

# Introduction to Robot Operating System

Building an understanding of Robot Operating System

# Agenda

- Introduction to Robotics programming
- Introduction to ROS
- Installing ROS

# Agenda

- ROS Architecture
- ROS Communication Model
- ROS Client Libraries
- ROS Concepts

# Agenda

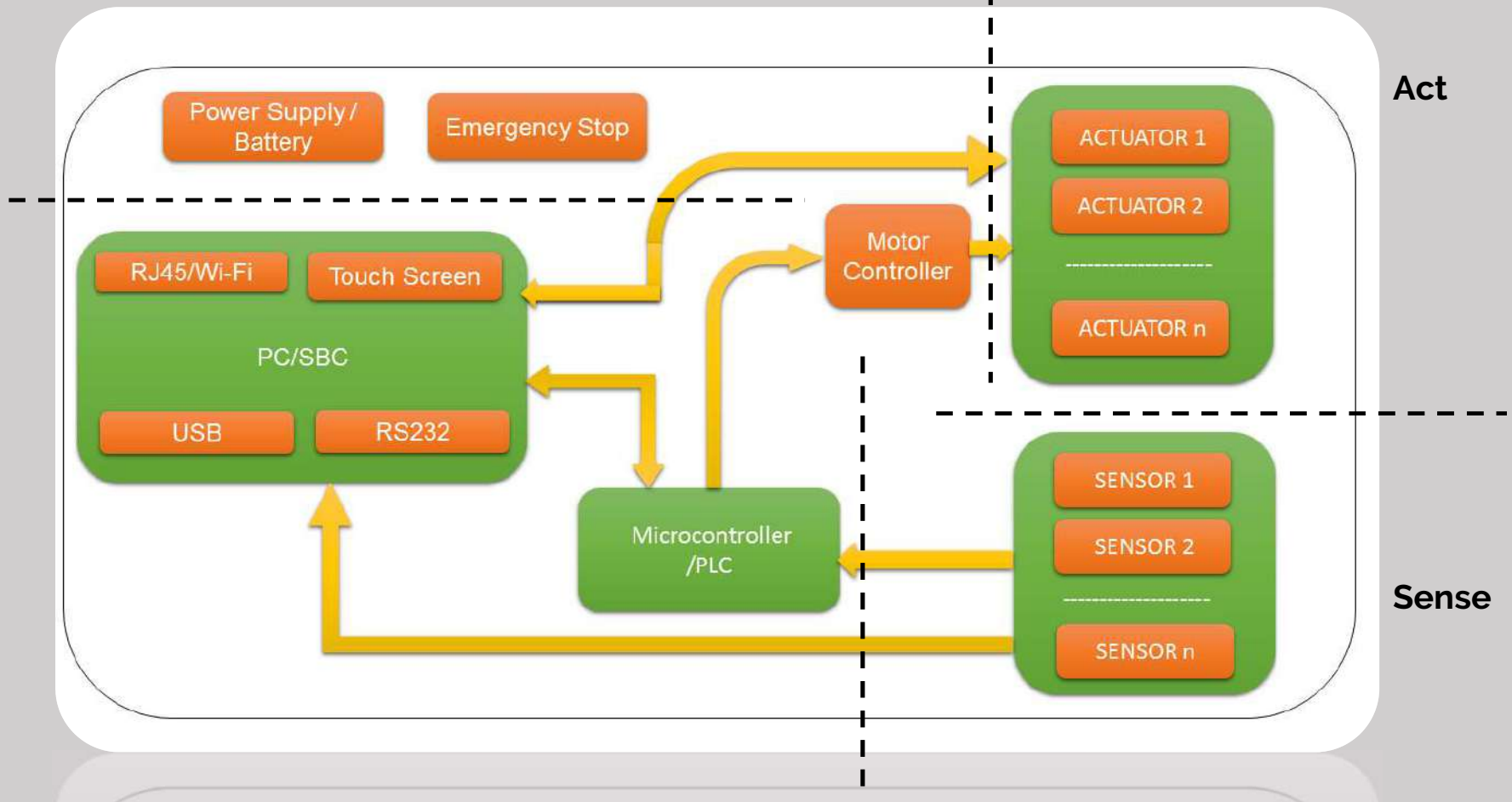
- ROS Build System
- ROS TurtleSim projects

# Preferred software

- Ubuntu 20.04 LTS
- ROS Noetic or higher
- Terminator application:
  - \$ **sudo apt-get install terminator**
- gedit / nano text editor

# What is a robot? Inside view

Think



Act

Sense

# Robot Components

- Main Robot components
  - **Sensors:** Camera, Wheel encoders, Ultrasonic sensors, etc.
  - **Actuators:** Servo, Stepper, DC motors, etc.
  - **Computer**
    - **PC:** Intel NUC, Nvidia TX2
    - **Microcontroller platform:** Arduino board, Texas Launchpad, ARM controller based boards



RC Servo Motor



Stepper Motor



DC Gear Motor



Industrial Servo Motor



Dynamixel Smart Actuator





Velodyne LiDAR



UTM – 30LX, Laser Scanner



Kinect, 3D Depth Sensor



Intel RealSense, 3D Depth Sensor

# Computing units for Robots

- X86, X86\_64 Based PC
  - Intel NUC
  - Industrial PC
- Single board Computer
  - Nvidia TK1, TX1, TX2
  - Raspberry Pi
  - Odroid



Intel NUC



Raspberry Pi



Nvidia TX2



Odroid



Industrial PC

# What is Robot Programming

- **Robot Programming:**
  - Modelling the behavior of a robot using a programming language
  - Robot sensor values as input to the model
  - Robot actuators as the output of the model.
- **Programming languages: C++, Python etc.**

# How should I choose a programming language for my robot?

- Threading
- High-level and Object oriented
- Low level device control
- Ease of prototyping
- Inter-process communication
- Performance
- Community support
- Availability of third-part libraries
- Existing robotics software framework support

# Programming: Different Approaches

- Use existing software frameworks to program robots
  - Quick prototyping
  - Good for academic research
  - Need fine tuning for robotic products
  - Example: **Arduino framework (low level)**, **Robot Operating System (high level)**
- Create everything from scratch using a programming languages
  - Time consuming
  - Better in the long run
  - Preferred Programming languages: **C++** , **Python**

# FREE Robotic Software frameworks

- **Robot Operating System:** Communication Middleware
  - Website: <http://www.ros.org/>
- **Note:** ROS is not a real Operating System but a meta operating system
- **Open-CV:** Computer Vision library
  - Website: <https://opencv.org/>
- **PCL:** Point Cloud Library
  - Website: <http://pointclouds.org/>



# FREE Robotic Software frameworks

- **Gazebo:** Robot simulator
  - Website: <http://gazebo.org/>



- **Open-Rave:** Robot framework for motion planning
  - Website: <http://openrave.org/>



# Proprietary Robotic Software frameworks

- **Webots:** Robot simulator
  - Website: <https://cyberbotics.com/>
- **V-REP:** Robot simulator
  - Website: <http://www.coppeliarobotics.com/>
- **Actin:** Robot simulation framework
  - Website: <https://www.energid.com/actin>





# Popular Robotic Programming Languages

- C++



- Python



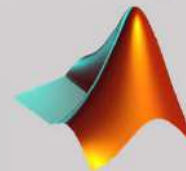
- Java



- C#/.NET



- MATLAB



# Prerequisites to learn ROS

- Knowledge in Linux terminal commands
- Knowledge in C++ or Python
- Do you have these skills ??



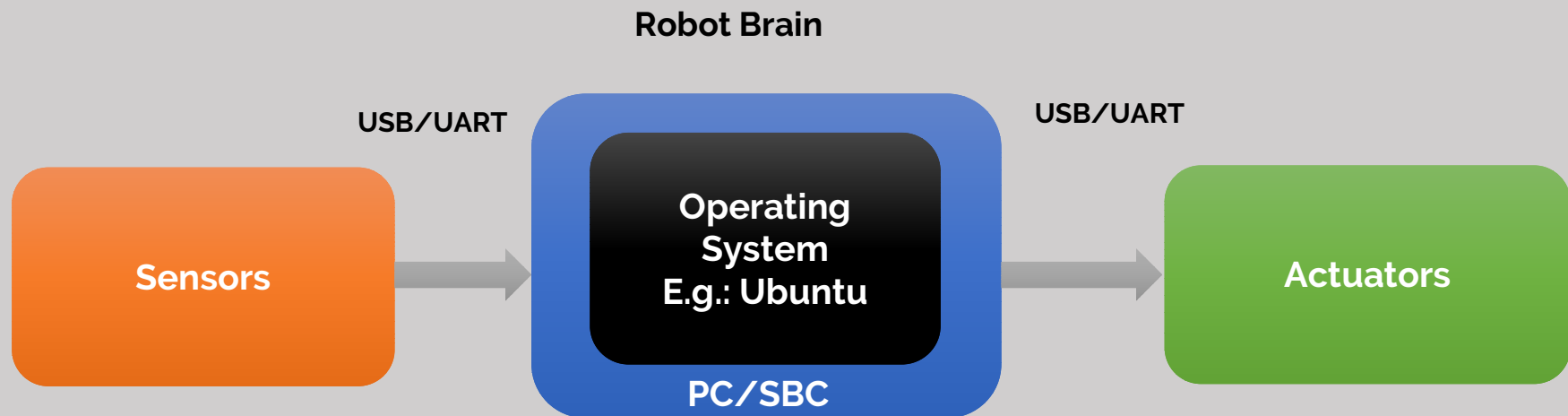


# Why we need an OS in Robots?

Discussing the importance of an OS in a robot

# Why we need an OS in Robots?

- Operating system (OS) is running on robot brain/PC
- OS is responsible for scheduling the tasks in a computer
- It can receive input from robot sensors | Perform computation | Take decision | Send command to actuator



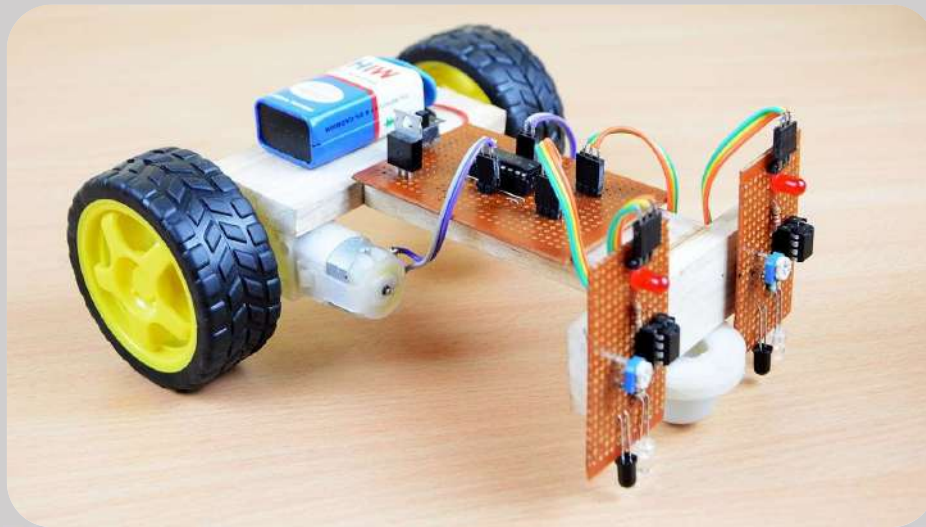
# Why we need an OS for Robots?

