瑞森可

Baxter ™

研究版•操作手册

作者	版本	编订日期
售后部	2.0	2020/02/13

内容提要

Baxter 机器人是总部位于波斯顿的美国 Rethink Robotics 公司推出的一款协作型机器人，湖南瑞森可机器人科技有限公司已完全拥有 Baxter 品牌和技术。Baxter 配备了基于 ROS（Robot Operating System）的软件开发套件 (SDK)，是一个安全、经济且强大的平台。

目前全球大多数实验室和学校都配有 Baxter，该机器人广泛应用于机械手臂运动规划、双臂柔顺协调控制、机器视觉、人机交互等领域的科研和教学活动。

此操作手册共分三个部分，分别是 Baxter 设置，工作站设置，运行示例。

Baxter 设置部分包括：硬件配置、工作空间选择、夹持器安装等。

工作站设置包括：Ubuntu 及 ROS 安装、依赖软件包安装、SDK 安装等。

运行示例包括：使能机器人、运行示例程序。

通过使用该手册，初学者能够快速熟悉 Baxter 的基本控制和工作流程，对其软硬件有一定初步认识。

目 录

1 Baxter 安装及设置

1.1所需硬件及工具

- Baxter 科研版机器人本体
- 1/2 英寸扳手
- 17mm 扳手
- 27mm 扳手（用于底座安装）
- 平头螺丝刀
- 电动平行夹持器或气动夹持器
- Baxter 底座
- 内六角扳手一套
- USB 键盘
- 路由器及网线

