GaiTech
www.gaitech.net

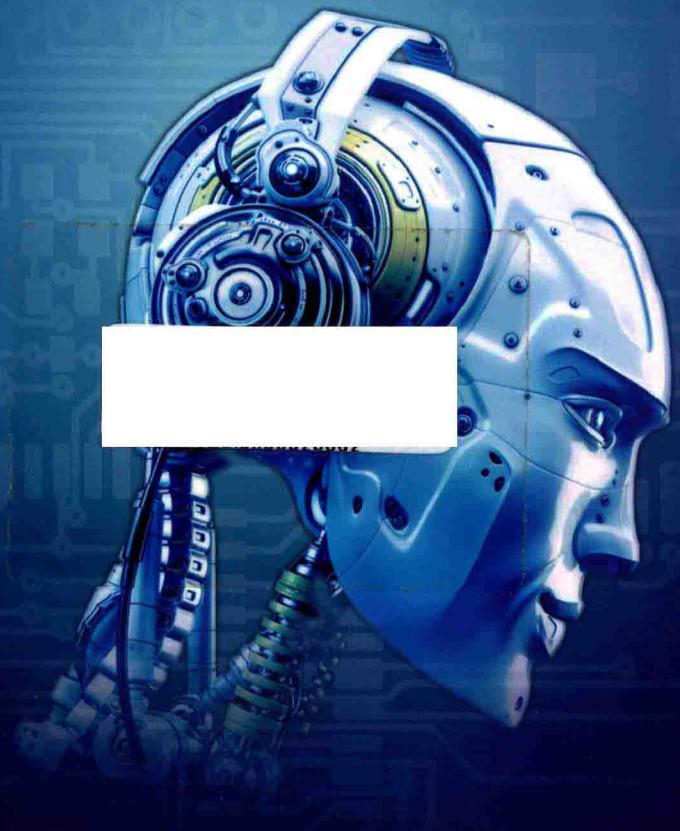
上海硅步科学仪器
有限公司支持出版

ROS By Example

A Do-It-Yourself Guide to the Robot Operating System

ROS入门实例

[美] R. 帕特里克·戈贝尔◎著
[墨] J. 罗哈斯　刘柯汕　彭也益　刘振东　李家能　黄玲玲◎译



中山大學出版社
SUN YAT-SEN UNIVERSITY PRESS

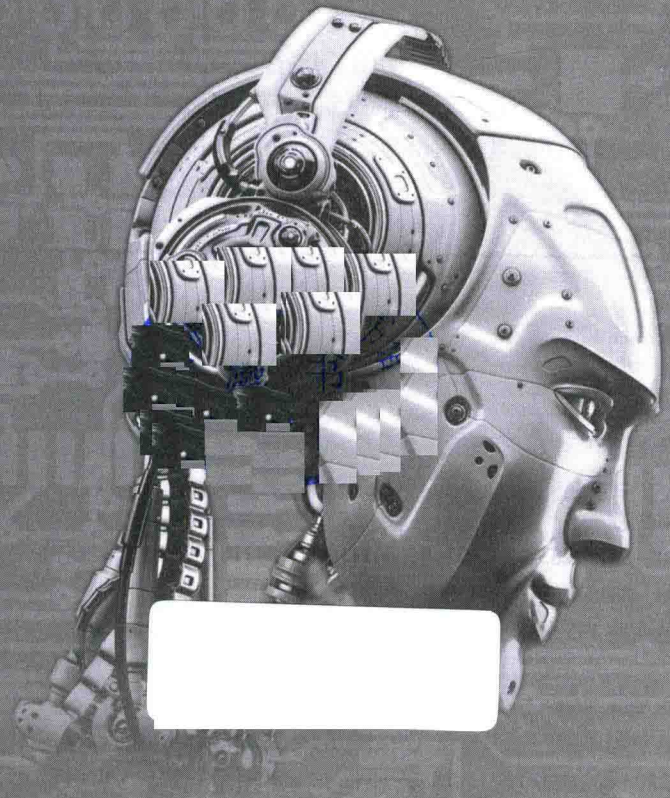
GaiTech
www.gaitech.net

上海硅步科学仪器
有限公司支持出版

ROS By Example
A Do-It-Yourself Guide to the Robot Operating System

ROS入门实例

[美] R. 帕特里克·戈贝尔◎著
[墨] J. 罗哈斯　刘柯汕　彭也益　刘振东　李家能　黄玲玲◎译



中山大學出版社
SUN YAT-SEN UNIVERSITY PRESS
·广州·

图书在版编目（CIP）数据

ROS 入门实例/（美）戈贝尔（Goebel，R. P.）著；（墨）罗哈斯（Rojas，J.）等译. —广州：中山大学出版社，2016. 1
ISBN 978 - 7 - 306 - 05511 - 8

