



Introduction to ROS (Robot Operating System) with Arduino

References

All content on ROS has been adapted from www.ros.org , please visit for more tutorials and information.

All content on Arduino has been adapted from www.arduino.cc , please visit for more information.

Contents

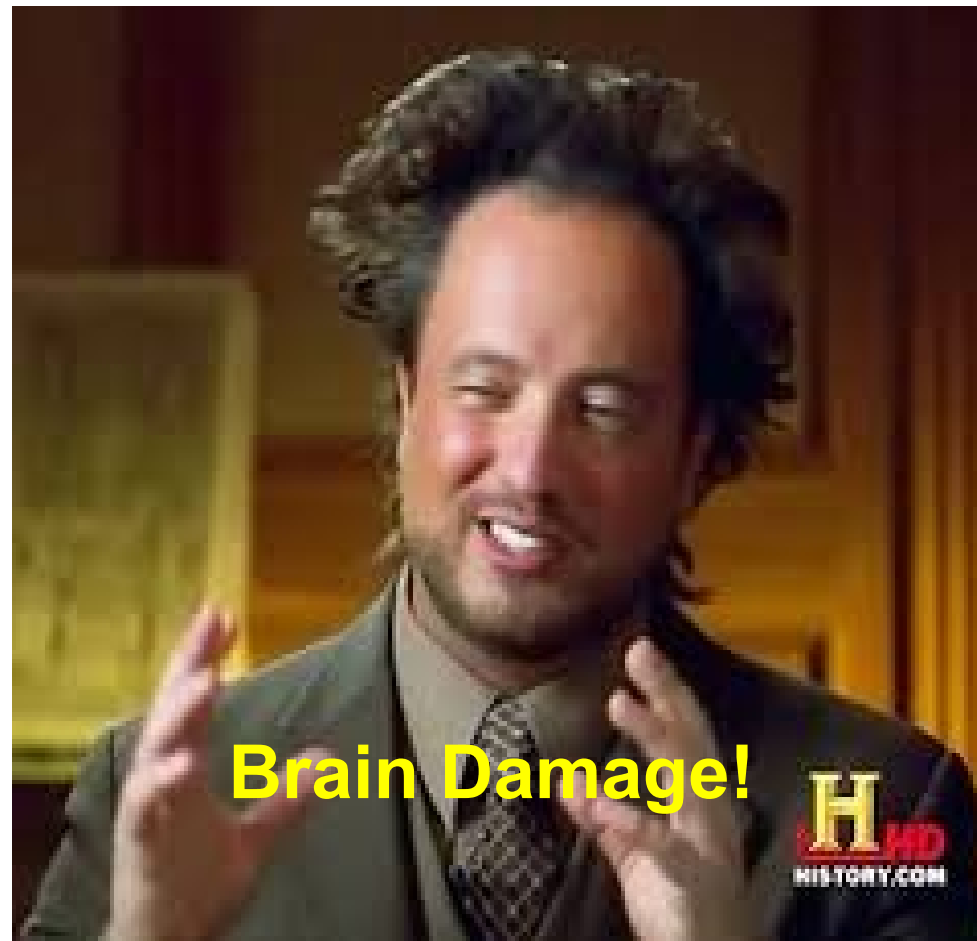
- Introduction to ROS
- Need for a meta-OS for Robots
- Important Definitions and Concepts
- Introduction to Arduino
- Using ROS nodes with Arduino
- Examples



The PR2 robot
Supported by ROS

Statutory Warning!

Using other people's code, without understanding it properly, will cause...



For
Humour
Purpose
Only!

Robot Operating System

- ROS provides libraries and tools to help developers create robot applications.
- It provides hardware abstraction, device drivers, libraries, visualizers, message-passing, package management, etc.
- ROS is licensed under an open source, BSD license.
- ROS is a meta-OS , NOT a complete standalone OS.



