP 地址】:11311
```

```
$ ROS_HOSTNAME = 【树莓派的 IP】
```

PC 配置如下:

```
$ ROS_MASTER_URI = http://【PC 的 IP 地址】:11311
```

```
$ ROS_HOSTNAME = 【PC 的 IP】
```

ROS_MASTER 运行在远程 PC 上。

让环境生效:

```
$ source ~/.bashrc
```

3. 安装树莓派系统

由于树莓派 3B+后运行 Ubuntu meta 系统, 会出现彩虹屏幕, 无法使用, Ubuntu 官方也没有给出解决方案。网络上有人给出的一些解决方法, 虽然可以解决彩虹屏幕, 但是后续使用中会发现很多问题, 如树莓派相机无法驱动、网路问题等。所以 TB3 官方推荐使用 Raspbian 系统。

接下来介绍预装了 ROS 和 TB3 的【集成 Raspbian】系统的安装方法。

3.1 系统说明:

该系统是 turtlebot3 官方制作, 它基于 Raspbian Stretch 与桌面, Raspbian 基于 Debian Stretch:

解决了树莓派 3B+ 系统不兼容问题

删除了 Wolfram, Mathematica, Minecraft Pi 和 Oracle Java SE 等非自由软件

删除了 libreoffice 以减少镜像大小

使用 raspi-config 启用 SSH 和 Camera 功能

更改密码: **turtlebot**

3.2 下载镜像:

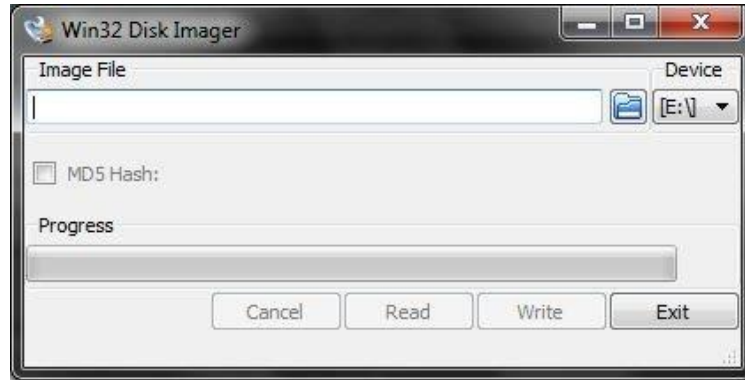
在京天的百度云盘中, 包括经过反复测试的镜像文件和系统烧写软件:

链接: https://pan.baidu.com/s/1at_3nTAWthj3D_RjKL0zoQ

提取码: edjq

3.2 安装镜像:

- 先把下载的文件解压成.img 文件
- 格式 Miscro SD 卡为 FAT32 格式
- 启动 Win32DiskImager



d.

- e. Image File 处选择 Raspbian 映像文件
- f. Device 处选择盘符为你读卡器的盘符
- g. 点 Write, 然后点一下 Yes 确定操作, 开始系统写入
- h. 写入完成, 提示成功
- i. Micro SD 卡插入树莓派, 接通电源启动

注意: 如果第一次开机链接不上 wifi, 就重新配置一次。

3.4 开机并连接 wifi

完成后, 把 SD 内存卡插入树莓派中, 连接鼠标键盘和 HDMI 显示器, 给树莓派供电开机。

密码是 **turtlebot**。