1.2选择合适的工作空间

Baxter 机器人在工作时应保留足够的空间以免机械臂碰到障碍物停止运动，Baxter 工作空间分布参考：图 1.1-图 1.4。

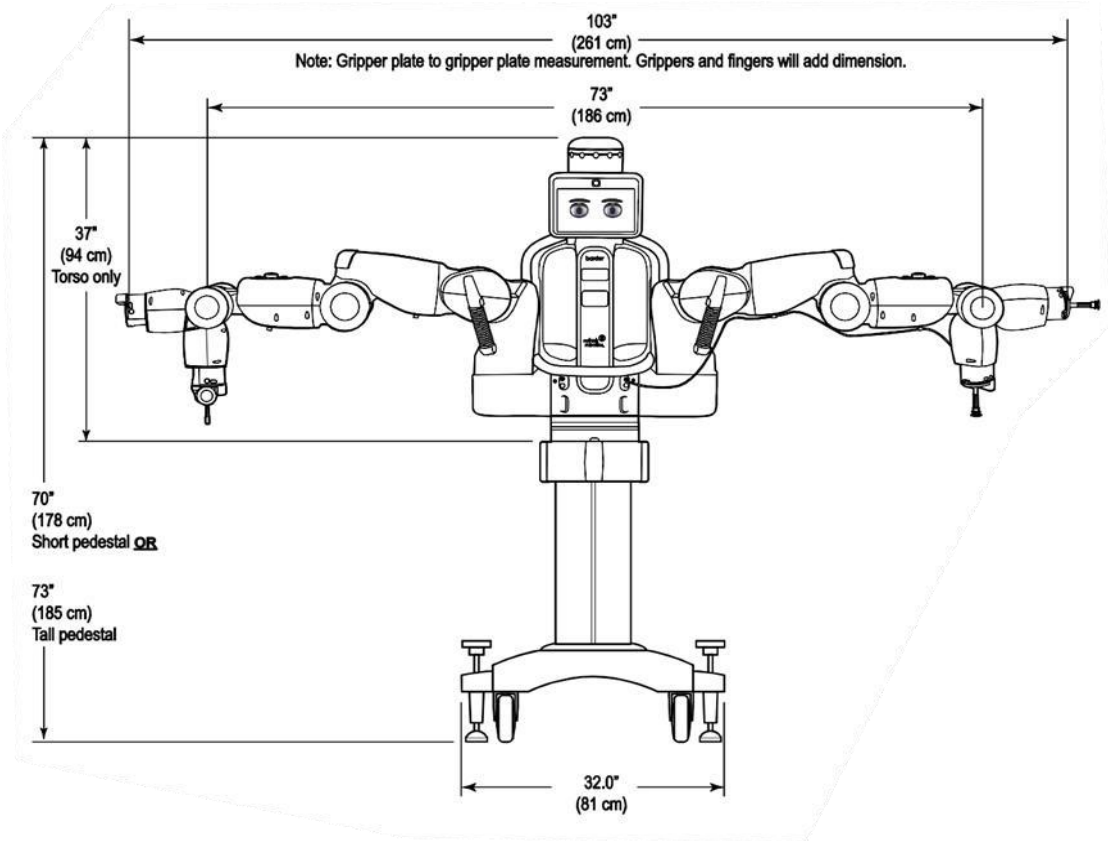


图 1.1 Baxter 正视图

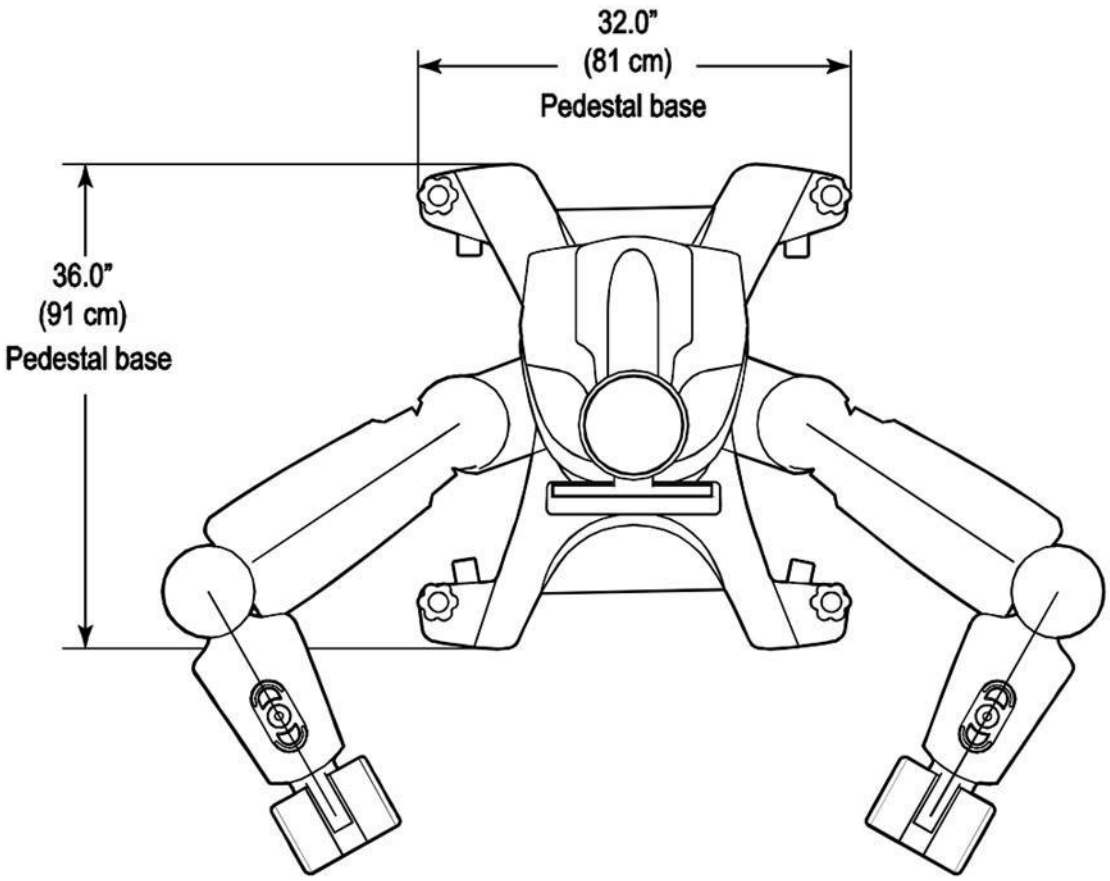


图 1.2 Baxter 俯视图

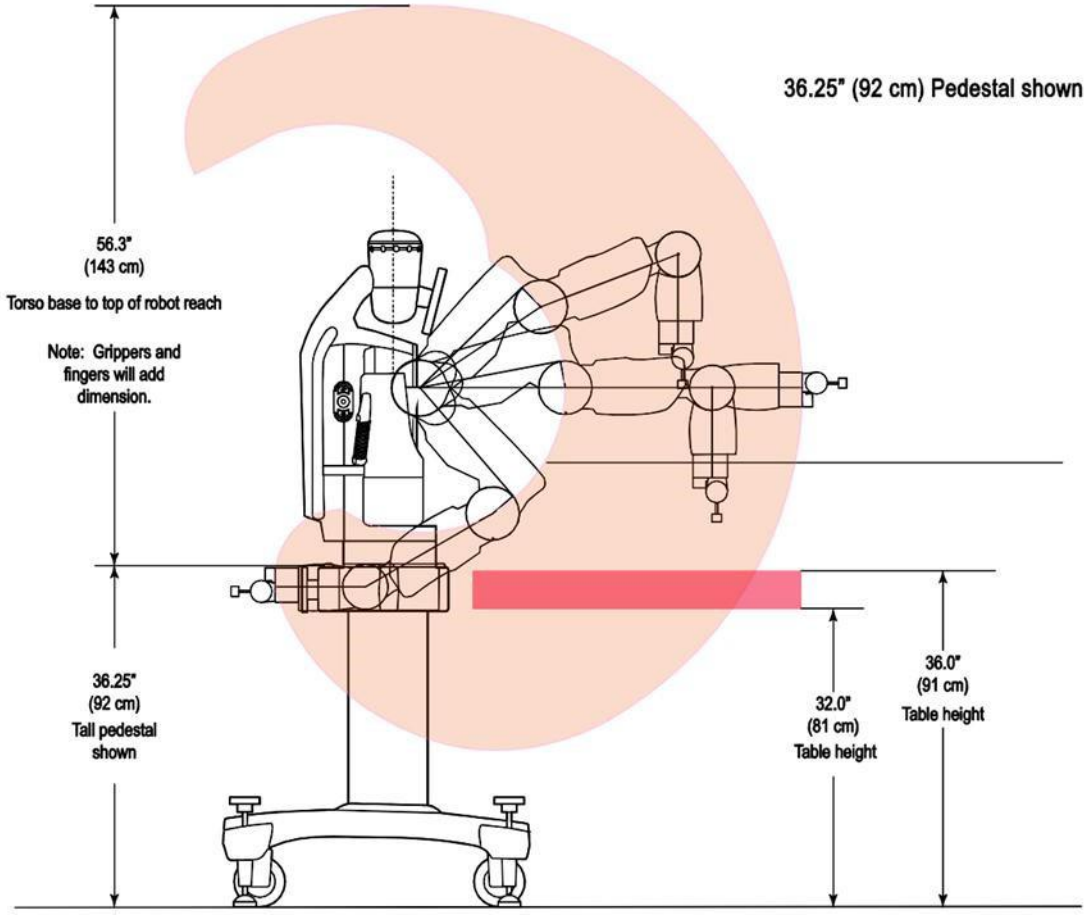


图 1.3 Baxter 侧视图

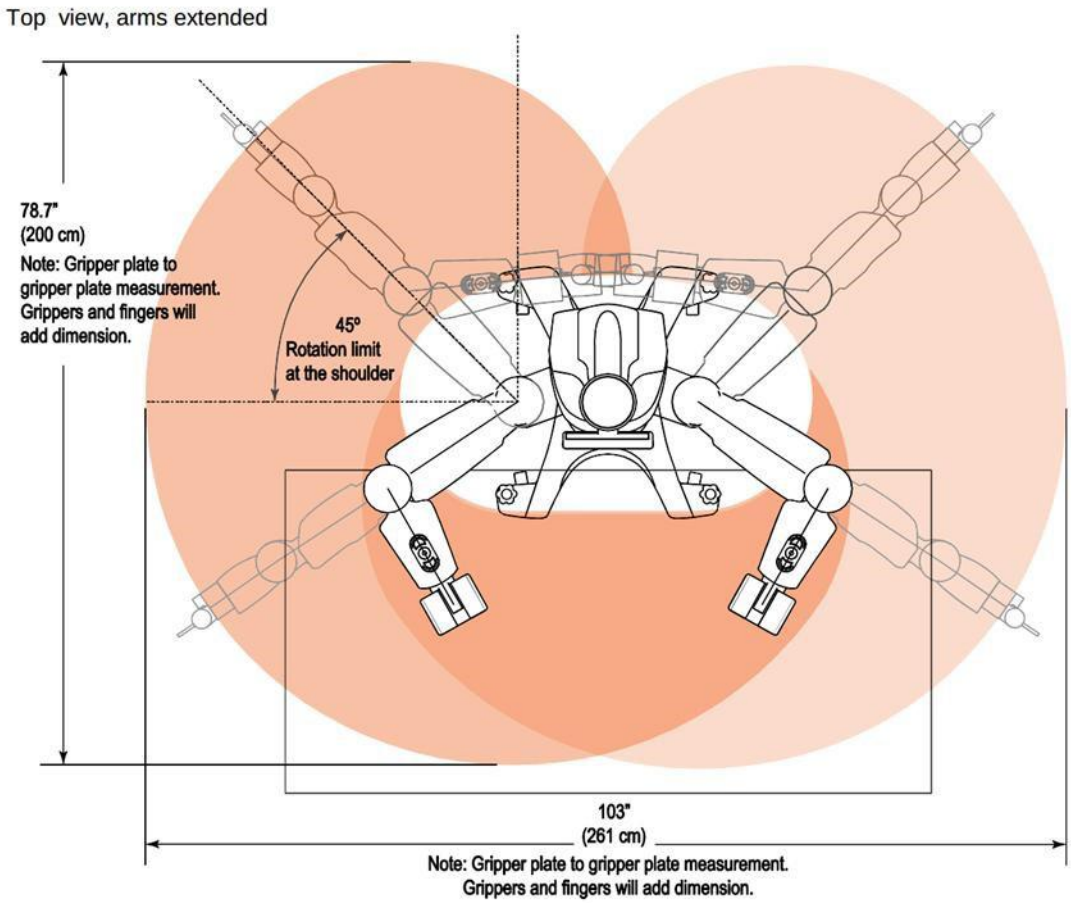


图 1.4 Baxter 机械臂运动范围

1.3 Baxter 安装

首先参考图 1.5 选择底座安装高度。

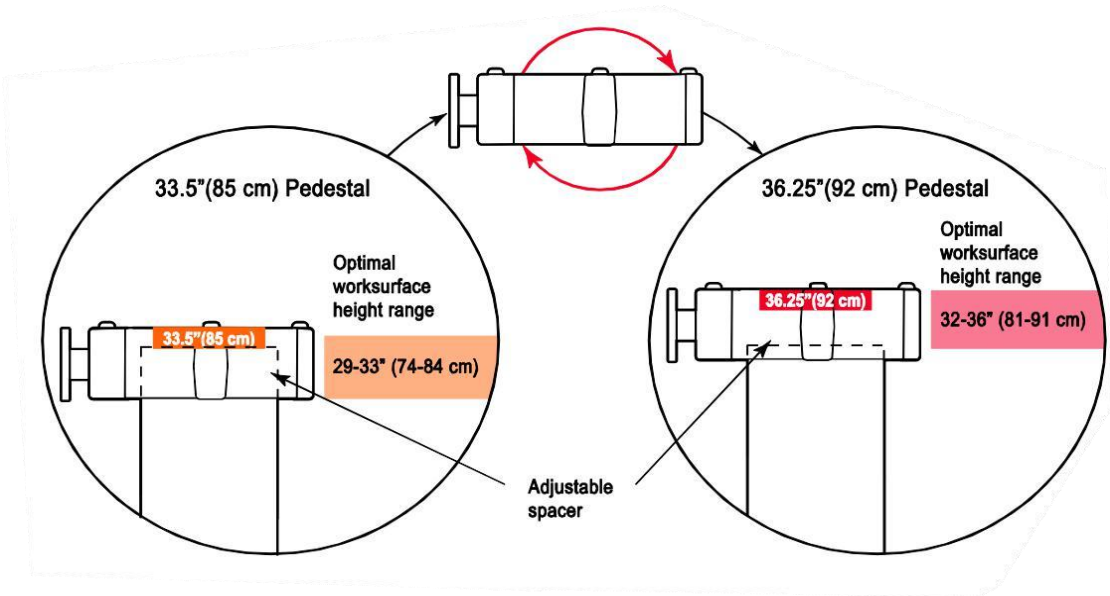


图 1.5 Baxter 底座安装高度

然后将 Baxter 本体吊装到底座上，如图 1.6 所示。