Ⅰ. ①R…　Ⅱ. ①戈… ②罗…　Ⅲ. ①机器人—程序设计　Ⅳ. ①TP242

中国版本图书馆 CIP 数据核字（2015）第 262110 号

出 版 人：徐　劲
策划编辑：周建华　黄浩佳
责任编辑：黄浩佳
封面设计：曾　斌
责任校对：廖丽玲
责任技编：何雅涛
出版发行：中山大学出版社
电　　话：编辑部 020 - 84110283，84113349，84111997，84110779
　　发行部 020 - 84111998，84111981，84111160
地　　址：广州市新港西路 135 号
邮　　编：510275　　传　真：020 - 84036565
网　　址：http://www.zsup.com.cn　　E-mail:zdcbs@mail.sysu.edu.cn
印 刷 者：广东省农垦总局印刷厂
规　　格：880mm × 1230mm　1/16　15.25 印张　440 千字
版次印次：2016 年 1 月第 1 版　2016 年 1 月第 1 次印刷
定　　价：58.00 元

0 前　言

本书指导你如何利用编程使你的机器人做一些神奇的事情。比如，利用识别人脸或其他实际物体的技术，让机器人在你的房间里自主导航，或者使其对你的口头命令做出反应。ROS（the Robot Operating System，机器人操作系统）是美国加利福尼亚州的 Willow Garage 创造出来的，现在由机器人研究开源组织（the Open Source Robotics Foundation，OSRF）维护运营。多亏了 ROS，我们在这本书中将使用当前领域中最先进的机器人软件。

ROS 的首要任务就是为机器人提供一个标准化的、开源的编程框架，以便在各种现实和虚拟环境中实现机器人控制。OSRF 当然不是第一个做出这种努力的组织。事实上，在维基百科上搜索，显示有超过 15 个机器人软件机构。但 Willow Garage 决不仅仅是一个只放出免费软件的程序员团队。在大量资金、专业知识和一系列精确的开发计划的推动下，Willow Garage 成功地在机器人学家中点燃了 ROS 编程热，机器人研究者们可以使用在短短数年里由其他 ROS 用户贡献出的大量 ROS package 来编程。ROS 目前的软件已经涵盖了导航定位（SLAM）、3D 物体识别、动作规划、多关节机械臂运动控制、机器学习，甚至可以让机器人打桌球。

与此同时，Willow Garage 也设计并造出了一款名为 PR2 的机器人来更好地演示它的操作系统，这款机器人售价 400 000 美元。其使用了当前最先进的机器人硬件，包括两枚立体摄像头、一对激光感应器，拥有 7 个自由度的机械臂和一套万向轮驱动系统。但是，只有少数人有机会在真正的 PR2 上运行 ROS，包括 11 个得到免费 PR2 的研究机构，作为公测。然而，你并不需要一部 PR2 来领略 ROS 的强大，因为现在已经有支持更加低成本平台和组件的 package 被创造出来了，这些平台有：iRobot Create，TurtleBot，Arduino，WowWee Rovio，LEGO NXT，Phidgets，Arbot-iX，Serializer，Element，Robotis Dynamixels。

ROS 奠定在一个指导原则上——“不重复发明车轮”[①]。成千上万的聪明人已经在机器人编程这条道路上走了超过 50 个年头，何不将这些聪明的大脑才智集中在一起呢？幸运的是，现代互联网为我们提供了一个绝佳的分享代码的媒介。现在，很多大学、公司以及个人在网上公开地分享他们的 ROS 源代码资源。并且在有了免费的云平台（如 Google Code，GitHub）的情况下，任何人都可以轻易地且不需要花一分钱就能分享自己的 ROS 代码成果。