Robots supported by ROS

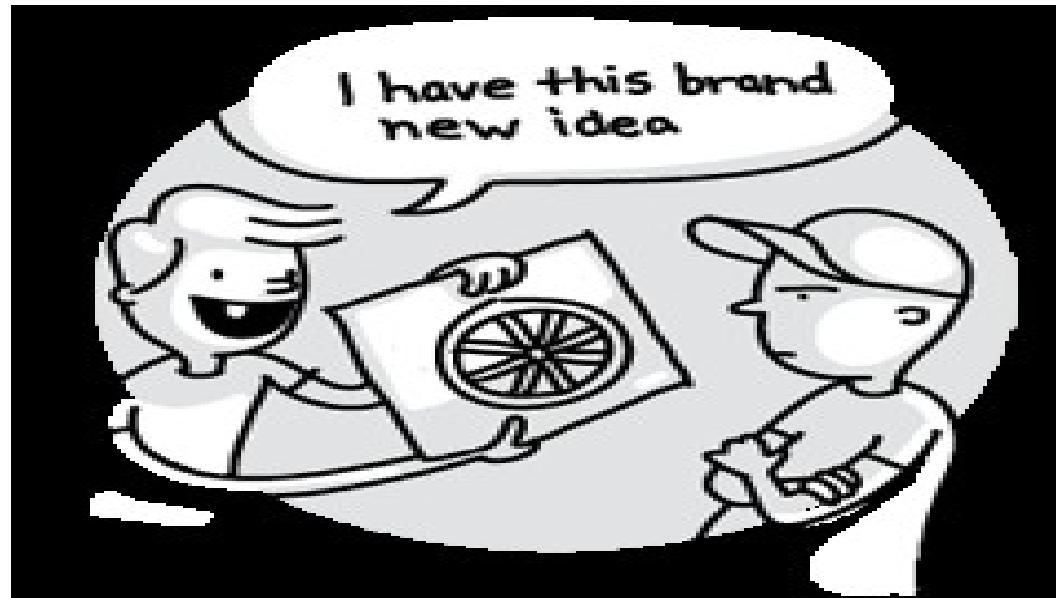


Most commercial robots support ROS. You can make your own robots to run with ROS too!



Why ROS?

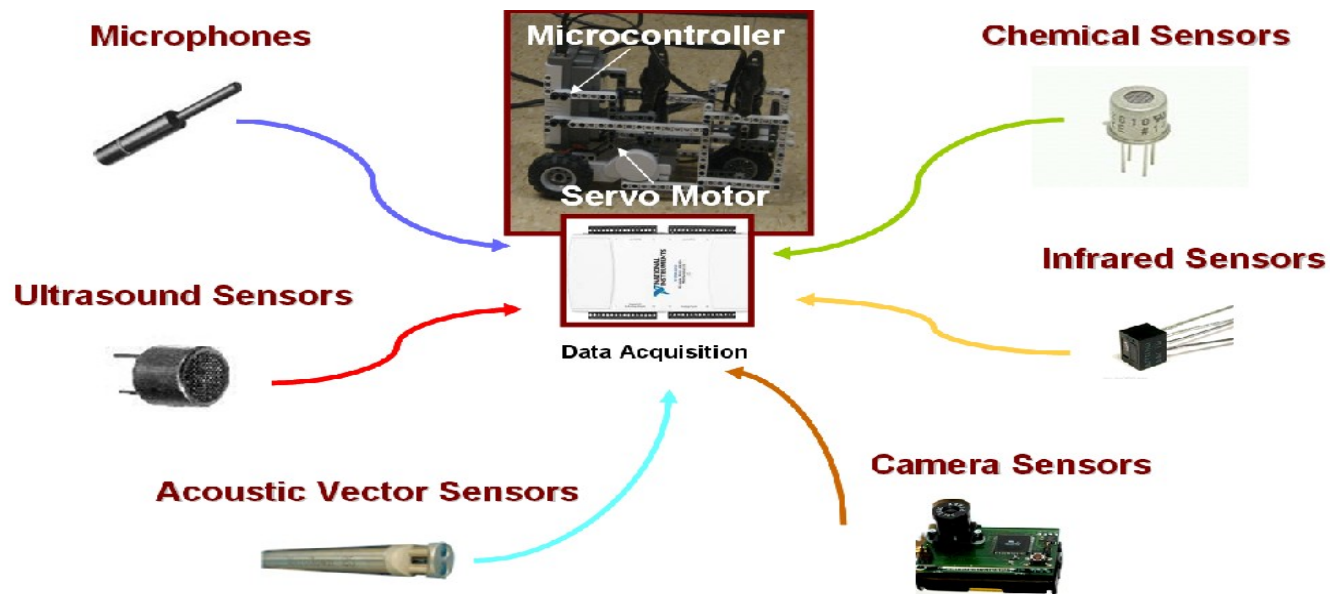
Robotics Engineers keep re-inventing the wheel, by writing software for standard equipment again and again.



