

|  |
| --- |
| 浙江重德智能科技有限公司 |
| **XBot-U使用手册** |
| **User Manual of XBot-U** |

**Version 1.1.4**

**修订记录**

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **修订内容** | **修订人** |
| 2019.05.07 | 1.0.0 | 基于旧版用户手册调整文档结构 | 席静 |
| 2019.05.19 | 1.0.1 | 进一步调整了文档结构  整合了主从关系设置部分内容  调整了产品组成部分内容 | 韩济阳 |
| 2019.05.11 | 1.0.2 | 更新了自主导航部分 | 韩济阳 |
| 2019.05.15 | 1.0.3 | 调整文档结构  更新了驱动机器人部分的内容  更新了人脸识别部分的内容  对文档重新排版 | 韩济阳 |
| 2019.05.16 | 1.0.4 | 更新了人脸识别部分的内容  更新了语音交互部分的内容  为图片添加了序号  简化了使用已知地图部分的描述 | 韩济阳 |
| 2019.05.21 | 1.0.5 | 更新了机器人分解图  更新了人脸注册章节，增加了人脸注销  对文档进行初步校验与修改 | 席静 |
| 2019.05.22 | 1.0.6 | 更新充电时间和待机时间  添加3.2.3对话和语音控制 | 韩济阳 |
| 2019.05.27 | 1.0.7 | 更新语音交互部分的内容 | 王晓云 |
| 2019.06.03 | 1.0.8 | 更新语音交互部分，对应最新的talker代码 | 王晓云 |
| 2019.07.02 | 1.0.9 | 更新app使用说明  更新四、五章节导航相关文档 | 席静 |
| 2019.07.02 | 1.1.0 | 校阅并发布 | 汪鹏 |
| 2019.09.27 | 1.1.1 | 1. 调整“配置您的个人计算机”章节，删减主从配置的非必要内容 2. 增加3.2章节 | 席静 |
| 2020.07.27 | 1.1.3 | 修改语音交互和自定义关键词说明为新版talker | 王晓云 |

目录

[1. 快速认识XBot 4](#_Toc47028934)

[1.1. 产品组成 5](#_Toc47028935)

[1.1.1. XBot-U产品清单 5](#_Toc47028936)

[1.1.2. 机器人主机 5](#_Toc47028937)

[1.1.3. 机器人控制平板 6](#_Toc47028938)

[1.1.4. Realsense RGBD摄像头 6](#_Toc47028939)

[1.1.5. XBot-U机器人标准充电器 7](#_Toc47028940)

[1.2. 产品特点 7](#_Toc47028941)

[2. 初步上手XBot-U机器人 9](#_Toc47028942)

[2.1. 使用前须知 9](#_Toc47028943)

[2.1.1. 认识急停开关 9](#_Toc47028944)

[2.1.2. 如何开机与关机 9](#_Toc47028945)

[2.1.3. 开机常规检查 10](#_Toc47028946)

[2.1.4. 了解机器人配置信息 11](#_Toc47028947)

[2.1.5. 了解更多设备支持 12](#_Toc47028948)

[2.2. 组装机器人硬件 13](#_Toc47028949)

[2.3. 开机并使用平板简单控制机器人 14](#_Toc47028950)

[3. 深入了解和使用机器人 15](#_Toc47028951)

[3.1. 启动功能节点 16](#_Toc47028952)

[3.1.1. 机器人功能包简介 16](#_Toc47028953)

[3.1.2. 启动常用功能 16](#_Toc47028954)

[3.1.3. 驱动机器人底盘 17](#_Toc47028955)

[3.1.4. 驱动摄像头 17](#_Toc47028956)

[3.1.5. 驱动激光雷达 18](#_Toc47028957)

[3.2. 节点状态查询与控制 19](#_Toc47028958)

[3.2.1. 核心传感器数据查询 19](#_Toc47028959)

[3.2.2. 红外与超声传感器 21](#_Toc47028960)

[3.2.3. 键盘控制检测 22](#_Toc47028961)

[3.2.4. 云台控制 23](#_Toc47028962)

[3.2.5. 摄像头数据检测 24](#_Toc47028963)

[4. 智能交互 25](#_Toc47028964)

[4.1. 人脸识别 25](#_Toc47028965)

[4.1.1. 注册人脸 25](#_Toc47028966)

[4.1.2. 识别人脸 30](#_Toc47028967)

[4.1.3. 人脸注销 31](#_Toc47028968)

[4.2. 语音交互 32](#_Toc47028969)

[4.2.1. 播放指定的语音文件 32](#_Toc47028970)

[4.2.2. 将输入的文字转化为语音 33](#_Toc47028971)

[4.2.3. 对话和语音控制 33](#_Toc47028972)

[4.2.4. 自定义对话内容 34](#_Toc47028973)

[4.3. 使用XBot助手控制您的机器人 36](#_Toc47028974)

[4.3.1. UXbot助手APP的安装 37](#_Toc47028975)

[4.3.2. UXbot助手APP的使用准备 38](#_Toc47028976)

[4.3.3. 设置 38](#_Toc47028977)

[4.3.4. XBot-U机器人状态查看与控制 40](#_Toc47028978)

[4.3.5. XBot-U机器人摄像头图像查看 42](#_Toc47028979)

[4.3.6. 人脸管理与识别 44](#_Toc47028980)

[5. 机器人SLAM与自主导航 48](#_Toc47028981)

[5.1. 启动建图程序 48](#_Toc47028982)

[5.2. 移动机器人完成建图 50](#_Toc47028983)

[5.3. 保存地图 51](#_Toc47028984)

[5.4. 配置导航地图参数 52](#_Toc47028985)

[5.5. 导航 53](#_Toc47028986)

[5.6. 进一步的应用开发 55](#_Toc47028987)

[6. XBot仿真与教学 56](#_Toc47028988)

[6.1. 启动XBot模拟器 56](#_Toc47028989)

[6.2. SLAM仿真 57](#_Toc47028990)

[6.3. 已知地图与导航仿真 58](#_Toc47028991)

[7. 常见问题 60](#_Toc47028992)

[7.1. 执行权限 60](#_Toc47028993)

[7.2. 配置您的个人计算机作为从机 60](#_Toc47028994)

[7.2.1. 安装Ubuntu操作系统 61](#_Toc47028995)

[7.2.2. 安装ROS 62](#_Toc47028996)

[7.2.3. ROS环境配置 63](#_Toc47028997)

[7.2.4. ROS安装测试 63](#_Toc47028998)

[7.2.5. 从机连接到xbot主机 64](#_Toc47028999)

[7.2.6. 在从机上部署xbot功能包 70](#_Toc47029000)

[7.2.7. 在主机上安装sshd 73](#_Toc47029001)

[7.3. 调节机器人的输出音量 75](#_Toc47029002)

[8. 联系我们 76](#_Toc47029003)

[8.1. 参考链接 76](#_Toc47029004)

[8.2. 联系我们 76](#_Toc47029005)

# 快速认识XBot

XBot系列机器人是重德智能科技出品的一款机器人平台解决方案，其中包含了机器人的硬件平台、软件平台和操作系统三方面的支持。

该系列经历了XBot1与XBot-U两代机器人的开发与改进，目前已完全支持室内环境中的所有的机器人ROS相关传感器接入和机器人应用。该平台自带有运动控制系统、二维激光雷达点云测距、超声测距、红外测距、高清人脸识别摄像头、RGBD深度高清摄像头以及机器人视觉伺服云台，可以满足您对室内移动机器人的大部分需求。

在人脸识别方面，XBot-U机器人搭载了由腾讯优图和中科院软件所联合推出的人脸识别盒子，用户按照SDK规定的接口调用方式访问人脸识别盒子，轻松获取当前的人脸识别结果。

在语音识别和语音交互方面，XBot也毫不逊色。依靠科大讯飞的强大语音处理能力，机器人上搭载了我们为XBot-U机器人特别定制的语音交互功能，它能够与人脸识别、机器人运动规划等功能联动起来，实现更多更复杂的功能。

XBot-U机器人平台将是您室内移动机器人应用的不二选择。



图1–1 XBot1(左)与XBot-U(右)

## 产品组成

### XBot-U产品清单

您购买的XBot-U智能机器人平台将包含以下产品：

| **产品清单** | **数量** |
| --- | --- |
| XBot-U机器人主机 | 1台 |
| 机器人控制平板（已预装控制APP） | 1台 |
| Realsense RGBD摄像头（附固定螺钉） | 1台 |
| XBot-U机器人标准充电器 | 1台 |
| XBot-U使用手册及配套教程（电子版，即本文档） | 1份 |

### 机器人主机

XBot-U机器人主机由很多部件构成，其硬件组成如下图所示：

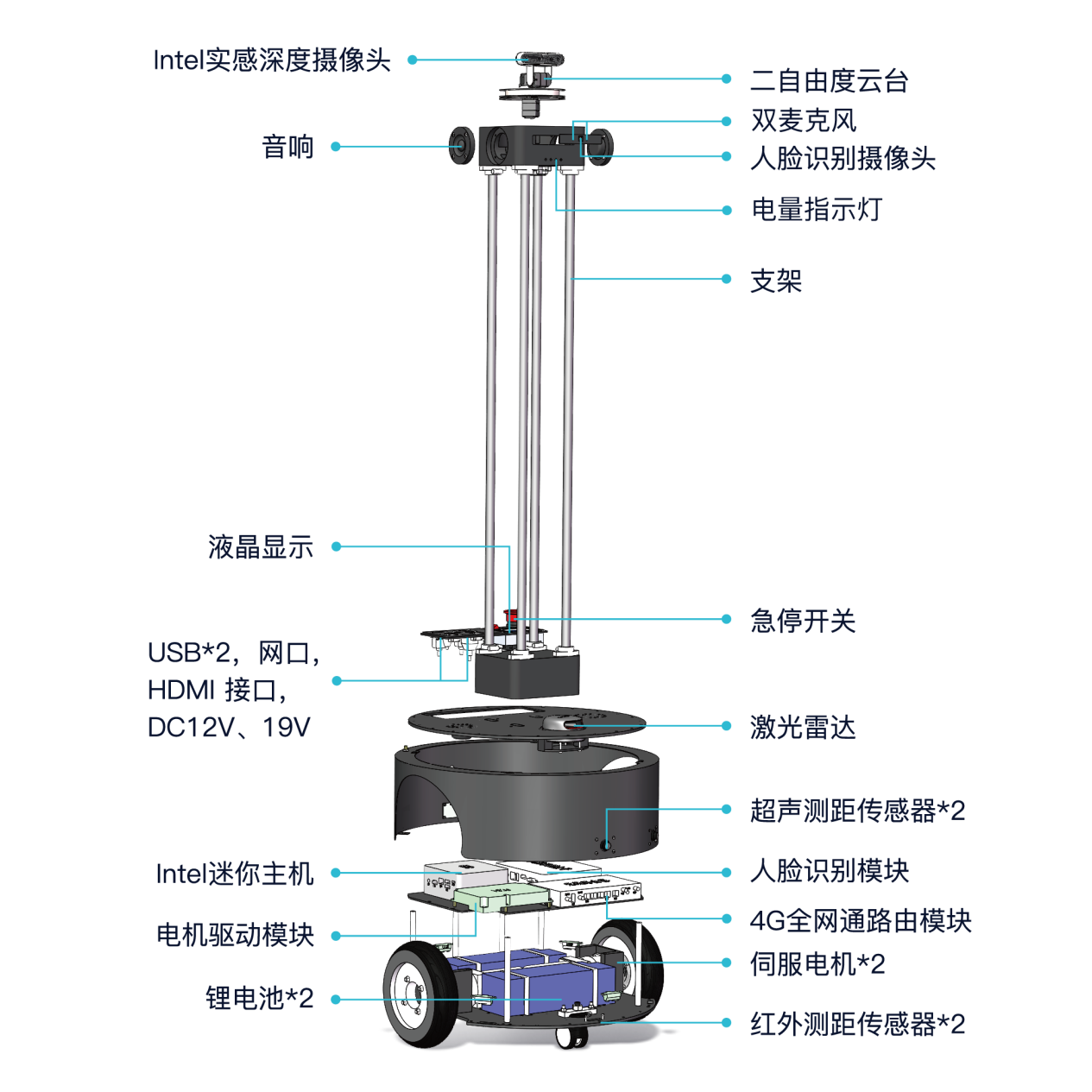


图1–2 XBot-U机器人分解图

### 机器人控制平板

配套的机器人控制平板可以帮助您在不方便携带PC端的情况下对XBot-U机器人进行简单运动控制、人脸注册管理等操作。你也可以在自己的Android设备中下载并安装UXBot助手APP实现相应的功能。关于机器人控制平板与UXBot助手APP的使用，可参看4.3部分。

### Realsense RGBD摄像头

附带Realsense RGBD摄像头可以帮助您获取机器人四周的图像与深度信息。 使用前将Realsense RGBD摄像头安装到XBot-U机器人顶端的云台上即可。具体的安装方法请参看2.2部分。

### XBot-U机器人标准充电器

XBot-U机器人随机附送标准充电器一只，当机器人顶部的电量指示灯只剩下一个灯亮时，说明机器人电量即将耗尽，此时您可使用该充电器对机器人进行充电。

XBot-U机器人标准充电器输入为100-240Vac 1.5A，输出为29.4Vdc 3A。需要充电时，请**先行关闭机器人电源（按下船型开关），**然后打开充电口保护盖，插入充电器，然后再将充电器插头插入插线板即可。充电器指示灯红色表示正在充电，绿色表示已充满，机器人一次正常充满预计耗时10小时左右。你也可以通过下方的液晶显示面板查看机器人的电量情况。

机器人的正常工作电压为20.5V~29.4V，当液晶显示面板指示在29.4V表示满电，**接近20.5V时则表示即将没电，请您及时为机器人充电。**

## 产品特点

* **稳定、可靠的运动控制**

XBot-U机器人运动电机采用先进的PID鲁棒性控制算法，提供稳定、可靠的机器人运动控制，配以高减速比的高精度电机，机器人运动速度可控制到0.01m/s的精度，最小速度达0.01m/s，最大速度达2m/s。具有加速时间短，制动效果明显等的多方面优秀特性。

* **完备的驱动软件支持**

我们为XBot-U机器人提供完备的驱动软件，采用国际通用的驱动软件框架和通信协议，能够提供50Hz频率以上的数据心跳包传输和快速精准地数据编码解码功能，使机器人的运动状态控制精度到达20ms以上，从而机器人能够更加迅速地响应用户算法的控制。

* **自主建图定位与导航**

XBot-U机器人具备室内环境下的自主建图定位与导航功能，该功能让机器人在室内实现完全自主的同步建图和定位，从而机器人能够根据用户需求，在任意位置之间自由穿梭行走，同时在导航过程中精准避障，全自主规划行走路径。

* **超长续航与自主充电**

XBot-U机器人配备高达26Ah的超大容量电池，续航时间最高可达24小时。未来还将支持用户预约返回充电和自主返回充电模式，实现机器人智能管理自身的能量，在能量不足时自动返回充电桩充电。

* **高性能计算能力**

XBot-U配备高性能的CPU计算能力，搭载Intel i5高性能处理器、大内存主机，能满足大多数场景下的计算需求。

* **ROS系统全支持**

XBot-U机器人软件框架专为ROS系统定制，可运行ROS系统下的所有软件和算法，运动控制和规划算法完全支持ROS系统协议，为更多的学习和开发者提供通用的算法验证和应用落地的平台。

* **搭载高精度人脸识别盒子**

由腾讯优图技术支撑的人脸识别盒子集成在机器人底盘内部，使机器人拥有国际前沿最为先进的人脸识别功能，识别率高达99.7%以上。经过重德对XBot-U机器人的移植和定制，用户只需要注册人脸、获取人脸识别结果两个简单的步骤，即可轻松实现人脸识别。

* **搭载语音交互模块**

XBot上集成了科大讯飞的语音识别、语音合成、语音交互等多种强大的语音类智能模块，经过对机器人系统的移植，出厂的机器人已经具备简单的对话功能，而且还能根据用户的配置，实现指定场景下的语音交互。

# 初步上手XBot-U机器人

本章将指导您配置自己的个人计算机，并对XBot-U机器人进行初步的上手使用。请您务必在使用XBot-U机器人之前阅读本章节。**当您暂时不使用XBot-U机器人时，请务必切断电源。如果机器人正在连接鼠标、键盘、显示器等设备，请按下急停开关，以防止误操作导致机器人移动，拖拽您的设备。**