注意：机器人本体在人工搬运时，只有其黑色的铁制框架部分可以着力，切勿直接着力于其手臂关节进行搬运。

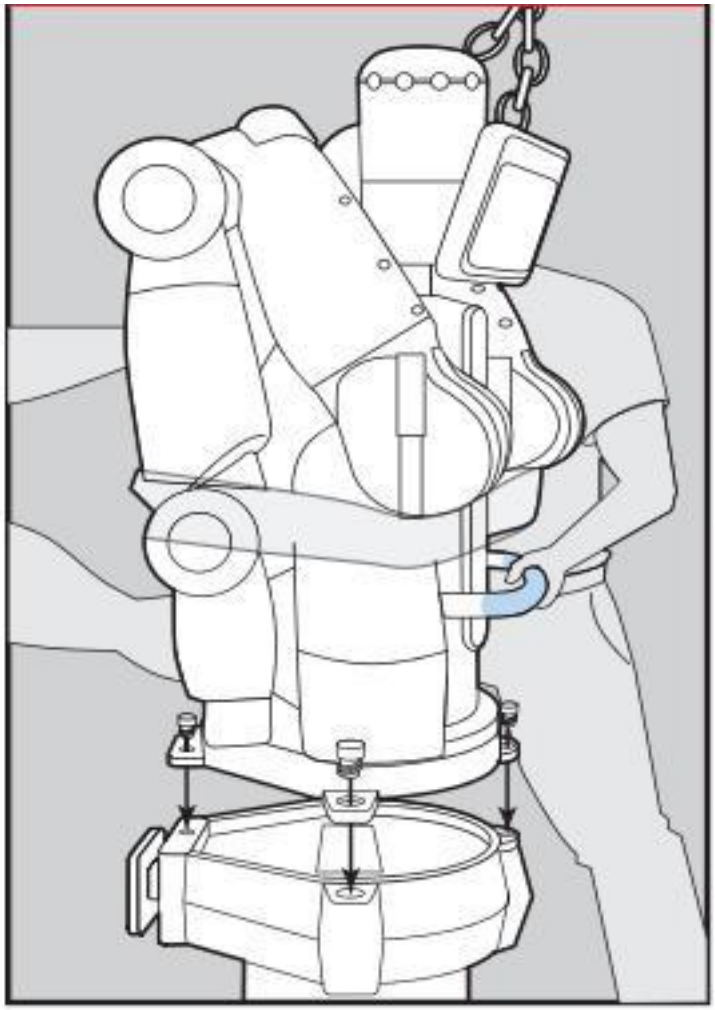


图 1.6 吊装 Baxter

1.4安装夹持器

1.4.1 安装电动夹持器

参考图 1.7-图 1.9 所示步骤。

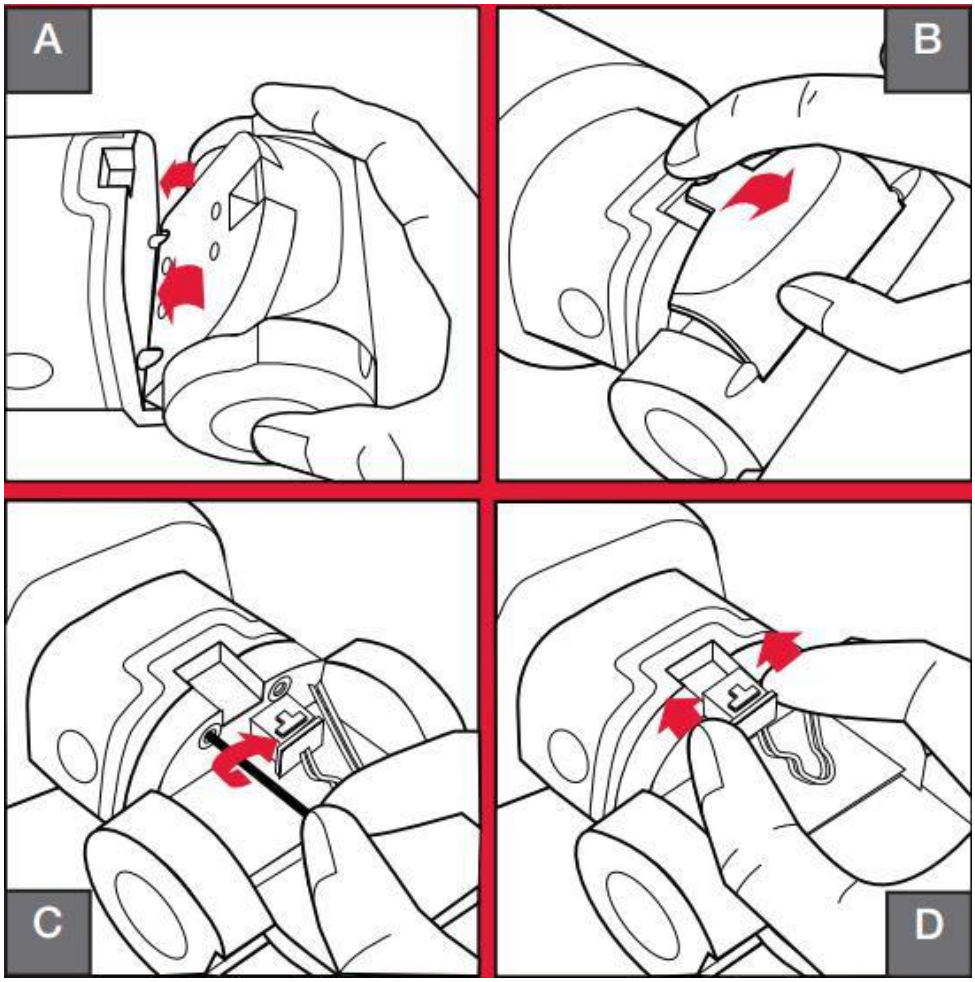


图 1.7 安装电动夹持器基座

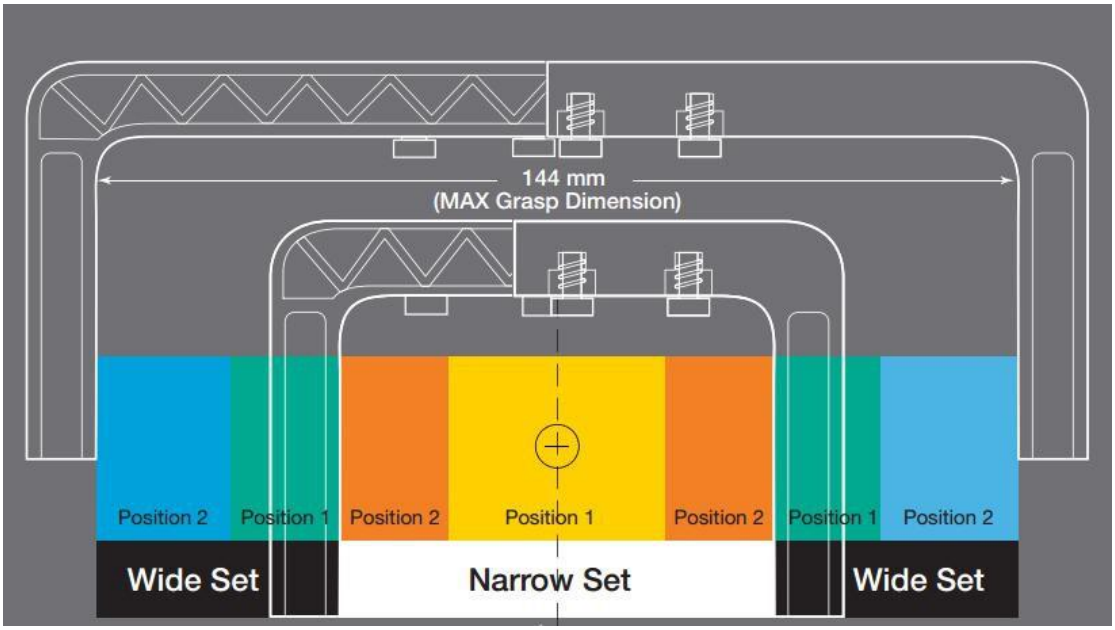


图 1.8 选择不同开度的手指

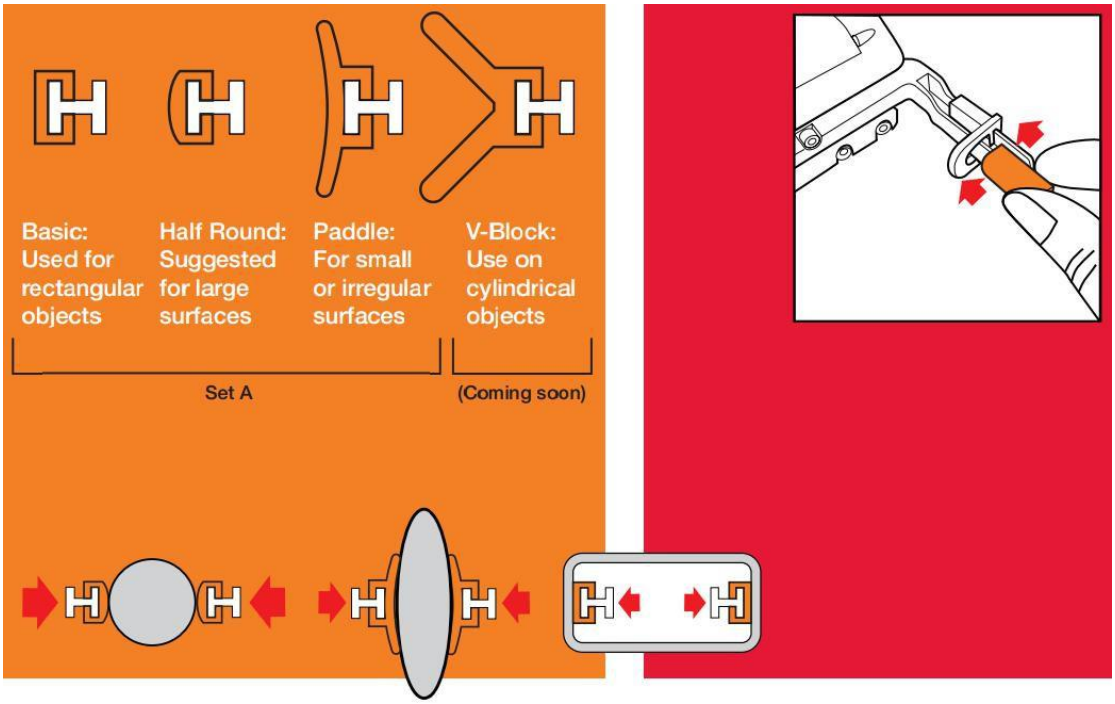


图 1.9 选择指套

1.4.2 安装气动夹持器

参考图 1.10-图 1.13 所示步骤。

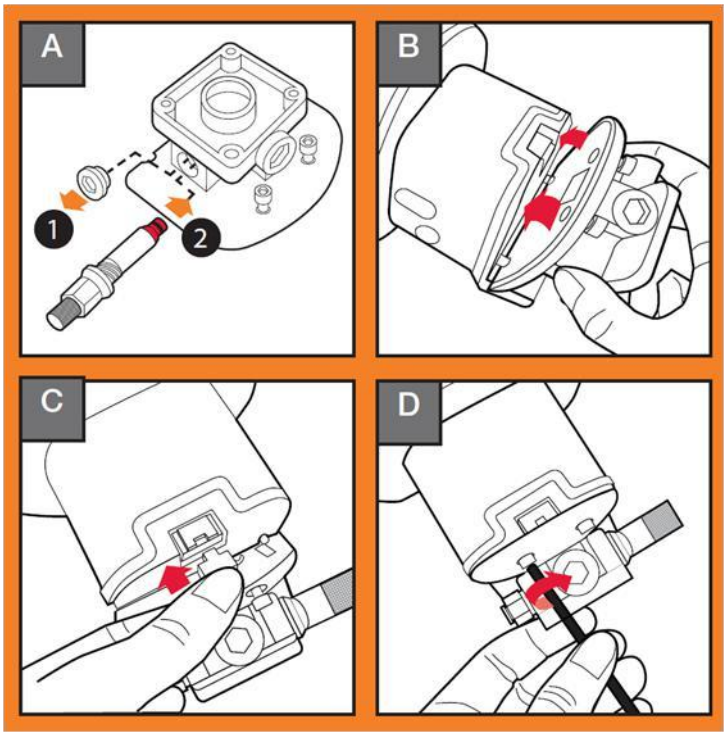
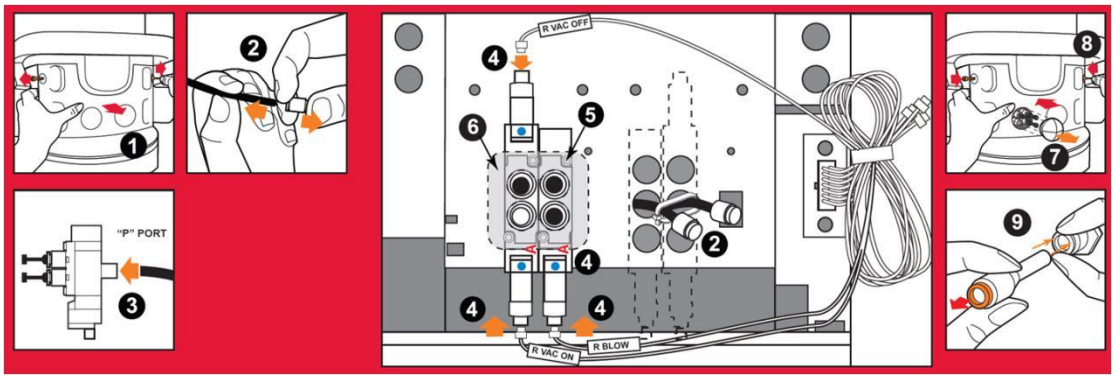
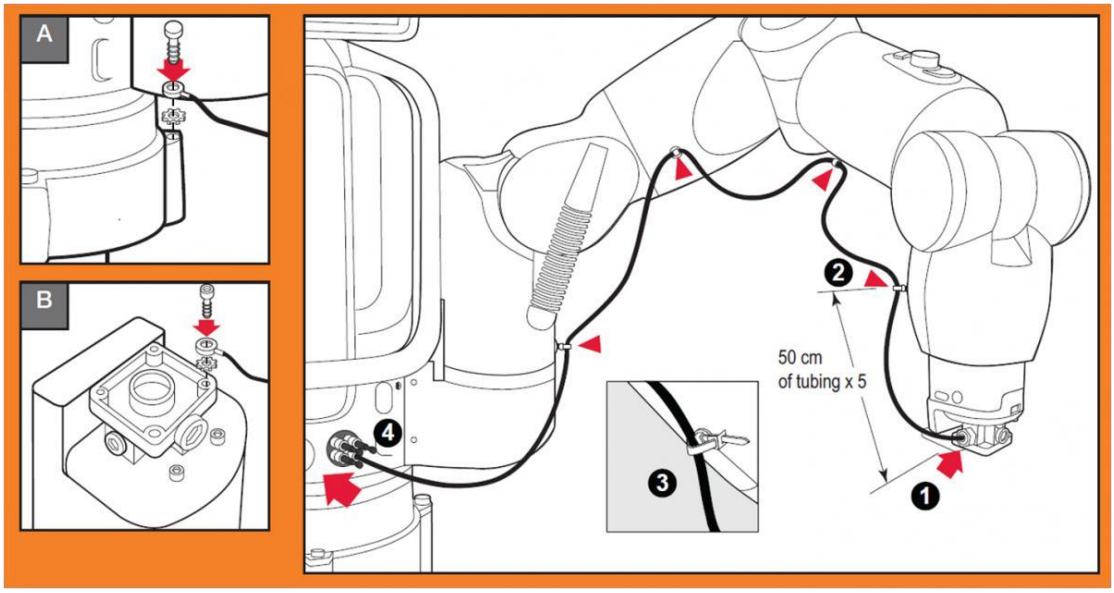