ROS helps you avoid this problem by standardising the way code is written for different platforms. It also helps people who wish to bypass hardware intricacies (Not recommended!).

Why ROS?

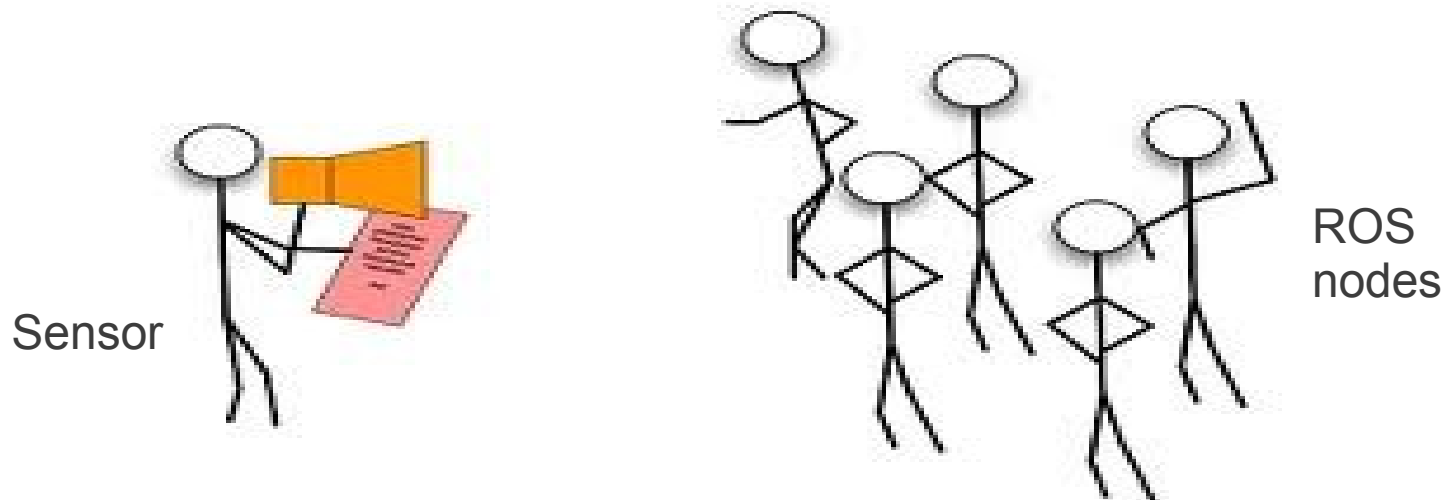
Robots are getting more and more complex, and require the integration of a plethora of sensors and actuators.



ROS supports a wide range of sensors (you can also write a ROS driver for your sensor) and actuators. This helps in quick implementation of the algorithm and cuts down time required for integrating hardware.

Why ROS?

