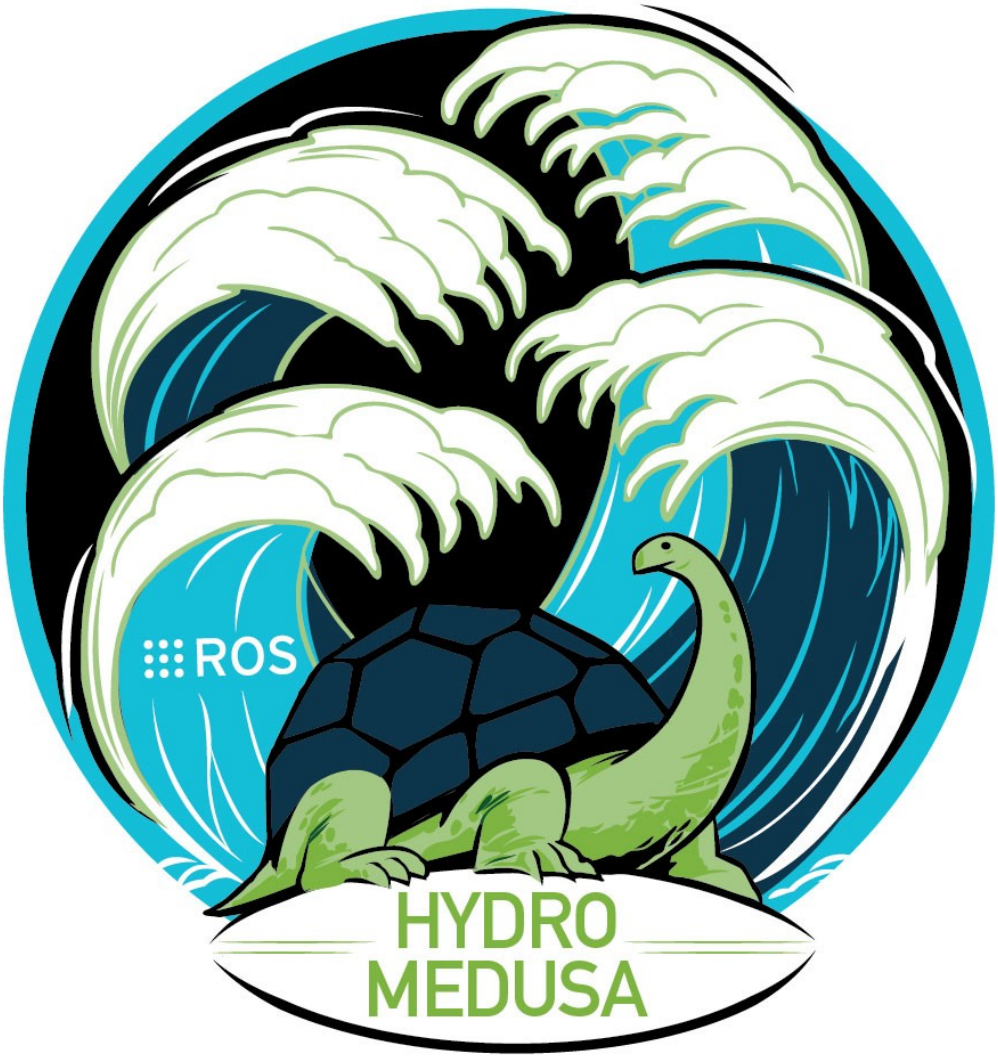

Tutorial 1

Introduction to ROS

Ralph SEULIN

Le2i UMR CNRS 6306 – http://le2i.cnrs.fr
ralph.seulin@u-bourgogne.fr

v1.3 - September 2015

1- Introduction

This tutorial is mainly inspired from the workshop of GDR Robotique @Montpellier – July 2013.

Special thanks to Stéphane Magnenat & Francis Colas – Autonomous Systems Lab. ETH Zurich

1.1- References

- ROS Wiki - http://wiki.ros.org/
- ROS Tutorials - http://wiki.ros.org/ROS/Tutorials
- Suggested Courses – Summer course on ROS Framework 2013

  http://mediawiki.isr.ist.utl.pt/wiki/Summer_course_on_ROS_framework_2013
- Support - http://answers.ros.org

1.2- Tips

- **ROS Distribution version**

  ROS Hydro is the only distribution for TurtleBot2 development during Robotics Pojects

  !!! **Always select Hydro in tutorials or documentation on the ROS wiki !!!**

  

- **Buildsystem**

  A buildsystem generates executable files from source files. Catkin is the only buildsystem used for new development. (Unless you need to compile "old fashioned" packages with rosbuild)

  !!! **Always select catkin in tutorials or documentation on the ROS wiki !!!**

  

  Consult the ROS Wiki for further information: http://wiki.ros.org/catkin_or_rosbuild

2- ROS Configuration

We already performed some settings on Workstations and TurtleBot2 Netbooks. Every workstations and netbooks are already configured with ROS HYDRO. Workspaces are configured to have catkin and rosbuild working alongside.