首次开机后可能无法连接 wifi?

这是因为不同国家的 wifi 协议不同, 需要先配置 wifi 所在区域, 步骤:

- (1) 点击右上角 **【网络信号图标】**, 电机 **【click here to set Wifi-country】**。
- (2) 在弹出的对话框中 **【country】** 右边的菜单中选择 **【CN china】**。保存关闭。
- (3) 再点击 **【网络信号图标】** 就会出现当前 wifi 列表。选择你自己的 wifi, 输入密码。

4. OpenCR 固件配置

方法一: 通过 shell 脚本

连接上 OpenCR 与 Ubuntu 电脑用 USB 线连接起来:

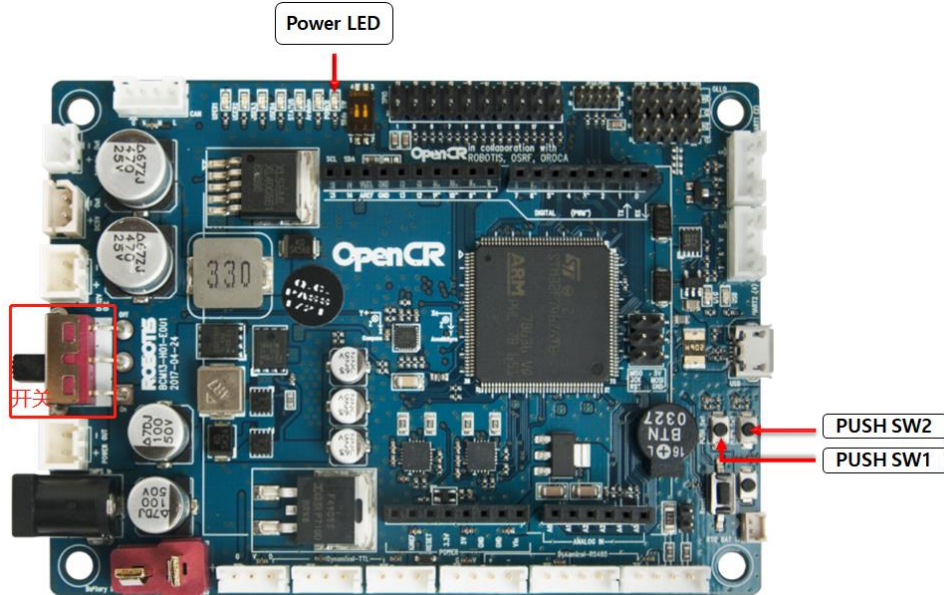
- **TurtleBot3 汉堡, 连接:**
- `export OPENCNCR_PORT=/dev/ttyACM0`
- `export OPENCNCR_MODEL=burger` #如果是汉堡设为 burger, 如果是华夫派设为 waffle
- `rm -rf ./opencnrcr_update.tar.bz2`
- `wget`
`https://github.com/ROBOTIS-GIT/OpenCR-Binaries/raw/master/turtlebot3/ROS1/latest/opencnrcr_update.tar.bz2 && tar -xvf opencnrcr_update.tar.bz2 && cd ./opencnrcr_update && ./update.sh $OPENCNCR_PORT $OPENCNCR_MODEL.opencnrcr && cd ..`
- ***** 注意最后这句比较长, 从 wget....一直到 cd .. 是一整句, 不要断开。**

方法二: 通过 arduino 程序

OpenCR 的 arduino 程序配置流程见后文“10.安装 Arduino 配置 OpenCR 库”。

测试 OpenCR 是否烧写成功

1. 连接机器人两个舵机和 OpenCR,
2. 接通 12V 电源（电源适配器或者电池）
3. 打开 OpenCR 的开关，这时电机会抱死（即加了扭矩，用手拧不动了）
4. 然后长按 sw1（或者 sw2）几秒钟，小车会向前走或者原地旋转。这就说明程序烧好了



5. 远程连接配置

很多初学者在这个步骤经常出问题，希望大家仔细阅读，细心输入命令。
步骤：

(1) 首先需要检查【笔记本电脑】和【树莓派】的 IP 地址。

打开终端（ctrl+alt+t），在终端中输入 ifconfig。