## 使用前须知

### 认识急停开关

为了保证机器人运行的绝对安全，每台机器人都配备工业级标准的旋转红色急停按钮，用于紧急情况下给机器人电机断电。**在任何情况下，拍下该按钮，都将能够切断机器人电机电源，保证运行安全。**

### 如何开机与关机

每一台XBot-U机器人都内置了Ubuntu操作系统，相当于一台小型的计算机，因此，您可以像使用个人计算机电脑一样使用机器人。



图2–1 机器人底盘



图2–2 人脸识别摄像头与电量指示灯

机器人底盘上红色船型开关为机器人的启动开关。按下启动开关后，机器人电源接通，此时机器人上的液晶面板会实时显示当前机器人的电压、功率等信息。经过一小段时间的Ubuntu操作系统启动及机器人初始化命令执行之后（约30秒），云台位置归位，机器人前方的指示灯点亮，整个机器人完成启动。

与此同时，XBot-U也会打开自身的无线热点。您可在PC或者手机的Wifi列表中查看到以xbot-network-\*命名的WiFi热点，其中星号是您XBot-U机器人的编号。

机器人的关机只需要按下底盘上的启动开关即可。此时机器人掉电，所有功能均不可用。

### 开机常规检查

一台正常启动的XBot-U机器人，启动完成之后，我们一般需要检查机器人的如下状态：

1. 机器人电源液晶显示板，查看机器人的电压，功率等信息。
2. 检查急停开关状态，根据即将的操作合理选择按下或者弹起急停开关。
3. 激光雷达：默认开机初始化时启动了激光雷达，开机启动后，激光雷达正常应该处于不停的旋转状态。
4. 电量指示灯，电量指示灯指示机器人的当前电量，从左至右每一颗灯表示25%的电量，以亮2颗灯为例，即表示当前机器人还有25%~50%的电量，因此当只有一颗灯亮时，请您及时充电。
5. 云台状态，上电后，机器人初始化程序默认复位水平云台和竖直云台到0度。

### 了解机器人配置信息

以下项目在机器人出厂时均已配置好，**不需要您再次配置**，您只需了解这些信息即可，以方便您更好的使用本产品。

#### 路由器信息

每一台XBot上均配备有一台工业级的路由器，路由器支持双频以及3G网络。默认情况下，机器人是不连接外部网络，只在自身的局域网内运行和通讯。

路由器的配置信息如下：

|  |  |
| --- | --- |
| 管理ip地址 | 192.168.8.1 |
| 管理密码 | admin |
| 局域网ip段位 | 192.168.8.100~192.168.8.250 |
| 无线网ID | xbot-network-\*(\*为机器人编号) |
| 无线网密码 | xbot1234 |

#### 机器人主机信息

机器人主机信息如下：

|  |  |
| --- | --- |
| 固定Ip地址 | 192.168.8.101 |
| ROS\_MASTER\_URI | http://192.168.8.101:11311 |
| ROS\_HOSTNAME | 192.168.8.101 |
| 系统用户 | xbot |
| 用户密码 | xbot1234 |

### 了解更多设备支持

在XBot所搭载的大部分基础传感器和设备之外，机器人还留有丰富的接口环境提供给使用者进行扩展使用，让XBot具有更多的可能性。

#### USB通用设备

XBot具备两个USB3.0接口，该接口直接连接到机器人的大脑，即机器人主板上。依据外接设备的特性和功能，在设备厂商支持ROS驱动以及通用USB接口的前提下，用户可以将该设备接入到机器人上。

在使用机器人时，可以将鼠标和键盘通过USB3.0口接入，进行系统操作。

还可以使用指令：

$ lsusb

来查看设备的接入状态，接入之后能够看到列表当中有该设备的唯一接口编号，然后根据厂商提供的ROS驱动程序，将启动入口代码贴入launch文件中进行统一启动。

#### HDMI接口

除了通用USB接口之外，机器人还留有HDMI显示接口。某些需要从显示器上查看机器人的实际运行效果的时候，可以使用HDMI连接线连接电脑和显示器进行测试开发。

#### 预留的电源接口

有些设备除了需要连接接口进行数据传递之外，还可能需要提供一定的电源。在此，Xbot非常人性化的提供了两种标准电压的电源接口，即：

19V 5A x1

12V 5A x1

## 组装机器人硬件

当您拿到XBot-U机器人时大部分的组装工作已经完成，您只需执行如下两步：

1. 将两根天线接到底盘金色天线接头上。
2. 将Realsense RGBD摄像头安装到到机器人顶部的云台上。下面介绍如何安装Realsense RGBD摄像头。

首先，使用配套的固定螺丝将Realsense RGBD摄像头固定到机器人顶部的云台上，然后将摄像头通过Type-c接口与机器人相连，Realsense RGBD摄像头安装完毕。

安装好后的Realsense RGBD摄像头如下图：



图2–3 Realsense RGBD摄像头1



图2–4 Realsense RGBD摄像头2

## 开机并使用平板简单控制机器人

组装完机器人之后，您就可以开始使用了。最简单的使用方法就是通过搭载的平板上的UXbot助手APP来控制机器人。

UXbot助手APP目前可以通过app来查看机器人的状态、控制机器人移动、查看摄像头图像、人脸注册等功能。

app功能一直在持续升级改进，后续将为大家提供更多的功能。

1. 首先，我们需要开机启动机器人，按机器人底盘上的红色船型开关。按照2.1.3进行开机检查。
2. 开机正常后，拿出平板，首先设置wifi网络。将wifi连接到机器人的无线网络：“xbot\_network-bj\*”。wifi密码参考2.1.4.1章节。
3. 找到UXbot助手 APP并打开应用，当APP显示“已连接到ROS服务器”之后，就可以通过APP来使用机器人了。

**APP控制机器人人运动时，因为是人工控制状态，机器人不会自动避障，因此建议将机器人放到宽敞空旷的场所进行，遇到紧急情况，请按下机器人急停按钮。**如果对APP使用需要详细指导，可以参考4.3章节详细说明。

# 深入了解和使用机器人

作为一个技术爱好者或者研究人员，通过app控制机器人显然并不能满足您的要求，而且当前app功能的开发进度是滞后于机器人本身的。那么我相信您一定需要了解机器人的ROS 命令来深度了解和使用机器人。最简单直观的办法就是外置显示器和鼠标键盘对机器人进行操控，具体步骤如下：

1. 准备工作

由于机器人不带显示屏，为了更好的交互，我们建议您使用机器人自带的HDMI、USB3.0接口连接外部显示器与键鼠，对机器人进行操控。

您需要准备一台带HDMI接口的显示器，无线鼠标键盘一套（推荐无线设备，有线设备亦可）。在开机之前，将显示器的HDMI接口接入到机器人底盘上的HDMI接口上。将无线鼠标的发射器插入到机器人USB3.0口上。（此时您可以简单的将机器人看做一个普通的台式机主机去理解）

1. 开机并进行开机常规检查。具体操作详见2.1.2和2.1.3章节。
2. 开机之后，显示器就会显示Ubuntu操作系统的界面，此时我们就可以像使用一台普通的ubuntu操作系统的计算机那样使用机器人了。后续就是了解机器人的各种命令，并在ubuntu命令终端输入命令来查询或者操控机器人了。

说明：

如果您的身边一时很难找到一台显示器和鼠标键盘，那么您也可以通过您的个人PC电脑连接机器人的wifi，通过ssh远程控制机器人的方式进行操控。关于ssh的使用方法，是linux的基础知识，这里略去不谈。

此外，全部ssh远程控制需要在每开启一个命令窗口的时候，都进行ssh远程登录，比较麻烦，因此您还可以利用ROS提供的主从机制，设置XBot端（ROS主机）与PC端的主从关系，从而通过您的个人计算机对机器人进行控制。可参考7.2配置您的个人计算机作为从机章节。

## 启动功能节点

### 机器人功能包简介

在XBot-U机器人的工作空间（~/catkin\_ws/src/）中，我们已经部署了机器人的相关程序包：

1. 驱动包合集xbot（包括bringup，description，driver，msgs，node，safety-controller，tools和相关文档说明）
2. 人脸识别包（xbot\_face）
3. 运动规划包（xbot\_navi）
4. 语音交互包（xbot\_talker）

### 启动常用功能

在机器人上，打开命令终端输入下方的指令，可以通过我们编写好的一键启动脚本（xbot\_bringup/launch/xbot-u.launch文件）来启动XBot-U机器人常用功能。

$ roslaunch xbot\_bringup xbot-u.launch

该脚本里包含了Xbot的常用功能。输入该指令后后，机器人会启动底盘节点（包括机器人电机控制器，电机码盘，超声波传感器，红外传感器，电压电流检测器）、激光雷达、Realsense深度摄像头节点信息、人脸识别摄像头视频流等。

**注意，该脚本是默认启动后自动执行的，即开机时所有功能都已经为您启动好。**您也可以使用下面的指令将其全部关闭。

$ sudo service xbot stop

该指令需要您输入密码以确认权限。

### 驱动机器人底盘

如果你不想启动那么多功能，只想启动XBot-U机器人的部分功能，您也可以手动输入下面命令来启动XBot。

$ roslaunch xbot\_bringup xbot.launch

该脚本所驱动起来的传感器包括机器人电机控制器，电机码盘，超声波传感器，红外传感器，电压电流检测器等。xbot.launch脚本所启动的功能相比xbot-u.launch脚本要少些，主要集中在机器人的运动控制方面。

启动该脚本后，你将在终端上看到以下内容，表示脚本启动成功：

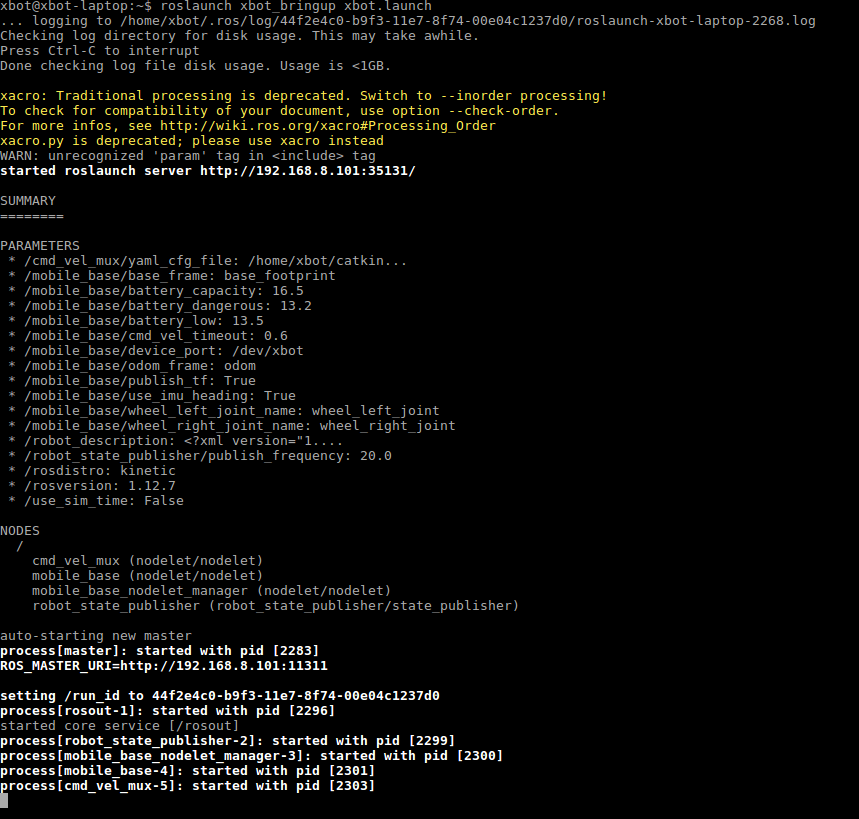
r

图3–1 xbot.launch脚本启动成功

### 驱动摄像头

XBot-U机器人配带有Intel RealSense 摄像头

使用下面指令可单独启动Realsense摄像头。

$ roslaunch xbot\_bringup realsense.launch

### 驱动激光雷达

使用下面指令可以单独启动Rplidar激光雷达。

$ roslaunch xbot\_bringup rplidar.launch

输入指令后，终端将显示如下信息，表示雷达启动成功。当雷达启动时输入，可以使雷达关闭。

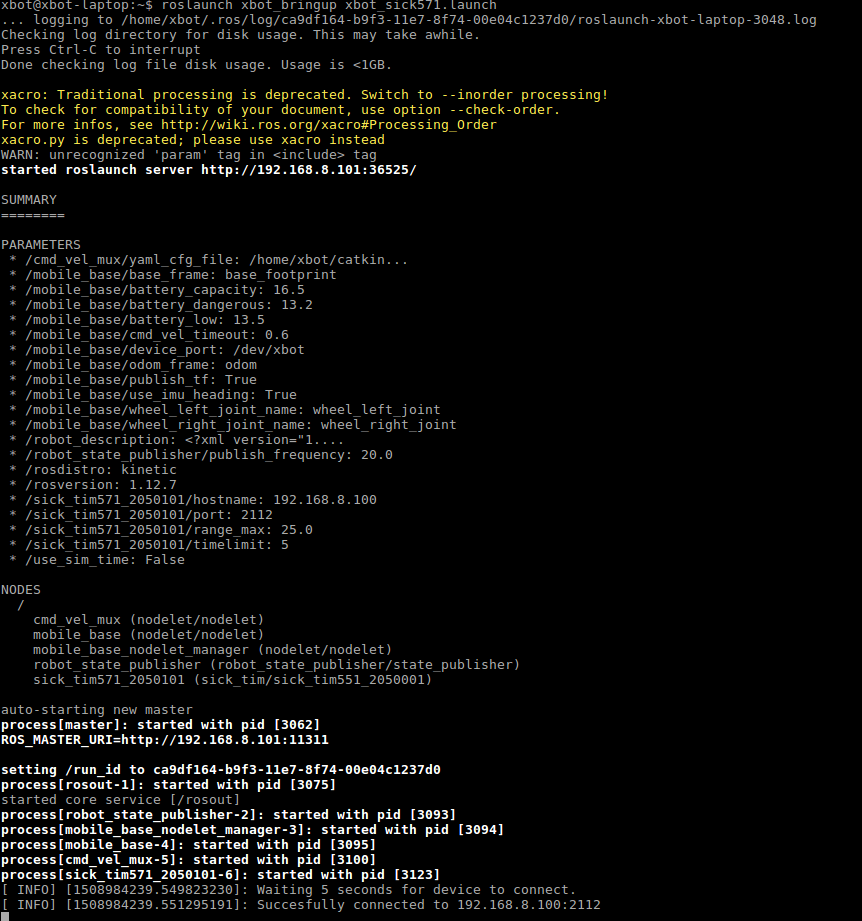


图3–2 雷达启动成功

如果您后续还需要启动其他节点，也可以使用不同指令启动相应的节点。

## 节点状态查询与控制

### 核心传感器数据查询

键入命令

$ rostopic echo /mobile\_base/sensors/core

即可查看相关传感器返回数据，也可以看到电机返回数据，返回数据信息格式如下：

---  
header:   
 seq: 1262  
 stamp:   
 secs: 1555999055  
 nsecs: 768794824  
 frame\_id: ''  
time\_stamp: 23422 #时间戳  
left\_encoder: 0 #  
right\_encoder: 0 #  
ischarging: True #是否正在充电  
battery\_percent: 41 #剩余电量百分比  
front\_echo: 4461 #前方超声  
rear\_echo: 1873 #后方超声  
front\_infrared: 2598 #前方红外  
rear\_infrared: 2382 #后方红外  
motor\_disabled: False #急停开关状态  
left\_motor\_current: 1.31226110458 #  
right\_motor\_current: 1.15603709221 #  
error\_state: 0 #错误状态码  
version: 52 #版本号  
---

输入命令:

