ROS Primer

Author: Sterling McLeod

December 7, 2016

Introduction

The purpose of this assignment is for you to become familiar with the Robot Operating System (ROS) framework. You will go through basic ROS concepts and implement two very useful features of the framework, publishing and subscribing. All ROS packages must be located in a ROS workspace. The workspace allows you to compile many ROS packages with one command and makes locating packages with command-line tools much easier. The build command for each ROS package is configured using the cmake tool - each package has a corresponding CMakeLists.txt file that you will need to edit to build the package. After successfully building and running a ”Hello World” ROS program (also called a ROS node), you will implement nodes that communicate with each other using the Publish/Subscribe pattern of communication.

System Info

Operating System, ROS Version: Ubuntu 14.04, Indigo
Extra Hardware needed: None

Supplemental Material

Visit http://wiki.ros.org/ROS/Tutorials for a list of tutorials on ROS. For this assignment, the following tutorials will be most relevant:

1. Installing and Configuring Your ROS Environment
2. Creating a ROS Package
3. Building a ROS package
4. Understanding ROS Nodes
5. Understanding ROS Topics
6. Writing a Simple Publisher and Subscriber (C++)
7. Writing a Simple Publisher and Subscriber (Python)

Tasks

Task 1. Create a ROS workspace.

The workspace should be in the user's home directory.

Task 2. Create a ROS package in your ROS workspace.

1. Name the package "ros_practice"
2. Edit the CMakeLists.txt and package.xml files for ros_practice to add the following dependencies:
   (a) roscpp
   (b) rospy
   (c) std_msgs
   (d) nav_msgs
3. Create a main.cpp file that does the following:
   (a) Creates a ROS node titled "Test Node"
   (b) Prints "Hello World!" to the console
4. Edit CMakeLists.txt to compile your main.cpp into an executable named "my_first_node"
5. Compile your ROS node
6. Run your ROS node using the *rosrun* command

Task 3. Create a custom msg in your ROS package.

1. The name of the msg should be MyMsg.msg. It should contain the following attributes:
   (a) ID - integer value unique to each MyMsg
   (b) message - String value containing a word or sentence, e.g. "hey this is a MyMsg"

Task 4. Create two nodes that publish and subscribe messages.

1. Name the nodes "node_a" and "node_b".
2. "node_a":
   (a) Publishes: std_msgs/String msg on topic "topic_a"
   (b) Subscribes: "topic_b"
3. "node_b":
   (a) Publishes: ros_practice/MyMsg msg on topic "topic_b"

   (b) Subscribes: "topic_a"

4. The callback function for each Subscriber should print the topic name and all attributes of a msg
5. Run both nodes simultaneously
6. Run "rqt_graph" and save a copy of the graph (screenshot or image icon in top-right)

Helpful commands

Use the ***rostopic*** command to help you debug these nodes.
rostopic list - Displays all topics currently running on the network
rostopic echo *topic_name* - Displays the msgs being published on a topic. Example: *rostopic echo topic_a*