```
+ ~ ifconfig
lo        Link encap:本地环回
          inet 地址:127.0.0.1 掩码:255.0.0.0
          inet6 地址: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  跃点数:1
          接收数据包:7751 错误:0 丢弃:0 过载:0 帧数:0
          发送数据包:7751 错误:0 丢弃:0 过载:0 载波:0
          碰撞:0 发送队列长度:1000
          接收字节:1289503 (1.2 MB)  发送字节:1289503 (1.2 MB)

wls8     Link encap:以太网 硬件地址 b8:81:98:e5:13:d9
          inet 地址:192.168.3.87 广播:192.168.3.255 掩码:255.255.0
          inet6 地址: fe80::94f6:43f1:4a9a:a678/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  跃点数:1
          接收数据包:376302 错误:0 丢弃:0 过载:0 帧数:0
          发送数据包:468743 错误:0 丢弃:0 过载:0 载波:0
          碰撞:0 发送队列长度:1000
          接收字节:384786603 (384.7 MB)  发送字节:554146437 (554.1 MB)
```

分别记录下来【笔记本电脑 IP】和【树莓派 IP】，千万不能搞混了，切记！切记！切记！

(2) 配置环境变量

在终端打开

```
$ gedit ~/.bashrc
```

TURTLEBOT 的.bashrc 文件末尾添加如下两行：

```
$ export ROS_MASTER_URI=http://【笔记本电脑 IP】:11311
```

```
$ export ROS_HOSTNAME=【树莓派 IP】
```

笔记本 PC 的.bashrc 文件末尾添加如下两行：

```
$ export ROS_MASTER_URI=http://【笔记本电脑 IP】:11311
```

```
$ export ROS_HOSTNAME=【笔记本电脑 IP】
```

解释：ros 中只有一个节点是主节点，ROS_MASTER_URI 即表示主节点是谁，我们这里设置在笔记本 PC 上。ROS_HOSTNAME 表示本机的 IP 地址。所以两个文件中的 ROS_MASTER_URI 应该是相同的，都是【笔记本电脑 IP】。而 ROS_HOSTNAME 为各自的 IP 地址。这里错了就运行不了。

(3) 让环境生效：

```
$ source ~/.bashrc #修改完 IP 一定要 source 一下
```

提示：最好把所有的终端都重启一下。

6. ROS 操作机器人

6.1 启动 TurtleBot3

[Remote PC]运行 ROS core (主节点)

```
$ roscore
```

[TurtleBot3 SBC]给 LiDAR 连接到 ttyUSB0 的权限

```
$ sudo chmod a+rw /dev/ttyUSB0
```

[TurtleBot3 SBC]启动 launch 文件

```
$ roslaunch turtlebot3_bringup turtlebot3_robot.launch
```

如果模型是汉堡启动成功后你将看到下面打印信息

SUMMARY

=====

PARAMETERS

- * /rostdistro: kinetic
- * /rosversion: 1.12.13
- * /turtlebot3_core/baud: 115200
- * /turtlebot3_core/port: /dev/ttyACM0
- * /turtlebot3_core/tf_prefix:
- * /turtlebot3_lds/frame_id: base_scan
- * /turtlebot3_lds/port: /dev/ttyUSB0

NODES

/

- turtlebot3_core (roserial_python/serial_node.py)
- turtlebot3_diagnostics (turtlebot3_bringup/turtlebot3_diagnostics)
- turtlebot3_lds (hls_lfcd_lds_driver/hlds_laser_publisher)

```
ROS_MASTER_URI=http://192.168.1.2:11311
```

```
process[turtlebot3_core-1]: started with pid [14198]
process[turtlebot3_lds-2]: started with pid [14199]
process[turtlebot3_diagnostics-3]: started with pid [14200]
[INFO] [1531306690.947198]: ROS Serial Python Node
[INFO] [1531306691.000143]: Connecting to /dev/ttyACM0 at 115200 baud
[INFO] [1531306693.522019]: Note: publish buffer size is 1024 bytes
[INFO] [1531306693.525615]: Setup publisher on sensor_state [turtlebot3_msgs/SensorState]
[INFO] [1531306693.544159]: Setup publisher on version_info [turtlebot3_msgs/VersionInfo]
[INFO] [1531306693.620722]: Setup publisher on imu [sensor_msgs/Imu]
[INFO] [1531306693.642319]: Setup publisher on cmd_vel_rc100 [geometry_msgs/Twist]
[INFO] [1531306693.687786]: Setup publisher on odom [nav_msgs/Odometry]
[INFO] [1531306693.706260]: Setup publisher on joint_states [sensor_msgs/JointState]
[INFO] [1531306693.722754]: Setup publisher on battery_state [sensor_msgs/BatteryState]
[INFO] [1531306693.759059]: Setup publisher on magnetic_field [sensor_msgs/MagneticField]
[INFO] [1531306695.979057]: Setup publisher on /tf [tf/tfMessage]
[INFO] [1531306696.007135]: Note: subscribe buffer size is 1024 bytes
[INFO] [1531306696.009083]: Setup subscriber on cmd_vel [geometry_msgs/Twist]
[INFO] [1531306696.040047]: Setup subscriber on sound [turtlebot3_msgs/Sound]
[INFO] [1531306696.069571]: Setup subscriber on motor_power [std_msgs/Bool]
[INFO] [1531306696.096364]: Setup subscriber on reset [std_msgs/Empty]
[INFO] [1531306696.390979]: Setup TF on Odometry [odom]
[INFO] [1531306696.394314]: Setup TF on IMU [imu_link]
[INFO] [1531306696.397498]: Setup TF on MagneticField [mag_link]
[INFO] [1531306696.400537]: Setup TF on JointState [base_link]
[INFO] [1531306696.407813]: -----
[INFO] [1531306696.411412]: Connected to OpenCR board!
[INFO] [1531306696.415140]: This core(v1.2.1) is compatible with TB3 Burger
[INFO] [1531306696.418398]: -----
[INFO] [1531306696.421749]: Start Calibration of Gyro
[INFO] [1531306698.953226]: Calibration End
```

提示: 如果要分别启动激光雷达传感器, Raspberry Pi 摄像头, 英特尔®实感™MR200 或核心, 请使用以下命令。