$ rostopic echo /mobile\_base/sensors/extra

可以显imu数据和云台角度数据等。数据格式如下：

header:   
 seq: 83  
 stamp:   
 secs: 1556078520  
 nsecs: 464847185  
 frame\_id: ''  
yaw\_platform\_degree: 120 #水平云台角度  
pitch\_platform\_degree: 120 #竖直云台角度  
sound\_is\_mutex: True #喇叭静音  
acc\_x: 4.1225104332 #imu加速度  
acc\_y: 0.0 #imu加速度  
acc\_z: -1.64187479019 #imu加速度  
gyro\_x: 14.162528038 #imu角速度  
gyro\_y: -3.26288318634 #imu角速度  
gyro\_z: -69.6071166992 #imu角速度  
mag\_x: 723.599975586 #imu磁力计  
mag\_y: 1228.80004883 #imu磁力计  
mag\_z: 9830.25 #imu磁力计  
yaw: -3.7101585865 #imu\_偏航角  
pitch: -0.398658037186 #imu\_俯仰角  
roll: -179.962387085 #imu\_翻滚角  
q1: -0.00033110845834 #imu\_四元数（q1-q4）  
q2: 0.999469876289   
q3: -0.0323674604297  
q4: 0.00349123124033  
error\_state: 0 #错误代码  
time\_stamp: 29931 #时间戳  
version: 101 #版本号  
---

### 红外与超声传感器

键入rqt\_plot，在该界面中，上面的框中敲入/mobile则下面自动出现很多传感器备选项，末尾以echo结尾的，是超声传感器，以infrared结尾的，是红外测距传感器，我们加上这两种传感器，可查看对应屏幕显示的波纹变化。

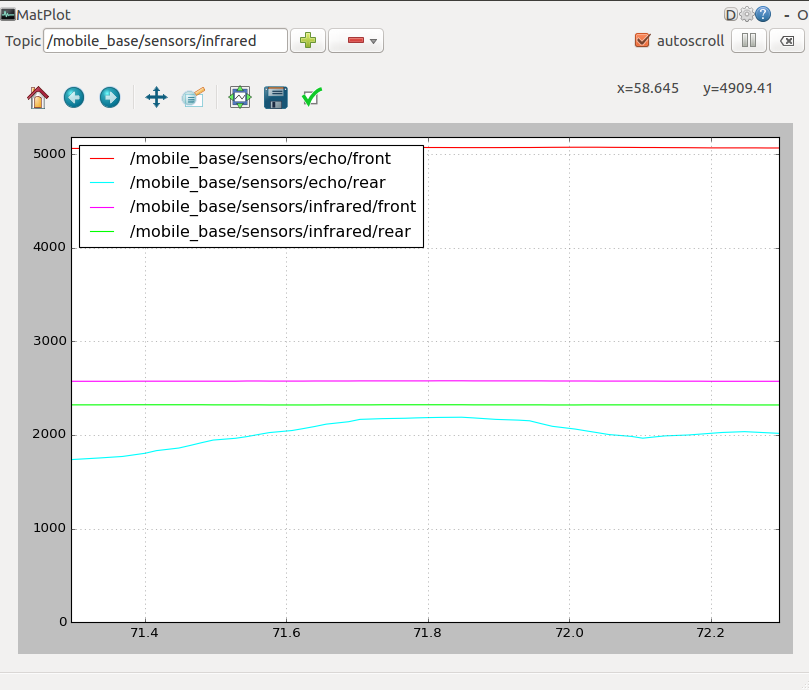


图 –3传感器返回图像

### 键盘控制检测

键入命令

$ rosrun xbot\_tools keyboard\_teleop.py

按照命令窗口提示的信息按键，可以控制机器各方向移动。

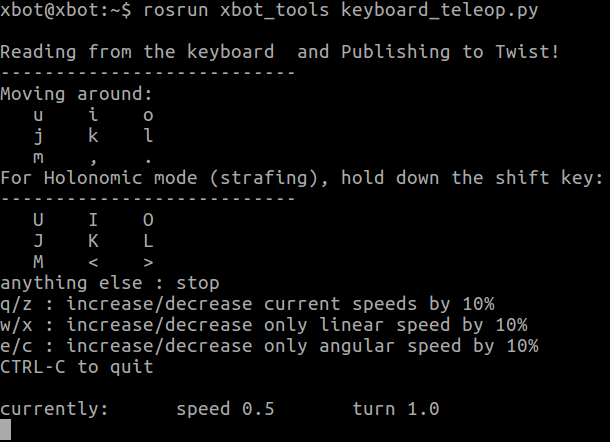


图 –4检查机器人移动

**注意：**当您控制机器人运动发现机器人无反应时，请检查两件事情：

1. 鼠标是否定位在上图所示的命令窗口。
2. 急停按钮状态。按下则无法运动是正常的，需要让急停按钮弹起再试。

### 云台控制

键入以下命令，修改data后的值，可以控制竖直云台俯仰指定角度，模拟抬头低头的动作。data的值域范围为[-60,30]。

$ rostopic pub -1 /mobile\_base/commands/pitch\_platform std\_msgs/Int8 “data: **0**”

键入以下命令，修改data后的值，可以控制水平云台左右旋转指定角度，模拟左右转头动作。data的值域范围为[-90,90]。

$ rostopic pub -1 /mobile\_base/commands/yaw\_platform std\_msgs/Int8 “data: **0**”

查看舵机偏转角是否运转正常。

### 摄像头数据检测

如何查看摄像头图像呢，键入命令：

rqt\_image\_view

选择color/image\_raw即可查看上面的摄像头显示情况，而选择camera/image则可以通过下部的摄像头查看情况：

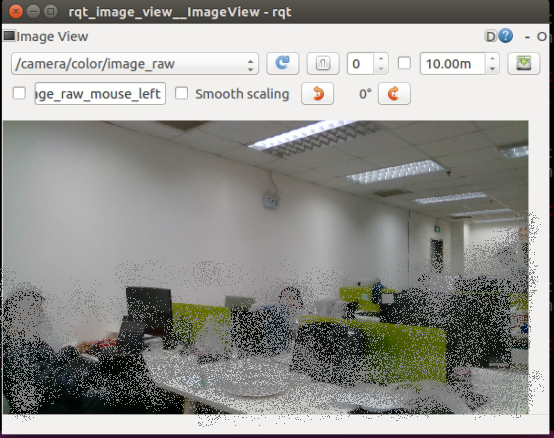


图 –5检查机器人摄像头运转

以上是人脸识别摄像头图像的查看,更换topic即可查看realsense摄像头的彩色摄像头图像.与摄像头相关topic如下：

realsense-rgb图像：/camera/color/image\_raw  
人脸识别摄像头：/xbot/camera/image

# 智能交互

机器人上配备有人脸识别、语音交互与平板交互等三项智能功能，下面将逐一为您介绍机器人这三项功能的使用。

## 人脸识别

人脸识别功能主要分为注册人脸，管理人脸和识别人脸三个部分。接下来将依次为您介绍它们的使用方法。

### 注册人脸

注册人脸分为人脸图片注册和实时拍照注册两种方式。

首先，将您的机器人连接显示器与键盘鼠标。因为机器人的开机自启动程序已经占用了人脸识别摄像头，所以要先在机器人的终端上输入如下指令终止全部服务：

$ sudo service xbot stop

释放人脸识别摄像头之后，继续输入下面指令启动ros：

$ roscore

接下来就是任意从下面的两种人脸注册方式中选择一种进行人脸注册。目前支持使用摄像头交互注册和使用已有的照片注册两种方式，命令分别如下：

#使用摄像头交互进行注册:

rosrun xbot\_face face\_register.py camera [camera\_index]

#使用照片进行注册:

rosrun xbot\_face face\_register.py image [user\_name]

1. 使用摄像头交互注册

使用摄像头交互注册，需要根据设备情况填写[camera\_index]参数，该参数表示设备的id。我们可以通过以下命令查询：

ls /dev

执行后会得到以下结果：

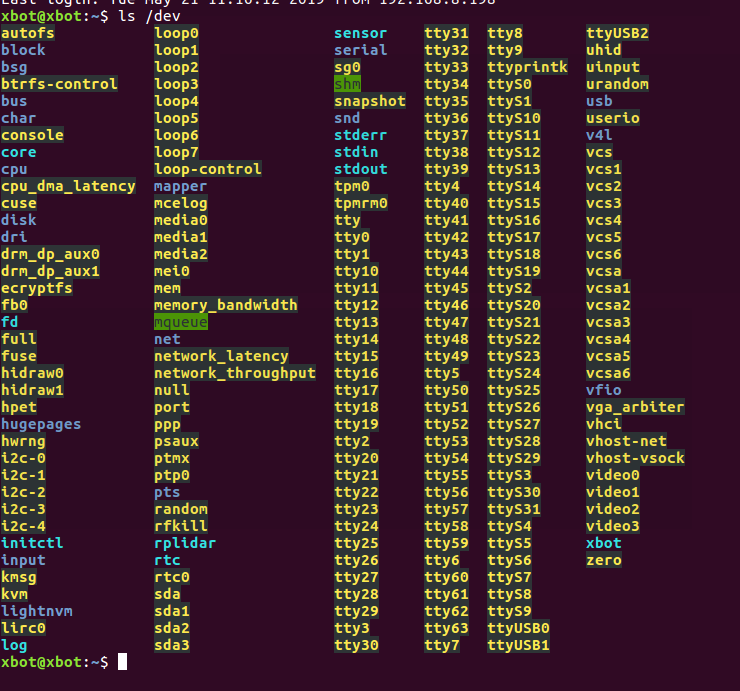


图 4–1设备查询结果

其中，video\*这些设备就是摄像头设备，在机器人上一般有4个摄像头。其中video3是人脸注册摄像头，我们一般用该摄像头进行人脸注册，传入摄像头id号3，则命令如下：

rosrun xbot\_face face\_register.py camera **3**

执行以上命令就会打开一个摄像头窗口，在窗口中能够清楚的看到图像信息。此时您需要用鼠标点击摄像头窗口（此时窗口就处于激活状态），然后将人脸置于窗口中间，调整人脸在窗口中的位置和清晰度，合适时，按下R键，此时摄像头窗口的画面将会定格。（友情提示，使用非人脸，会提示人脸数据提取失败，以下示意图为处理后，请勿模仿）



图 4–2摄像头窗口采集人脸照片

然后返回到终端，命令窗口会提示您输入姓名拼音，如"xjpcnew03"(注意要加双引号)并回车，如果下一行返回的是0，表示注册成功，其后只需要鼠标点击摄像头窗口并按Q或者直接在终端键入ctrl+c即可退出人脸识别程序，此时就完成了一次人脸注册。

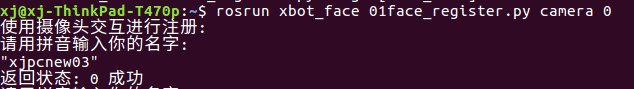


图 4–3 摄像头交互注册成功

注册的信息可以通过在浏览器中直接输入接口http://192.168.8.141:8000/management/userids返回，查看返回结果是否包含刚刚注册的id：xjpcnew03。

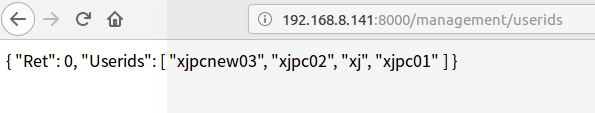


图 4–4 人脸注册用户查询

1. 使用照片进行注册

使用已有的照片进行人脸注册，需要设备上已经有了待注册用户的照片，此时我们只需要键入以下命令，并输入待命名的用户名（如xjimage01）：

rosrun xbot\_face face\_register.py image xjimage01

回车执行命令之后，会弹出一个文件选择窗口，提示选择人脸照片文件：

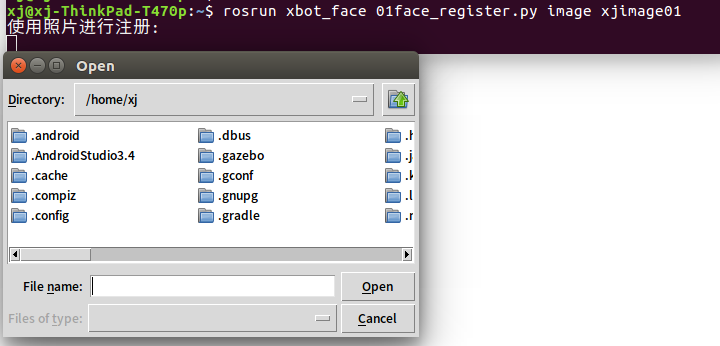


图 4–5 照片注册-选择照片

根据人脸照片路径，选择照片即可完成注册。注册成功，返回0.

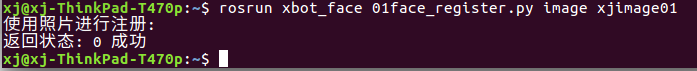


图 4–6 照片注册-成功提示

其它常见的人脸注册返回值如下表定义：

|  |  |
| --- | --- |
| Ret（Int） | 备注 |
| 0 | 正常 |
| 1 | 请求超时 |
| 2 | 识别/核身解析结果错误 |
| 3 | 识别解析结果错误 |
| 4 | 识别解析相似度错误 |
| 5 | 输入信息错误 |
| 6 | 无该Userid对应信息 |
| 7 | 获取注册人脸图片数据错误 |
| 8 | Base64解码错误 |
| 9 | 人脸数据提取失败（使用非人脸进行注册时会提示该信息） |
| 10 | 删除人脸失败 |
| 11 | 图片过大，不超过初始配置大小（1280,720） |
| 12 | 文件不存在 |
| 13 | 人脸图片打开失败 |
| 14 | 人脸已存在（注册人脸时method=normal情况下，userid重复） |
| 15 | 未检测到网卡 |
| 16 | 输入信息不合法 |
| 17 | 一键开门失败 |
| 18 | 文件读取失败 |
| 2019 | 恢复出厂设置失败 |
| 2020 | 数据清除失败 |
| 2021 | 获取日志列表失败 |
| 22 | MAC地址不匹配 |

### 识别人脸

注册完人脸之后，如果摄像头画面中出现有注册过的人脸，机器人就能够识别出相应的信息。

注意，如果您之前停止了机器人的服务，现在需要使用<2.5.1 启动常用功能>中的一键启动脚本将它们重新启动起来。一键启动指令为：

$ roslaunch xbot\_bringup xbot-u.launch

我们在机器人或者从机上运行下面的指令，即可查看机器人当前识别的人脸结果。

$ rostopic echo /xbot/face\_result

执行以上命令之后，会一直不停的输出以下格式的信息：

$ face\_exist: False

$ name: "UNKNOWN"

$ confidence: 0.0

当无人脸置于机器人人脸识别摄像头视阈范围，那么则会输出：

$ face\_exist: False

$ name: "UNKNOWN"

$ confidence: 0.0

当机器人的人脸识别摄像头捕获到人脸信息时，会进行识别，并输出识别结果、匹配的注册用户名以及把握度信息。

$ face\_exist: True

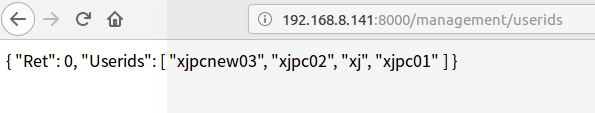
$ name: "xijing"

$ confidence: 0.632823

### 人脸注销

1. 注销指定用户

注销指定用户，建议在注销之前，通过http://192.168.8.141:8000/management/userids接口查询已经注册的用户id。下图中的Userids后面的数组记录的即为注册用户的id。



假设删除xjnew03用户，那么执行命令，返回为0则表示注销成功：

rosrun xbot\_face face\_logout.py xjnew03



图 4–7 注销用户成功

再次访问http://192.168.8.141:8000/management/userids接口查询已经注册的用户，已经没有xjnew03，该用户已经成功被注销。

1. 注销所有用户

注销所有用户则只需要输入参数all：

rosrun xbot\_face face\_logout.py all

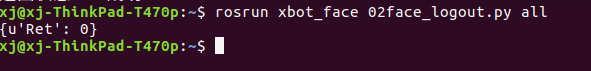


图 4–8 注销所有用户成功

## 语音交互

机器人中已经配置好了语音交互模块，使用该模块只需在机器人或者从机上调用相应的service，并修改相应的请求参数，即可打开内置的语音交互模块。

由于机器人出厂前已经进行了语法构建，所以xbot\_talker目录下已经存在cache文件目录，用户无需进行构建语法操作，可直接进行4.2.1的测试。若没有cache文件，则需要在终端输入

roslaunch xbot\_talker build\_grammar.launch

进行语法构建，cache文件夹里包含语法生成的文件以及各种识别日志等缓存文件和调试信息，可定期删除此文件，此文件删除后，需要重新运行上述命令，用户修改命令词后，也需要重新运行此命令。