- Robot software is running on Operating System
- Low level control
- Support for robot framework
  - E.g.: Robot Operating System (ROS), Open-CV, PCL etc.
- More flexibility in programming robots
- Secure
- Realtime processing



# When we don't need an OS in a robot?

- Brain only have a microcontroller
- Brain only using digital circuits



# List of Operating System used in Robots

- GNU/Linux variants

- Ubuntu (<https://www.ubuntu.com/>)
- Debian (<https://www.debian.org/>)



- Windows CE, 7 & 10



- Mac OS X



- QNX Real time operating system (<https://blackberry.qnx.com/en>)



- VxWorks: (<https://www.windriver.com/products/vxworks/>)



# Computing units for Robots

- X86, X86\_64 Based PC
  - Intel NUC
  - Industrial PC



Intel NUC



Raspberry Pi

- Single board Computer
  - Nvidia TK1, TX1, TX2
  - Raspberry Pi
  - Odroid



Nvidia TX2



Odroid



Industrial PC

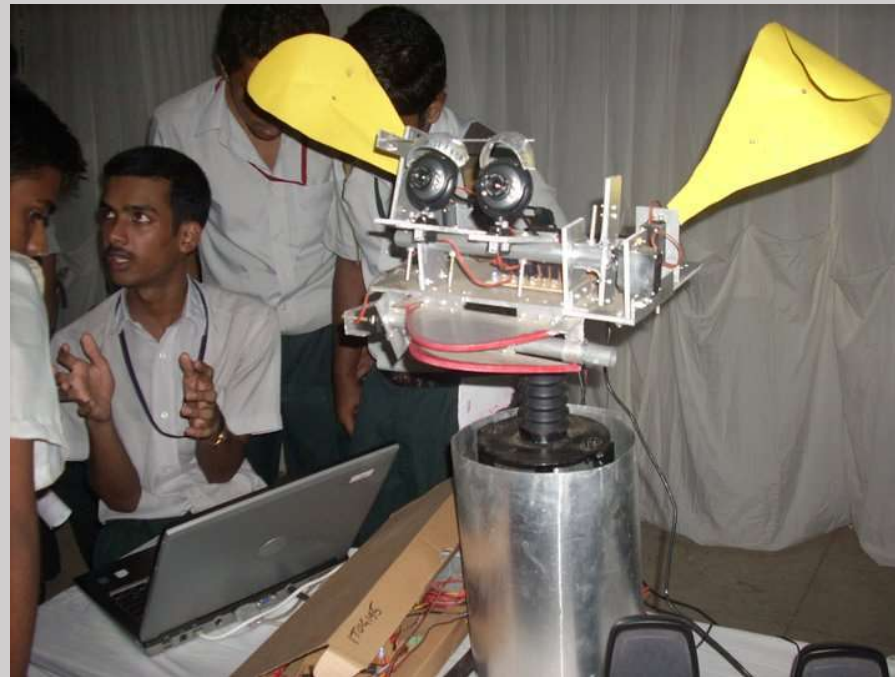




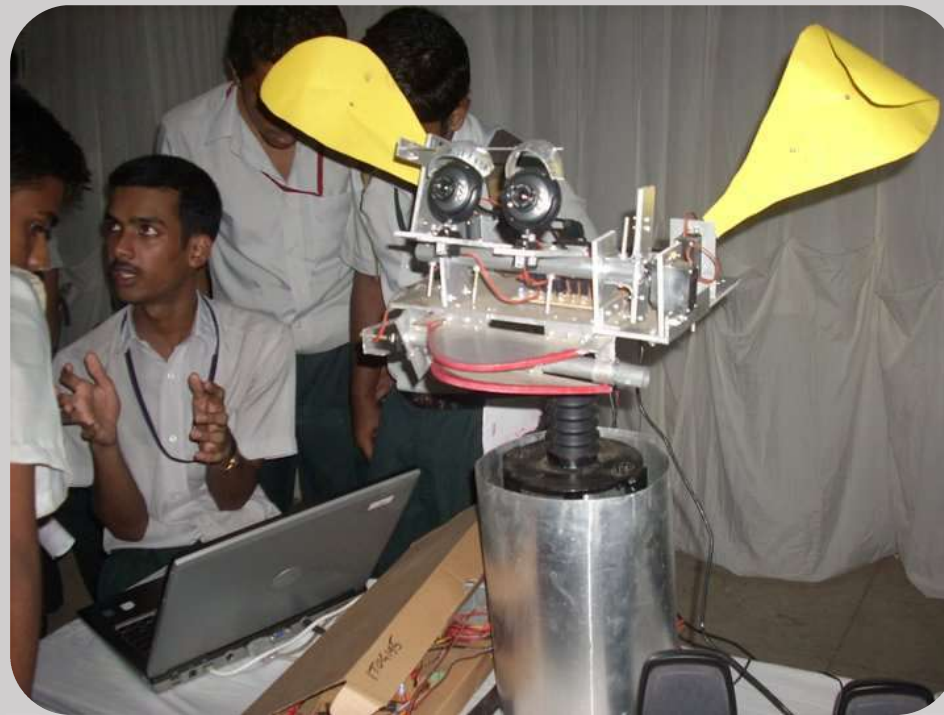
# What is Robot Operating System (ROS)

Discussing brief idea of ROS

# Let's start with a story



# A robot working without R.O.S



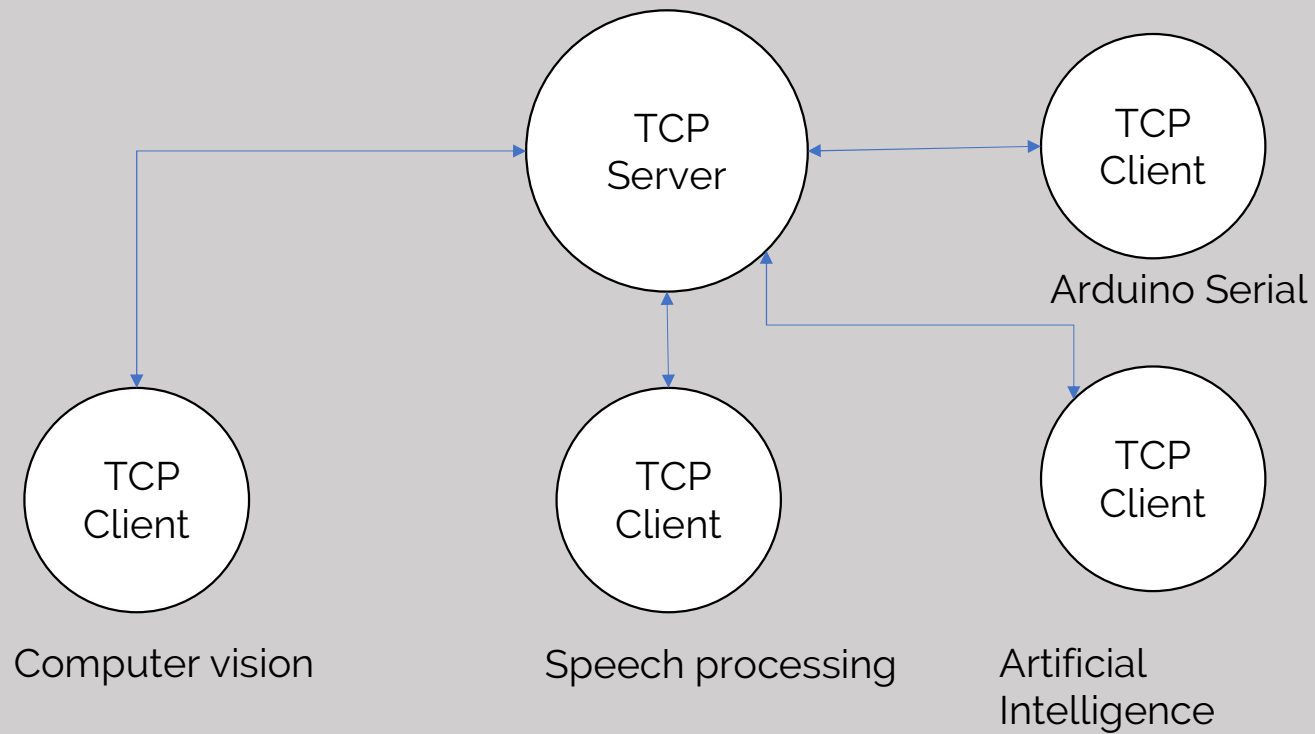
- **Type of Robot** : Social Robot
- **Sensors** : Dual VGA camera, Microphone
- **Actuators** : 12 RC – Servos
- **Robot Brain** : 2 X Laptops
- **Capabilities** :
  - Interacting with people using natural language using AI, speech recognition/synthesis and express emotions
  - Face detection, motion detection and object detection
- Programming languages : Python, C++
- Communication of processes using Python Server, client architecture



# Challenges Faced !!!!

- Interprocess communication from different machines
- Synchronization of robot task
- Lack of Modularity
- Difficult to reuse the code
- Difficult to develop application on top of the existing codes

# Software Communication diagram



# A robot working on R.O.S



•**Type of Robot** : Autonomous Mobile Robot(Chefbot)

•**Sensors** : Kinect 3D sensor, Microphone, Ultrasonic, IMU

•**Actuators** : DC-Geared motor with Encoder

•**Robot Brain** : Intel NUC mini PC

•**Capabilities** :

- Autonomous navigation :One of the application is food serving
- Interacting with people using natural language using AI, speech recognition/synthesis and express emotions

• Programming languages : Python, C++

• Completely running on R.O.S platform





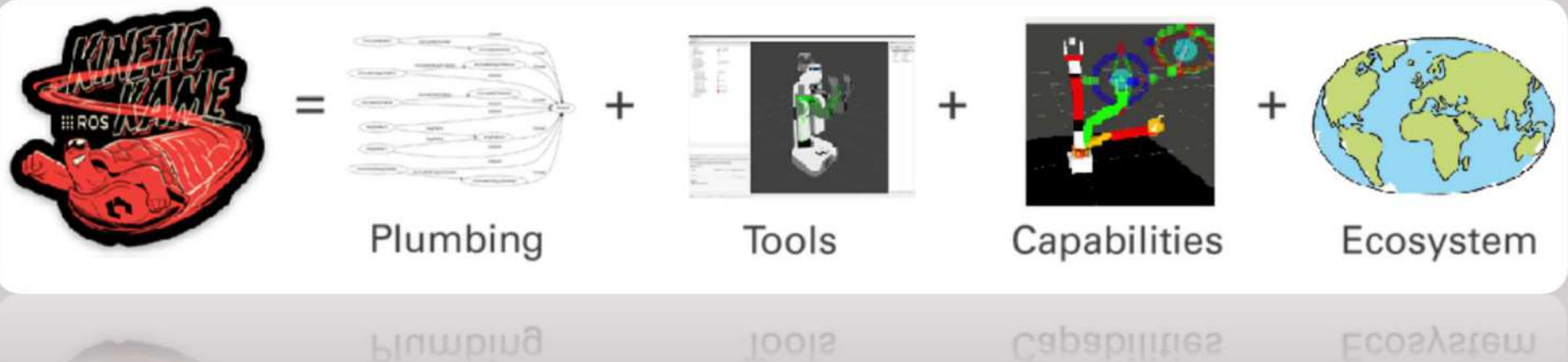
# Issues Faced !!!!