```
$ roslaunch turtlebot3_bringup turtlebot3_lidar.launch
$ roslaunch turtlebot3_bringup turtlebot3_rpicamera.launch
$ roslaunch turtlebot3_bringup turtlebot3_realsense.launch
$ roslaunch turtlebot3_bringup turtlebot3_core.launch
```

6.2 在 RVIZ 中加载 Turtlebot3

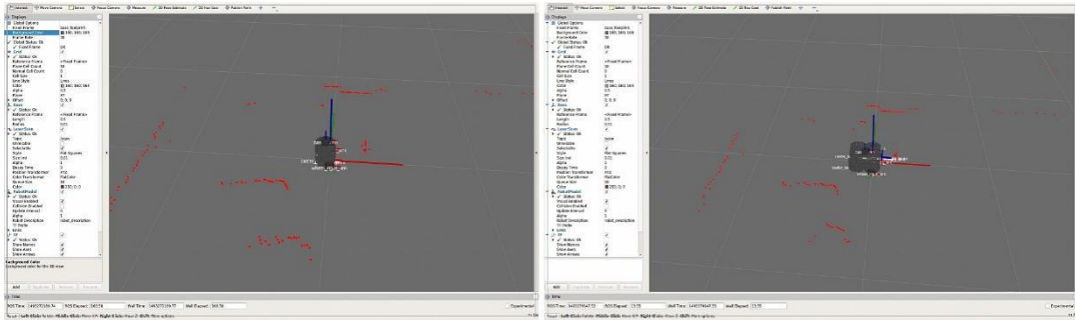
[远程 PC]启动机械手状态发布者并运行 RViz。

```
$ export TURTLEBOT3_MODEL=${TB3_MODEL} //${TB3_MODEL}是你正在使用的模型
的名称 burger, waffle, waffle_pi
```

```
$ roslaunch turtlebot3_bringup turtlebot3_remote.launch
```

打开一个新的终端窗口, 然后输入以下命令。

```
$ rosrun rviz rviz -d `rospack find turtlebot3_description`/rviz/model.rviz
```



6.3 控制

键盘控制

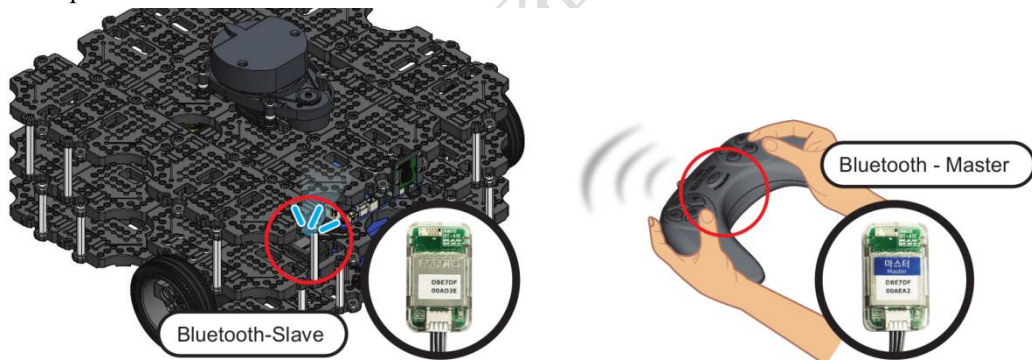
[Remote PC] 启动进行简单远程操作测试

```
$ roslaunch turtlebot3_teleop turtlebot3_teleop_key.launch
```

[Remote PC] 如果文件成功启动，可以用‘w s a d’按键控制前进、后退、左转、右转。

RC100

用于 TurtleBot3 Burger, Waffle 和 Waffle Pi 的 OpenCR 固件中包含 ROBOTIS RC-100B 控制器的设置。该控制器可与蓝牙模块 BT410 一起使用。TurtleBot3 Waffle Pi 包括此控制器和蓝牙模块。使用 RC-100 时，无需执行特定的节点，因为该 turtlebot_core 节点/cmd_vel 在直接连接到 OpenCR 的固件中创建了一个主题。



6.4 ROS 基础示例

注意： 执行此示例之前，请确保在 turtlebot3 上先执行 [Bringup](#) 指令，并在桌上测试机器人时要小心，因为机器人可能会掉落。

使用交互式标记移动

可以通过 RViz 上的交互式标记移动 TurtleBot3。您可以使用交互式标记将 TurtleBot3 旋转或旋转。

[Remote PC] 打开新终端并启动远程文件。

提示：在执行此命令之前，您必须指定 TurtleBot3 的型号名称。该 `{TB3_MODEL}` 是你正在使用的模型的名称 burger, waffle, waffle_pi。如果要永久设置导出设置，请参阅“导出 TURTLEBOT3_MODEL”页面。