### 播放指定的语音文件

您还可以让机器人播放语音文件，mode:1 表示播放模式为将文字转化为语音具体的指令为：

$ rosservice call /xbot/play "loop: false

$ mode: 1

$ audio\_path: '~/catkin\_ws/xbot\_talker/assets/wav/welcom.wav'

$ tts\_text: ''"

其中audio\_path后面是您想要播放的语音文件的路径，可自行修改。

### 将输入的文字转化为语音

您可以让机器人播放您输入的文字，只需要在机器人终端中输入如下指令启动/play服务即可，mode:2 表示播放模式为将文字转化为语音：

$ rosservice call /xbot/play "loop: false

$ mode: 2

$ audio\_path: ''

$ tts\_text: '请和我打招呼'"

其中您可以将‘请和我打招呼’替换为任何您想要播放的文字。语音播放成功后，终端会输出相应的提示：

$ success: True

在输入指令时，您可以使用Tab键来自动补全指令，以避免指令输入错误。补全指令后您只需修改相应参数即可。

### 对话和语音控制

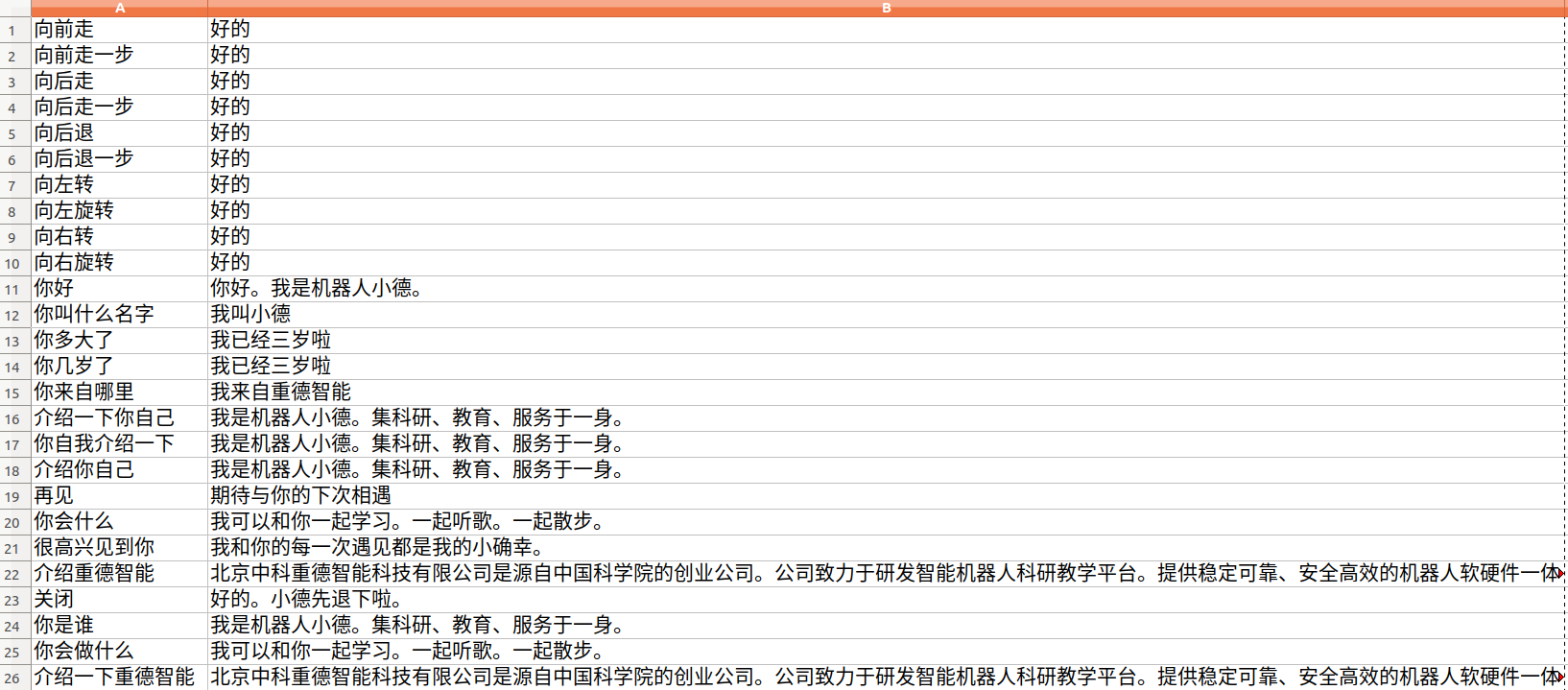
/chat服务提供了与机器人进行对话以及通过语音控制机器人的交互功能。运行以下命令，即可进行交互：

$ rosservice call /xbot/chat "start\_chat: true"

执行该指令后，机器人会在一段时间内发出“嘟——嘟——”声，您在听到声音之后即可以开始与机器人交谈。

结束对话时需要告诉机器人“关闭”。在机器人接收到语音输入的关闭指令后，会结束对话。

初始配置的机器人已经能够回答一些简单的问题，包括：你好、你多大了、你是谁、你叫什么名字、你会什么、介绍一下你自己等等，如果要设置更多问答以及更丰富的交互场景，您可以按照4.2.4小结的教程进行自定义对话内容。

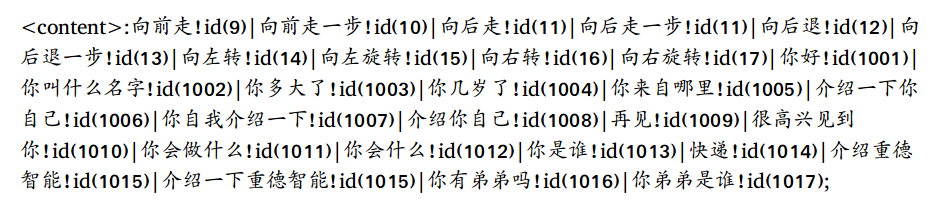
初始配置的部分对话内容如下，用户可输入含有关键词的语音，与机器人进行交互。例如，如果问机器人“你几岁了？”，由机器人就会根据已定义好的回答，回应“我已经三岁啦”；命令机器人“向前走”，机器人就会回答“好的”并向前移动一小段距离。

### 自定义对话内容

用户可以通过修改以下两个配置文件自定义对话内容。

1. 修改xbot\_talker/userconfig/grammar.bnf语法文件。

bnf语法使用一种结构描述了用户可能说出的语言范围和构成模式。简单来说，机器人只能够识别该文件中定义的关键词。该文件可用WPS打开并编辑。打开文件后，用户可直接在的最后添加自定义的关键词，例如，添加关键词“你有弟弟吗”“你弟弟是谁”：



关键词之间用“| ”分隔开，“!id”后面小括号里的数字，依次往后加1即可，最后用分号“；”结束。

1. 修改xbot\_talker/defaultconfig/answer\_dic.csv文件，添加新的交互词典。

打开answer\_dic.csv表格文件，在文件末尾添加自定义的交互信息。

每一行代表一组问答，第一栏代表“问答的关键词”，必须与grammar.bnf里的定义一致，否则无法进行有效识别。第二栏代表机器人回答的语句，语句最后以句号结尾，若语句分为好几句，中间也应尽量以句号作为停顿，避免使用英文逗号作为标点。例如：

第三栏数字代表机器人要执行的动作，比如前进，后退，向左旋转，向右旋转等。目前机器人可进行的动作以及数字编码都可参照answer\_dic.csv前面的设定进行设定。例如0代表机器人不执行任何动作，1代表关闭对话，3代表向前走一步，4代表向后退一步……

第四栏数字代表动作模式，目前支持的数字模式：0表示在执行动作前播放语音和3只播放语音不执行任何动作。

修改完以上两个配置文件之后，需要先进性语法构建：需要在终端输入：

roslaunch xbot\_talker build\_grammar.launch

接着重新启动xbot-u.launch后，便可按照4.2.3的内容进行测试。

## 使用XBot助手控制您的机器人

目前我们提供了Android应用”UXbot助手APP“来与XBot进行交互。 UXbot助手是面向用户的操作终端，方便用户实时掌控XBot状态以及对XBot进行交互操作。用户可以从UXbot助手上了解当前XBot-U机器人的电量，摄像头俯仰角度，摄像头平台的旋转角度。用户也可以通过UXBot助手来调节XBot的摄像头俯仰角度、摄像头平台旋转角度。用户可以通过界面上的摇杆控件，来控制XBot-U机器人进行移动，还可以实时查看XBot上的摄像头拍摄到的图像。

### UXbot助手APP的安装

请登录重德智能官网，下载app安装。目前app仅支持andriod系统。

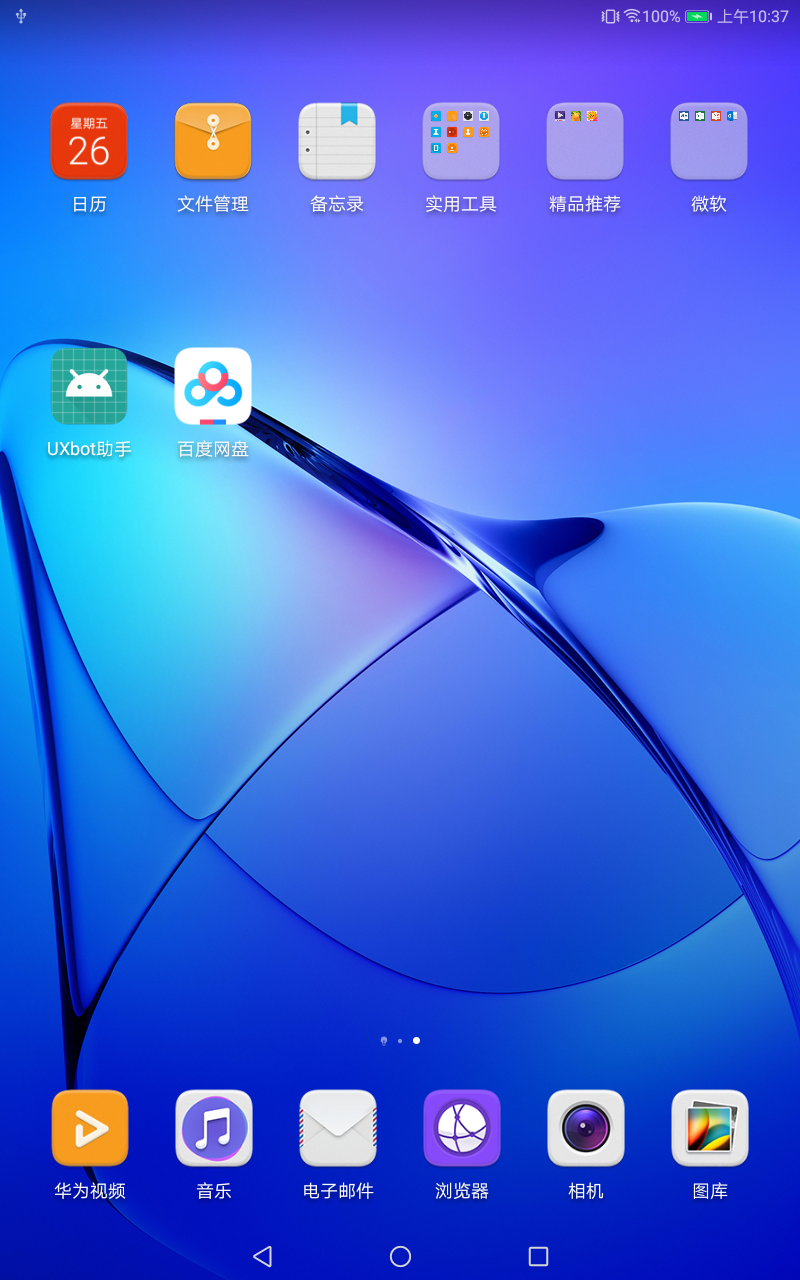
APP安装时，请注意：

1. 在手机进行APP安全检测时，请选择信任该APP，勾选“我已充分了解该风险，继续安装”，且选择“继续安装”。
2. 在安装过程中，请选择“信任此应用”给APP赋权限。请注意一定要给“调用摄像头”的权限，否则人脸模块功能将不可用。



图 –9 安装APP

按照提示，完成安装即可。“UXBot助手APP”的图标如下图：



### UXbot助手APP的使用准备

由于UXbot助手APP是XBot-U机器人的工具APP。在使用之前，我们需要如下操作：

1. 首选要开启XBot-U机器人电源开关，等待XBot-U机器人完成启动（大约30秒左右）。启动后，我们可以观察激光雷达传感器开始不停的旋转，电源指示灯也亮起。
2. 拿出手机或者平板，在WLAN网络中，找到“xbot\_network-bj\*”类似命名的无线网络，并连接。\*这里指代的是机器人的出厂编号，该编号可以从机器人底盘的二维码上查看得到。

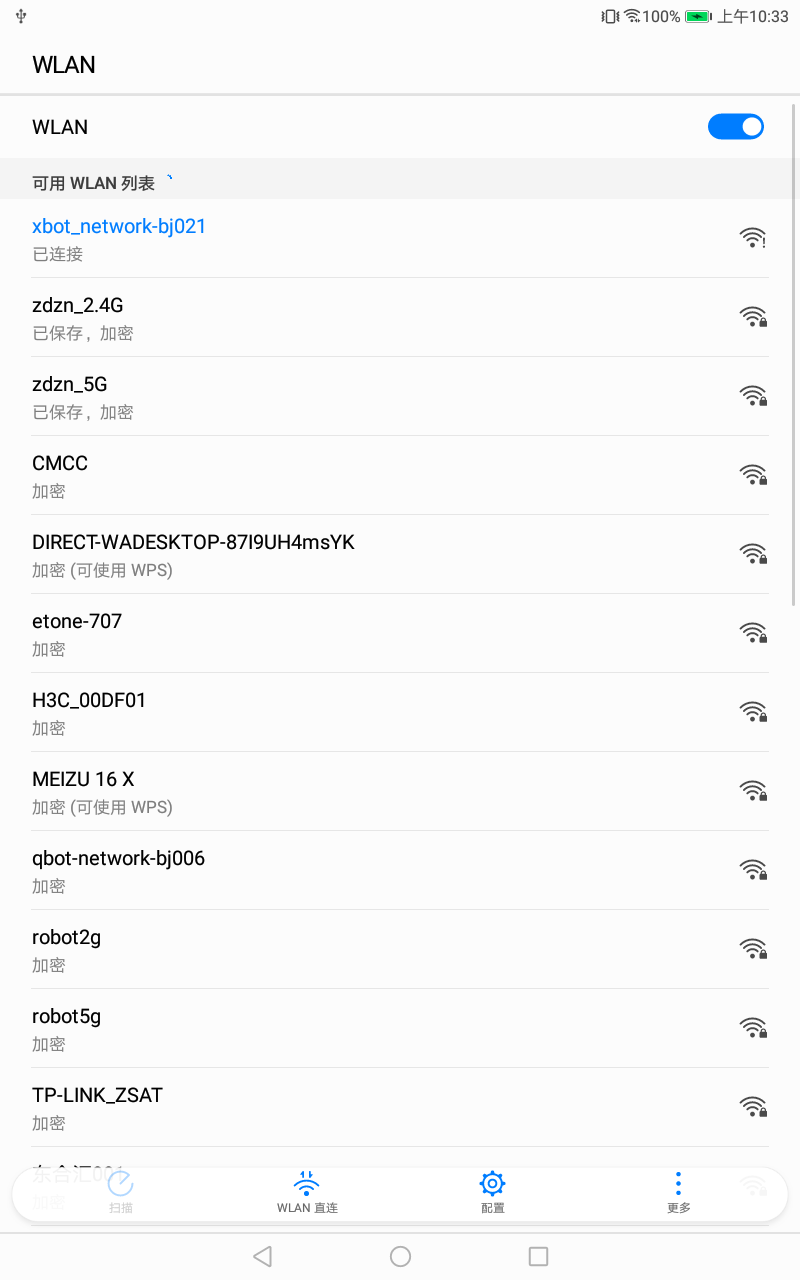


图 –10wifi连接

1. 连接网络之后，就可以启动UXbot助手APP了。
2. 启动之后，APP弹出“ROS服务器连接成功”，则说明APP已经成功与机器人连接，下面就可以开始使用“UXBot助手APP”来控制机器人了。