- ~~Interprocess communication from different machines~~
- ~~Synchronization of robot task~~
- ~~Lack of Modularity~~
- ~~Difficult to reuse the code~~
- ~~Difficult to develop application on top of the existing codes~~

- ~~Interprocess communication from different machines~~
- ~~Synchronization of robot task~~
- ~~Lack of Modularity~~
- ~~Difficult to reuse the code~~
- ~~Difficult to develop application on top of the existing codes~~

# What is Robot Operating System?

- Open Source Robotic Software framework
- Robotics Middleware, not a real operating system
- Need a host operating system to run
- ROS Equation



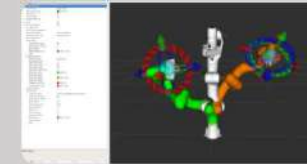
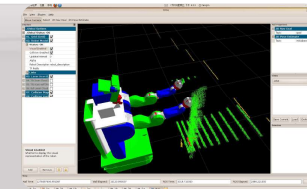
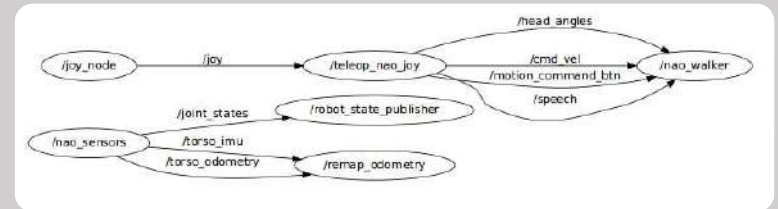
# What is Robot Operating System?

- **Plumbing/Communication middleware:**
  - Inter-process communication

- **Tools:**
  - Visualization and Debugging of robot data
  - Example: Rviz, Rqt

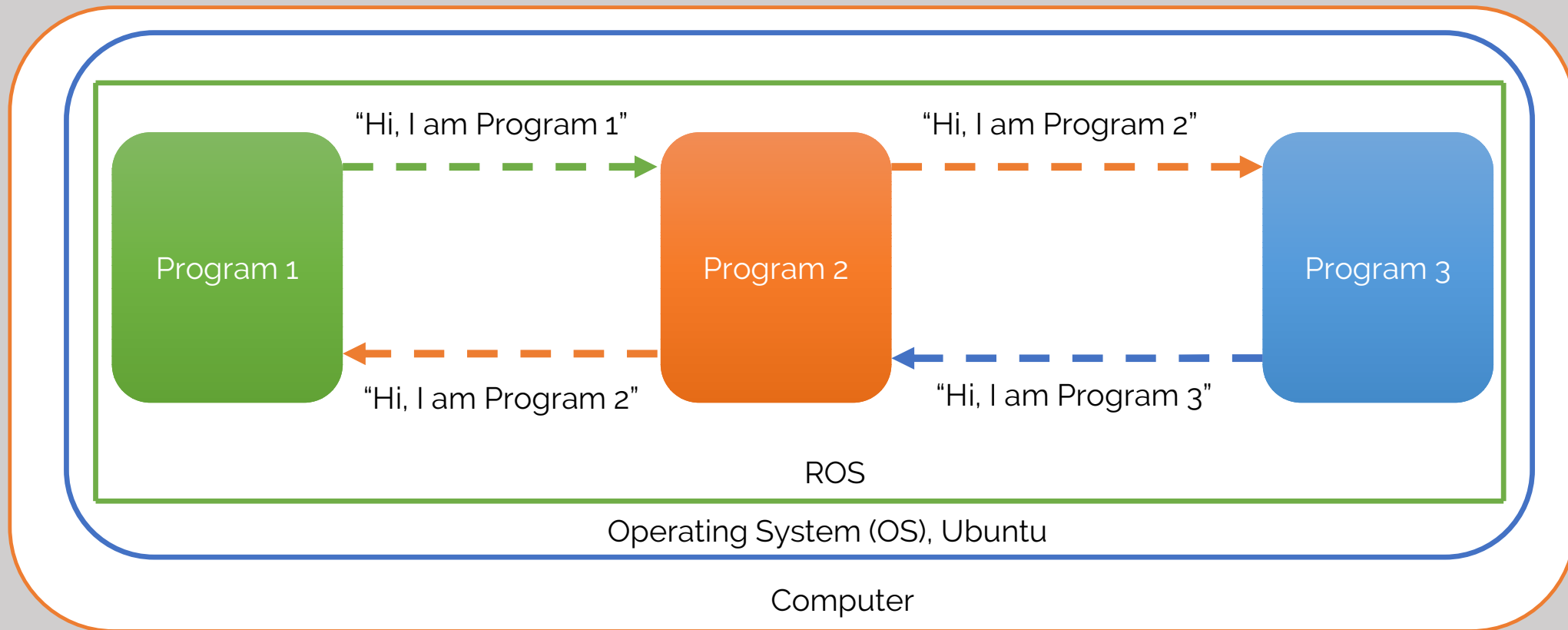
- **Capabilities:**
  - Robot manipulation and navigation
  - Example: MoveIt!, and navigation stack

