ROS Basics

Topic Covered: Nodes, messages and Topics

Part 1

100DaysofROS series

By
M.N.Subhash
@malladisubhash

WHAT IS ROS

BASICS

Meta-operatingsystem

Open-source framework works with the support of main OS

Framework

Composition of basic tools and ready-made components

No need to reinvent the wheel

Reusability of the Components

Basic Components

ROS Master
ROS topics
ROS nodes

Meta Operating System

ROS is the collection of libraries, tools, nodes and topics working together to develop a robotic application, It is a system which is dependent on the underlying main OS to carryout the tasks to develop any application

Framework

ROS is a middleware suite which is responsible for the communication between different programmes(nodes) and help the developer tobuild and reuse the code among the robotic applications

No need to reinvent the wheel

As the main structure of ROS is madeup of nodes and these nodes can be reused again in different applications to add a desired feature from one robotic application to another, which reduces the development time.

Basic Components of the ROS

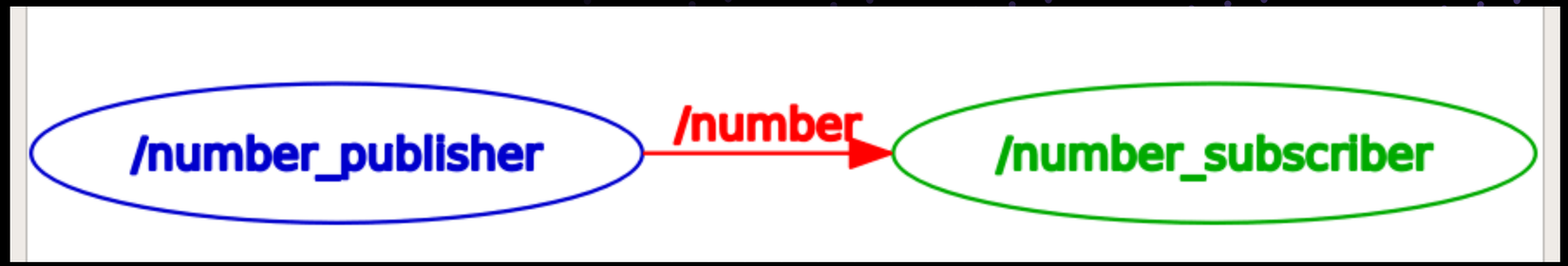
ROS Master - It is responsible for the monitoring of the communication between the nodes

ROS topics - It is the channel which is exsiting between the nodes for the exchange of messages between them

ROS Nodes - These are the processes where the computation happens and then the resultant output is exchanged with other nodes

ROS Computation Graph

- ROS has mainly three levels of concepts which are responisble for its functioning they are ROS Filelevel system, Computation graph system and community level system
- The Computational graph is the peer-to-peer network of ROS processes that are processing data together
- The basic Computation Graph concepts of ROS includes following parts nodes, Master, Parameter Server, messages, services, topics, and bags
- rqt_graph provides a GUI plugin for visualizing the ROS computation graph.
- The basic rqt_graph of publisher subsciber nodes is as follows



- In the above figure
- **number_publisher**: Is the publisher node
- **number**: It is the topic through which information is tranmitter
- **number_subscriber**: Is the subscriber node

Source : http://wiki.ros.org/ROS/Concepts

Nodes

- Node is an executable program file in a package that uses ROS to communicate with other nodes to exchange information
- It is written either in C++ or python programing language using client library roscpp and rospy
- Once the nodes are written their execution starts with running the command ROSCORE
- **ROSCORE** :- is the combination of three things Master (Which Provides names service for ROS) + rosout (Which is to print the output) + Parameter server
- **ROSNode**:- which is used to get the information the node which is running
- **ROSRUN**:- Which is the command to run the node
- The basic ROSnode-command line tools are
  a) **rosnode list** - To list the node running.
  b) **rosnode info** - To get the information of the node.
  c) **rosnode kill** - To kill a running node.

Source: http://wiki.ros.org/ROS/Tutorials/UnderstandingNodes, http://wiki.ros.org/rosnode

ROS Topics

- These are the channels through which the data/information is send b/w the nodes.
- The nodes publish and subscribe to the topics to exchange the information.
- Topics are unidirectional, streaming communication channels
- The information which is passed through the channels is the ROS messages
- The topics allow only a particular message types and the nodes subscribed to the topic allow only those message types

ROS Messages

- ROS uses simplified message description language for discribing the datavalues (message)
- These are derived datatypes which are similar to arrays,functions and pointers in C++
- Message descriptions are stored in .msg files in the msg/ subdirectory of a ROS package
- There are two parts to a .msg file: fields and constants. Fields are the data that is sent inside of the message. Constants define useful values

Source : http://wiki.ros.org/msg

About Me

I am currently pursuing my degree at Sir M Visvesvaraya Institute of Technology, Bangalore.
The domains I am interested in are Electronics and Robotics.
My interest in Robotics started when I was in my second year, Currently improving my skills in ROS and took a 100daysofROS challenge to focus mainly on the hardware implementation of ROS
I put regular updates of my progression of challenges on Twitter
As a part of the challenge, I will be documenting the important things and will be making a concise version of them
Follow me to catch up with ROS and learn how to implement it on hardware.

LinkedIN profile: linkedin.com/in/malladisubhash
Twitter Handle: @mns2610
GitHub ROS repository link : https://github.com/malladi2610/100_days_of_ROS