当app显示“ros服务器未连接”提示时，首先请检查机器人是否正常启动，然后检查手机或者平板是否连接了机器人WLAN网络，**再将当前的app进程在后台清理掉，最后重启app。**此时一般就能够显示“ROS服务器连接成功”提示信息了。

此时，就可以正常使用app来控制机器人了。

### 设置

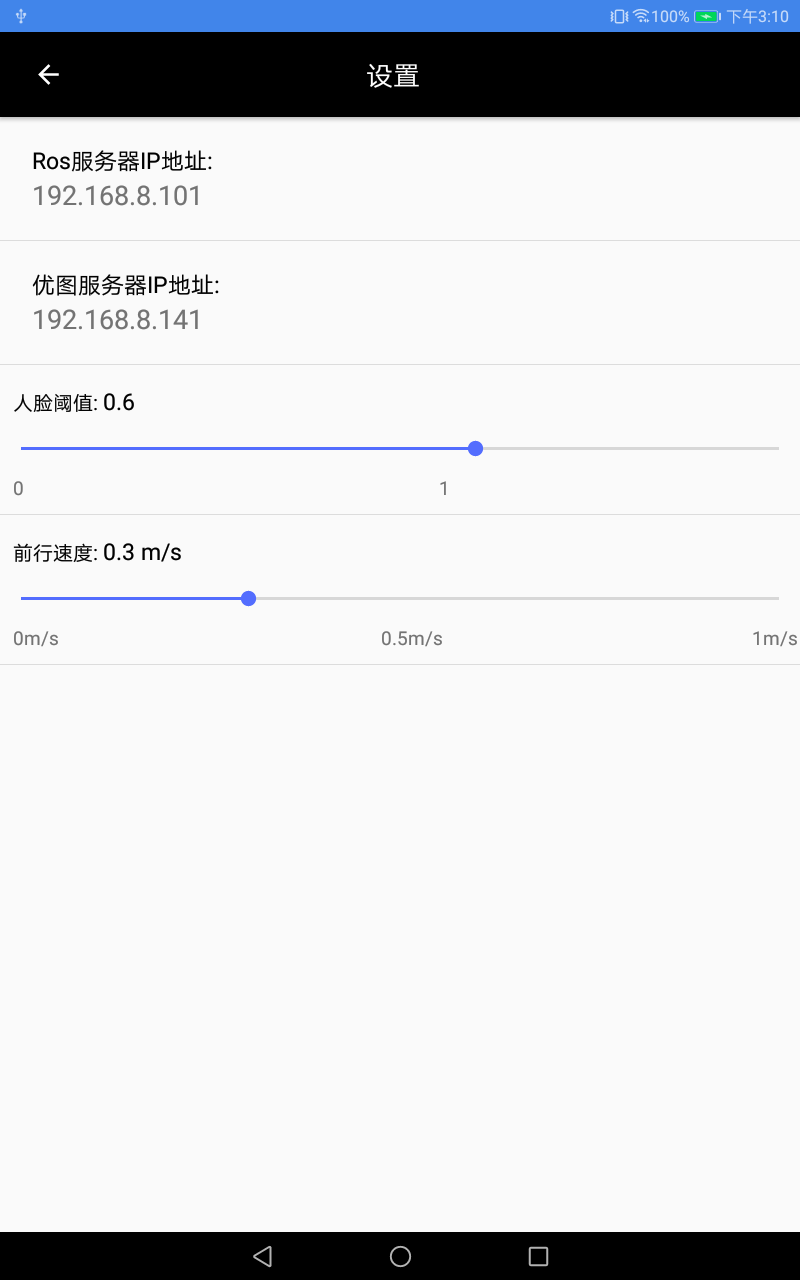


图 –11设置

#### ROS服务器IP地址

Ros服务器IP地址默认是192.168.8.101.一般不需要修改，除非当XBot-U机器人主机的IP地址修改，否则不需要修改。

#### 优图服务器IP地址

优图服务器（有时也被称为人脸识别盒子）IP地址默认是192.168.8.141.一般不需要修改，除非当XBot-U机器人中内置的人脸识别盒子中的IP地址修改，否则不需要修改。

#### 设置人脸识别阈值

设置人脸识别时，当返回值confidence的返回值高于人脸识别的阈值设置时，我们认定为识别成功。

#### 设置遥控基准速度

“前进速度”即设置机器人运动时的速度。配合“方向控制轮盘”来控制机器人的运动。

### XBot-U机器人状态查看与控制

当UXBot助手连接上ros服务器后，可以在"状态"界面实时查看到Xbot的状态，包括电量，摄像头俯仰角度，摄像头平台旋转角度(即云台角度)等。用户可以通过关闭或打开电机电源，来操作电机电源的开关。当电机电源处于开启状态时，可以在“控制”界面使用摇杆控件来操作机器人运动。当电机电源关闭时，不能使用摇杆控件来操作Xbot进行移动。用户还可以手动调节云台角度和摄像头角度。

点击app底部的“状态”标签，可以查看XBot-U机器人状态。

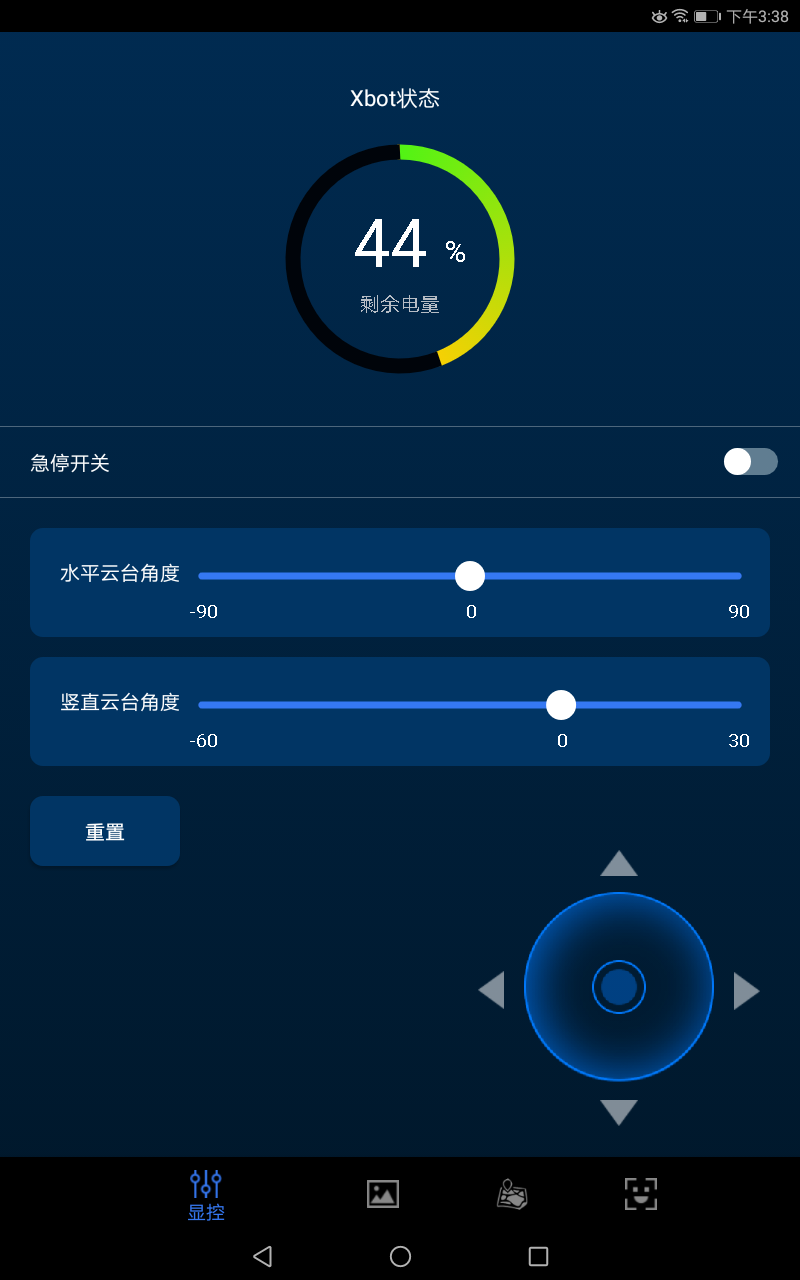


图 –12显控面板

#### 电池电量显示

“剩余电量”显示的即为机器人剩余的电池电量状态。当电量小于25%时，请注意需要连接电源充电。

#### 水平云台垂直云台位置显示与控制

“水平云台角度”：显示的即为当前水平云台的角度信息。用手拖动滑块可以改变角度，机器人会随之调整水平云台角度，实现类似人类“转脖子”的动作。水平云台左右旋转的角度是90度。

“垂直云台角度”：显示的即为当前垂直云台的角度信息。用手拖动滑块可以改变角度，机器人会随之调整垂直云台角度，实现类似人类“低头抬头”的动作。垂直云台向下可以低头60度，向上仰头30度。

#### 重置xbot云台状态

“重置”按钮，可以将“水平云台角度”和“垂直云台角度”重置为0度。

#### 方向控制轮盘

图1中的圆形区域即为“方向控制轮盘”，用来控制机器人移动时的方向。

* 0度方向：右转
* 90度方向：前进
* 180度方向：左转
* 270度方向：后退

说明：“方向控制轮盘”只控制方向，运动的速度需要在“设置”中进行设置。

#### 急停开关控制

图1中，“急停开关”开启时，相当禁用了机器人运动功能，此时操作“方向控制轮盘”，机器人不动。

“急停开关”关闭时，操作“方向控制轮盘”，机器人按照指示的方向运动。

### XBot-U机器人摄像头图像查看

#### “人脸识别摄像头”图像查看



图 –13人脸摄像头图像回传

#### “深度摄像头”RGB-彩色图像查看

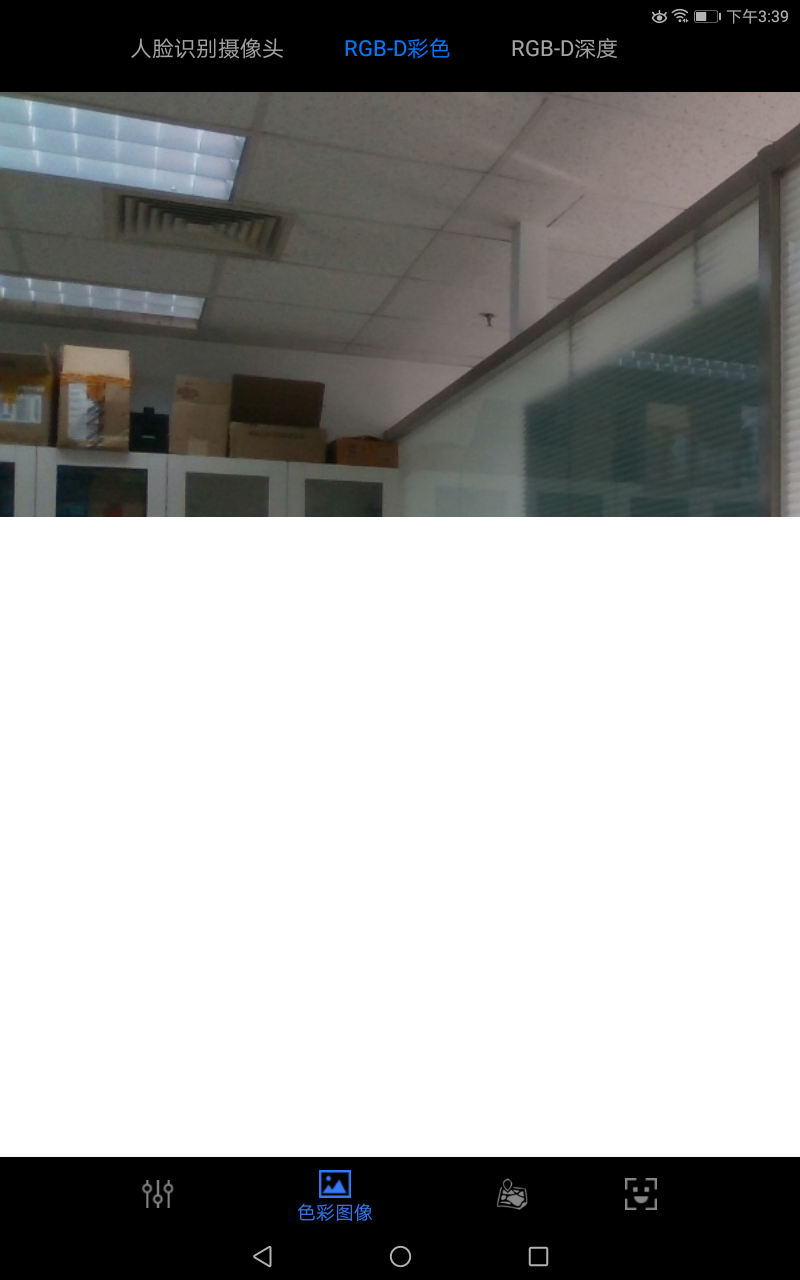


图 –14RGB摄像头图像回传

#### “深度摄像头”深度图像查看

待实现……

### 人脸管理与识别

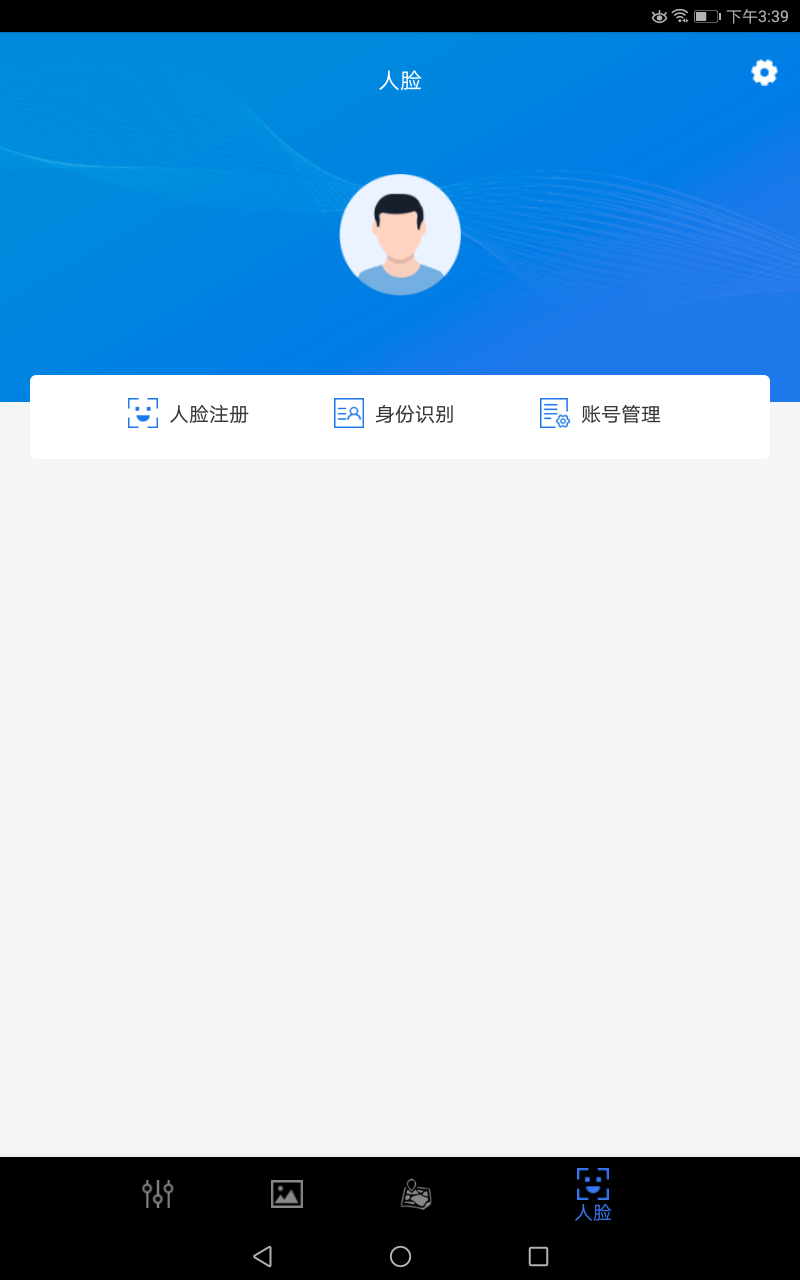


图 –15人脸功能

#### 人脸注册

当需要给机器人录入新的注册用户是，就可以按照以下操作步骤即可完成人脸注册：

人脸--->人脸注册--->输入用户ID--->采集人脸图像--->确认注册。

注册完成之后，才能在身份认证中被识别出来。

#### 身份识别

身份识别在机器人应用中一般结合应用场景，与语音识别、运动导航等融合使用。APP中只提供基本的检测功能，即识别出人脸图像所对应的注册用户的ID。该功能的使用很简单，只需要按照如下步骤操作即可：