- **Ecosystem:**
  - World wide ROS developers
  - ROS Wiki
  - ROS Q & A



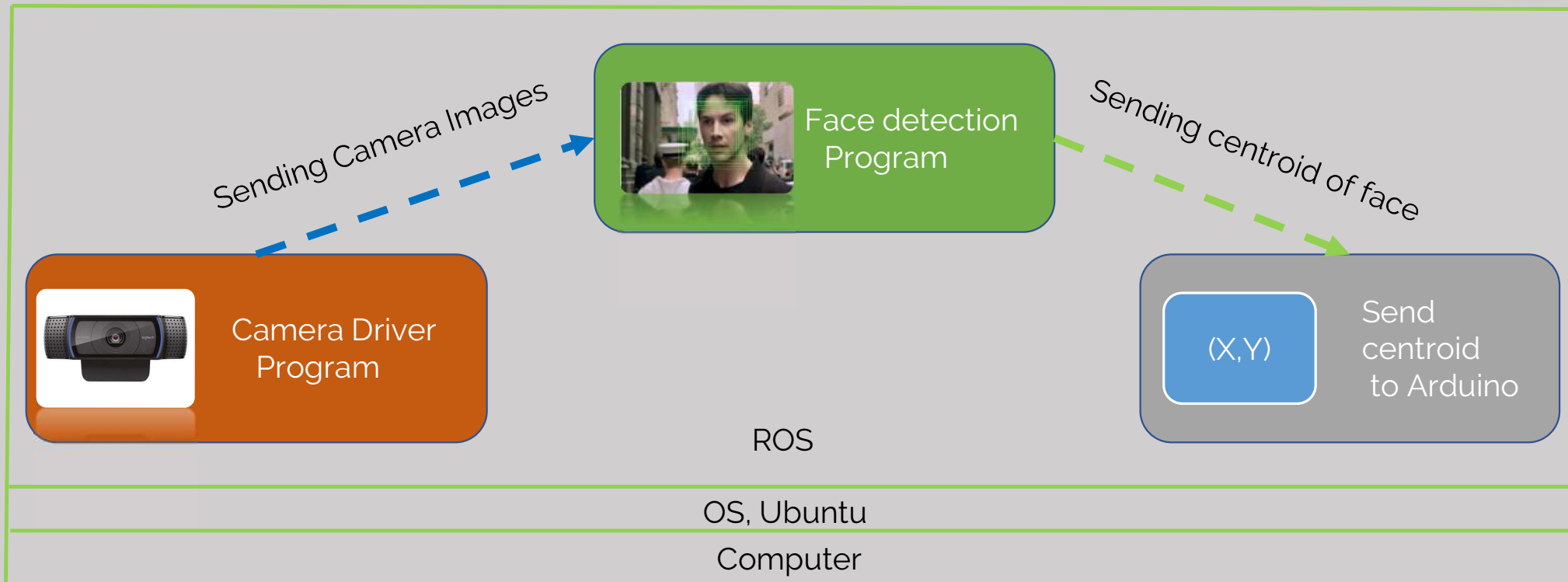
# Working of ROS: Plumbing

- Inter-process Communication



# Working of ROS: Plumbing/Middleware

- Example: Face Detection in ROS

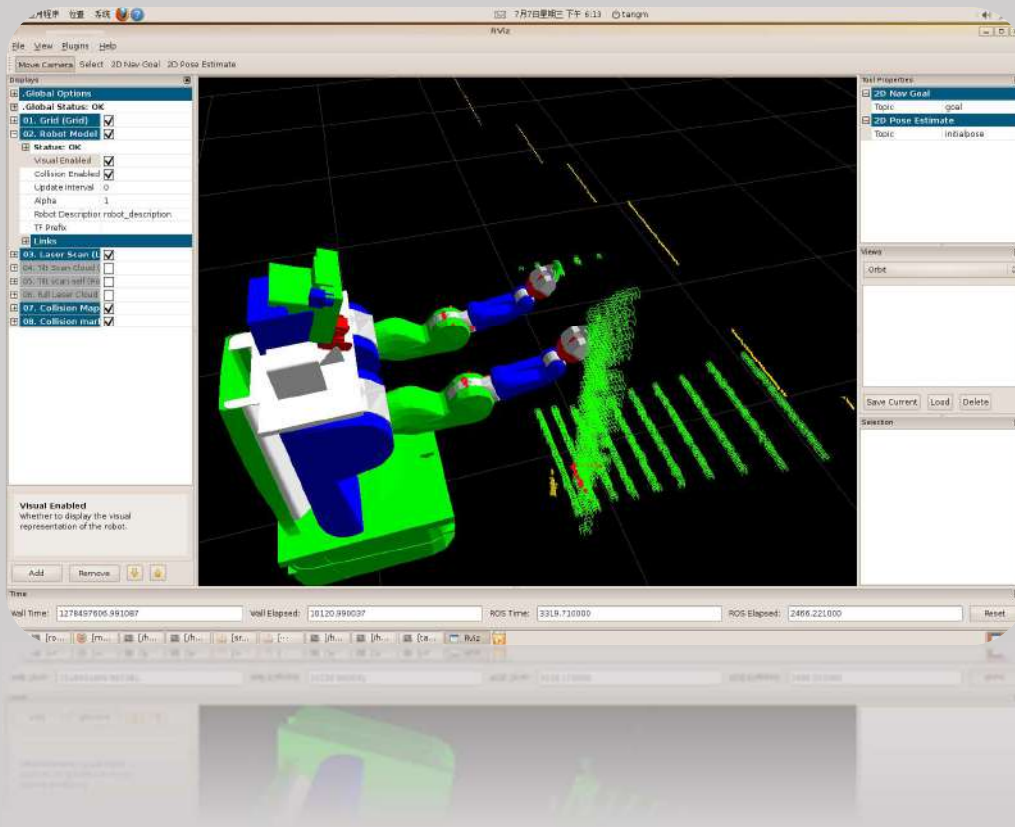


# Working of ROS: Tools

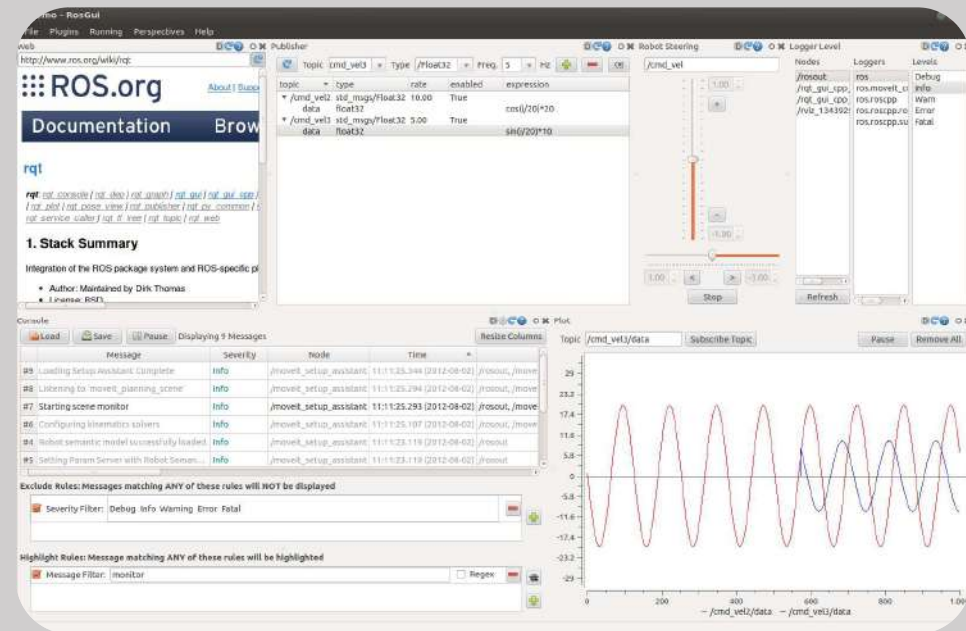
- **Rviz (ROS Visualizer)**
  - 3D Visualization tool in ROS
  - <http://wiki.ros.org/rviz>
- **Rqt (ROS Qt)**
  - GUI framework in ROS based on Qt
  - <http://wiki.ros.org/rqt>
- **ROS Command line tools**
  - rostopic
  - roslaunch
  - <http://wiki.ros.org/ROS/CommandLineTools>

# Working of ROS: Tools

Rviz



rqt





# Working of ROS: Capabilities

- **ROS Navigation**

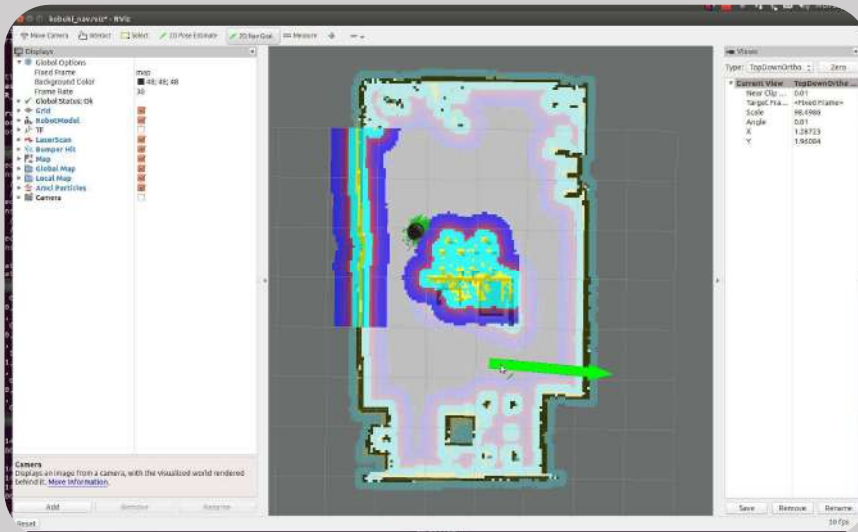
- Collection of software of ROS to move a mobile robot from A to B

- **ROS MoveIt!**

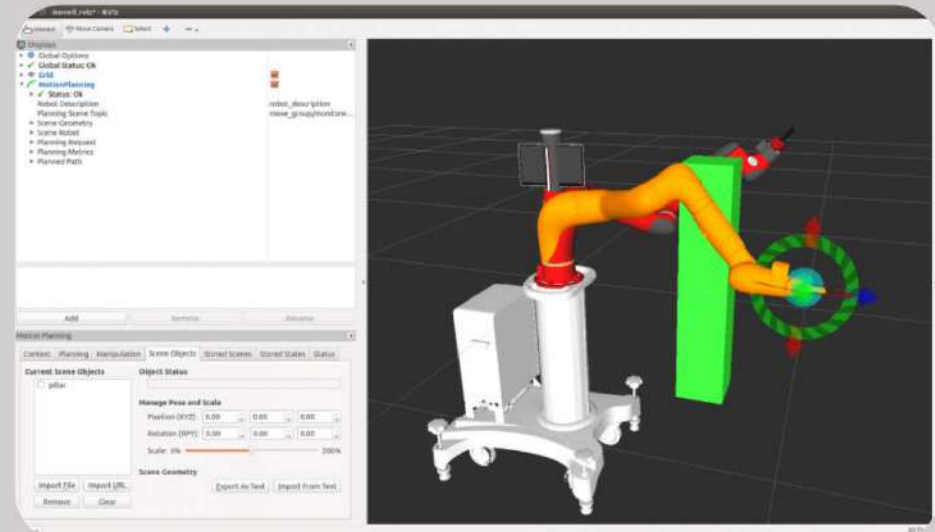
- Software in ROS for planning the motion of a robotic arm