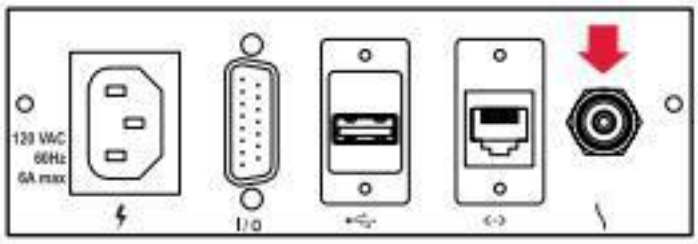
图 1.10 安装气动夹持器基座



1.11 安装电磁阀



1.12 连接夹持器地线及气管



1.13 气泵通过背部气孔（直径 6mm）向机器人提供气源

1.5连接急停开关及电源

Baxter 支持通用电源接口，工作电压：90 - 264V AC (47 - 63Hz)，最大功耗 720W。

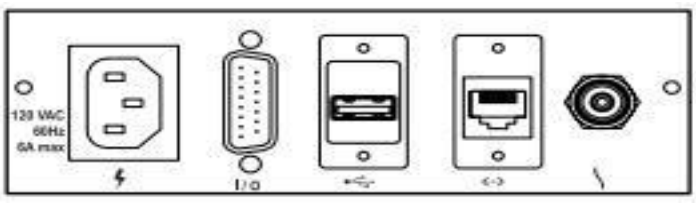


图 1.14 将急停开关连接至 I/O 端口

1.6打开电源

至此，Baxter 机器人硬件配置已经完成，打开背部开关后，约 3 分钟后会显示图 1.15 的欢迎界面。

图 1.15 欢迎界面

2 工作站设置

2.1 Ubuntu 安装

Ubuntu 系统（14.04 系统或者 16.04 系统）的安装方法有很多（使用者可以去自己上网查找一种安装方法来安装 Ubuntu 系统）。在这里我们使用 U 盘来制作 U 盘启动盘安装 Ubuntu16.04、Ubuntu14.04 系统，本手册以安装 Ubuntu16.04 系统为例：

2.1.1 Ubuntu16.04 系统镜像

参考下载地址:http://releases.ubuntu.com/

2.1.2 制作 U 盘启动盘

1、工具

①、Rufus 工具　　下载链接：https://rufus.ie/（本手册示例以 Rufus 工具来制作 U 盘启动盘）

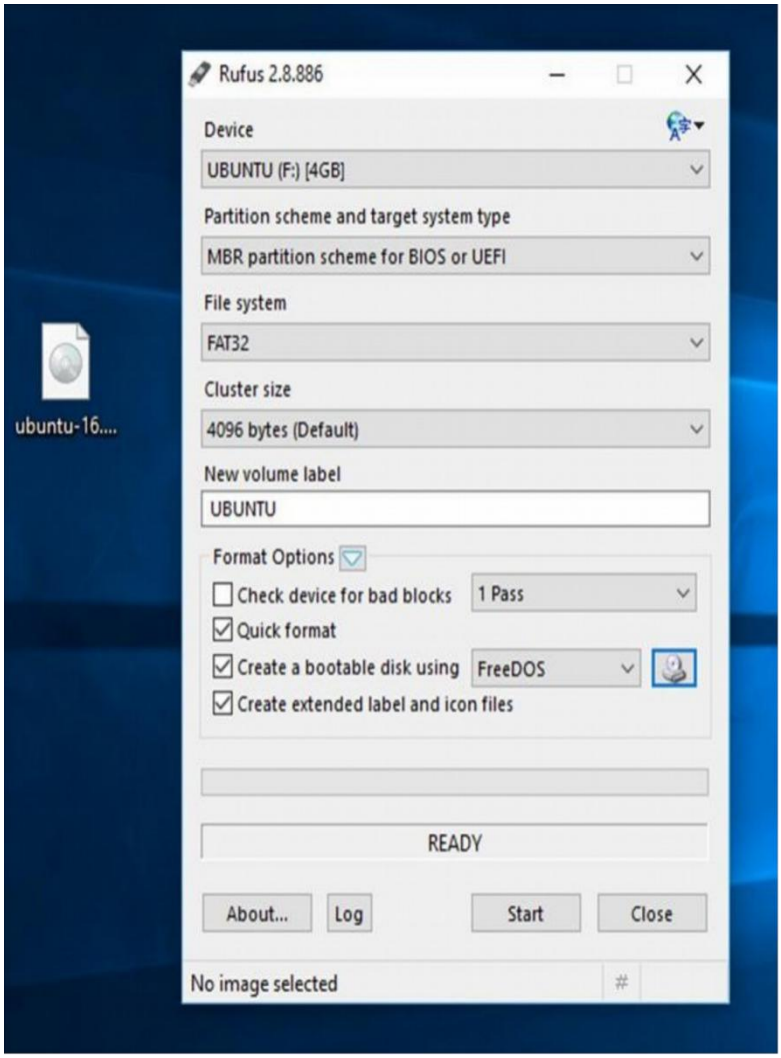
②、Etcher 工具　　下载链接：https://www.balena.io/etcher/