More infos on catkin: http://wiki.ros.org/catkin/Tutorials

The workspaces are :

- for catkin ~/ros/hydro/catkin_ws
- for rosbuild ~/ros/hydro/hydro_workspace

After ROS installation, the last step is to configure the ROS environment. The **~/.bashrc** file has to be configured by adding lines to run correctly ROS commands. These lines launch a script that sets environment variables.

2.1- To Do

- Study tutorial on ROS environment

  !!! **ROS environment is already set on the computer: do not create again a workspace !!!**

  http://wiki.ros.org/ROS/Tutorials/InstallingandConfiguringROSEnvironment

- Check environment variables that contain “ROS” value

  $ export | grep ROS

- Study the ~/.bashrc file

  To edit the ~/.bashrc file

  $ gedit ~/.bashrc

- Study documentation on environment variables:

  http://www.ros.org/wiki/ROS/EnvironmentVariables (the first two sections in particular)

2.2- Additional Informations

Several versions of ROS can eventually coexist (GROOVY & HYDRO for example) and be selected using scripts (be careful not to mix the ROS distributions).

Example of ROS distribution selection in the ~/.bashrc file:

```
#You can switch between GROOVY & HYDRO by activating one set of source and select ROS_DISTRO=xxxx

#Example with GROOVY active and HYDRO disabled

##ROS GROOVY

ROS_DISTRO=groovy

source /opt/ros/groovy/setup.bash

##ROS HYDRO

#ROS_DISTRO=hydro

#source /opt/ros/hydro/setup.bash
```

- ROS packages are searched and detected only from ROS_PACKAGE_PATH value, which often involves adding a package path to this variable in the ~/.bashrc file.
- Modifications in the ~/.bashrc file are taken into account by doing the following :
  - close the terminal
  - reopen a new shell (terminal window)

3- Exploration

ROS provides useful command line tools.

3.1- To Do

- Use *rospack* to obtain the list of packages and the dependences of *roscpp*, *sensor_msgs*, *gmapping packages*.
- Use *roscd* to quickly navigate from one package to another one.
- Use *rosmsg* to obtain the list of messages and find description of the following messages: *sensor_msgs/Joy*, *sensor_msgs/LaserScan*, *nav_msgs/Odom*...
- Study messages documentation on the package website (example with http://wiki.ros.org/sensor_msgs)
- Study general information on messages on the ROS wiki website - http://wiki.ros.org/msg

3.2- Informations

- Messages are defined in the *msg* folder in a package. Their definition has to be compiled to make them useable.
- Messages documentation provided by a package is described on the package website.

4- Bags

ROS provides a powerful tool to record messages in order to replay them later. *rosbag* tool is able to record and play a full working ROS sequence.

4.1- To Do

- Study available *rosbag* commands
- Extract the following informations on the provided bag files:
  - duration, start and end time
  - messages types and amount
- Replay one bag file.
- While a bag file is being replayed, record a new bag file with only selected topics
- Use and study the *rqt_bag* tool (*rqt_bag* supersedes *rxbag*, which is deprecated since ROS GROOVY)

5- Topics and nodes

ROS provides also two tools to analyse working nodes and used topics (published or suscribed).

5.1- To Do

- While a bag file is replayed, use *rosnode* to get the list of working nodes and related informations.
- Use *rosnode ping* command to test nodes.
- Kill */rosbag* then */rosout*.
- While a bag file is replayed on the one hand, and on the other, some topics are recorded , use *rqt_graph* to study ROS computation graph (*rqt_graph* supersedes *rxgraph*, which is deprecated since ROS groovy).

5.2- Additional informations

- */rosout* is part of the *roscore* that agregates log messages. It is restarted automatically if it is killed.

6- Visualisation

ROS provides visualisation tools for standard data analysis:

rqt_plot - http://wiki.ros.org/rqt_plot

rviz - http://wiki.ros.org/rviz

6.1- To Do

- Use simulation time instead of computer's system clock – http://wiki.ros.org/Clock
- Play the bag file *summit_outdoor.bag* with forcing the clock to be published
- Use *rostopic* to display the “x” value of the robot position in the */odom* message
- With the same bag, use *rqt_plot* to display the same value, try to display several value in the same time. Select *QwtPlot* in the plot options (*rqt_plot* supersedes *rxplot*, which is deprecated since ROS groovy).
- Start *rviz*, from the rviz package (use *rosrun* command) to visualize the following topics:

  /tf

  /scan
- Study messages that *rviz* can display and related options. Study rviz menus.

6.2- Advanced

- Check global options of rviz – study “Fixed Frame”.
- Save configuration file. Set a *default config.* file.

Informations

- Check out other powerful tools based on *rqt* - a Qt-based framework for GUI development for ROS :
  - http://wiki.ros.org/rqt
  - http://wiki.ros.org/rqt/Plugins

7- Useful Command Line tips

Locate a file

$ locate file.txt

Edit a file

With a GUI :

$ sudo gedit file.txt

$ sudo geany file.txt

Without a GUI (during SSH)

$ sudo nano file.txt

Remote Terminal

$ ssh login@IP (example: ssh turtlebot@192.168.0.100)

Remote Terminal with X display

$ ssh -X login@IP (example: ssh -X turtlebot@192.168.0.100)