# Working of ROS: Capabilities

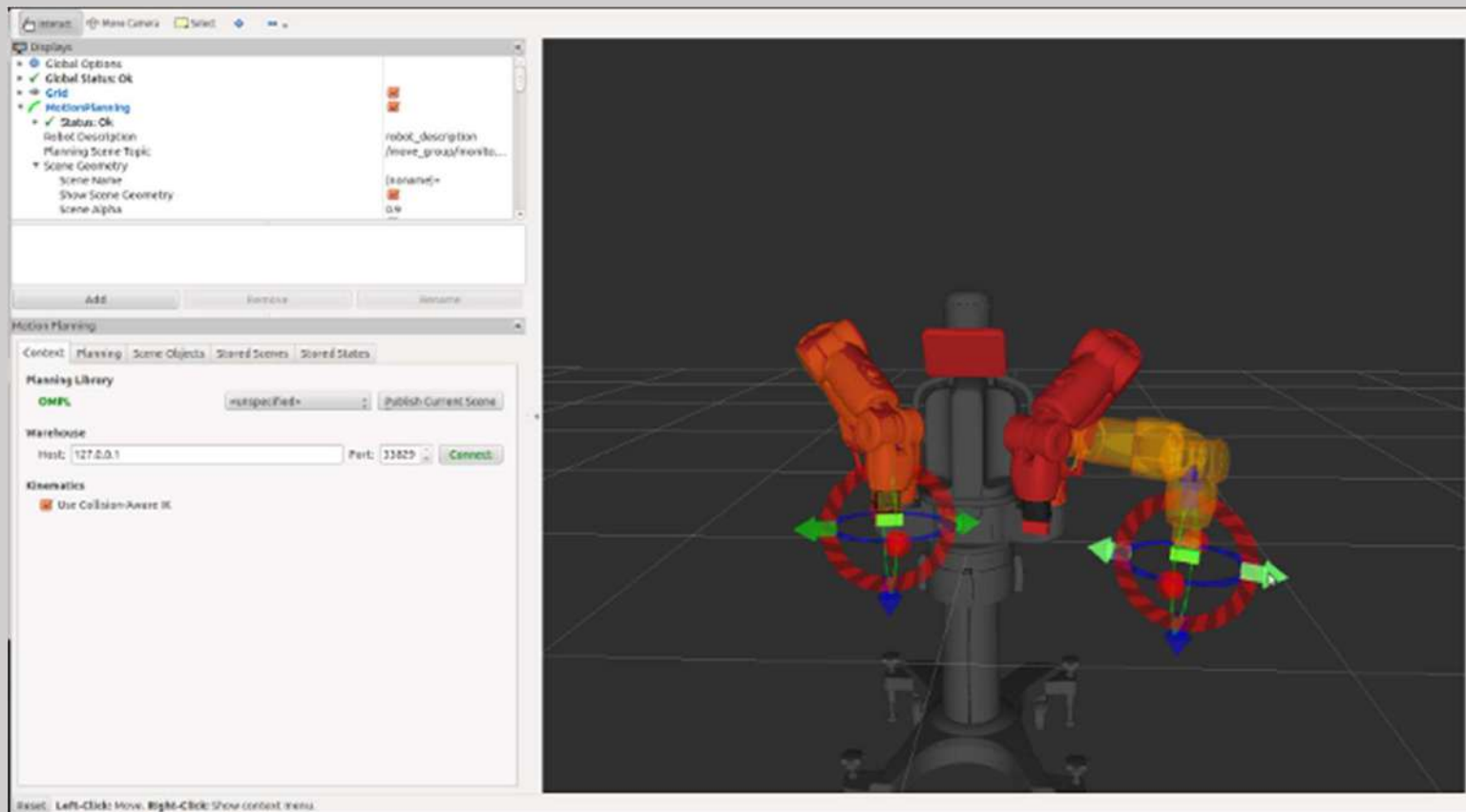
## ROS Navigation



## ROS MoveIt!



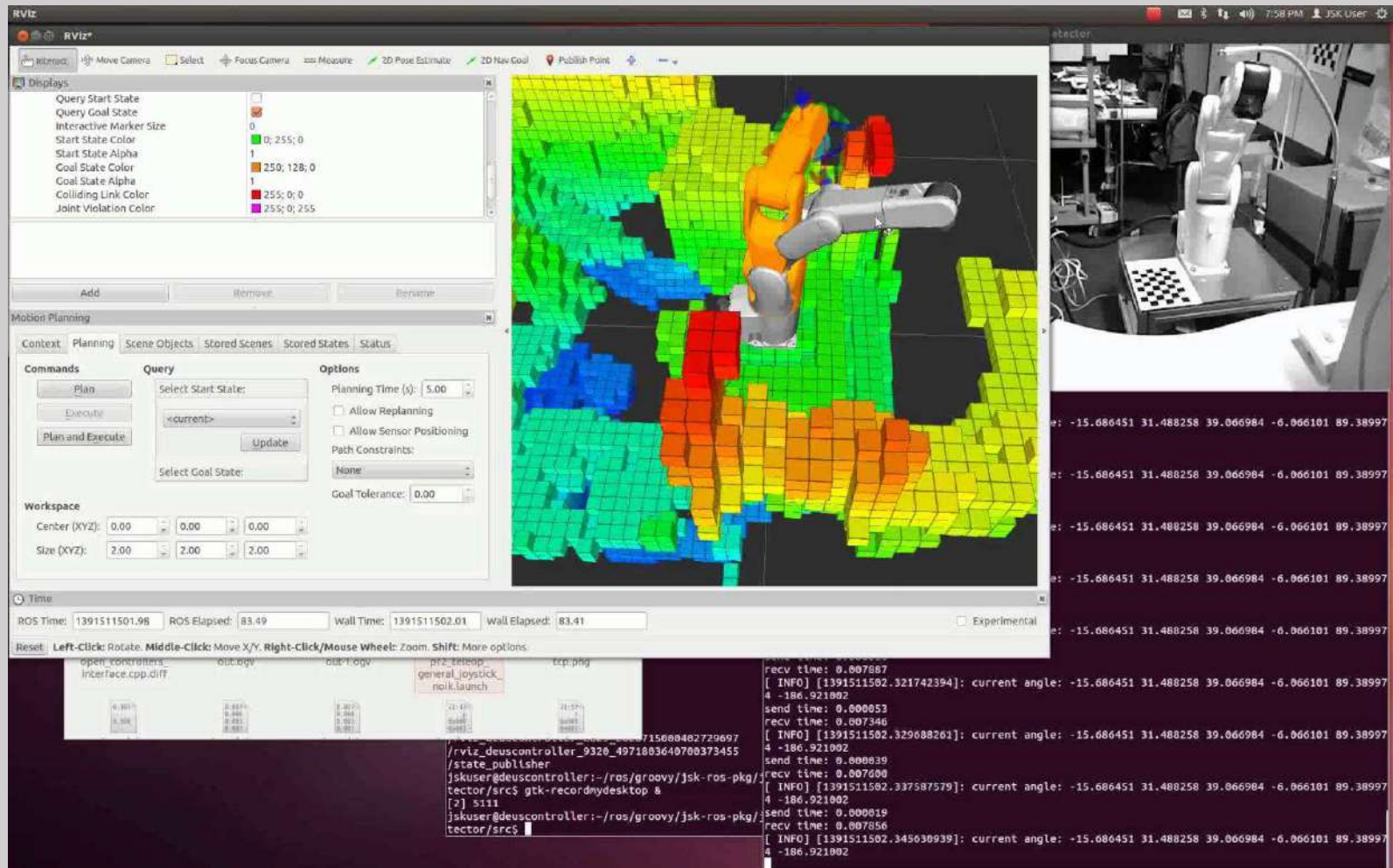
# ROS Manipulation



# ROS Manipulation



# ROS Perception












# ROS Perception



# Working of ROS: Ecosystem

- Worldwide ROS developers
  - <http://wiki.ros.org/Metrics>







1.	 United States	100,711 (20.08%)
2.	 China	90,120 (17.97%)
3.	 Japan	45,834 (9.14%)
4.	 Germany	39,590 (7.89%)
5.	 India	20,632 (4.11%)
6.	 South Korea	16,683 (3.33%)
7.	 United Kingdom	12,784 (2.55%)
8.	 Taiwan	11,809 (2.35%)
9.	 Canada	11,685 (2.33%)
10.	 France	11,651 (2.32%)

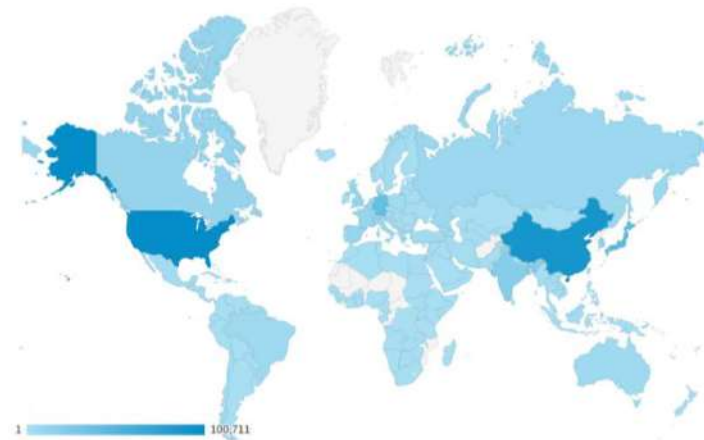




# Working of ROS: Ecosystem

- ROS Wiki: <http://wiki.ros.org>
- ROS Q&A: <https://answers.ros.org/questions/>

1.	 United States	100,711 (20.08%)
2.	 China	90,120 (17.97%)
3.	 Japan	45,834 (9.14%)
4.	 Germany	39,590 (7.89%)
5.	 India	20,632 (4.11%)
6.	 South Korea	16,683 (3.33%)
7.	 United Kingdom	12,784 (2.55%)
8.	 Taiwan	11,809 (2.35%)
9.	 Canada	11,685 (2.33%)
10.	 France	11,651 (2.32%)



Source: Google Analytics  
Site: wiki.ros.org in July 2017



# Supported Host Operating System

- Full Support
  - Ubuntu/Linux
  - Debian



- 
- Experimental
    - OSX
    - Gentoo



# ROS Features



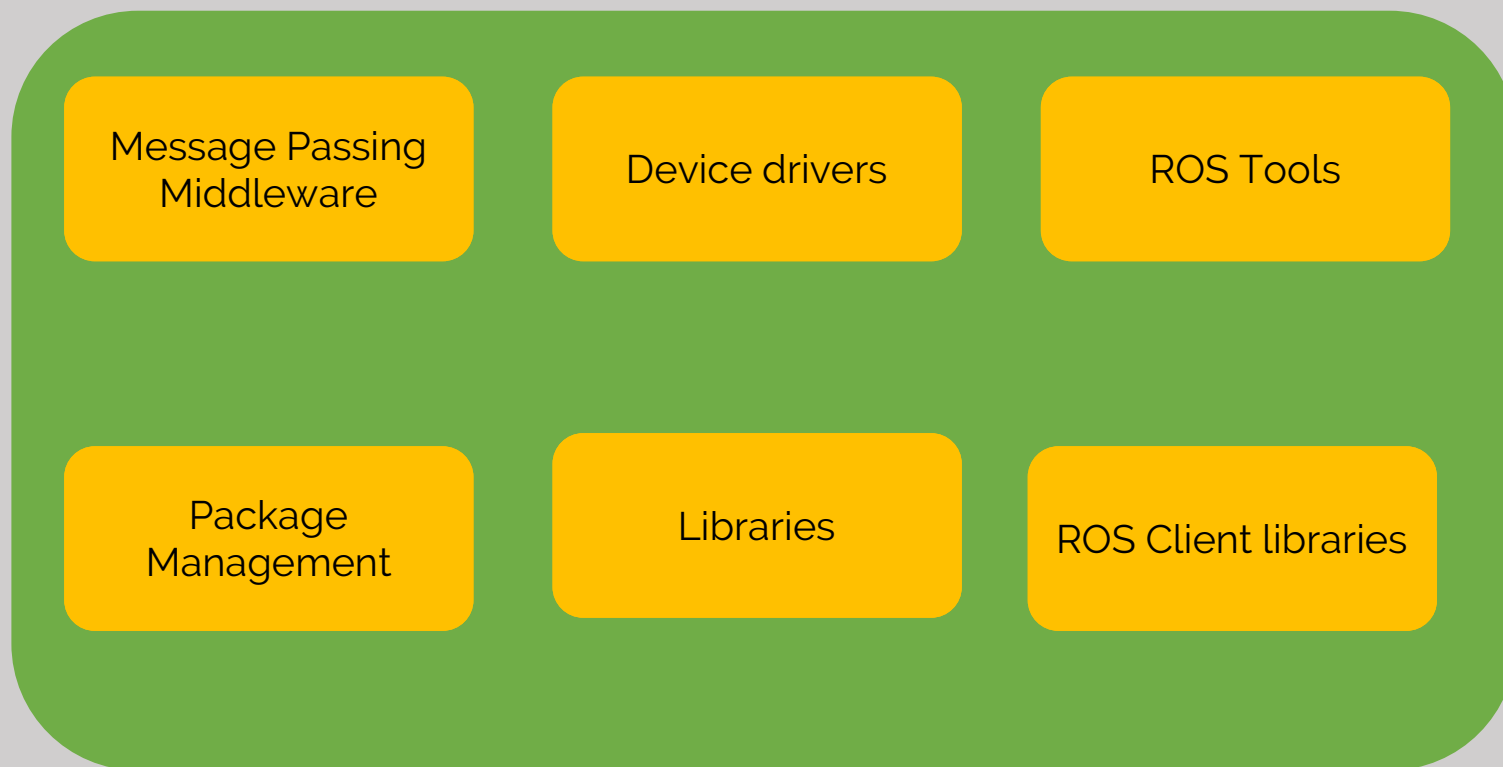
- Provide Hardware Abstraction
- Low level device control
- **Message-Passing between process**
- **Package Management, Modularity**

# ROS Features



- Powerful **tools** for debugging, plotting, and visualizing various kind of data ( rviz , rqt\_gui)
- Vibrant developer community
- Open source BSD license for commercial and closed source products

# ROS - Block Diagram



# Before R.O.S

## Robotic Research & Software development

- No common software development platform for robots
- Code reuse is nearly impossible
- All codes are specific to the each robot
- No active community support

# **Before R.O.S**

## **Robotic Research & Software development**

- Robotic software development time is high because all code should write from scratch
- High cost for robotic simulators
- Less open source simulators
- Difficult to collaborate with robotic projects

# **After R.O.S**

## **Robotic Research & Software development**

- A common software platform to develop robotic applications
- The main goal of ROS is code re-use
- Open source simulators like Gazebo community become very active after R.O.S
- Collaboration in robotic development with universities and companies



# Who created ROS?

Discussing the brief history of ROS



# Who started ROS?

- 2007 : ROS project started under the name ***switchyard*** by the Stanford Artificial Intelligence Laboratory in support of the Stanford AI Robot STAIR (STanford AI Robot).
- Original Author : **Morgan Quigley**
- Main publication: ROS: an open-source Robot Operating System

## **ROS: an open-source Robot Operating System**

Morgan Quigley\*, Brian Gerkey†, Ken Conley†, Josh Faust†, Tully Foote†,  
Jeremy Leibs‡, Eric Berger†, Rob Wheeler†, Andrew Ng\*

\*Computer Science Department, Stanford University, Stanford, CA

†Willow Garage, Menlo Park, CA

‡Computer Science Department, University of Southern California

Computer Science Department, University of Southern California  
Los Angeles, California 90089, USA

# Who started ROS?

- **Morgan Quigley**
- Currently working at Open Robotics (<https://www.openrobotics.org/>)
- <http://people.osrfoundation.org/morgan/>
- **PhD** under Prof: Andrew Ng
  - Stanford Artificial Intelligence Lab



# Who created ROS?

- 2008 – 2013 : **Willow Garage**, a robotics research institute/incubator, USA
- Started ROS project
- Significant development of ROS happened here.
- <https://www.willowgarage.com/>



# Who created ROS?

- **PR2**: The robot fully running using ROS
- **Turtlebot**: Education & Research robot using ROS



# Who created ROS?

- 2013 – Now : Open Source Robotics Foundation or Open Robotics
- <https://osrfoundation.org>
- <https://www.openrobotics.org/>



# Who created ROS?

- Create and support Open software and hardware
- Research, Education and Product development
- Main products: ROS & Gazebo simulator