人脸--->身份识别--->将人脸置于采集图框--->识别成功提示且输出用户ID。



图 –16人脸识别功能

#### 账号管理

账号管理用于对已经注册的用户进行查看、编辑、删除等管理操作。app通过如下列表的方式展示机器人上已经注册过的用户信息。我们还可以通过APP添加注册用户，编辑注册用户信息，删除注册用户等操作。



图 –17人脸注册用户查询

# 机器人SLAM与自主导航

我们在最初的XBot的软硬件设计上，都早已充分考虑到了XBot对于目前绝大部分建图算法和开源程序的支持特性。

在硬件上，XBot具备标准的双轮差分系统，前后各配置一个万向轮，适应大多数室内的运行环境；激光雷达方面，XBot可以与市面上大部分的激光雷达兼容运行，支持所有使用二维激光雷达建图的程序。

在软件上，XBot以成为一款最适合中国ROS学者和研究人员使用的机器人软硬件平台为目标，从最原始的驱动软件编写逻辑和优化方法，到其node Package的设置，都按照ROS控制标准好系统标准而完成，以方便您在本平台上测试调试任何在ROS算法程序。

XBot-U机器人具有运动规划功能，可以根据已知地图或运动中建立的地图自主生成运动轨迹。当您打开RViz可视化窗口后，可以直接在窗口上指定目标点，让机器人进行运动规划，并导航至该点。

**在运行导航程序之前，请在您的个人计算机上搭建ROS主从环境，具体操作请参考7.2章节。**

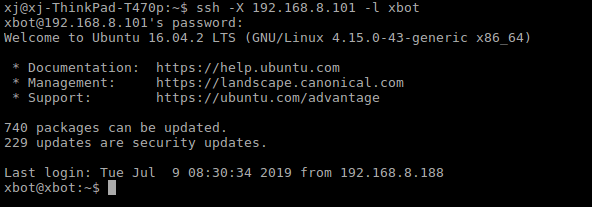
## 启动建图程序

以下操作均在从机端进行。

1. 连接到XBot-U机器人自带的WiFi，连接方法与前面主从配置部分相同。
2. 把机器人摆放至合适的位置准备建图 ( 根据任务需求以及场地实际定制)
3. 以机器人为主机，在从机端远程SSH登录到主机上

$ ssh xbot@192.168.8.101

4.按照提示输入机器人密码，当@后的主机名变成xbot时表示当前已经连上机器人主机，后续在xbot@xbot 下执行的所有命令都可以认为等同于在xbot的命令终端执行的。



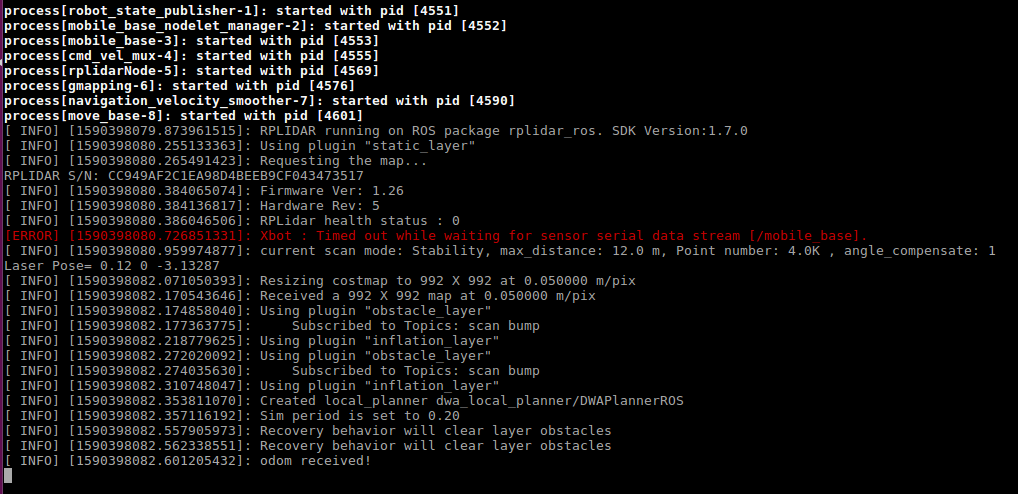
1. 给机器人定义一个原点，即每次建图和导航时，机器人初始放置的位置和方向，记下来。
2. 【从机端】打开终端，运行xbot\_navi/launch目录下的build\_map.launch程序。

$ roslaunch xbot\_navi build\_map.launch #使用gmapping建图

需要注意的是，XBot-U机器人开机时，已经启动了激光雷达，但是在build\_map.launch文件中，仍然写有启动雷达的语句，故此可能出现执行上述指令导致雷达停转的情况，此时您可以直接观察激光雷达是否运转，如果停止运转，请按下**ctrl+c终止之前的进程后，再次执行roslaunch xbot\_navi build\_map.launch 命令一次，观察机器人的激光雷达运转起来则表示建图程序启动成功 。**

说明：

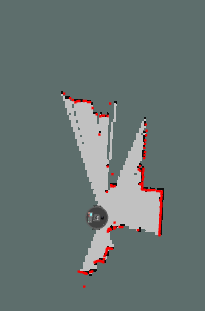
在运行过程中会出现下图红色所示的错误提示，一两条Timed out while waiting for serial date stream不会影响机器人的运行，这是因为在数据通信时，偶尔有一两条数据传输不完整，解析过程中会丢弃这一条数据。



1. 在【从机端】新开一个终端，在从机端运行rviz查看远端机器人建立的地图信息：（这次可以不用ssh连接到机器人上）

$ roslaunch xbot\_navi rviz\_build\_map.launch

此时可以看到机器人实体和周围的环境的图像。



## 移动机器人完成建图

在RViz中，中间区域显示的是XBot-U机器人建立的地图信息，选择上方MoveCamera指令可以调整地图视角，其中，按住鼠标左键拖动可旋转地图。滚动鼠标滚轮可以放大和缩小地图，按住鼠标滚轮拖动可以调整地图位置。

在建图过程中，我们需要控制机器人移动，使机器人搭载的激光雷达扫描整个环境，从而建立地图信息。控制机器人移动的方式很多，可以新开终端输入键盘命令去驱动机器人运动，也可以使用UXbot助手APP的运动轮盘去控制，甚至可以人工推着机器人运动(此时必须是机器人双轮着地运动，不可搬运离地)。目的就是为了让激光雷达能够扫描到整个场地。  
下面介绍通过键盘命令控制机器人移动。

在从机上打开一个新的终端（之前的build\_map和rviz\_build\_map命令窗口都不要关闭），输入如下指令：

$ rosrun xbot\_tools keyboard\_teleop.py

**当焦点在该终端上时，**通过键盘上的按键，可以手动控制机器人的移动，按键与机器人运动对照关系如下表：

|  |  |
| --- | --- |
| 按键 | 运动 |
| i | 向前 |
| k | 停止 |
| , | 向后 |
| j | 顺时针旋转 |
| l | 逆时针 |

需要注意的是，建图需要机器人移动，因此接口不适合接入有线设备（如显示器、有线鼠标键盘等），因此在通过键盘命令驱动机器人运动时，建议采用ssh远程控制或者在PC从机上控制的方式。且**在手动控制机器人移动时，机器人不会自主避障，如果将要发生碰撞，可以通过机器人地盘上的红色按钮紧急制动。**

注意：一次建图不成功可以反复进行多次建图。

## 保存地图

1. 建图完成后不要关闭build\_map.launch，通过ssh远程主机的方式，终端运行保存地图的程序。首先，cd到想要保存地图的目录，然后通过下面的指令保存地图：

$ ssh xbot@192.168.8.101

$ roscd xbot\_navi/map #为了能把地图保存到xbot\_navi/map目录下

$ rosrun map\_server map\_saver -f <filename>

举例：rosrun map\_server map\_saver -f 807

其中，<filename>是地图的名称(没有 <> ),为方便说明，下面使用 "807" 替代。

1. 至此，会在该目录下得到两个文件,分别为 807.pgm 和 807.yaml，即为保存好的地图。
2. 建议在建图过程中多次运行以上指令保存不同阶段的地图来选用最合适的一张,切记保存地图时的文件名不能相同,否则会自动替换文件。
3. 注意，此处的地图是保存在运行以上指令的机器和目录当中的。

地图保存完成后，建图过程中打开的窗口可以关闭。

## 配置导航地图参数

机器人的导航功能需要知道当前的空间环境，空间环境是通过之前创建的地图来描述的。所以，首先我们需要修改导航程序中的地图参数。

在主机上，打开catkin\_ws/src/xbot/xbot\_navi/launch/下的demo.launch文件进行编辑。

具体操作为，在从机上执行：

ssh [xbot@192.168.8.101](mailto:xbot@192.168.8.101)

roscd xbot\_navi/launch

vim demo.launch

修改文档中的map\_file参数，将上述红色字体地方修改为刚刚保存的地图名，并保存。

<arg name="map\_file" default="$(find xbot\_navi)/map/**mapname**.yaml" />

## 导航

我们通过一个最简单直观的方式，来实际运行下导航功能。

1. 将机器人放置到之前建图时定义的原点位置，保持与启动建图时一致的位置和方向。
2. 配置好导航地图参数之后，我们检查之前打开的所有终端窗口（或者结束各运行的命令，尤其是建图命令）是否关闭，如果没有关闭，建议关闭所有终端窗口。
3. 接着，新开终端，通过远程主机，启动导航功能：

ssh [xbot@192.168.8.101](mailto:xbot@192.168.8.101)

roslaunch xbot\_navi demo.launch

注意：在导航期间，该命令窗口一直保持运行状态，不要关闭和结束运行。

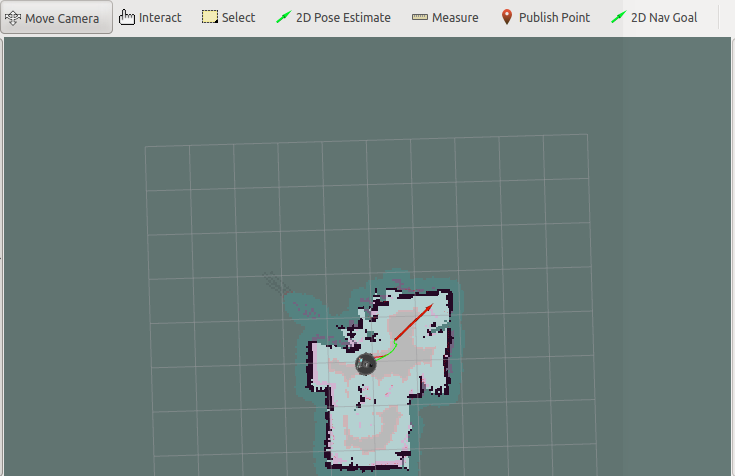
1. 在从机的命令终端，直接运行导航展示程序：

roslaunch xbot\_navi rviz\_demo.launch

此时，从机上会出现如下的界面：



1. 在上图所示的rviz界面中，点击按钮，待鼠标状态改变后，在地图上标记原点的位姿（位置和方向）。
2. 接着，在工具栏点击按钮，在地图可行的区域（粉灰色区域）设置导航目标点的位置。



1. 设置完目标点之后，稍等几秒，rviz中地图上会出现如上图所示的绿色路线，这是导航程序规划的全局路径。稍等片刻，机器人会按照规划的路径自主导航到设定的目标点，完成导航。

## 进一步的应用开发

除了以上基础功能之外，为了给使用者和开发者一个综合所有基础功能的演示应用的demo，我们在xbot\_navi包当中提供了一个demo.launch的启发程序，其中结合了包括机器人基础运动、人脸识别、语音对话交互、机器人SLAM、机器人自主导航等功能在内的几乎所有基础功能。用户可以自行查看该启发程序的源代码，程序逻辑和功能实现，并在此基础上逐步开发实现自己想要的机器人场景应用。

# XBot仿真与教学

我们为ROS初学者和XBot-U机器人的使用者提供了一套仿真环境，可以实现模拟器上对机器人的操作。

**注意，仿真环境与之前配置的从机环境不相同。**

首先下载、安装依赖和编译ROS-Academy-for-Beginners

$ cd ~/catkin\_ws/src

$ git clone <https://github.com/DroidAITech/ROS-Academy-for-Beginners.git>

$ cd ~/catkin\_ws

$ rosdep install --from-paths src --ignore-src --rosdistro=kinetic -y #安装依赖

$ catkin\_make #编译

在运行模拟器前，请确认你的gazebo在7.0版本以上，通过以下命令查看

$ gazebo -v

如果**版本低于7.0**，请升级gazebo

$ sudo sh -c 'echo "deb http://packages.osrfoundation.org/gazebo/ubuntu-stable `lsb\_release -cs` main" > /etc/apt/sources.list.d/gazebo-stable.list'