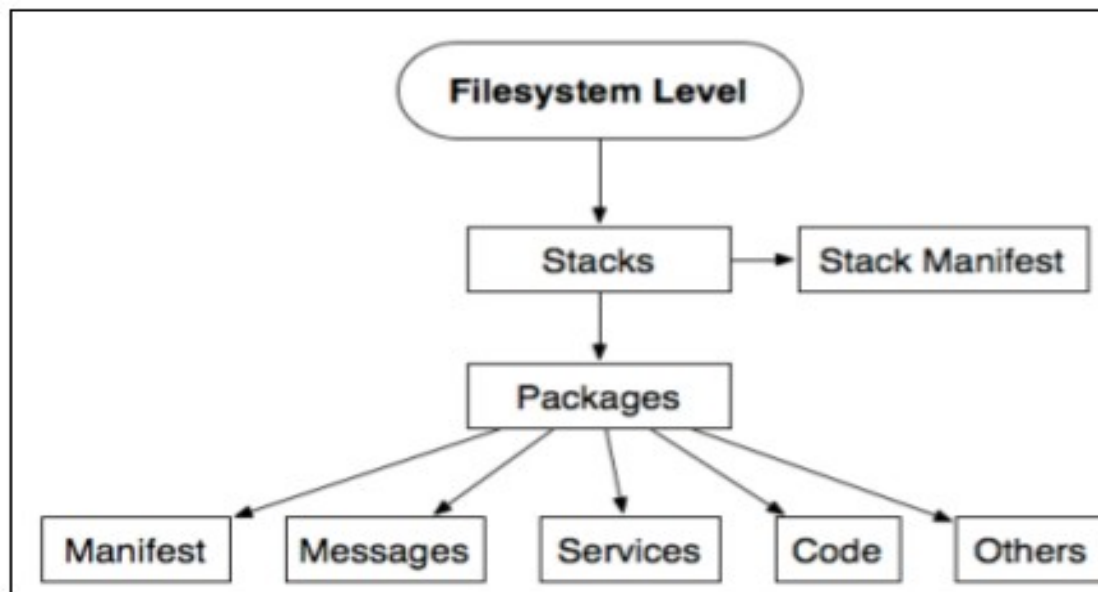
ROS supports parallel computing and offers all the benefits of an OS. Example : One can publish data received from sensors to different sub-programs running.



In short the days of making Line-Followers each year at IIT-kanpur are over! :)
We can now quickly develop intelligent robots.