搭乘 ROS 这趟列车最精彩、最令人兴奋的部分或许就在于可以和来自全世界数以千计的志同道合者们一同研究机器人。你不用再担心自己苦心研究多年的内容同时别人也在进行同样的研究，更妙之处在于：当你发现你的努力正在为一项快速发展领域作出一丝贡献时，你会由衷地感到欣慰。

0.1 关于本书纸质版和电子版的说明[②]

本书的纸质版和电子版除了存在几处重要的不同外，其他部分几乎是一样的。两者的页面排版是完全一样的，但是多数电子版读者会把文档的开始页作为第一页而忽略了本书在正文中设置

① 译者注：即减少重复劳动。

② 外文原版书中，纸质版和电子版同时发行，考虑到国内的实际情况，作出了一些修改。本书超链接都以脚注的形式标注，方便读者查阅。

的页码。书中的图片和代码在电子版中是彩色的而在纸质版中却是黑白的，这样可以降低书本的制作成本。电子版的文本中有很多可以直接用电脑鼠标点击相关网页的超链接，在纸质版中这些超链接被附上了下划线和标号。要访问这些超链接，可以查看书后的超链接，那里有对应的链接地址。

拥有最新的版本：如果你想获知本书以及其附带代码的更新信息，请加入我们的 ros-by-example Google Group。

0.2 从 GROOVY 到最新版本的演变

如果你首次接触 ROS 或本书时使用的是 Hydro，那你可以略过本章。然而，如果你通过本书的旧版本接触过 Groovy 或者更早版本的 ROS，那你需要注意一些版本之间的变化。你可以通过查阅官方提供的 ROS 的 Groovy 版本和 Hydro 版本差异列表（Groovy →Hydro migration page[①]）来了解不同版本间的差异。以下是一些会影响本书中代码的变化。

0.2.1 ROS 版本变化

- 现在 catkin 的 build system 需要把一些包提交到 ROS build farm 中，从而把这些包转化成可以通过 apt-get 来安装的 Debian 包。因此，本书中用到的位于 ros-by-example 中的栈需要被转化成一个 catkin meta package。你可以通过查阅位于 ROS 的 Wiki 上的 catkin 指南，得到创建自己的 catkin 包的命令。你也可以通过查阅本书第 4 章获得。
- catkin 要求所需模块所在的包位于地址 package/src/package_name 下。对 ros-by-example 而言，这部分代码要求把一些位于某些包的 nodes 子目录下的 Python 文件移动到对应的 src/package_name 目录下。具体而言，rbx1_vision 包要求把 face_detector. py，good_features. py 和 lk_tracker. py 三个文件从目录 rbx1_vision/nodes 移动到目录 rbx1_vision/src/rbx1_vision。
- 在 Python 脚本中，我们不再需要导入 roslib 并运行 roslib. load_manifest（）。因此在 ros-by-exampe 代码中，这些代码全都从 Python 节点中删除了。
- 诸如 rxconsole，rxplot 等在 wxWidgets 中的 ROS 工具，都已经被改写成 Qt 版本，并改名为 rqt_console，rqt_plot 等。它们的基本功能都被保留下来了，不过你现在需要使用新的版本（在 Groovy 中两个版本都是可用的）。你若想得到详尽的工具列表，请查阅 Wiki 中的 rqt_common_plugins 页面。
- 多亏了 David Lu，现在 ROS 的导航栈可以使用分层的代价地图了。虽然这个内容超出了本书的范围，但是这个方法让我们可以在特定情景下创建自定义的代价地图。幸运的是，ROS Hydro 仍然可以用以前的代价地图格式，而不需要在这个时候考虑分层。
- 另外一个关于 costmap 的变化是，现在在 RViz 中使用和显示它们时是把它们看作 maps 而不是 grid squares。这个变化使我们在使用以前的配置文件时，要在 RViz 中改变障碍物的显示类型，而不是以前使用的 grid squares 类型。在 Hydro 版本中，配置文件 ros-by-example 已经完成了这个更新。
- tf 库现在已经被更新为 tf2 库。这个新库是原来旧库的一个超集，但并没有为本书用到的代码提供任何新的功能。因此，我们在本书中会继续引用 tf 库。
- 新的 tf2 库的一个缺点是删除了命名中出现的 tf 前缀，这使得在第 10 章中引用到的 skeleton_markers 和 pi_tracker 包都需要重新编码。