③、Unetboot 工具　下载链接 https://unetbootin.github.io/

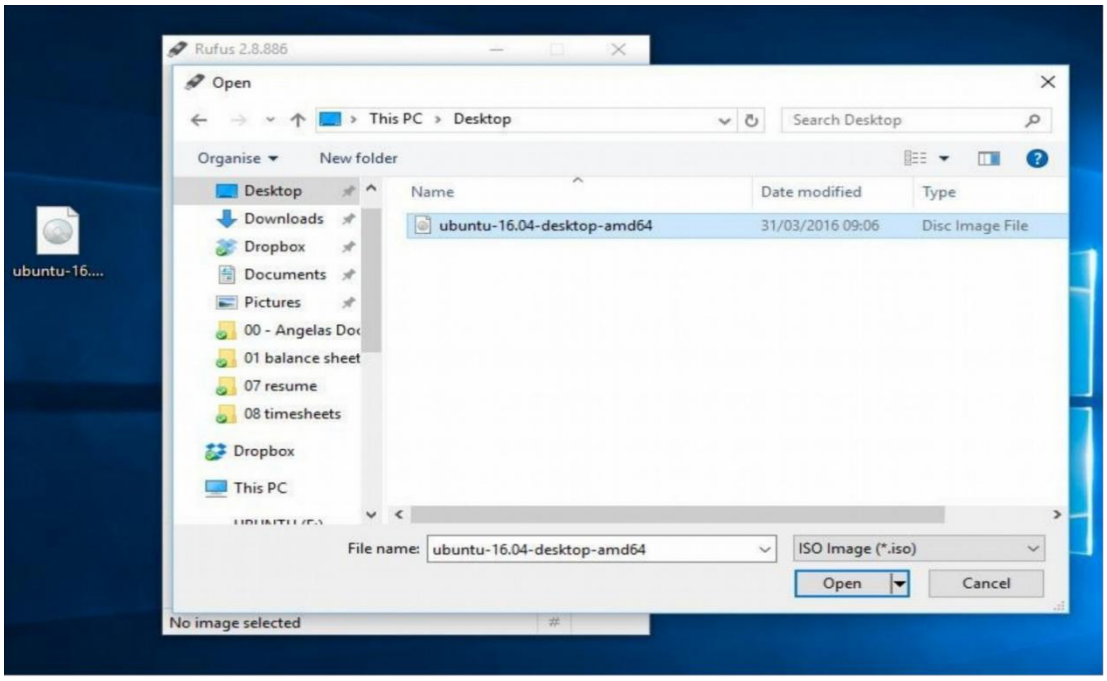
2、下面以 Rufus 为例,介绍如何制作 USB 启动盘

①、打开 Rufus 的下载链接：选择 Rufus 3.5（1 MB）进行下载。

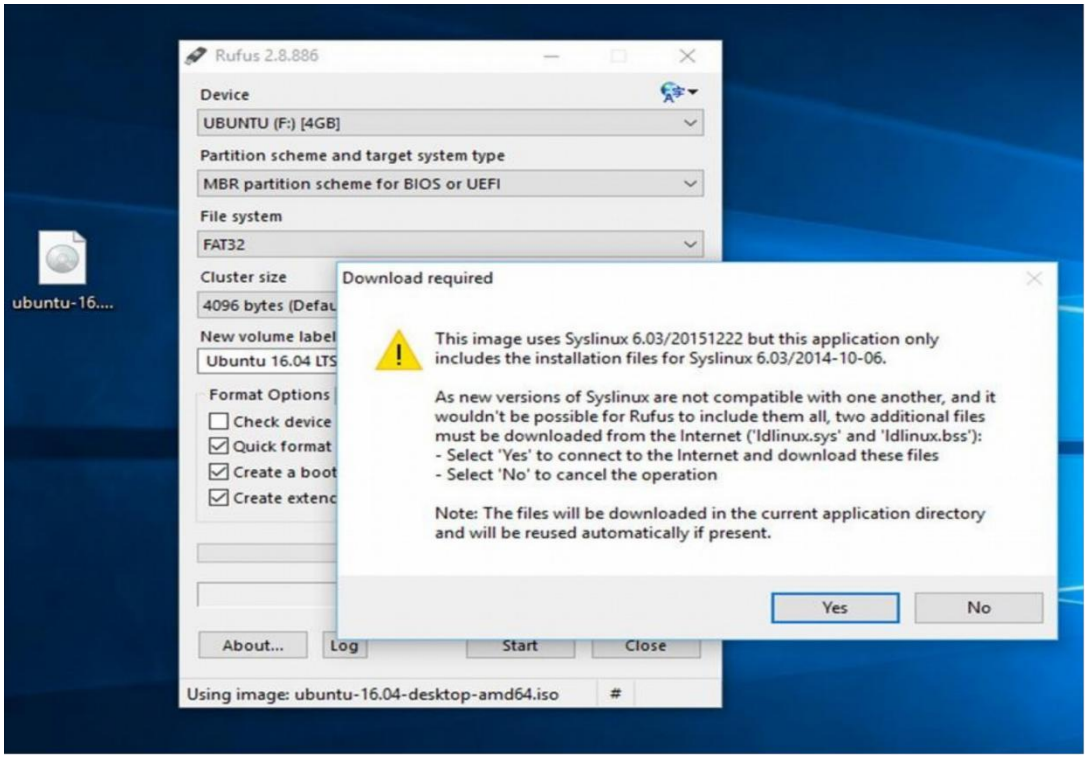
②、下载完以后，打开 Rufus 选择 USB 设备。

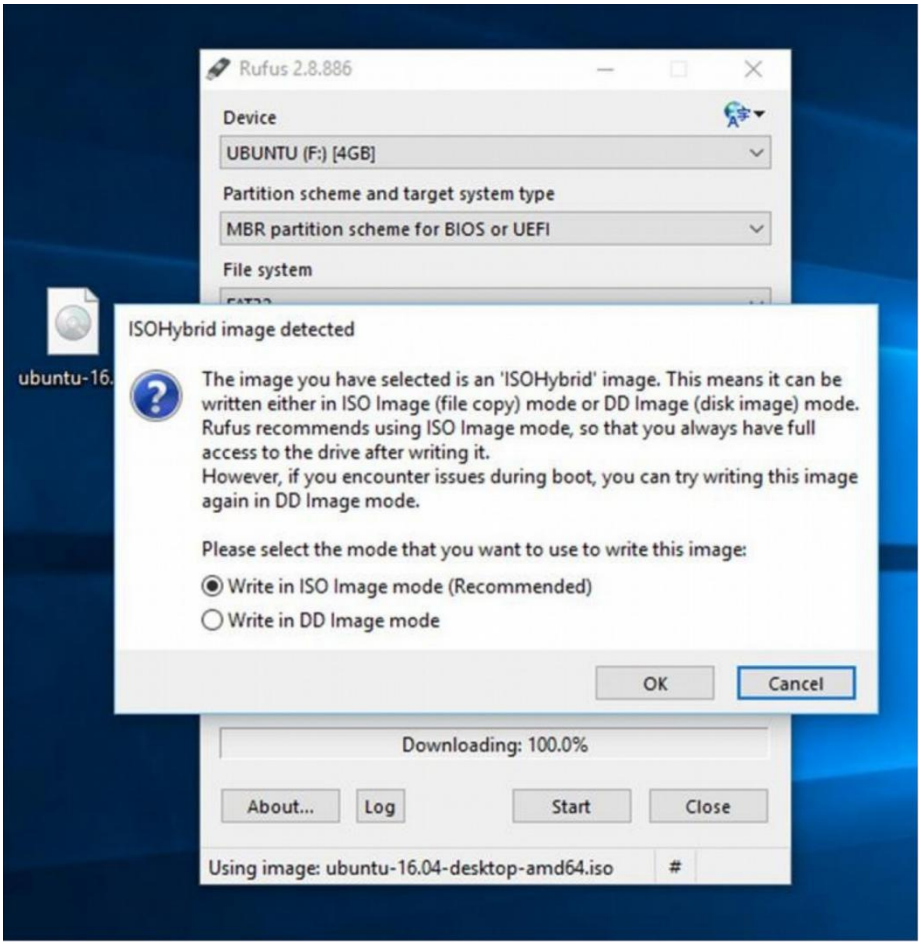


③、选择下载好的 Ubuntu16.04 系统镜像。



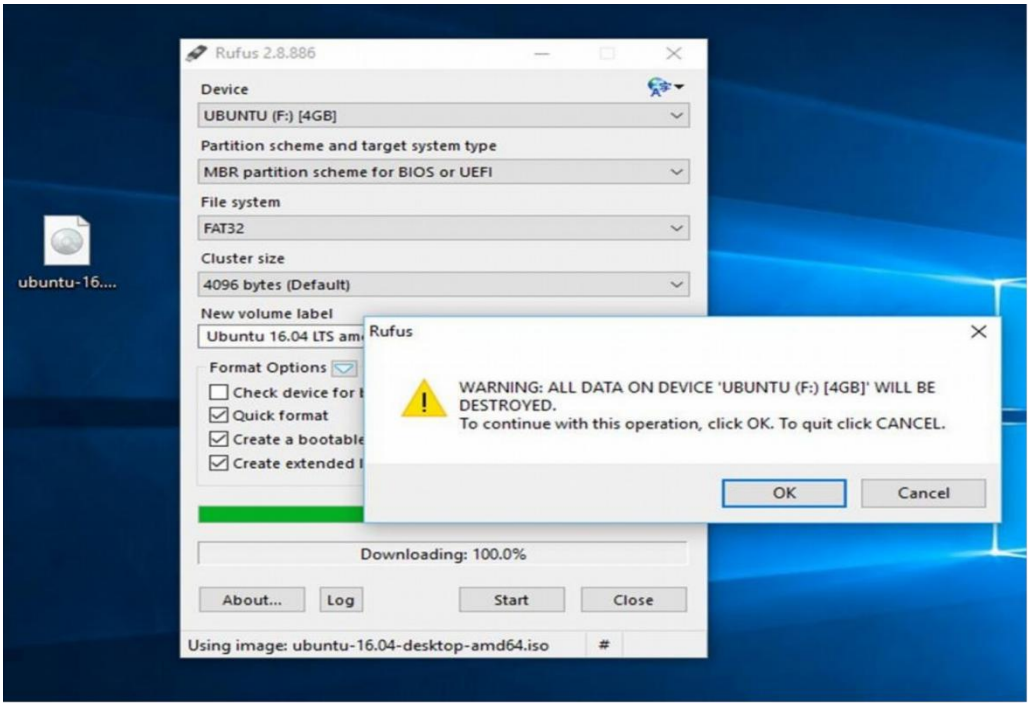
④、点击“Yes”，下载 Syslinux 软件。



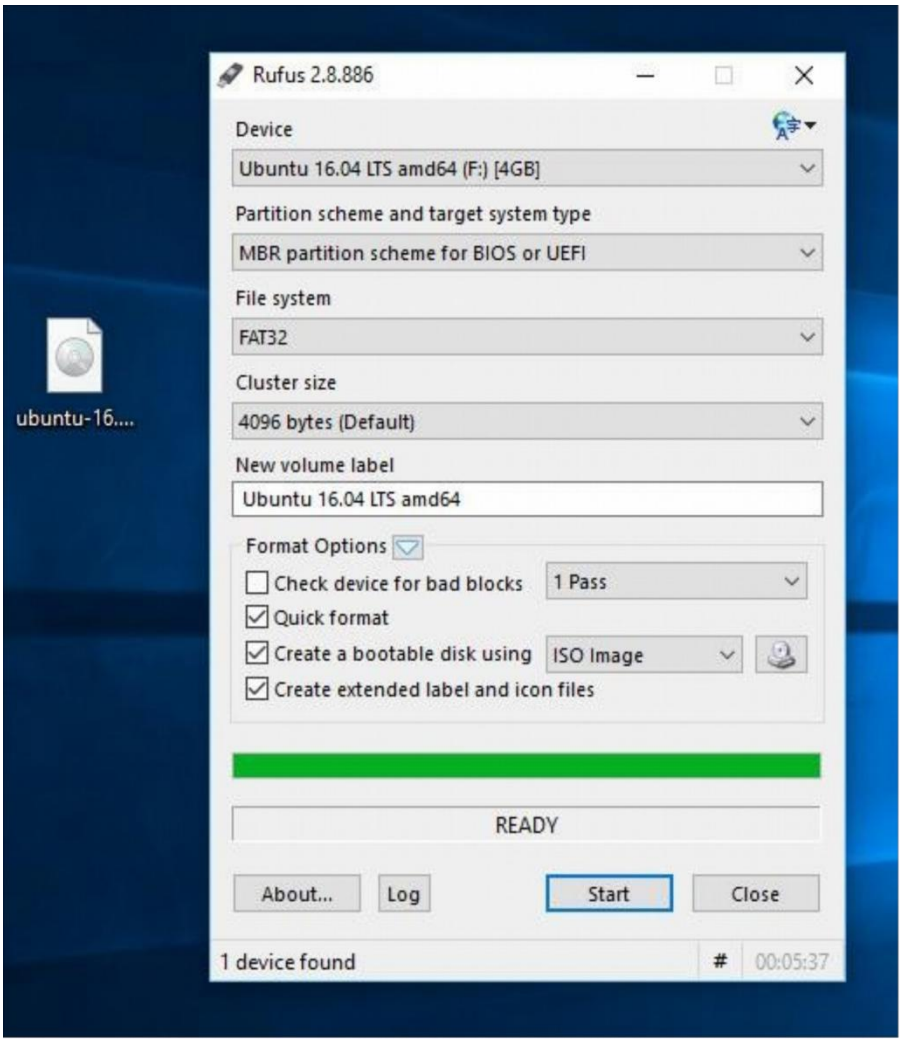


⑤、点击 “OK”，以 ISO 方式写入。

⑥、确认 USB 设备。



⑦、写入完成后，重启电脑，选择 U 盘启动完成系统安装。



2.1.3 参考网站

U 启动盘制作：

①、在 windows 系统下使用 Rufus 工具制作 Ubuntu16.04 系统 U 盘启动盘：

https://tutorials.ubuntu.com/tutorial/tutorial- create-a-usb-stick-on-windows#0

②、在 Mac OS 苹果电脑下使用 Etcher 工具制作 Ubuntu 系统 U 盘启动盘：

https://tutorials.ubuntu.com/tutorial/tutorial-create-a- usb-stick-on-macos#0

③、使用 Rufus 3.4 工具来制作 Ubuntu 系统 U 盘启动盘方法：

https://ywnz.com/linuxjc/3978.html

④、使用 Etcher 来制作 Ubuntu 系统启动盘的方法：

https://ywnz.com/linuxjc/3010.html

Ubuntu 系统安装：

①、启动 U 盘安装 Ubuntu 前的电脑 BIOS 设置方法：

https://ywnz.com/linuxjc/3803.htm

②、详细的 Ubuntu 系统安装图解教程：

https://ywnz.com/linuxaz/2588.html

③、使用 Rufus 工具 win10 与 Ubnutu16.04 双系统安装教程：

https://www.jianshu.com/p/e8fbf2aef5f2

④、使用 Rufus 工具 win10 与 Ubnutu16.04 双系统安装教程：

https://blog.csdn.net/auto1993/article/details/64965255

⑤、安装 Ubuntu 系统到自己电脑方法：

https://tutorials.ubuntu.com/tutorial/tutorial- install-ubuntu-desktop#0

⑥、Ubuntu 系统安装到自己电脑说明：

https://help.ubuntu.com/community/GraphicalInstall

2.2ROS 简介与安装

ROS（Robot Operating System）是一个开源的机器人操作系统，或者说是次级操作系统，需运行在机器操作系统之上（例如： Linux 系统），因此也认为是一类用于机器人的开源的元操作系统。ROS 提供库和工具来帮助软件开发人员创建机器人应用程序，为机器人软件开发者提供类似操作系统所提供的功能， 例如：硬件抽象描述、底层驱动程序管理、公共功能的执行、计算机视觉、进程间的信息传递、程序发行包管理，同时它也提供用于获取、编译、编写和跨计算机运行代码所需的工具和库函数。ROS 是根据开源 BSD 许可证授权的。

2.2.1 ROS 简介

ROS 是一种分布式的处理框架，这使得可执行文件能够被独立设计，并且在运行时松散耦合。ROS 可以分为两层，下层是上述的操作系统层，上层则是广大用户群贡献的能够实现不同功能的功能包，例如：机械臂运动规划、自主导航定位，传感器插件、仿真工具等等。

ROS 的主要特点为：

①、精简与集成：ROS 建立的系统具有模块化的特点，各模块中代码可以单独编译，而且编译使用的 Cmake 工具可以很容易的实现精简的理念。ROS 集成了很多现在已经存在的开源库，例如：OpenCV、OpenRAVE、Player 等。

②、支持多种编程语言：为满足不同编程者的需求，ROS 采用了语言中立性的框架结构，即不限于 某一种编程语言。ROS 支持语言包括 C++、Python、Octave 和 LISP 等。

③、免费并且开源：ROS 遵循 BSD 协议,对个人、商业应用以及修改都是免费的。

④、点对点设计：ROS 的点对点设计以及服务和节点管理器等机制可以分散由计算机视觉和语音识别等功能带来的实时计算压力。

⑤、2009 年 Willow Garage 开源 ROS 以来，ROS 已经发行了 10 多个版本。目前常用的是 Indigo 和 Kinetic，本手册选择 Indigo 和 Kinetic 版本作为 Baxter 的 ROS 系统进行配置和讲解。

2.2.2 ROS 版本

①、ROS indigo 仅支持 Saucy（Ubuntu 13.10）和 Trusty（Ubuntu 14.04）的 debian 软件包。

②、ROS kinetic 仅支持 Wily （Ubuntu 15.10） 、Xenial（Ubuntu 16.04）和 Jessie（Debian 8）的 debian 软件包。

③、我们已经编译好多个 Ubuntu 的平台下的 Debian 的软件包，安装方法如：

直接安装编译好的软件包比从源码编译安装更加高效，这也是我们在 Ubuntu 系统上的首选安装方式。

2.2.3 Ubuntu 系统中安装 ROS 的步骤和方法

第一步：配置 Ubuntu 软件库

（可跳过，直接从第二步开始）

配置你的 Ubuntu 软件仓库(repositories)以允许"restricted"、"universe"和"multiverse"这三种安装模式。 你可以按照下面网站完成配置。

https://help.ubuntu.com/community/Repositories/Ubuntu

第二步：添加 sources.list

目的是为了设置你的电脑可以从 Packages.ros.org 接收软件。