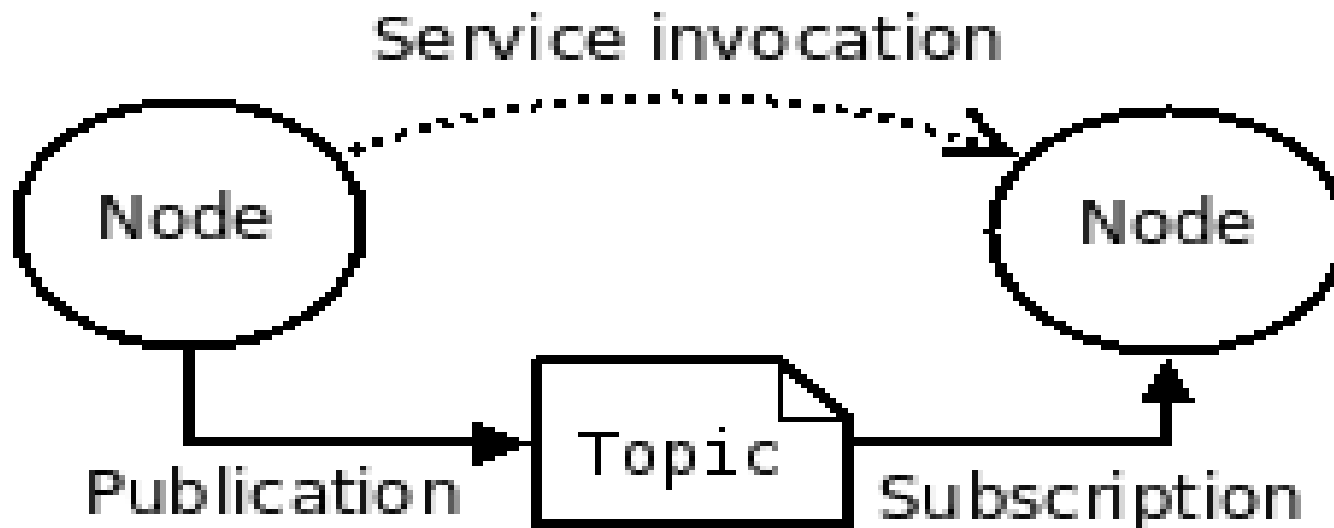
Important Definitions

- Packages: A package has the minimum structure and content to create a program withing ROS.
- Nodes: Nodes are executables (runtime processes). A package may have several Nodes.
- Stacks: Several packages with some common goal combined together are called Stacks. Eg. Navigation Stack.



Important Definitions

- Message: Nodes communicate with each other through messages.
- Topics: Each message must have a name to be routed by ROS. When a message is published, it is called a Topic.



ROS Installation

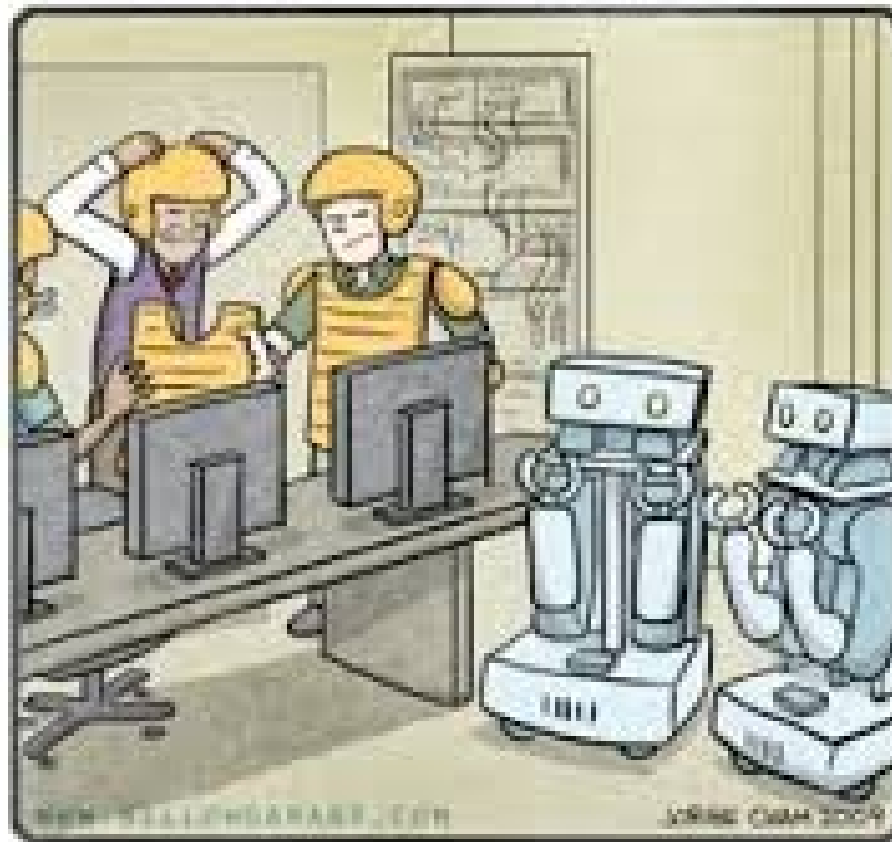
- Refer to www.ros.org for download instructions.
- Recommended OS : **Ubuntu 13.04** (Warning! ROS doesn't run properly on Windows or Ubuntu 13.10)
- ROS can also be installed on Embedded Systems like BeagleBone, BeagleBoard, R-pi, etc.



For
Humour
Purpose
Only!

