Ubuntu install of ROS Kinetic

Installation

ROS Kinetic ONLY supports Wily (Ubuntu 15.10), Xenial (Ubuntu 16.04) and Jessie (Debian 8) for debian packages.

Configure your Ubuntu repositories

Configure your Ubuntu repositories to allow "restricted," "universe," and "multiverse." You can follow the Ubuntu guide for instructions on doing this.

Setup your sources.list

Setup your computer to accept software from packages.ros.org.

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

Set up your keys

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

If you experience issues connecting to the keyserver, you can try substituting hkp://pgp.mit.edu:80 or hkp://keyserver.ubuntu.com:80 in the previous command.

Installation

First, make sure your Debian package index is up-to-date:

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

There are many different libraries and tools in ROS. We provided four default configurations to get you started. You can also install ROS packages individually.

In case of problems with the next step, you can use following repositories instead of the ones mentioned above ros-shadow-fixed

- Desktop-Full Install: (Recommended) : ROS, rqt, rviz, robot-generic libraries, 2D/3D simulators, navigation and 2D/3D perception

  -
    ```
    sudo apt-get install ros-kinetic-desktop-full
    ```
  - or click here

- Desktop Install: ROS, rqt, rviz, and robot-generic libraries

    -
      ```
      sudo apt-get install ros-kinetic-desktop
      ```
    - or click here
- ROS-Base: (Bare Bones) ROS package, build, and communication libraries. No GUI tools.

    -
      ```
      sudo apt-get install ros-kinetic-ros-base
      ```
    - or click here
- Individual Package: You can also install a specific ROS package (replace underscores with dashes of the package name):

    -
      ```
      sudo apt-get install ros-kinetic-PACKAGE
      ```
    - e.g.

    -
      ```
      sudo apt-get install ros-kinetic-slam-gmapping
      ```

To find available packages, use:

```
apt-cache search ros-kinetic
```

Initialize rosdep

Before you can use ROS, you will need to initialize rosdep. rosdep enables you to easily install system dependencies for source you want to compile and is required to run some core components in ROS.

```
sudo rosdep init
rosdep update
```

Environment setup

It's convenient if the ROS environment variables are automatically added to your bash session every time a new shell is launched:

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

If you have more than one ROS distribution installed, ~/.bashrc must only source the setup.bash for the version you are currently using.

If you just want to change the environment of your current shell, instead of the above you can type:

```
source /opt/ros/kinetic/setup.bash
```

If you use zsh instead of bash you need to run the following commands to set up your shell:

```
echo "source /opt/ros/kinetic/setup.zsh" >> ~/.zshrc
source ~/.zshrc
```

Dependencies for building packages

Up to now you have installed what you need to run the core ROS packages. To create and manage your own ROS workspaces, there are various tools and requirements that are distributed separately. For example, rosinstall is a frequently used command-line tool that enables you to easily download many source trees for ROS packages with one command.

To install this tool and other dependencies for building ROS packages, run:

```
sudo apt-get install python-rosinstall python-rosinstall-generator python-wstool build-essential
```

Ubuntu

Install Gazebo using Ubuntu packages

Default installation: one-liner

1. Install
2. curl -ssL http://get.gazebosim.org | sh
3. Run
4. gazebo

Install gazebo_ros_pkgs

Choose the method you would prefer. The easier and faster is installing it from packages but installing from source means you can more easily debug and submit bug patches ;-)

A. Install Pre-Built Debians

The gazebo_ros_pkgs packages are available in:

- ROS Kinetic:

```
sudo apt-get install libsdformat4
sudo apt-get install libgazebo7
```

```
sudo apt-get install ros-kinetic-gazebo-ros
```

```
sudo apt-get install ros-kinetic-gazebo-ros-pkgs ros-kinetic-gazebo-ros-control
```

Turtlebot Installation

ISO Installation

This instruction is for users with a netbook with no operating system.

Download ISO & Create Startup USB

Download TurtleBot ISO with the following link and create startup USB disk using Startup Disk Creator or similar tools.

- ubuntu-14.04.2-desktop-amd64-turtlebot-RC0.iso, sha512sum

Install Ubuntu with TurtleBot ISO

Boot from turtlebot ISO usb and follow normal ubuntu installer guide. This will also install turtlebot packages with fancy turtlebot background image.

Debs Installation

These instructions are intended for users with a netbook pre-installed with Ubuntu Trusty.

Ubuntu Repo Setup

Follow the generic Ubuntu instructions for a ros-indigo-desktop-full installation.

Ubuntu Package Install

In addition, you will need to install the following debs for TurtleBot (please update this if you find any errors):