```
$ sudo sh -c'echo“ deb http://packages.ros.org/ros/ubuntu $ {lsb_release -sc）main”>
/etc/apt/sources.list.d/ros-latest.list'
```

同时你也可以选择性的安装国内的 mirrors 镜像源例如 UTSC， 这样速度比较快。

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

第三步：添加 keys

（如这步链接已失效可跳过，不影响后面）

```
$ sudo apt-key adv --keyserver 'hkp://keyserver.ubuntu.com:80' --recv-key
C1CF6E31E6BADE8868B172B4F42ED6FBAB17C654
```

如果您在连接到密钥服务器时遇到问题，可以尝试将上一个命令中“hkp://keyserver.ubuntu.com:80”替换“hkp://pgp.mit.edu: 80”或“hkp://keyserver.ubuntu.com:80”另外，您可以使用 curl 而不是 apt-key 命令，如果您位于代理服务器后面，则可能会有所帮助：

```
$ curl -sSL
'http://keyserver.ubuntu.com/pks/lookup?op=get&search=0xC1CF6E31E6BADE8868B172B
4F42ED6FBAB17C654' | sudo apt-key add -
```

第四步：安装 package

①、首先，确保您的 Debian 软件包索引是最新的：

```
$ sudo apt-get update
```

②、在 ROS 中，有很多不同的函数库和工具。我们为你提供了四种默认安装方式来帮助你学习，你也可以单独安装 ROS 包。

1、桌面完整版安装：**(推荐)**包含 ROS、rqt、rviz、通用机器人函数库、2D/3D 仿真器、导航以及 2D/3D 感知功能。

```
$ sudo apt-get install ros-indigo-desktop-full（14.04 系统）

$ sudo apt-get install ros-kinetic-desktop-full（16.04 系统）
```

2、桌面版安装：包含 ROS、rqt、rviz 以及通用机器人函数库。

```
$ sudo apt-get install ros-indigo-desktop（14.04 系统）

$ sudo apt-get install ros-kinetic-desktop（16.04 系统）
```

3、基础版安装：（简版）包含 ROS 核心软件包、构建工具以及通信相关的程序库，无 GUI 工具。

```
$ sudo apt-get install ros-indigo-ros-base（14.04 系统）

$ sudo apt-get install ros-kinetic-ros-base（16.04 系统）
```

4、单个软件包安装：你也可以安装某个指定的 ROS 软件包（使用软件包名称替换掉下面的 PACKAGE）。

```
$ sudo apt-get install ros-indigo-PACKAGE（14.04 系统）
```

```
$ sudo apt-get install ros-kinetic-PACKAGE（16.04 系统）
```

例如：

```
$ sudo apt-get install ros-indigo-slam-gmapping（14.04 系统）
$ sudo apt-get install ros-kinetic-slam-gmapping（16.04 系统）
```

5、要查找可用软件包,请运行：

```
$ apt-cache search ros-indigo（14.04 系统）
$ apt-cache search ros-Kinetic（16.04 系统）
```

第五步：初始化 rosdep

在开始使用 ROS 之前你还需要初始化 rosdep。rosdep 可以方便在你需要编译某些源码的时候为其安装一些系统依赖，同时也是某些 ROS 核心功能组件所必需用到的工具。

```
$ sudo rosdep init
$ rosdep update
```

第六步：配置 ROS 环境：

如果每次打开一个新的终端时 ROS 环境变量都能够自动配置好（即添加到 bash 的会话中）那会方便很多：

①、配置 Ubuntu14.04 系统 indigo 版本 ROS 环境方法：

```
1、$ echo "source /opt/ros/indigo/setup.bash" >> ~/.bashrc
2、$ source ~/.bashrc
```

②、配置 Ubuntu16.04 系统 kinetic 版本 ROS 环境方法：

```
1、$ echo "source /opt/ros/kinetic/setup.bash" >> ~/.bashrc
2、$ source ~/.bashrc
```

注意事项：

①、如果你安装有多个 ROS 版本，~/.bashrc 必须只能用当前使用版本所对应的 setup.bash。

②、如果你只想改变当前终端下的环境变量，可以执行以下命令：

```
$ source /opt/ros/indigo/setup.bash（14.04 系统）
$ source /opt/ros/kinetic/setup.bash（16.04 系统）
```

③、如果你使用 zsh，替换其中的 bash，你用以下命令来设置你的 shell：

```
$ echo "source /opt/ros/indigo/setup.bash" >> ~/.zshrc
$ source ~/.zshrc（14.04 系统）
$ echo "source /opt/ros/kinetic/setup.bash" >> ~/.zshrc
$ source ~/.zshrc（16.04 系统）
```

第七步：安装 rosinstall

到目前为止，你已经安装了运行核心 ROS 包所需的内容。为了创建和管理自己的 ROS 工作区，我们需要各种各样的工具，但这些是独立发布的，需要单独安装。例如：rosinstall，rosinstall 是 ROS 中一个独立分开的常用命令行工具，且经常使用，它使你能够通过一条命令就可以给某个 ROS 软件包下载很多源码树。