$ wget http://packages.osrfoundation.org/gazebo.key -O - | sudo apt-key add –

$ sudo apt-get update

$ sudo apt install gazebo7

另外建议建议在本地Ubuntu下运行仿真程序。虚拟机对Gazebo的兼容性存在问题，可能会有错误或卡顿。

## 启动XBot模拟器

输入以下命令启动XBot模拟器：

$ roslaunch robot\_sim\_demo robot\_spawn.launch

随后Gazebo会启动，如果是第一次启动，可能需要等待几分钟，等待gazebo从服务器上下载模型。

图片包含 天空, 户外

已生成极高可信度的说明

图6–1 Gazebo上软件博物馆模型

Gazebo正常启动，你就能看到XBot-U机器人模型和软件博物馆的场景了。

启动键盘控制程序，你就可以控制机器人前后左右移动了

$ rosrun robot\_sim\_demo robot\_keyboard\_teleop.py

## SLAM仿真

在XBot的SLAM仿真中提供了Gmapping、Karto、Hector等常见SLAM算法的demo，本节以启动和可视化Gmapping为例，介绍运行方法。

首先确保已经启动XBot模拟器，然后输入

$ roslaunch slam\_sim\_demo gmapping\_demo.launch

启动gmappign后，启动RViz查看建图效果

$ roslaunch slam\_sim\_demo view\_slam.launch

你可以再启动键盘控制程序，移动机器人，就能够看到地图逐渐建立的过程。

$ rosrun robot\_sim\_demo robot\_keyboard\_teleop.py

运行效果如下图所示：

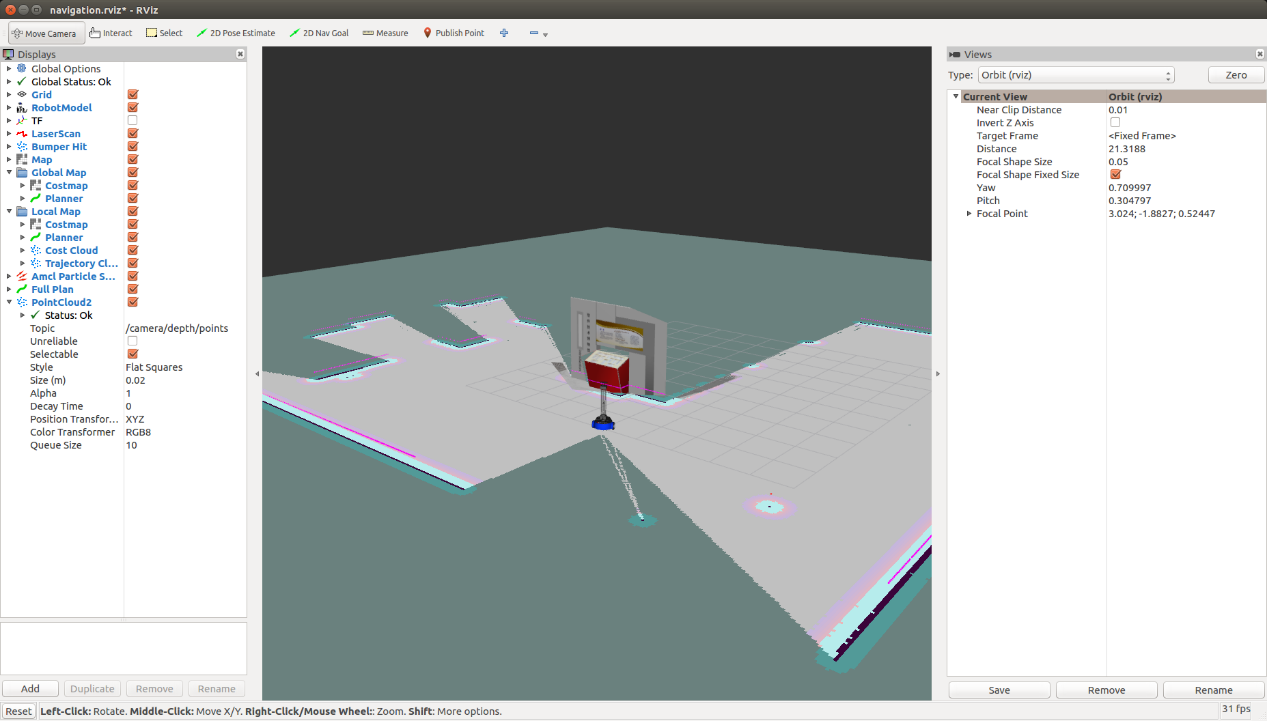


图6–2 RViz建图界面

在slam\_sim\_demo中还有其他SLAM算法，启动方法和Gmapping相同。

## 已知地图与导航仿真

许多情况下，我们已经建立好了场景地图，只需要机器人执行定位和导航的任务，我们提供了AMCL（定位）、Map\_server（已知地图）、和导航相结合的仿真环境。

首先启动XBot模拟器，然后输入启动amcl与导航仿真

$ roslaunch navigation\_sim\_demo amcl\_demo.launch

启动RVIz，查看定位效果

$ roslaunch navigation\_sim\_demo view\_navigation.launch

如图所示，绿色箭头表示粒子，点击RViz上方工具栏中的2D Nav Goal，然后在地图上确认目标点位置和方向，机器人就会执行导航任务。

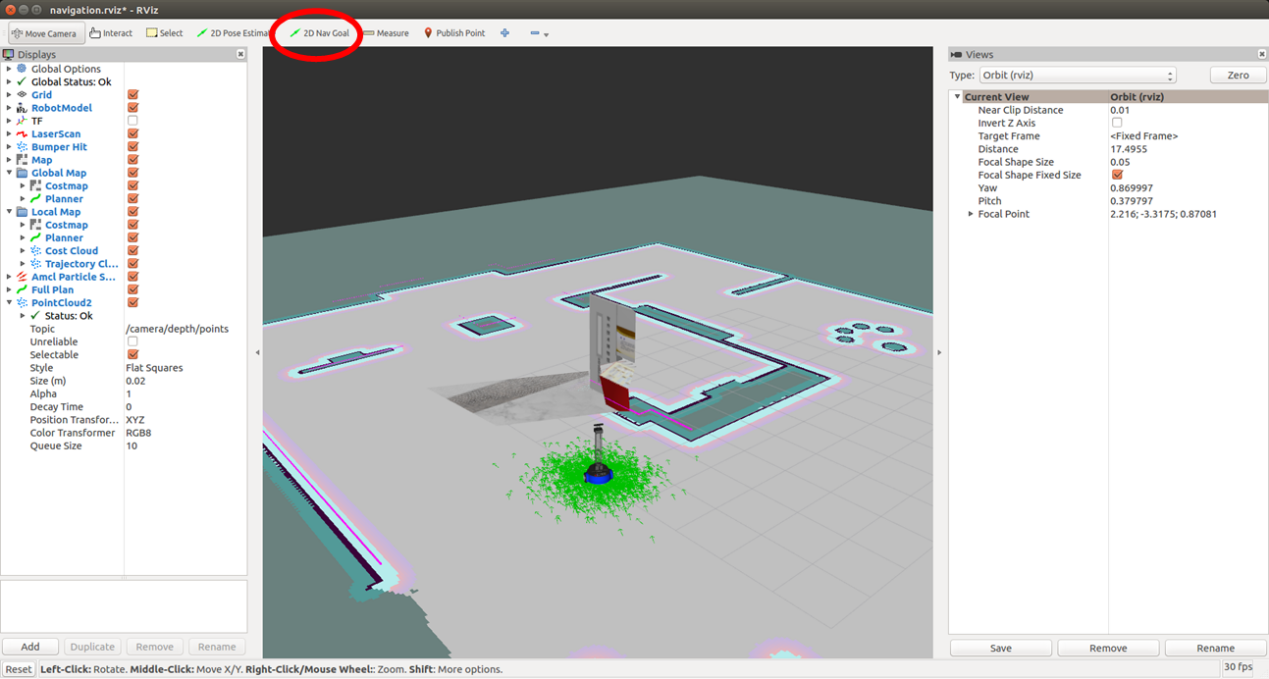
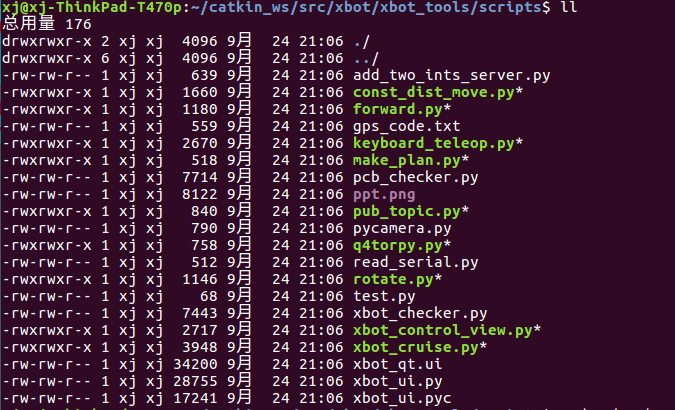


图6–3 输入目标点控制机器人移动

# 常见问题

## 执行权限

1. 在通过rosrun运行某个py脚本时，会出现没有任何反应的情况，此时首先进入py文件所在的目录，通过ll命令查看文件的权限。没有执行权限的，请给py文件设置可执行权限。



具体的操作命令为：sudo chmod +x \*.py

不了解linux文件权限，以及不会查询、设置命令的可以自行搜索了解。

## 配置您的个人计算机作为从机

机器人主机本身没有配备显示器，虽然我们也可以通过UXbot app对机器人进行简单控制和状态查看，或者使用自带的HDMI接口外接显示器，但是当我们想进一步对xbot进行了解、研究、或者进行建图、运动规划等高级功能时，这两种方法都有各自的不便之处，因此，我们推荐您通过配置个人计算机（PC端）来实现与机器人的无线通信。配置后的个人计算机与XBot-U机器人构成主从机关系（XBot-U为主机，PC端为从机），同时从机可以SSH远程登录到XBot上，这样，您就可以使用您的个人计算机实现对机器人的控制。

对个人计算机环境的配置主要包括：

1. 安装Ubuntu操作系统
2. 安装ROS环境
3. 配置ROS环境
4. 从机连接到XBot-U主机
5. 部署xbot功能包

### 安装Ubuntu操作系统

Ubuntu是当前ROS运行的标准平台，建议您在笔记本或带无线网卡的台式机上安装Ubuntu系统。

**注意：XBot-U机器人主机上已安装好Ubuntu，您只需要在PC上安装。**



图7–1 Ubuntu系统桌面

每一个ROS版本都存在对应的Ubuntu版本，XBot支持Kinetic、Indigo等主流ROS版本，我们建议您选择Ubuntu16.04与ROS Kinetic。

|  |  |
| --- | --- |
| Ubuntu版本与ROS版本对照表 | |
| ROS版本 | 首选Ubuntu版本 |
| Lunar | Ubuntu 17.04 |
| **Kinetic**(推荐) | **Ubuntu 16.04**(推荐) |
| Jade | Ubuntu 15.04 |
| Indigo | Ubuntu 14.04 |
| Hydro | Ubuntu 13.04 |

清华大学开源镜像网站中Ubuntu16.04下载链接如下：

<https://mirrors.tuna.tsinghua.edu.cn/ubuntu-releases/xenial/ubuntu-16.04.6-desktop-amd64.iso>

### 安装ROS

安装好Ubuntu系统后，我们可以开始准备安装ROS环境。

① 在Ubuntu系统上，确认初始环境配置正确。

打开Ubuntu的设置->软件与更新->Ubuntu软件->勾选关键字universe, restricted, multiverse三项。



图7–2 软件和更新

“下载自”一栏可以选择速度较快的源，推荐使用国内高校的镜像。

②添加sources.list

打开终端，输入

$ sudo sh -c '. /etc/lsb-release && echo "deb http://mirrors.ustc.edu.cn/ros/ubuntu/ $DISTRIB\_CODENAME main" > /etc/apt/sources.list.d/ros-latest.list'

我们默认添加中国科大的ROS镜像。

③添加keys

打开终端，输入

$ sudo apt-key adv --keyserver hkp://ha.pool.sks-keyservers.net:80 --recv-key 421C365BD9FF1F717815A3895523BAEEB01FA116

④ 系统更新

$ sudo apt-get update && sudo apt-get upgrade

⑤安装ROS

$ sudo apt-get install ros-kinetic-desktop-full # Ubuntu 16.04

请耐心等待直至系统安装配置结束。

### ROS环境配置

① 初始化rosdep

$ sudo rosdep init && rosdep update

② ROS环境配置

$ echo "source /opt/ros/kinetic/setup.bash" >> ~/.bashrc

③安装rosinstall

$ sudo apt-get install python-rosinstall

### ROS安装测试

首先启动ROS，输入代码运行roscore：

$ roscore

如果出现图9所示结果，则说明ROS已经正常安装和启动。

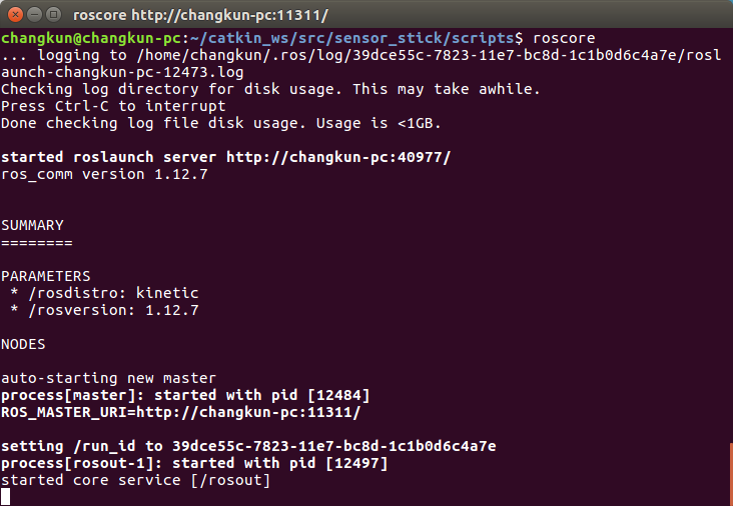


图7–3 ROS安装正常

### 从机连接到xbot主机

您的个人计算机通常可以通过“SSH连接”和“ROS主从机配置”两种方式控制XBot-U机器人，这两种方法各有用途。

“SSH连接”是linux系统的功能，其安装和使用就不一一详述。

“ROS主从机配置”即为ros计算机分布式主从通信，是需要安装ros环境之后才能配置。

两者在命令的层级上就不一样，ssh是linux的底层通信，ros主从机通信是ros主机和从机通过订阅话题和服务实现的，其使用的前提是主机的ros必须启动。

关于两者使用的区别，用户可以自己去尝试和查询资料。关于其他的ROS网络的配置教程，，请参考ROS官方wiki社区：<http://wiki.ros.org/ROS/NetworkSetup>

不管何种方式连接机器人，操作机器人的第一步就是开机。启动机器人后，机器人会启动随身WIFI。请在PC上链接WIFI热点**：xbot-network-\*，**其中后面的\*是机器人的网络唯一编号(编号在机器人底盘上二维码处可见)，初始密码**：xbot1234**。

1. 通过SSH远程连接到机器人

SSH是一种较为常用的远程登录方式，基础格式为：ssh <主机用户>@<主机IP地址>

想通过个人PC远程控制主机，那么我们需要在个人PC上打开一个终端，并输入：

$ ssh xbot@192.168.8.101

@符号之前是主机的用户名，@符号之后是主机的IP地址。按照机器人出厂设置，用户名默认为xbot，主机IP地址为192.168.8.101

接着，终端会提示输入密码：

xbot@192.168.8.101's password:

此时，需要输入xbot的主机密码即可。这里密码输入后不会有任何显示，输入完成直接按回车键即可。

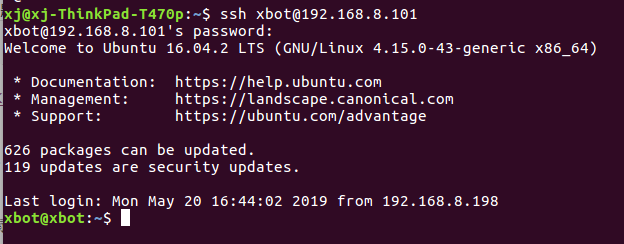


图 7–4 ssh成功登录

ssh远程登录成功后，可以看到如上图所示的信息。尤其最为关键的是看最下面一行的绿色字体部分(表示“用户@机器名”)，已经由个人PC电脑的用户名和机器名（xj@xj-Thinkpad-T470p）变成主机的用户名和机器名(xbot@xbot)，这是最为重要的ssh登录成功的验证手段。

以上方式为最简单的ssh登录，每次新打开一个终端，都需要输入该命令和密码。为了简化操作，您还可以通过配置ssh的公钥私钥来设置免密码登录。有兴趣可自行了解。

此外，在连接机器人的过程中，我们有时会需要打开图形界面，此时我们可以通过-X（X大写）打开图形界面：

$ ssh -X [xbot@192.168.8.101](mailto:xbot@192.168.8.101)

个人PC通过ssh连接机器人主机成功之后，就可以像在主机上一样，键入任何命令来操作机器人了。

1. ROS主从环境配置

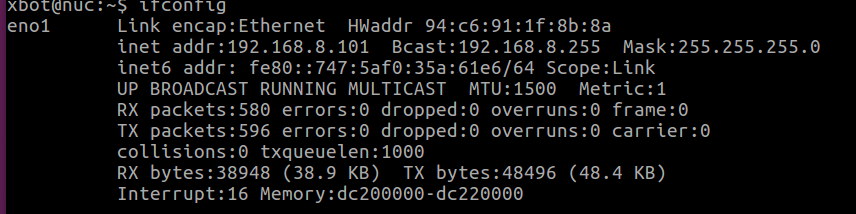
ROS的主从环境其实是一种分布式通信方式。这里我们讲述ROS主从环境的配置步骤：

1. 查看IP

启动机器人后，机器人会启动随身WIFI，在PC上链接机器人WIFI。

首先，使用下面的命令，分别在XBot与您PC上执行，获取xbot主机和您PC的IP地址：

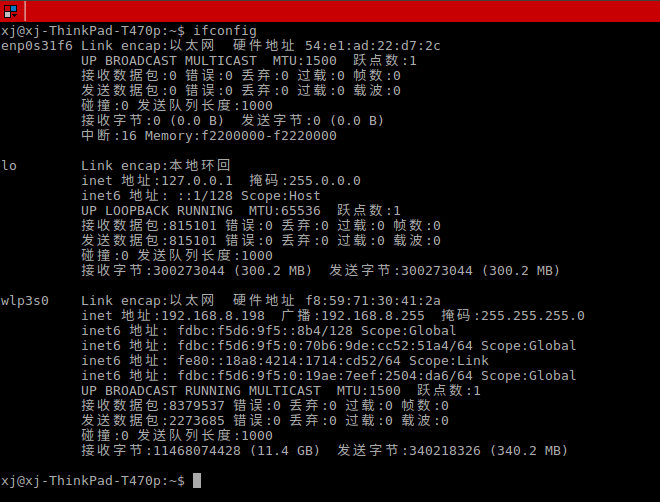
$ ifconfig

[](https://yt.droid.ac.cn/xbot-u/hq/uploads/8b8c1a3c5ac1ff926dd932d23851692c/1.png)图 7–5 通过ifconfig指令查看IP地址

XBot端显示如图 24所示（XBot-U机器人主机的默认IP已在出厂时设定为192.168.8.101）。

类似的，您的PC端也会显示出一个IP地址，应注意，在查看PC端IP时，请务必保证您的PC已经连接上了XBot-U机器人的wifi，同时，为避免以后每次都要重新查看ip,可先配置固定IP。