# Team: Open Robotics



**Morgan Quigley**

Chief Architect,  
Founder



**Brian Gerkey**

Chief Executive  
Officer, Founder



**Nathan Koenig**

Chief Technology  
Officer, Founder



**Tully Foote**

ROS Platform  
Manager

<https://www.openrobotics.org/team/>



# Why we use ROS for Robotics Programming?

Discussing the main features of ROS and the reason for using ROS in robotics research and companies



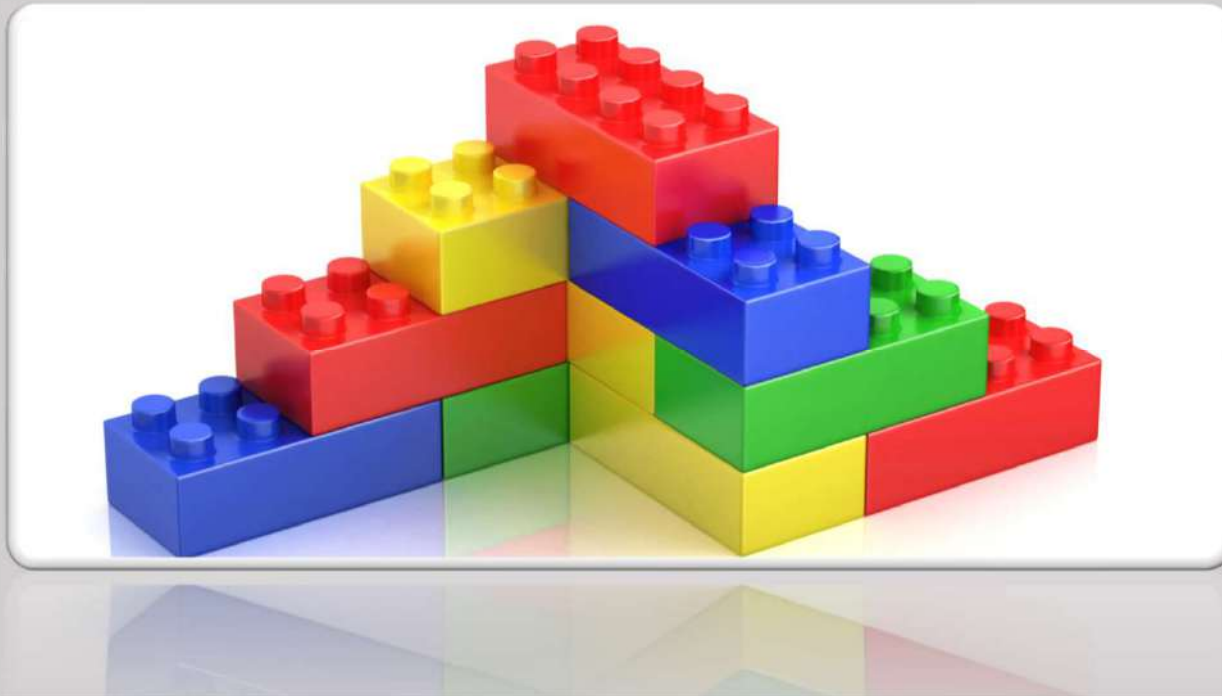
# Why we use ROS

- Code **reuse** in Robotics research and development



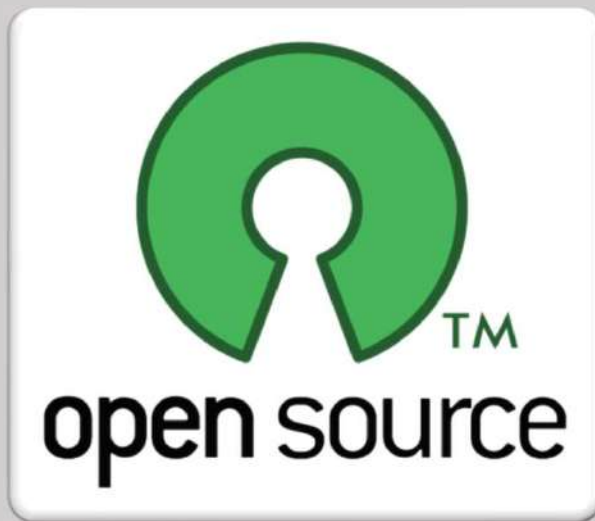
# Why we use ROS

- **Modularity** of code: [ **Nodes->Packages->Meta Packages-> Repositories** ]

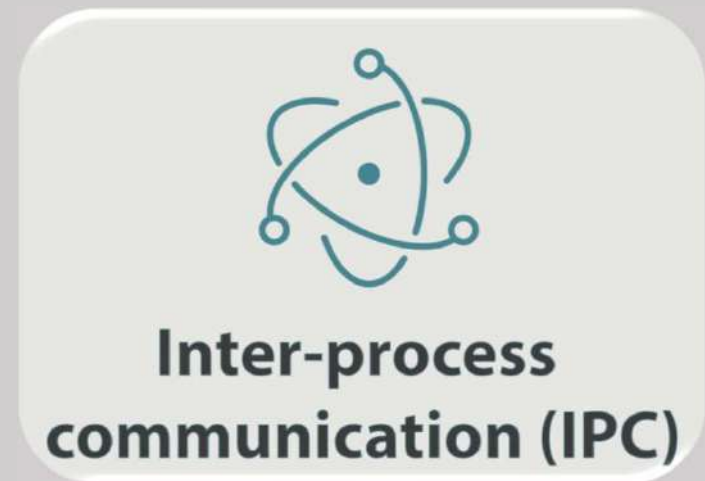


# Why we use ROS

- Free & Open Source framework to implement inter process communication



open source



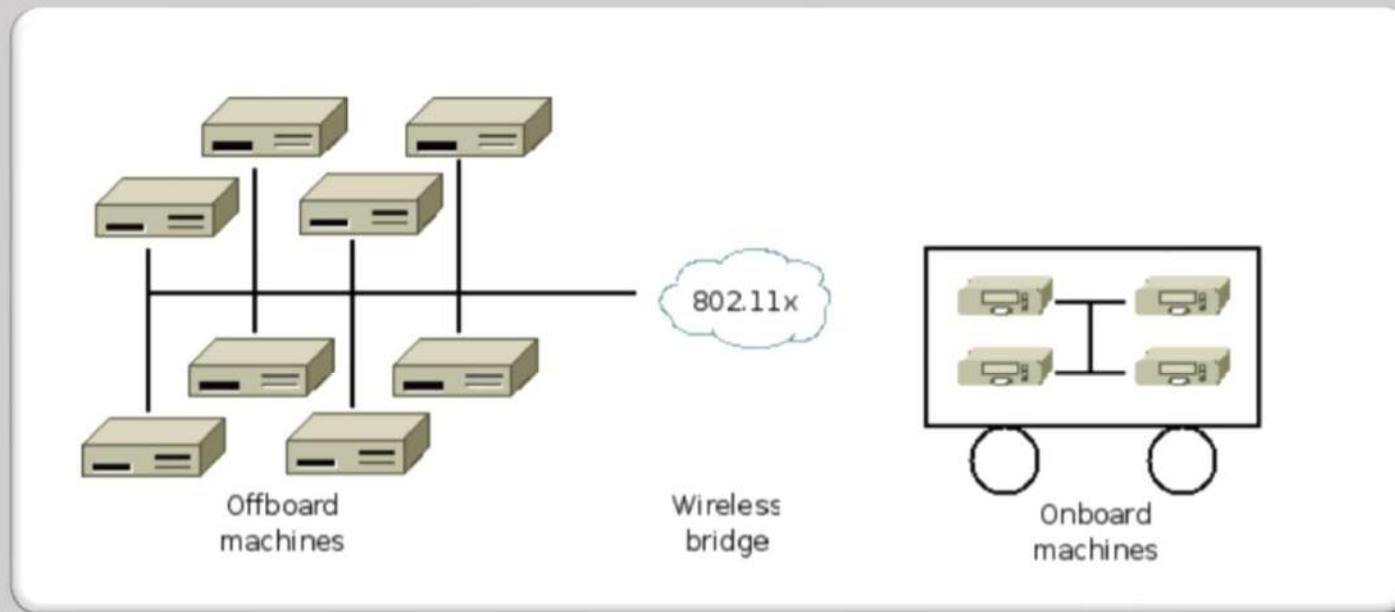
# Why we use ROS

- Great community support



# Why we use ROS

- Scalable framework, Distributed communication architecture



# Why we use ROS

- Popular programming language support for development: **C++**, **Python**
- Can create Agnostic libraries
- Easy test interface
- Various tools to visualize and debug robot data



# What is ROS Distribution

Discussing important concepts of ROS distributions

# What is a ROS Distribution?

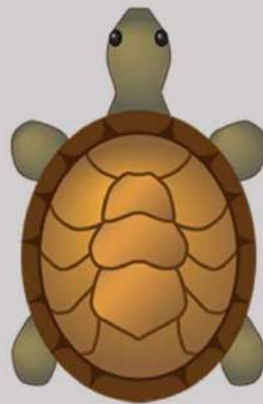
- Versioned set of ROS programs
- Similar to Linux Distribution: Ubuntu, Fedora, Kali Linux
- Relatively Stable set of ROS programs/packages
- Easy to maintain





# List of ROS Distributions

- 2010 – **ROS Box Turtle** : First ROS distribution
- Ubuntu support: 8.04,9.04,9.10 & 10.04
- Status: EOL (End of Life)



⌘Box Turtle

# List of ROS Distributions

- 2010 – **ROS C Turtle** : Second ROS distribution
- Ubuntu support: 9.04,9.10,10.04 & 10.10
- Status: EOL



# List of ROS Distributions

- 2011 – **ROS Diamondback** : Third ROS distribution
- Ubuntu support: 10.04,10.10 & 11.04
- Status: EOL



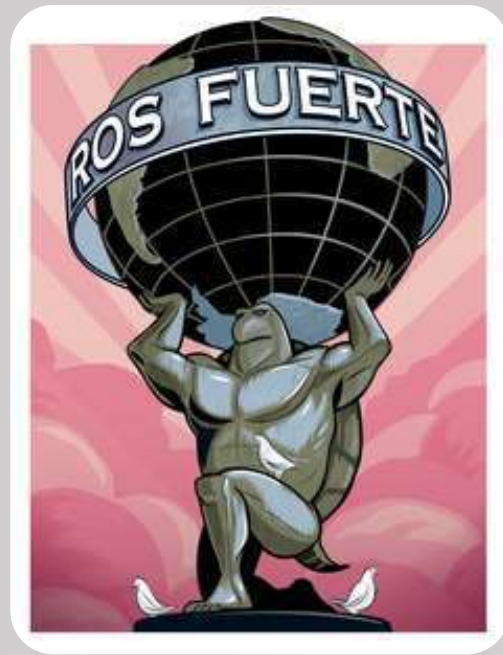
# List of ROS Distributions

- 2011 – **ROS Electric Emys**: Fourth ROS distribution
- Ubuntu support: 10.04,10.10, 11.04 & 11.10
- Status: EOL