Practical Demonstration

R.O.B.O.T. Comics



"I HAVE A BAD FEELING
ABOUT THIS DEMO."

Arduino with ROS

- Arduino is an open source Microcontroller development platform for casual developers.
- It is fairly easy to interface different sensors and actuators with Arduino, which makes it quite attractive.
- One can interface Arduino with ROS using the **rosserial** node.
- Refer the following link for installation

http://wiki.ros.org/rosserial_arduino/Tutorials/Arduino%20IDE%20Setup



Kinect with ROS

- ROS supports most known cameras like USB webcams and Depth cameras like Kinect.
- Install the OpenNI node in ROS to access different features of OpenNI.
- ROS also supports PointCloud2 message types, which helps simplify the code and algorithm.



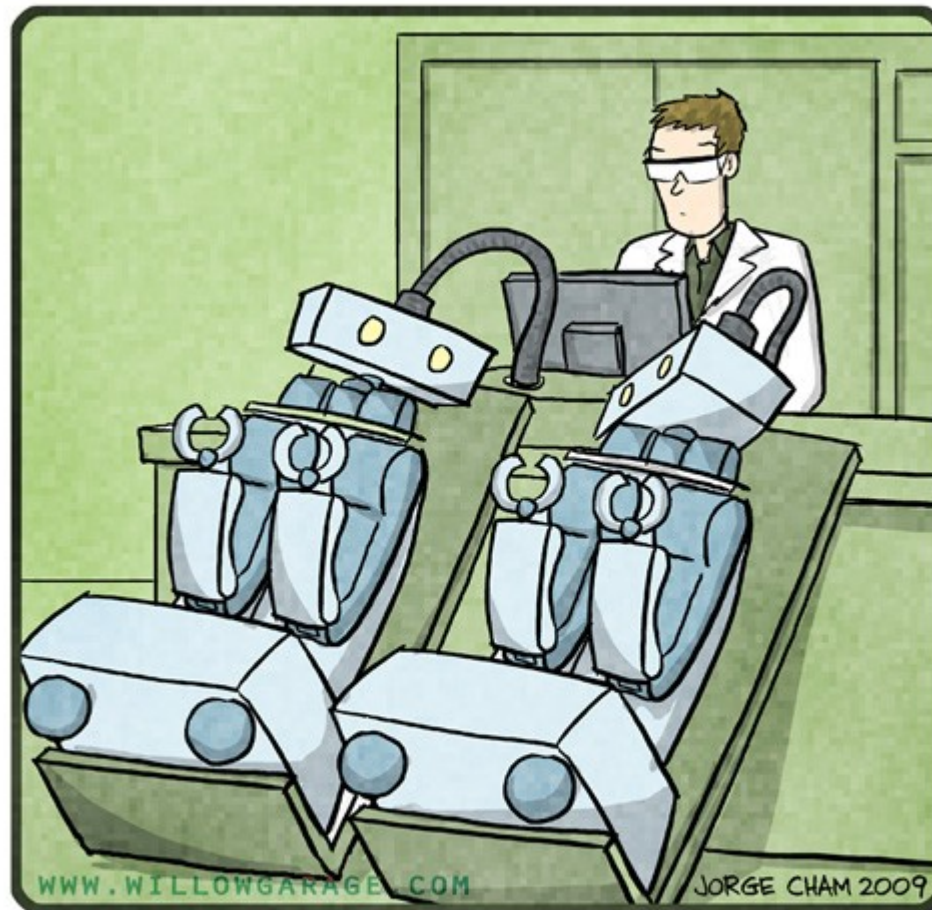
Standard Libraries with ROS

- One can easily integrate standard C++ libraries like OpenCV, Point Cloud Library, Qt, etc.
- ROS already comes installed with Boost, Eigen and VTK.
- All you need to do is edit the cmake file according to your need!



Practical Demonstration

R.O.B.O.T. Comics



"DO YOU EVER FEEL LIKE
YOU'RE IN THE MATRIX?"

Thank You

 **ROS.org**