然后使用ifconfig命令进行查看PC机的IP地址：



2.PC端与XBot端通信测试

在开始配置主从关系之前，先测试一下两台机器之间是否能够正常通信。在这里可以使用ping指令，首先测试从XBot端到PC端的连通。在XBot的终端里输入：

$ ping yourpc-ip

可以看到：

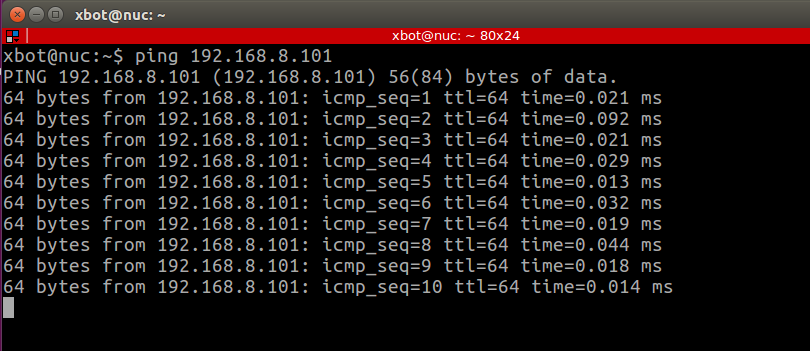


图7–6 通行正常1

从PC端到XBot端测试时，在PC端输入。

$ ping 192.168.8.101

连接成功的效果如下图：

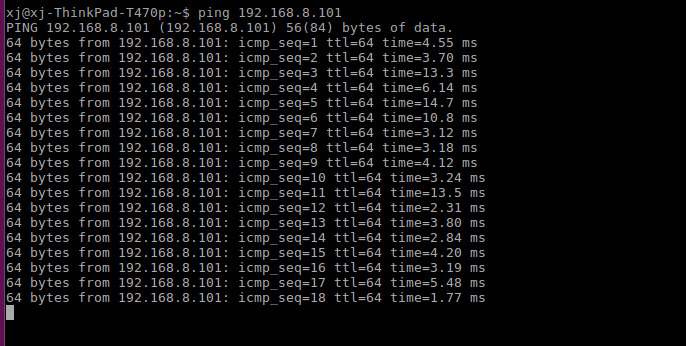


图7–7 通行正常2

双向ping通后，可以开始配置主从关系。

4.配置主从关系

在这里，我们设置Xbot为主机，PC端为从机。

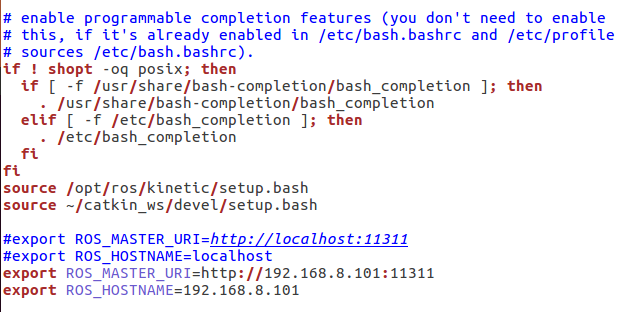
首先在XBot的终端上输入：

$ gedit ~/.bashrc

该指令打开.bashrc文件，如果文件中存在之前的配置，我们先将其注释掉，然后加入：

export ROS\_MASTER\_URI=http://192.168.8.101:11311

export ROS\_HOSTNAME=192.168.8.101



保存退出，并在终端上输入：

$ source ~/.bashrc

XBot上配置完毕。

在PC端上，我们也使用同样命令打开.bashrc文件：

$ gedit ~/.bashrc

如果存在之前的配置，则先将其注释掉。，再加入：

export ROS\_MASTER\_URI=http://192.168.8.101:11311 #ROS主机(Xbot)IP

export ROS\_HOSTNAME=192.168.8.198 #PC的IP,即本机的ip地址

保存退出后，在终端输入：

$ source ~/.bashrc

PC端配置完毕。

此时，主从关系已经全部配置完毕。我们可在XBot端启动ROS节点，然后再PC端上使用rostopic list查看相关话题，如果能够看到终端返回相关话题，则说明配置成功。返回的topic（部分）见下图：

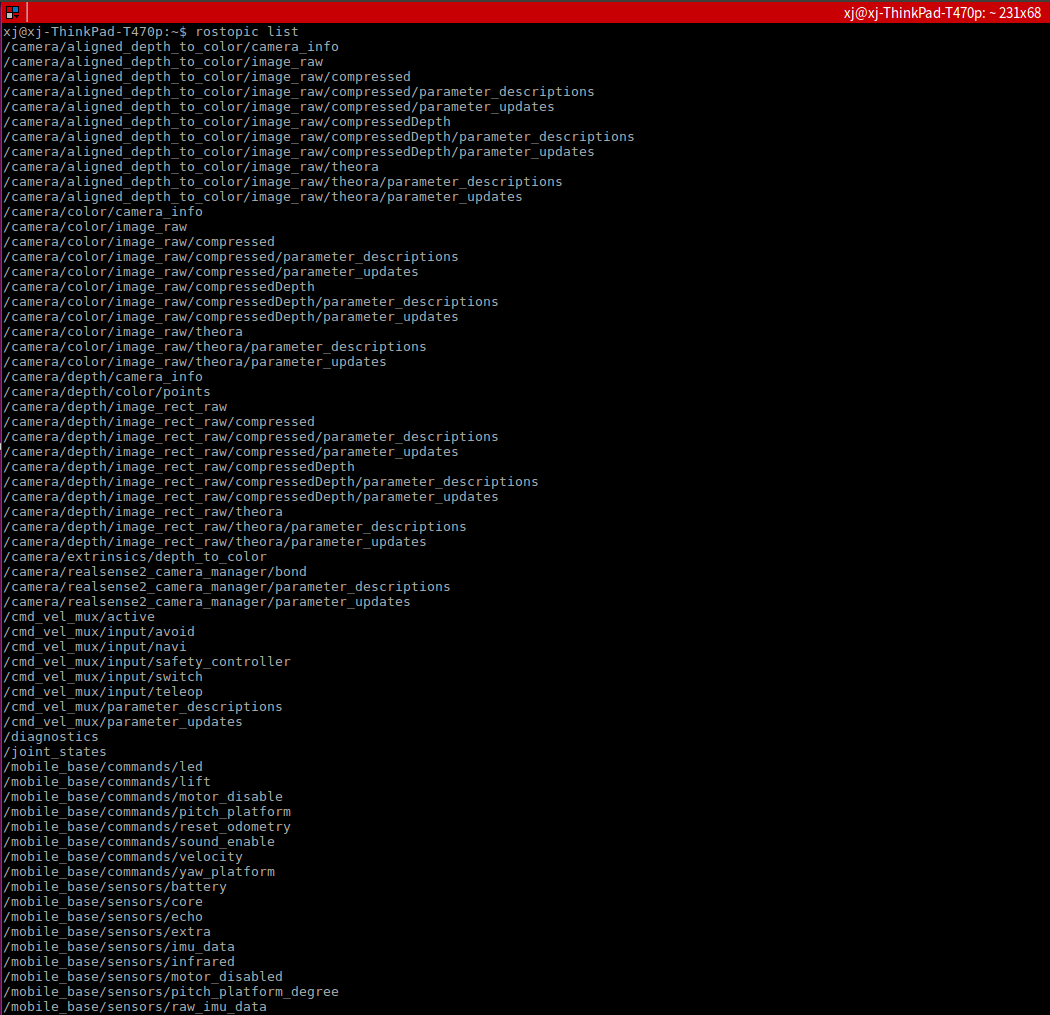


图7–8 从XBot-U上返回的Topic

### 在从机上部署xbot功能包

为了能够在从机上控制机器人，我们需要在您的个人计算机上部署与XBot-U机器人相同的工作空间。

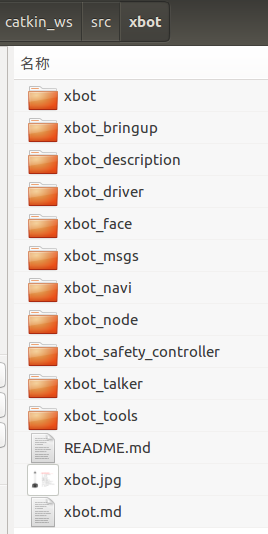
在PC上打开终端，新建工作空间caktin\_ws并对它初始化

$ mkdir -p catkin\_ws/src

$ cd catkin\_ws

$ catkin\_make #初始化工作空间

初始化完成后，将xbot机器人上的/home/xbot/catkin\_ws/src路径下的xbot功能包全部拷贝到catkin\_ws/src下。



功能包包括：

| **ROS软件包** | **主要功能** |
| --- | --- |
| xbot\_bringup | 驱动程序启动入口，包含多个launch文件 |
| xbot\_driver | 底层通讯驱动，收发数据 |
| xbot\_node | ROS数据封装与分发，运动解算 |
| xbot\_msgs | 软件包集所用到的所有ROS消息类型 |
| xbot\_safety\_controller | 机器人安全控制程序 |
| xbot\_description | ROS建模与urdf描述文件 |
| xbot\_tools | 用于调试、开发以及运行过程的一些工具软件 |
| xbot\_face | 人脸识别程序 |
| xbot\_navi | SLAM、导航规划程序 |
| xbot\_talker | 机器人语音交互和对话程序 |

为了防止由于文件权限导致的运行错误，强烈建议您进行py文件的可执行权限验证：

$ cd ~/catkin\_ws/src

$ find -name \*.py |xargs chmod +x

最后编译工作空间。

$ cd ~/catkin\_ws

$ catkin\_make

$ source devel/setup.bash

**注意**：source命令，编译完成后必须刷新一下工作空间的环境，否则可能找不到工作空间。许多时候我们为了打开终端就能够运行工作空间中编译好的ROS程序，我们习惯把source ~/tutorial\_ws/devel/setup.bash命令追加到~/.bashrc文件中，这样每次打开终端，系统就会刷新工作空间环境。你可以通过echo "source ~/tutorial\_ws/devel/setup.bash" >> ~/.bashrc命令来追加。

有时候如果我们需要单独编译某个功能包，我们也可以使用指令：

$ catkin\_make --pkg package\_name

其中，package\_name是要单独编译的功能包的名字。当您直接使用catkin\_make指令不能通过编译时，可以使用该指令来排查不能通过编译的功能包。

在xbot系列包中，当直接catkin\_make出现问题的时候，您可以按照以下顺序分开编译：

$ catkin\_make --pkg xbot\_msgs

$ catkin\_make --pkg xbot\_talker

$ catkin\_make --pkg xbot\_face

$ catkin\_make --pkg xbot\_navi

$ catkin\_make

$ source devel/setup.bash

### 在主机上安装sshd

为了远程操控主机，建议主机上安装并开通ssh server。默认机器人上已经安装，如果从机ssh远程登录机器人主机被拒绝，可以按照以下方法进行检查和安装。

1. 检查主机是否安装sshd

$ which sshd

如果安装，命令会返回sshd的安装路径，一般为/usr/sbin/sshd；

如果没有安装，则无返回值。

1. 如果没有安装，则先安装。

$ sudo apt-get install openssh-server

1. 安装完成之后，检查ssh服务是否已启动

$ ps aux | grep ssh

输出结果出现sshd，则说明安装成功且进程已经开启

1. 查看防火墙是否已关闭

$ sudo ufw status

IMG_257

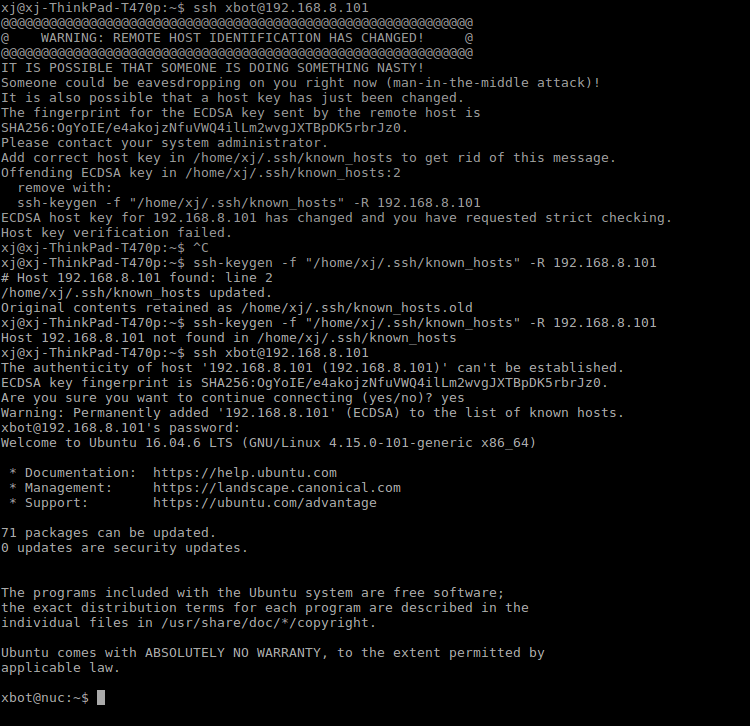
要求关闭防火墙，如果防火墙没有关闭，请先关闭防火墙。具体操作可自行搜索。

1. 验证ssh是否成功。

将从机连机器人网络xbot\_network-\*,并在从机的终端输入：

$ ssh xbot@192.168.8.101

如果能返回要求输入密码的界面，表示能正常连接。此处可能遇到许可等问题，属于linux基础知识，请自行搜索解决。

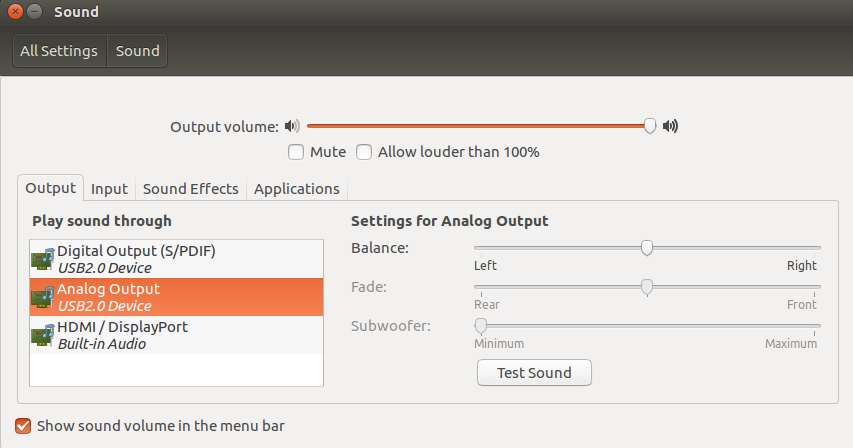


看到最后出现xbot@nuc打头的字符时，表示现在从机已经连上了主机。

## 调节机器人的输出音量

如果需要调节机器人喇叭音量，请通过外界显示器，点击ubuntu系统屏幕右上角喇叭图标，选择Sound Settings。

选择Output标签页，调节Output volume调节输出音量；点击Test Sound可以看到左喇叭和右喇叭分别测试的Test按钮，检测左右喇叭是否正常发音，方向是否准确，音量是否洪亮。



# 联系我们

## 参考链接

* 公司官网：[重德智能](https://www.droid.ac.cn/)
* 产品介绍：[XBot-U机器人网站介绍](https://www.droid.ac.cn/xbot_u.html)
* 配套自研课程：[中国大学慕课-机器人操作系统入门](https://www.icourse163.org/course/0802ISCAS001-1002580008)
* 公司ROS wiki：[ROSwiki xbot tutorials](http://wiki.ros.org/Robots/Xbot/tutorial/cn)
* 公司B站：<https://space.bilibili.com/443622831> **（含自制ROS课程）**

## 联系我们

如果您对我们的产品、使用手册文档有任何疑问，欢迎通过以下方式联系我们：

**服务邮箱：**[**service@droid.ac.cn**](mailto:service@droid.ac.cn)（注明XBot-U售后咨询）

公司ROS论坛(简称q站)：<https://q.droid.ac.cn/>

公司Github：<https://github.com/DroidAITech>

产品咨询：17610230188 (周一至周五9:00-18:00)

地址：北京市海淀区中关村南四街四号 中国科学院软件园区

更多内容请关注公司官网。