```
$ export TURTLEBOT3_MODEL=${TB3_MODEL}
```



```
$ roslaunch turtlebot3_bringup turtlebot3_remote.launch
```

[Remote PC]启动交互式标记文件。

```
$ roslaunch turtlebot3_example interactive_markers.launch
```

[Remote PC]使用 RViz 在 3D 模式下可视化模型。

```
$ rosrn rviz rviz -d `rospack find turtlebot3_example`/rviz/turtlebot3_interactive.rviz
```

障碍物检测

可通过 LDS 数据移动或停止 TurtleBot3。当 TurtleBot3 移动时，它会在检测到前方障碍物时停止。

[Remote PC]启动障碍文件。

```
$ roslaunch turtlebot3_example turtlebot3_obstacle.launch
```

点操作

TurtleBot3 可以通过 2D point (x, y)和进行移动 z-angular。例如，如果插入 (0.5, 0.3, 60)，TurtleBot3 将移动到点 (x = 0.5m, y = 0.3m)，然后旋转 60 度。

[Remote PC]启动 pointop 文件。

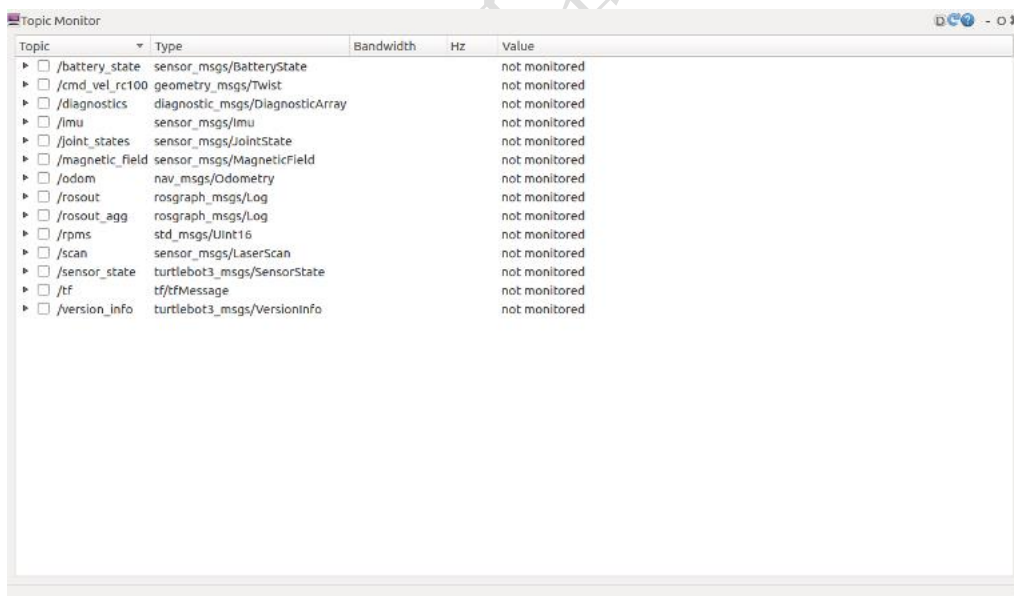
```
$ roslaunch turtlebot3_example turtlebot3_pointop_key.launch
```

主题监控

为了检查 TurtleBot3 的主题，我们将使用 ROS 提供的 rqt。rqt 是用于 ROS 的 GUI 开发的基于 Qt 的框架。rqt 是一种工具，通过显示主题列表中的所有主题，用户可以轻松查看主题状态。GUI 中有主题名称，类型，带宽，Hz 值。

[远程 PC]运行 rqt

```
$ rqt
```



提示：如果未显示 rqt，请选择 plugin-> Topics-> Topic Monitor。

首次运行 rqt 时，不监视主题值。要监视主题，请单击每个主题旁边的复选框。