①、要安装这个工具和其他构建 ROS 包的依赖项，请运行：

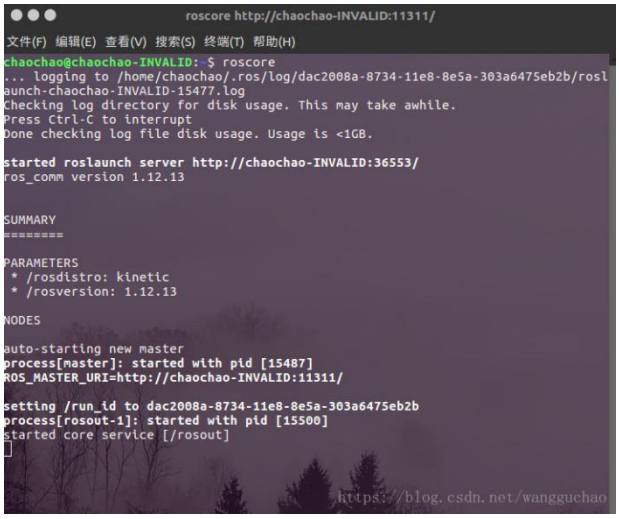
```
$ sudo apt-get install python-rosinstall（14.04 系统）

$ sudo apt-get install python-rosinstall python-rosinstall-generator python- wstool build-essential（16.04 系统）
```

②、至此，ROS 的安装基本完成，接下来测试 ROS 是否成功安装成功了。

第八步：测试 ROS 是否安装成功

打开终端运行面命令时会出现如图所示， 那么 ROS 就成功安装上了

```
$ roscore
```



2.2.4 运行 Demo

下面运行一个键盘控制海龟移动的 demo 来验证：

①、第一步，先安装示例 (备注：以 16.04 kinetic 为例)

```
$ sudo apt-get install ros-kinetic-turtlesim
```

②、第二步，打开一个终端测试 roscore 是否能正常启动，使用 ROS 必须 roscore 运行

```
$ roscore
```

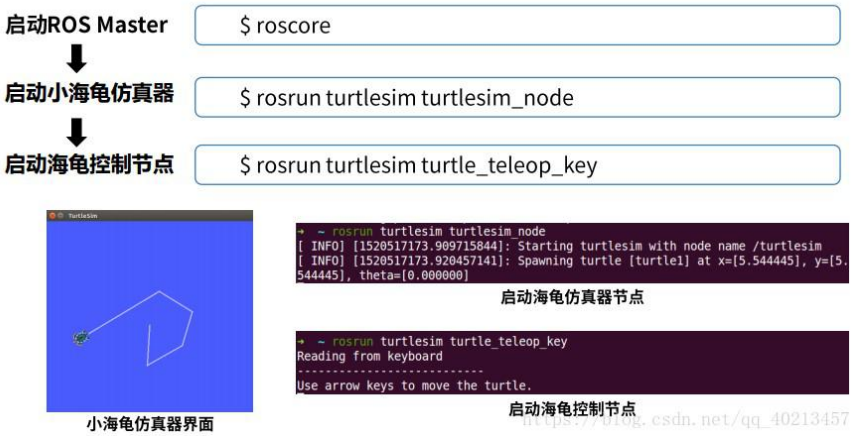
③、第三步，新开启一个终端，执行以下命令

```
$ rosrun turtlesim turtlesim_node
```

④、第四步，再开启一个终端，执行以下命令

```
$ rosrun turtlesim turtle_teleop_key
```

⑤、如下图所示：



2.2.5 卸载 ROS 的步骤方法

①、卸载全部 ROS

```
$ sudo apt-get autoremove --purge ros-*
```

②、或者卸载某个 ROS 版本(ROS 版本可以共存，每次需要切换)

```
$ sudo apt-get autoremove --purge ros-indigo（只卸载 Indigo）

$ sudo apt-get autoremove --purge ros-indigo（只卸载 Kinetic）
```

③、或者先卸载包

```
$ sudo apt-get purge ros-*
```

④、然后卸载依赖包、删除依赖，配置

```
$ sudo apt-get autoremove
```

⑤、检查 ~/.bashrc 以及 / opt/目录是否有 ROS 文件夹存在

2.3 Baxter SDK

安装完 Ubuntu（16.04 或者 14.04）系统和 ROS（ indigo 或者 kinetic）之后，我们将在工作站上安装 Baxter SDK，安装步骤如下：

2.3.1 所需硬件

1、电脑一台：

电脑的内存 4G 以上。

至少 20G 的可用硬盘空间。

2、USB 键盘一个。

3、路由器一个，网线两根。

2.3.2 第一步：创建 ROS 工作空间

```
$ mkdir -p ~/ros_ws/src
```

2.3.3 第二步：Source ROS

```
$ source /opt/ros/indigo/setup.bash  (14.04－ Indigo)
```

```
$ source /opt/ros/kinetic/setup.bash (16.04－Kinetic)
```

注意：在每次打开终端时，都先要运行上述命令才能运行 ros 相关命令，为了避免这一繁琐过程，可以事先在.bashrc 文件（该文件在当前系统的 Home 目录下）中添加这条命令。这样当你每次登录后，系统自动帮你执行这些命令配置好开发环境。

2.3.4 第三步：编译与安装

这一步要在创建的工作空间 ros_ws 中进行，所以先要改变工作目录路径，进入 ros_ws。

```
$ cd ros_ws/

$ catkin_make

$ catkin_make install
```

2.3.5 第四步：安装 SDK 依赖项

1）、ROS Ingido 版本（14.04－ Indigo）

```
$ sudo apt-get update

$ sudo apt-get install git-core python-argparse python-wstool python- vcstools python-rosdep ros-indigo-control-msgs ros-indigo-joystick-drivers
```

2）、ROS Kinetic 版本（16.04－Kinetic）

```
$ sudo apt-get update

$ sudo apt-get install git-core python-argparse python-wstool python- vcstools python-rosdep ros-kinetic-control-msgs ros-kinetic-joystick-drivers
```

2.3.6 第五步：安装 Baxter SDK

1、使用 wstool 空间工具，会自动在 Baxter Github 中检查所有需要的源，并安装在 ROS 工作空间（即 ros_ws）中。

```
$ cd ~/ros_ws/src/

$ wstool init

$ wstool merge https://raw.githubusercontent.com/RethinkRobotics/baxter/master/baxter_sdk.rosinstall

$ wstool update
```

这一过程安装时间比较长，如果电脑没有翻墙的话，很有可能安装不成功，所以进行此过程的时候最好先使用 VPN 连接！

2、Source ROS

```
$ source /opt/ros/indigo/setup.bash(14.04－ Indigo)

$ source /opt/ros/kinetic/setup.bash(16.04－Kinetic)
```

3、编译与安装

同样，这一步也需要在 ros_ws 路径下进行，因此先要进入 ros_ws 路径。

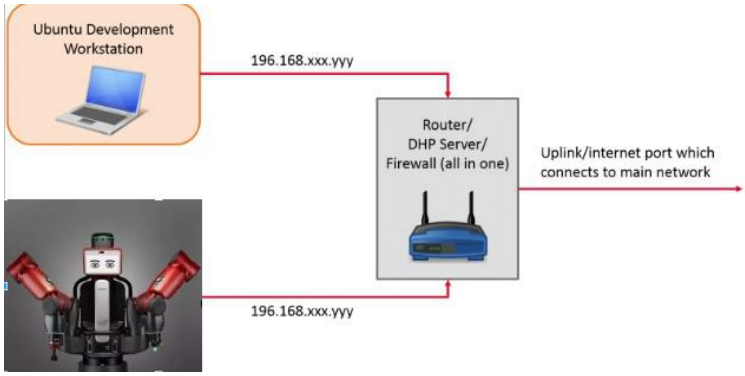
```
$ cd ~/ros_ws/

$ catkin_make
```

```
$ catkin_make install
```

2.4连接 Baxter

2.4.1 网络配置

如下所示，将 Baxter 机器人和电脑通过网线连接的方式链接到路由器局域网中，注意：请通过 Baxter 机器人电源线旁边的以太网端口连接到机器人。



Baxter 机器人网络连接参考网站：

http://sdk.rethinkrobotics.com/wiki/Networking

2.4.2 修改 Baxter.sh，配置 Baxter 通讯

1）、下载 Baxter.sh 脚本。

```
$ wget https://github.com/RethinkRobotics/baxter/raw/master/baxter.sh
$ chmod u+x baxter.sh
```

2）、请编辑 baxter.sh 脚本，进行必要的修改以描述您的开发 PC.

```
$ cd~/ros_ws/
$ gedit baxter.sh
```

3）、操作事项

①、把 your_ip="192.168.XXX.XXX"前面加一个#号

修改为：#your_ip="192.168.XXX.XXX"

②、去掉#your_hostname=" my_computer.local"前面的#号

修改为： your_hostname=" my_computer.local "

③、星号内为要修改的内容，robot_hostname 是控制箱的序列号

** robot_hostname=" robot_hostname.local" **

例如修改为：**robot_hostname="011510P0032.local"**

④、星号内为要修改的内容，your_hostname 是您的主机名。注意：您的主机名查找方法如下：

点击桌面最右上角的电源按键进入以后再点击关于这台计算器

进入这台机器人的详细信息界面如下图

设备名称就是您电脑的主机名

your_hostname=" your_hostname.local"

例如修改为：your_hostname=" cothink-Inspiron-5488.local"

4）、验证 ROS 版本

验证“ROS_version”字段是否与正在运行的 ROS 版本匹配，此字段默认应为 Ubuntu14.04 系统的 ROS“Indigo”版本，使用了 Ubuntu16.04 系统，对应 ROS “kinetic”版本的，请按下面方法修改：

ros_version="indigo"

修改为：**ros_version="kinetic"**

5）、请保存并关闭 baxter.sh 脚本。

6）初始化 SDK 运行环境：

每次打开新的终端，执行命令时，应该首先按下面步骤初始化 Baxter 机器人 SDK 运行环境

```
$ cd ~/ros_ws/
$ ./baxter.sh
```