① 地址：http://wiki. ros. org/hydro/Migration。

- 在某些安装好的 Hydro 中，可能会出现 openni_tracker 包损坏的情况。我们会在 10.9 节讨论 skeleton tracking 时提供解决方案。
- 以前在 URDF 模型中用到的 <include> 标签已经被删除了，并替换成了 <xacro：include>。所有的 URDF 文件都已经相应地更新了，包括 ros-by-example。

0.2.2 示例代码的变化

- 在 rbx1_dynamixels 包里面的 head_tracker.py 脚本已经更新成线程安全的了，这会防止在旧版本中出现的 random servo motions。
- 一个名为 object_follower.py 的脚本被添加到了 rbx1_apps 包里。这个脚本在 object_tracker.py 的基础上，还考虑了深度信息，使得机器人可以跟随一个移动的目标。
- 当 use_depth_for_tracking 参数被设置为 True 时，在 rbx1_vision 包里面的 face_tracker2.py 脚本可以对 depth information 进行正确的反应。在旧版本中，这个脚本中的一些错误使得这个参数被忽略了。

目　录

1 本书目标

ROS 是极其强大的，而且它还在迅速、持续地扩展和改进。但是许多新 ROS 用户所面临的一个挑战是不知道从哪里开始。开始 ROS 主要包含两个阶段：阶段 1 是学习基本的概念和编程技巧；阶段 2 则是运用 ROS 来控制你自己的机器人。

阶段 1 最好参考ROS Wiki[①]，在那里你可以找到一系列极好的安装指导[②]和新手教程[③]。这些教程都被成百上千的用户实际测试过，所以没必要在这本书中重复。阅读这些教程应该是使用这本书的预先准备工作。我们假设你已经学习过至少一次所有的教程。除此之外，阅读tf overview[④]和练习tf Tutorials[⑤]也很重要，它们可以帮助你理解 ROS 如何处理不同参照框架。如果你遇到了问题，可以访问ROS Answer 论坛[⑥]，那里有很多读者遇到并已经被解答过的问题。如果你没有找到自己所遇到的问题，那你可以在那里贴出自己的新问题与其他人探讨。（贴出新问题时请不要使用 ROS-users 邮件列表，这是为 ROS 新闻和通知预留的）

阶段 2 的主要内容：使用 ROS 让你的机器人做一些令人印象深刻的事情。每一个章节会给出与 ROS 不同方面相关的教程和示例代码。然后我们会将这些代码应用到一个真实世界的机器人，例如，一个可抬头、摇头的头部，或者一个照相机（如面部探测仪）。本书大部分章节并没有严格的先后阅读次序，读者可以根据自己喜欢的顺序完成本书教程。同时，这些教程也会以彼此为基础，所以到了本书的结尾部分，读者的机器人应该能够自主地导航到读者的家或者办公室，回应语音命令，甚至结合视觉和动作控制来进行面部跟踪或者在房间里跟随某个人。

在阶段 2 中，我们会接触到以下话题：

- 安装和配置 ROS（回顾）。
- 在不同层次的抽象层面控制一个可以移动的底部，从马达驱动器和车轮角度编码器开始，逐渐上升到路径计划和地图构建。
- 导航和 SLAM（即时定位与地图构建）使用激光扫描仪或者一个深度照相机（例如，微软 Kinect 或者华硕 Xtion）。
- 语音识别和合成，还有通过语音命令控制的机器人的一个应用。
- 机器人视觉，包括人脸检测和使用 OpenCV 进行颜色追踪，使用 OpenNI 进行骨架追踪，还有一个对进行 3D 视觉处理 PCL 的大致介绍。
- 结合机器人视觉和一个移动的底部创造两个应用，一个是对脸部和有颜色的物体进行跟踪，另一个是当某人在房间里走动时进行跟随。
- 使用 Dynamixel 伺服机控制一个可抬头摇头的照相机来跟踪一个移动物体。