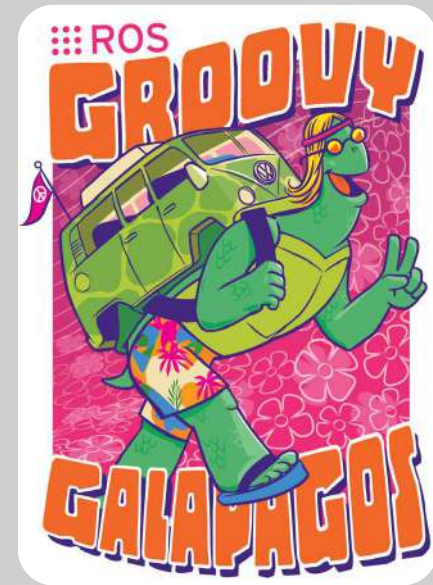
# List of ROS Distributions

- 2012 – **ROS Fuerte**: Fifth ROS distribution
- Ubuntu support: 10.04,10.10 & 12.04
- Status: EOL



# List of ROS Distributions

- 2013 – **ROS Groovy Galapagos**: Sixth ROS distribution
- Ubuntu support: 11.10, 12.04 & 12.10
- Status: EOL, July 2014



# List of ROS Distributions

- 2013 – **ROS Hydro Medusa**: Seventh ROS distribution
- Ubuntu support: 12.04, 12.10 & 13.04
- Status: EOL, May 2015



# List of ROS Distributions

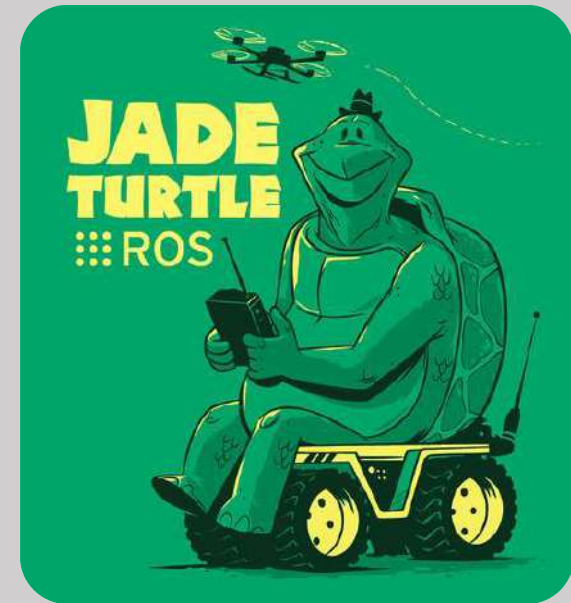
- 2014 – **ROS Indigo Igloo**: Eighth ROS distribution
- Ubuntu support: 13.10 & 14.04
- Status: April, 2019





# List of ROS Distributions

- 2015 – **ROS Jade Turtle**: Ninth ROS distribution
- Ubuntu support: 14.04, 14.10, & 15.04
- Status: May, 2017



# List of ROS Distributions

- 2016 – **ROS Kinetic Kame**: Tenth ROS distribution
- Ubuntu support: 15.10 & 16.04
- Status: April, 2021 , **Recommended**



# List of ROS Distributions

- 2017 – **ROS Lunar Loggerhead**: Eleventh ROS distribution
- Ubuntu support: 16.04, 16.10 & 17.04
- Status: May, 2019 ( EOL)



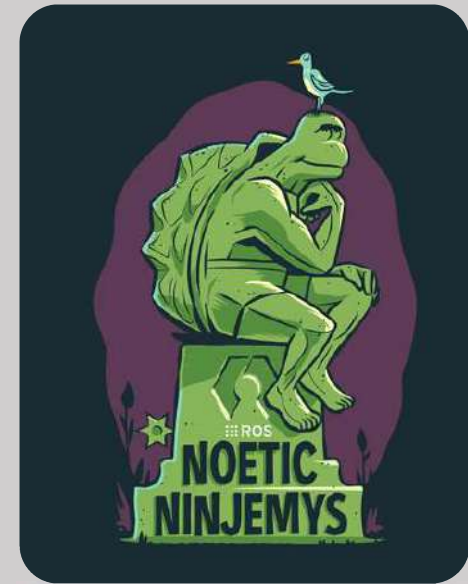
# List of ROS Distributions

- 2018 – **ROS Melodic Morenia**: Twelfth ROS distribution
- Ubuntu support: 17.10 & 18.04
- Status: May, 2023 (EOL)



# List of ROS Distributions

- 2018 – **ROS Noetic Ninjemys**: Thirteenth ROS distribution
- Ubuntu support: 20.04
- Status: May, 2025 (EOL), Latest
- List of ROS1 distributions
- <http://wiki.ros.org/Distributions>



# List of ROS 2 Distributions

- 2020 – **ROS 2 Foxy Fitzroy**
  - Ubuntu support: 20.04
  - Status: May, 2023 (EOL), Latest
- 
- **List of ROS 2 distributions**
  - <https://index.ros.org/doc/ros2/Releases/>





# Which are the robots supporting ROS?

Discussing list of robots working using ROS framework

# ROS supported Robots

- The Robots which are programmed using ROS
- Complete or partial ROS interface
- Research and commercial robots
- Open-Source reusable code and open hardware design
- List of ROS supported robots: <https://robots.ros.org/>





# ROS Supported Robots

- PR2(Personal Robot)
- <https://www.willowgarage.com/pages/pr2/overview>
- Processor : 2 X Quad Core i7 Xeon
- RAM : 24 GB
- Hard disk : 2 TB
- 2 x 7DOF arm
- Sensors : Laser scanners, Kinect



# ROS Supported Robots

- Turtlebot 2 : Mobile Robot
- <http://www.turtlebot.com/>
- Roomba Base
- Asus Xtion Pro Live
- Netbook loaded with ROS
- Educational and Research