2.4.3 查看、验证 SDK 安装环境

1、查看和验证 SDK 环境设置的命令是：

```
$ env | grep ROS
```

2、此时终端里面出现的字段是：

ROS_MASTER_URI –机器人的主机名。

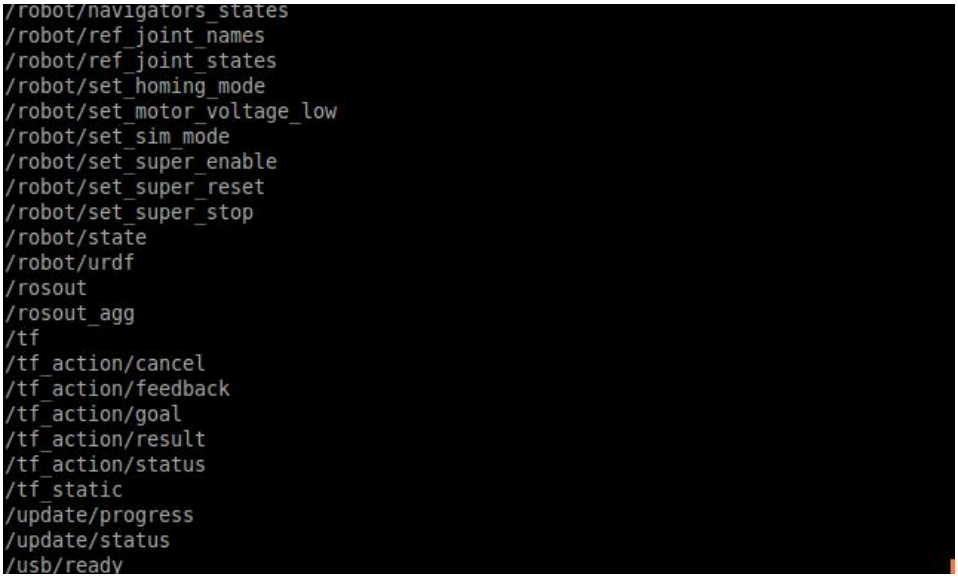
ROS_IP –工作站 IP（如果没有使用 ROS_HOSTNAME 模式这里应该包含工作站的 IP 地址，否则此项不可见）。

ROS_HOSTNAME –PC 的主机名（hostname）（如果没有使用 ROS_IP 模式， 这里应该包含你的 PC 的主机名（hostname），否则此项不可见）。

3、你可以尝试通过终端输入以下命令从机器人获取 rostopic 列表：

```
$ rostopic list
```

您可以从命令行输出中看到 rostopic 列表，如下所示：。



2.5 Hello Baxter

设计第一个 Baxter 的 ros 程序(以 14.04 系统 Ros Indigo 版本为例)。

（1）设置 ROS 环境

如果之前设置过 ROS 环境，那么在~/ros_ws 文件目录下应该会有一个 devel 文件夹，如果已经有了，可以跳过下面的代码（当然重复运行一次下面的代码也是没问题的），如果还没有此文件夹，那必须运行一遍下面的代码：

```
# Move to root of our catkin workspace
$ cd ~/ros_ws
$ source /opt/ros/indigo/setup.bash
$ catkin_make
```

```
# 如果是 16.04 系统，ROS 版本为 Kinetic，则执行以下代码：
$ cd ~/ros_ws
$ source /opt/ros/kinetic/setup.bash
$ catkin_make
```

接下来再 Source 一下 ROS 环境设置脚本，也就是 baxter.sh

```
# Source baxter.sh script
$ ./baxter.sh
```

（2）检查 ROS 连接

这一步，我们会检查 Baxter 机器人与我们的 PC 通信是否正常，也是我们用 PC 去控制 Baxter 的关键

（3）检查 PC 与 ROS Master（Baxter）的连接

这一步是检查 PC 能否给 ROS Master 发送数据。

还记得前文设置的 baxter_hostname 么？我们只要检查下是否能 ping 通即可。代码如下：

```
$ ping 011506P0014.local
```

如果显示有数据通信，就说明与 ROS Master 的连接正常，继续下一步。

（4）检查 Baxter 与开发工作空间 PC 的连接

这一步是想检查 PC 能否收到 ROS Master（Baxter）发过去的命令。

首先用 SSH 远程登录 ROS Master，在 ROS Master 上 ping 我们的 PC 机，看是否有通信。代码如下：

```
$ ssh ruser@<our ROS Master>
# 密码:
rethink
# 例子:
$ ssh ruser@011506P0014.local
#现在我们 SSH 登录 到了机器人，我们可以验证往回 Ping 开发 PC 是否可行
ruser@011506P0014.local:~$ ping <ROS_IP/ROS_HOSTNAME>
# 例子:
# ROS_IP
ruser@011506P0014.local:~$ ping 192.168.1.102
或者
ruser@011506P0014.local:~$ ping cothinkTab.local
```

如果能 ping 通，说明连接正常，可以退出远程登录了。

```
ruser@011506P0014.local:~$ exit
```

（5）查看 ROS Topic

如果双向通信没有问题了，我们就可以查看目前正在发生的 ROS Topic 了（关于 ROS Topic，请查看 ros 操作系统教程）

```
$ rostopic list
```

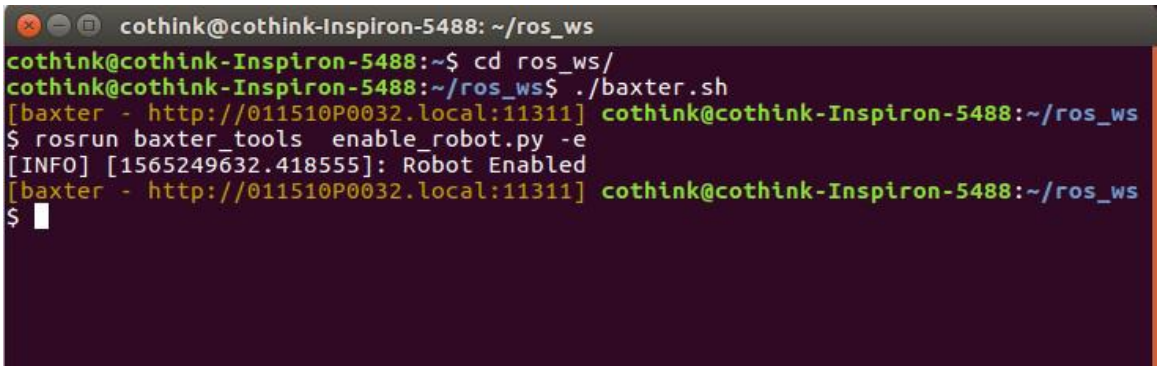
以上代码可以查看所有的 ROS Topic，如果要查看特定的 Topic，示例如下：

```
$ rostopic echo /robot/joint_states
```

（6）使能 Baxter

操作 Baxter 机器人之前，先要使能机器人。在 baxter_tools SDK 工具包中，提供了“使能 enable/反使能 disable/重置 reset/停止 stop”机器人的 API。这在急停按钮（E-STOP）按下后是必需执行的。 必须使能 Baxter 才能主动控制任何电机。使能机器人的代码如下：

```
$ rosrun baxter_tools enable_robot.py -e
```



如果输出的结果显示 Robot Enabled 表示已经成功使能 Baxter 机器人。

（7）运行一个示例程序

官方网站给我们提供了许多使用 Baxter_interface 软件包的 Baxter 示例程序，该软件包包含用于 Baxter Research Robot 开发的 Python 模块。我们来运行 joint_velocity_wobbler.py 看看效果，更多关于这个示例的信息，可以查看 Joint Velocity Wobbler Example Page

参考网址：(http://sdk.rethinkrobotics.com/wiki/Wobbler_Example)。

```
$ rosrun baxter_examples joint_velocity_wobbler.py
```

（8）自己编写一个程序来运行 Baxter

大多数编程语言的第一示例程序都是 hello world 之类的，本程序也不例外，我们将编写一个 Hello Baxter 程序，这个程序实现的功能是让 Baxter 机器人向我们挥手。

首先，当然还是使能 Baxter 机器人：

```
$ rosrun baxter_tools enable_robot.py -e
```

接着打开 gedit 编辑器，输入一下代码，并保存为 hello_baxter.py。

```
# Import the necessary Python modules
# rospy - ROS Python API
import rospy
# baxter_interface - Baxter Python API
import baxter_interface
# initialize our ROS node, registering it with the Master
rospy.init_node('Hello_Baxter')
# create an instance of baxter_interface's Limb class
limb = baxter_interface.Limb('right')
# get the right limb's current joint angles
angles = limb.joint_angles()
# print the current joint angles
print angles
# reassign new joint angles (all zeros) which we will later command to the limb
angles['right_s0']=0.0
angles['right_s1']=0.0
angles['right_e0']=0.0
angles['right_e1']=0.0
angles['right_w0']=0.0
angles['right_w1']=0.0
```

```
    angles['right_w2']=0.0
    # print the joint angle command
    print angles
    # move the right arm to those joint angles
    limb.move_to_joint_positions(angles)
    # Baxter wants to say hello, let's wave the arm
    # store the first wave position
    wave_1 = {'right_s0': -0.459, 'right_s1': -0.202, 'right_e0': 1.807, 'right_e1': 1.714, 'right_w0': -0.906, 'right_w1': -1.545, 'right_w2': -0.276}
    # store the second wave position
    wave_2 = {'right_s0': -0.395, 'right_s1': -0.202, 'right_e0': 1.831, 'right_e1': 1.981, 'right_w0': -1.979, 'right_w1': -1.100, 'right_w2': -0.448}
    # wave three times
    for _move in range(3):
        limb.move_to_joint_positions(wave_1)
        limb.move_to_joint_positions(wave_2)
    # quit
    quit()
```

在 Terminal 中运行编辑好的 python 程序：

```
$ python hello_baxter.py
```

是否看到 Baxter 在向你挥手？恭喜你，你已经完成了控制 Baxter 的第一个程序！

（9）参考网站

1）、Baxter 机器人设置参考网站：

http://sdk.rethinkrobotics.com/wiki/Baxter_Setup

2）、Baxter 机器人工作站设置参考网站：

http://sdk.rethinkrobotics.com/wiki/Workstation_Setup

3）、Baxterr 机器人网络设置参考网站：

http://sdk.rethinkrobotics.com/wiki/Networking

4）、Baxter 机器人操作参考网站：

http://sdk.rethinkrobotics.com/wiki/Hello_Baxter

5）、Baxter 机器人工作区范围参考网站：

http://sdk.rethinkrobotics.com/wiki/Workspace_Guidelines

6）、Baxter 机器人工作站系统要求参考网站：

http://sdk.rethinkrobotics.com/wiki/System_require