Module 1 - ROS

Lesson Objectives:

1. Learn fundamental concepts of ROS.
2. Develop basic operational understanding of ROS through application.
3. Design, build, and test a basic chat client using ROS.

Agenda:

1. ROS Introduction.
2. ROS Jupyter Notebook.
3. ICE1 Jupyter Notebook.

1 ROS Introduction.

Robotics Operating System (`https://www.ros.org/about-ros/`**):**

The Robot Operating System (ROS) is a flexible framework for writing robot software. It is a collection of tools, libraries, and conventions that aim to simplify the task of creating complex and robust robot behavior across a wide variety of robotic platforms. ROS is sometimes called a meta operating system because it performs many functions of an operating system, but it requires a computer's operating system such as Linux.

Why? Because creating truly robust, general-purpose robot software is hard. From the robot's perspective, problems that seem trivial to humans often vary wildly between instances of tasks and environments. Dealing with these variations is so hard that no single individual, laboratory, or institution can hope to do it on their own.

As a result, ROS was built from the ground up to encourage collaborative robotics software development. For example, one laboratory might have experts in mapping indoor environments, and could contribute a world-class system for producing maps. Another group might have experts at using maps to navigate, and yet another group might have discovered a computer vision approach that works well for recognizing small objects in clutter. ROS was designed specifically for groups like these to collaborate and build upon each other's work, as is described throughout this site.

Noetic Ninjemys (`http://wiki.ros.org/noetic`**):**

Catkin (`http://wiki.ros.org/catkin`**):**

Master (`http://wiki.ros.org/Master`**):**

Packages (`http://wiki.ros.org/Packages`**):**

Package.xml:

Nodes (`http://wiki.ros.org/Nodes`)**:**

Topics (`http://wiki.ros.org/Topics`)**:**

Messages (`http://wiki.ros.org/Messages`)**:**

Publisher and Subscriber:

Services: (`http://wiki.ros.org/Services`)

2 ROS Jupyter Notebook.

We will use a Jupyter Notebook to practice and provide more insight into the terms above.

1. On the master, open the Jupyter Notebook server:

```
dfec@master:~$ roscd usafabot_curriculum/Module1_ROS
dfec@master:~$ jupyter-notebook
```

2. Open the ROS Jupyter Notebook, "Module1_ROS.ipynb", and follow the instructions within the notebook.

3 ICE1 Jupyter Notebook.

The ICE1 Jupyter Notebook will guide you through implementation of a chat client using ROS.

1. On the master, open the Jupyter Notebook server (if it is not already open):

```
    dfec@master:~$ roscd usafabot_curriculum/Module1_ROS
    dfec@master:~$ jupyter-notebook
```

2. Open the ICE1 Jupyter Notebook, "ICE1_Talker.ipynb" and follow the instructions within the notebook.

Checkpoint. Take a screenshot or show the instructor the following:

1. List of running topics.
2. The *rqt_graph* for the nodes and topics currently running.
3. An echo of messages sent via the chat topic.
4. Show what type of message is sent over the chat topic (Hint: use the rostopic command).

4 Assignments.

□ (Before lesson 3) Go to `http://wiki.ros.org/ROS/Tutorials` on your master and work through ROS tutorials 2-16 on the master (remember, you only need to look at Python tutorials). This shouldn't take more than two hours.

5 Next time.

- Lesson 3: Quiz on ROS
- Lesson 4: Module 2 - Linux for ROS