考虑到 ROS 框架的深度和广度，我们在本书中省略了一些话题，尤其是我们并没有涉及眺

① 地址：http://wiki.ros.org/。
② 地址：http://wiki.ros.org/hydro/Installation。
③ 地址：http://wiki.ros.org/ROS/Tutorials。
④ 地址：http://wiki.ros.org/tf。
⑤ 地址：http://wiki.ros.org/tf/Tutorials。
⑥ 地址：http://answers.ros.org。

望台模拟器（Gazebo simulator[①]）、MoveIt[②]（正式称呼手臂导航[③]）、桌面对象操作[④]（tabletop object manipulation）、Ecto[⑤]、创建个人的 URDF 机器人模型、机器人诊断（robot diagnostics），或者任务管理器的使用（例如SMACH[⑥]）等内容。不过，从上面列出的话题列表可以看出，我们仍然有很多东西要学习。

① 地址：http://gazebosim.org/。
② 地址：http://moveit.ros.org/。
③ 地址：http://wiki.ros.org/arm_navigation。
④ 地址：http://wiki.ros.org/pr2_tabletop_manipulation_apps。
⑤ 地址：http://plasmodic.github.io/ecto/。
⑥ 地址：http://wiki.ros.org/executive_smach。

2 真实机器人和模拟机器人

虽然 ROS 可以用在多种类型的硬件上，但是并不是一定要有一个真实的机器人才能开始学习它。ROS 包括许多能在模拟环境中运行机器人的程序包，以便在真实的世界里冒险之前就可以测试你的程序。

2.1 Gazebo，Stage 和 ArbotiX 模拟器

在机器人操作系统中有许多方式可以模拟机器人。然而，最有经验的人会使用 Gazebo。首先，Gazebo 是一个功能齐全的 3D 物理模拟器，它比 ROS 出现要早，但现在已经与 ROS 良好地集成。其次，有人会使用 Stage。Stage 是一个简单一些的 2D 模拟器，它可以管理多个机器人和激光扫描仪等各种传感器。而经验没那么丰富的人会使用 Michael Ferguson 的 arbotix_python 包。这个程序包可以在一个差速传动机器人上运行一个假的模拟环境，但是没有传感器反馈或其他物理效果。我们将使用第三种方法来进行设置，因为这是最简单的一种方法，而且从我们的目的来看，我们也不需要物理效果的反馈。当然，你可以自由地探索 Gazebo 和 Stage，但是请准备好花一些时间在细节上做功课。特别是使用 Gazebo，需要具备相当高的 CPU 处理能力。