```
sudo apt-get install ros-kinetic-turtlebot
```

```
sudo apt-get install ros-kinetic-turtlebot-apps
sudo apt-get install ros-kinetic-turtlebot-interactions

sudo apt-get install ros-kinetic-turtlebot-simulator

sudo apt-get install ros-kinetic-kobuki-ftdi

sudo apt-get install ros-kinetic-ar-track-alvar-msgs
```

Source Installation

This is for people who'd like to hack on the turtlebot sources and contribute. For more information, please ask us for more information at <ros-sig-turtlebot AT googlegroups DOT com>.

Preparation

```
sudo apt-get install python-rosdep python-wstool ros-kinetic-ros
sudo rosdep update
```

Workspaces

We are using chained workspaces here, but you could just as simply merge them into one.

```
> mkdir ~/rocon
> cd ~/rocon
> wstool init -j5 src https://raw.github.com/robotics-in-concert/rocon/release/indigo/rocon.rosinstall
> source /opt/ros/indigo/setup.bash
> rosdep install --from-paths src -i -y
> catkin_make

> mkdir ~/kobuki
> cd ~/kobuki
> wstool init src -j5 https://raw.github.com/yujinrobot/yujin_tools/master/rosinstalls/indigo/kobuki.rosinstall
> source ~/rocon/devel/setup.bash
> rosdep install --from-paths src -i -y
> catkin_make

> mkdir ~/turtlebot
> cd ~/turtlebot
> wstool init src -j5 https://raw.github.com/yujinrobot/yujin_tools/master/rosinstalls/indigo/turtlebot.rosinstall
> source ~/kobuki/devel/setup.bash
> rosdep install --from-paths src -i -y
> catkin_make
```

Explore the Gazebo world

Description: Cruise around in the Gazebo world and use RViz to "see" what's in it.

Keywords: keyop, rviz

Tutorial Level: BEGINNER

Next Tutorial: Make a map and navigate with it

Bringup Turtlebot, see what the robot senses, teleoperate in Gazebo simulation.

Contents

1. Overview
2. Preparations
3. Make the TurtleBot move
4. See what the robot sees
5. Notes
6. What Next?

Overview

In this tutorial we will use the keyboard teleop tool from the turtlebot_teleop package to drive the TurtleBot around and use RViz to visualise the sensor data.

Preparations

First we need to install the turtlebot_teleop package. Since we will use other apps in the later tutorials, we will install all TurtleBot apps already.

For convenience we will additionally install a set of RViz launchers, which will bring up RViz already configured for our use case.

```
$ sudo apt-get install ros-kinetic-turtlebot-apps ros-kinetic-turtlebot-rviz-launchers
```

Make the TurtleBot move

First, bring up the TurtleBot simulation as described in the Gazebo Bringup Guide.

Next, in a second terminal* run the keyboard teleop tool.

```
$ roslaunch turtlebot_teleop keyboard_teleop.launch
```

For TurtleBot 2 you can also use the kobuki_keyop tool:

```
$ roslaunch kobuki_keyop keyop.launch
```

If you prefer other methods to move the robot, be sure to use geometry_msgs/Twist topic by the name /mobile_base/commands/velocity.

See what the robot sees

Use RViz to visualise various sensory information. In a new terminal* execute:

```
$ roslaunch turtlebot_rviz_launchers view_robot.launch
```

RViz won't show much at first, since the Gazebo world is empty. So, let's add some objects to it!

En este link estan los ejecutables para simular:
http://sauravag.com/2016/10/how-to-setup-turtlebot-simulator-in-ros-with-gazebo/