即使你真的拥有一个属于自己的机器人，在模拟器中运行本书中的例子仍然是一个不错的主意。一旦你对结果感到满意，你就可以在自己的机器人上试着运行了。

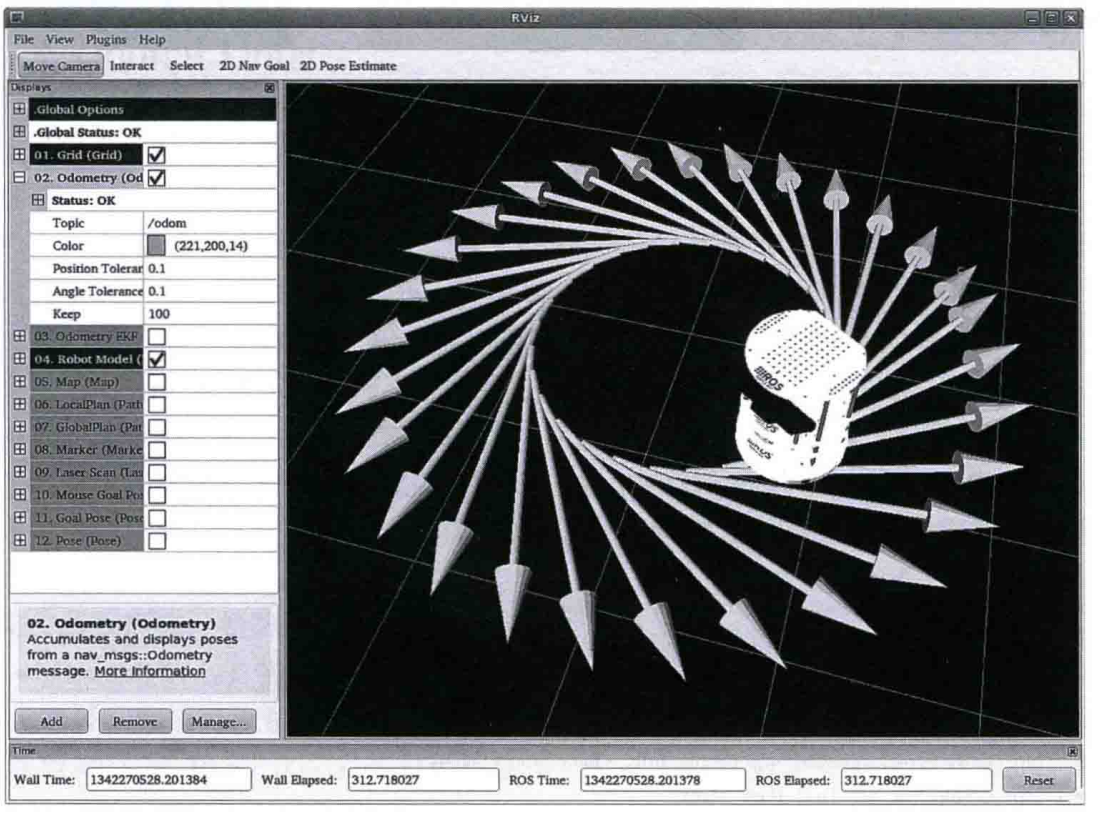
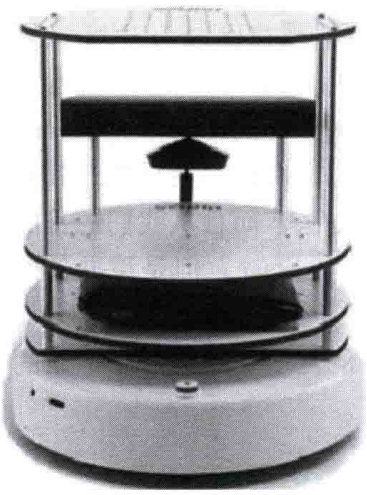


图 2－1

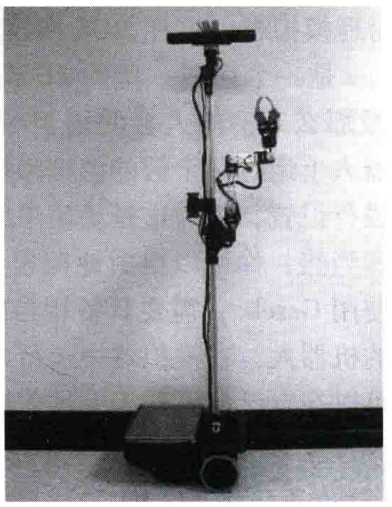
2.2 关于 TurtleBot，Maxwell 和 Pi 机器人的简介

就本书的目的来看，我们需要一个至少能在模拟环境中运行的机器人来测试我们的代码。ros-by-example 资料包中包含对两种测试机器人的支持：由Melonee Wise[①] 和Tully Foote[②] 创造的 Willow Garage TurtleBot[③]（图 2 - 2 - a）和本书作者自己制造的名叫Pi Robot[④]（图 2 - 2 - b）的机器人。

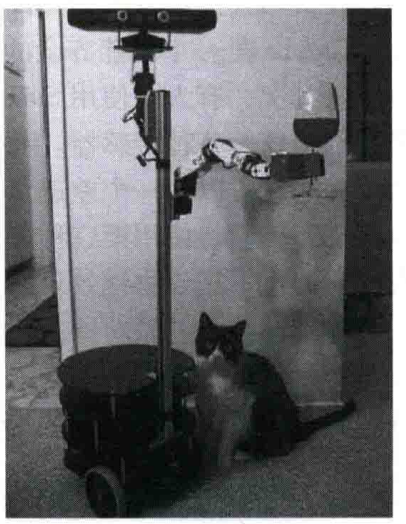
Pi 的灵感来自Michael Ferguson[⑤] 的Maxwell[⑥]（图 2 - 2 - c）机器人。而 Maxwell 反过来则是以佐治亚理工大学的EL-E[⑦] 机器人为原型。如果你拥有一个你自己机器人的 URDF 模型，你可以使用它来替代上面提到的机器人。在任何情况下，我们开发的绝大部分代码都可以运行在几乎任何支持基本 ROS 消息接口的机器人上。



（a）TurtleBot机器人



（b）Pi机器人



（c）Maxwell机器人

图 2 - 2

① 地址：http://www.meloneewise.com/index.ph。
② 地址：http://www.osrfoundation.org/people/tully - foote.html#_blank。
③ 地址：http://wiki.ros.org/TurtleBot#_blank。
④ 地址：http://www.pirobot.org/#_blank。
⑤ 地址：http://www.showusyoursensors.com/。
⑥ 地址：http://wiki.ros.org/maxwell。
⑦ 地址：http://www.ros.org/news/2010/03/robots - using - ros - georgia - techs - assistive - robots.html。

3 操作系统和 ROS 版本

ROS 可以在不同版本的 Linux、Mac OS X 运行，也可以部分地运行在微软 Windows 上。但是最方便的还是在 Ubuntu Linux 上使用，因为 Ubuntu Linux 是 OSRF 官方支持的操作系统。除此之外，Ubuntu 是免费的，而且可与其他系统共存。假如你没有使用过 Ubuntu，我们会在下一部分对于如何安装给出一些提示。

从超级计算机到 Beagleboard ，ROS 都可以在上面运行。但是我们开发的很多代码会要求一些 CPU 密集型处理。所以如果你使用笔记本或者台式机来运行这本书里面的样例，你会省去很多时间和减少可能的挫败。Turtlebot，Maxwell 和 Pi Robot 是被设计用来使你能够很容易地在机器人上放置笔记本。这意味着你可以直接在笔记本上开发代码，然后在机器人上测试自主行为。你甚至可以在开发的途中，重新通过长 USB 电缆用笔记本或台式机来控制机器人。

最新的 Ubuntu 版本是 13.04（Raring），最新的 ROS 版本是 Hydro。本书代码是在 ROS Hydro 和 Ubuntu 12.04（Precise）上测试的。Ubuntu 12.04（Precise）是目前长期支持的 Ubuntu 版本。

各种版本的 ROS 可以很好地兼容在你的机器上（如果你的操作系统支持的话），但是请确保当运行本书里面的命令和代码样例时，现行版本是 Hydro。

注意：对于连接到同一个 ROS 网络中的所有机器（包括机器人上的电脑和桌面工作站的电脑），请确保它们运行的 ROS 版本相同。因为 ROS 信息签名每一版本都有修改，不同版本之间基本无法使用对方的话题和服务。

3.1 安装 Ubuntu Linux

如果你已经在机器上安装并运行了 Ubuntu Linux，那很棒。但是如果你才开始，安装 Ubuntu 也并不困难。

最新的 Ubuntu 版本是上面提及的 13.04（Raring），但是本书基于 Ubuntu 12.04（Precise）。关于一系列与 ROS Hydro 官方兼容的 Ubuntu 版本，请查看ROS 安装指南[①]。

Ubuntu 可以被安装在一台使用 Windows 的机器上或者一台基于 Intel 的 Mac。假如你想保持原有系统完整的同时安装 Ubuntu，那么你可以在启动的时候选择操作系统。另一方面，如果你有另一台闲置的笔记本，你可以只安装 Ubuntu，专门用它来操作机器人（这样会运行得更快）。一般来说，不建议在虚拟机如 VMware 上安装 Ubuntu。尽管这是一个运行 ROS 学习指南的好方法，但是虚拟机很可能在运行图形密集型程序如 RViz 时卡住（事实上，这些程序有可能根本没办法运行起来）。

以下是一些我们在开始阶段会用到的链接：

- 在 windows 电脑上安装 Ubuntu 12.04[②]。
- 除此之外，你还可以试试Wubi Windowsinstaller[③]（请确保你选择的是 12.04 版本，因为默认版本是 12.10）。这个安装包运行在 Windows 下，它会像安装一个 Windows 应用程序

① 地址：http://wiki.ros.org/hydro/Installation/Ubuntu。

② 地址：https://help.ubuntu.com/12.04/installation-guide/i386/。

③ 地址：http://www.ubuntu.com/download/desktop/windows-installer。