# ROS Supported Robots



Baxter  
Collaborative Robot  
Rethink Robotics



REEM-C  
Full size biped humanoid robot  
PAL Robotics



Pepper  
Semi-humanoid robot  
Softbank Robotics



Tiago  
Service Robot  
PAL Robotics

# ROS Supported Robots



Robonaut – 2  
Robotic Astronaut  
NASA



REEM  
Full size humanoid robot  
PAL Robotics



Fetch  
Mobile Manipulation platform  
Fetch Robotics



TALOS  
Biped Robot  
PAL Robotics

# ROS Supported Robots



KINOVA-JACO  
Robotic Arm  
Kinova Robotics



ABB –Industrial Arm  
Industrial Arm  
ABB



UR3, UR5 & UR 10  
Collaborative Robot Arm  
Universal Robots

# ROS Supported Robots



Pioneer 3DX  
Differential drive robot  
Omron Adept Mobile Robots



Husky  
Unmanned Ground Vehicle (UGV)  
Clearpath Robotics



Fetch  
Mobile robotics platform  
Fetch Robotics

# ROS Robot Demos



# ROS Robot Demos



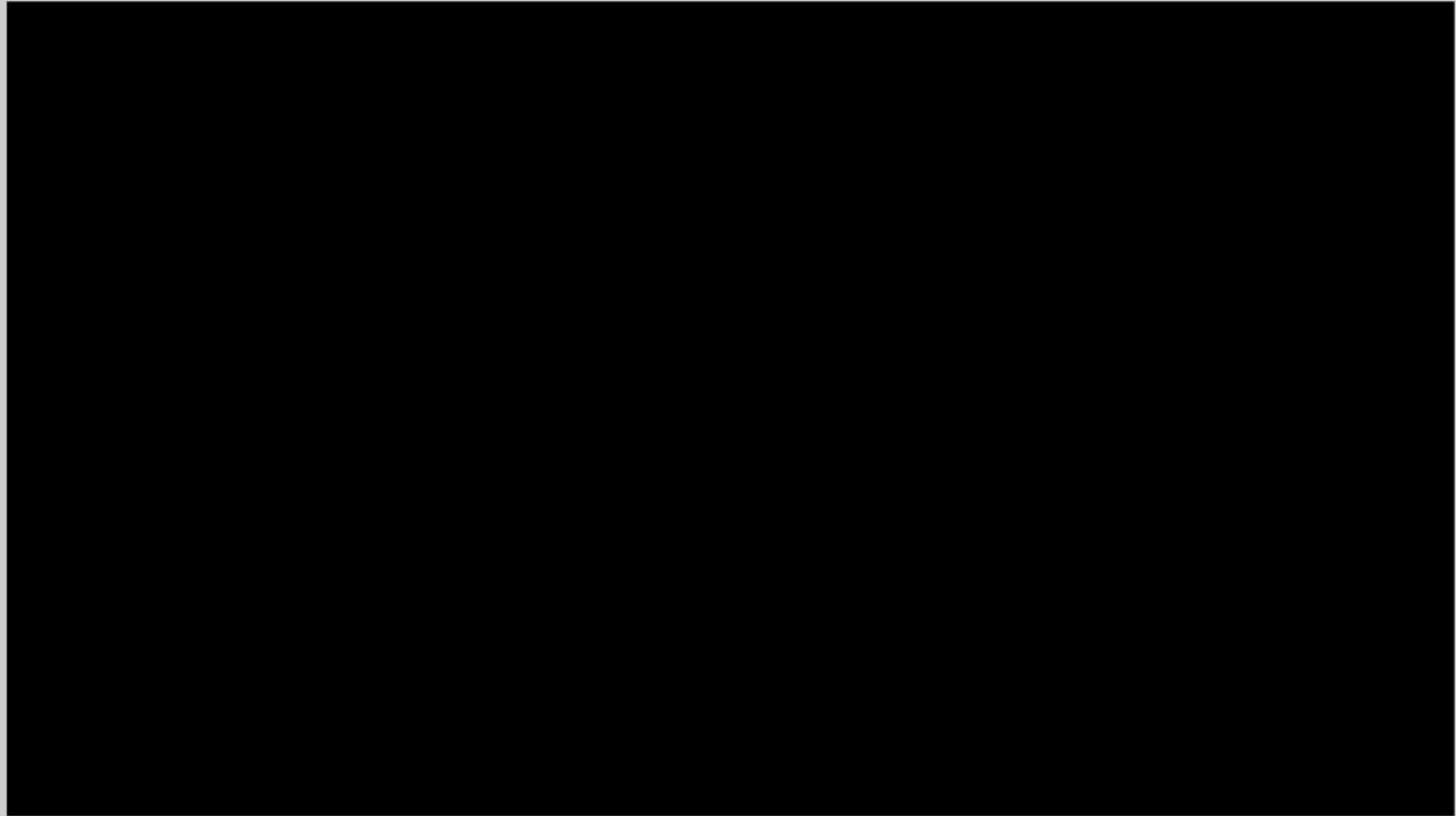


# ROS Robot Demos

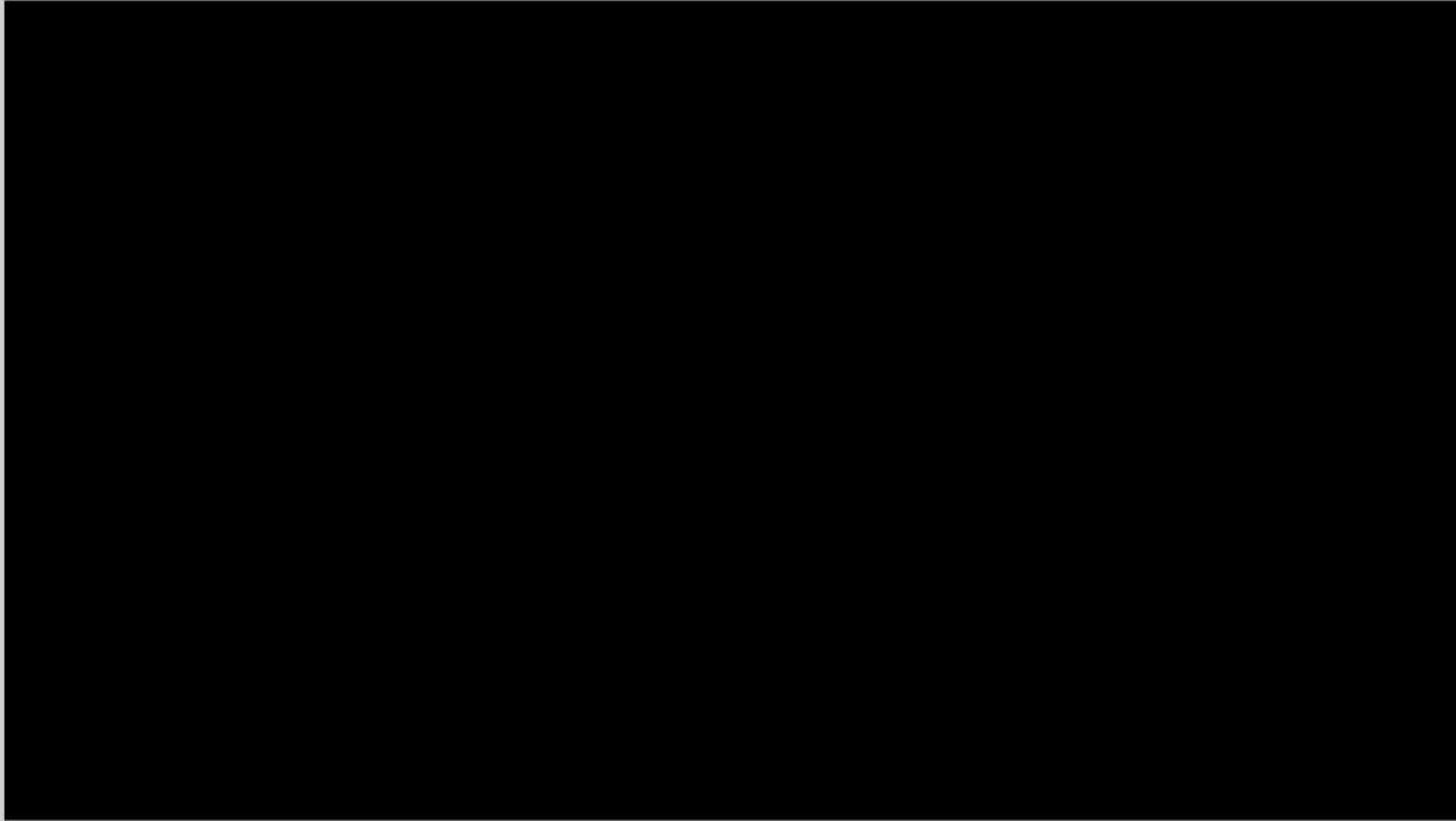
PR2 Making Popcorn at Automatica Fair 2012



# ROS Robot Demos



# ROS Robot Demos



# ROS Robot Demos



# Which are the Sensors supported by ROS??

Important sensors supported in ROS.

# ROS supported Sensors

- Sensors having a ROS interface
- Sensor data can be access from all ROS programs
- Sensor data can be visualized in Rviz
- List of ROS supported robots: <http://wiki.ros.org/Sensors>





Velodyne LiDAR



UTM – 30LX, Laser Scanner



Kinect, 3D Depth Sensor



Intel RealSense, 3D Depth Sensor

# Vision Sensors Supported in ROS



Velodyne LiDAR



UTM – 30LX, Laser Scanner



Kinect, 3D Depth Sensor



Intel RealSense, 3D Depth Sensor



# Vision Sensors Supported in ROS



ZED Camera, 3D Depth camera



USB Cam



Leap Motion, Hand tracker device



Orbbec Astra, 3D Depth camera

# Important GPS + IMU Supported in ROS



Micro Strain, IMU

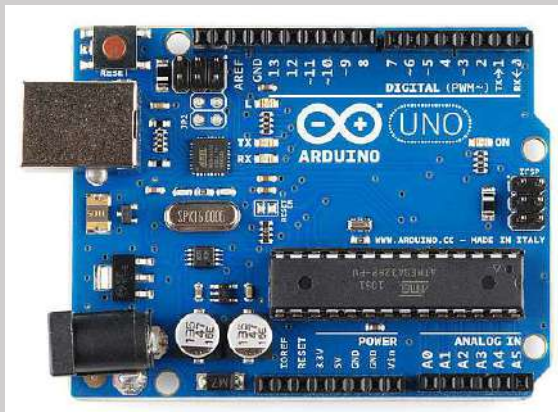


Applanix,  
IMU + GPS



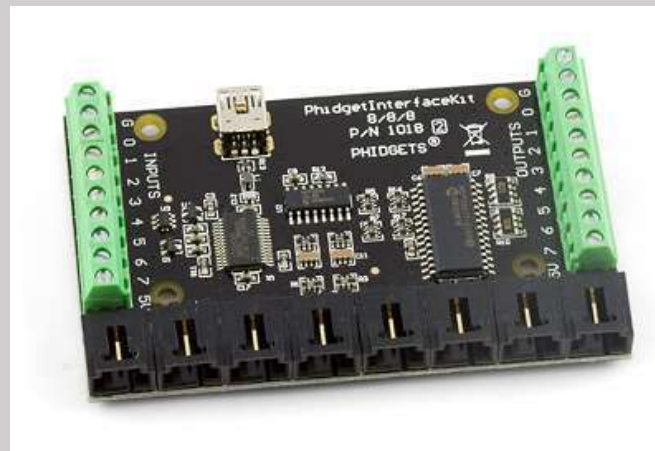
Xsens, IMU

# Important Sensor interface Supported in ROS



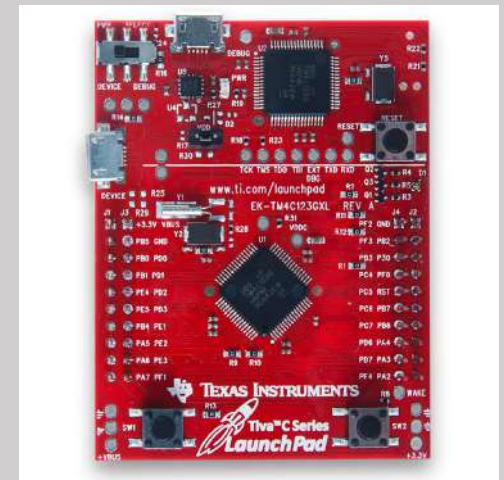
Arduino board

ROS Interface: **rosserial\_arduino**



Phidget

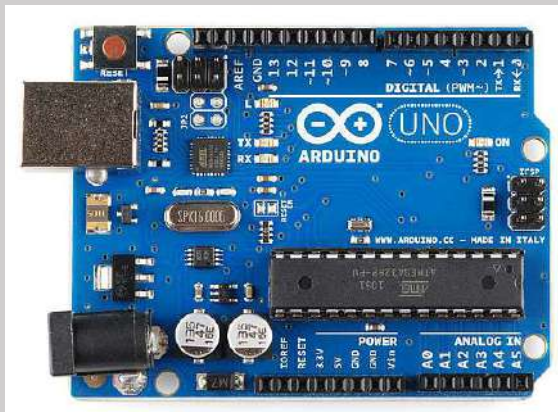
ROS Interface: **phidgets\_ros**



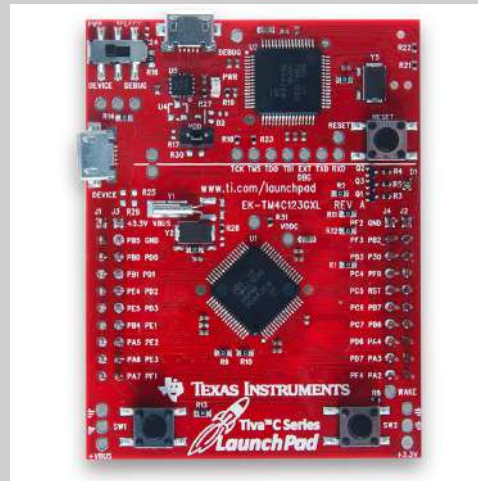
Tiva C Launchpad

ROS Interface : **rosserial\_tivac**

# Important Sensor interface Supported in ROS



Arduino board



Tiva C Launchpad



STM 32 Discovery

# Library and Simulators supported in ROS



# Who all using ROS?



# Who all using ROS?

## Silver Sponsors

ARM



dabit  
industries

Northwestern  
McCORMICK SCHOOL  
OF ENGINEERING  
MS in Robotics

QUALCOMM



## Bronze Sponsors

6 RIVER SYSTEMS

Apex.AI

AVIDBOTS

bonsai

BOSCH  
Invented for life

The Construct

MAGAZINO

NVIDIA

PAL  
ROBOTICS

PlusOne  
Robotics

rethink  
robotics.

Robotnik

Tango

VOY  
AGE

YUJINROBOT



# Who all using ROS?

- Robotics researchers/students
- Professionals working in Robotics companies
- Robotics Hobbyist
- Robot and sensor providers



# Jobs and research Opportunities

The screenshot shows the Glassdoor website interface. At the top, there's a navigation bar with links for Jobs, Company Reviews, Salaries, and Interviews. A search bar contains the text 'robotics engineer' and 'Los Angeles, CA'. Below the search bar, a list of job results is shown. The first job is 'Principal Robotics Engineer' at Hyperloop One, with an estimated salary of \$163k - \$230k. The second job is 'Technical Sales Engineer, Robotics - West Coast' at Sharp Electronics Corporation, with an estimated salary of \$57k - \$97k. The third job is 'SENIOR ROBOTICS ENGINEER' at Hyperloop Technologies, with an estimated salary of \$88k - \$123k. On the right side, a detailed view of the 'Principal Robotics Engineer' position is shown, including a list of preferred experience requirements.

**Principal Robotics Engineer**

Apply on Company Site Save

Job Company Rating Reviews

**PREFERRED EXPERIENCE:**

- PhD degree in robotics, computer science or electrical engineering.
- Experience with GPU programming, ROS
- Experience in writing codes for any of the following systems: sensor fusion, multi-target, tracking system, computer vision algorithms for object recognition.
- Experience with Agile or any other SW development methodologies.
- Experience with Python, shell scripts.
- Hands-on experience in working on any types of active and passive sensor data acquisition pipeline.
- Any publication records on related fields.

<https://www.glassdoor.com/>

# Future Scope

- Software is the key aspect in any robot. If there is no computer program running inside a robot, its just a piece of hardware.
- ROS allows programmers to prototype and deploy a robot faster than any other framework available now.

# Future Scope

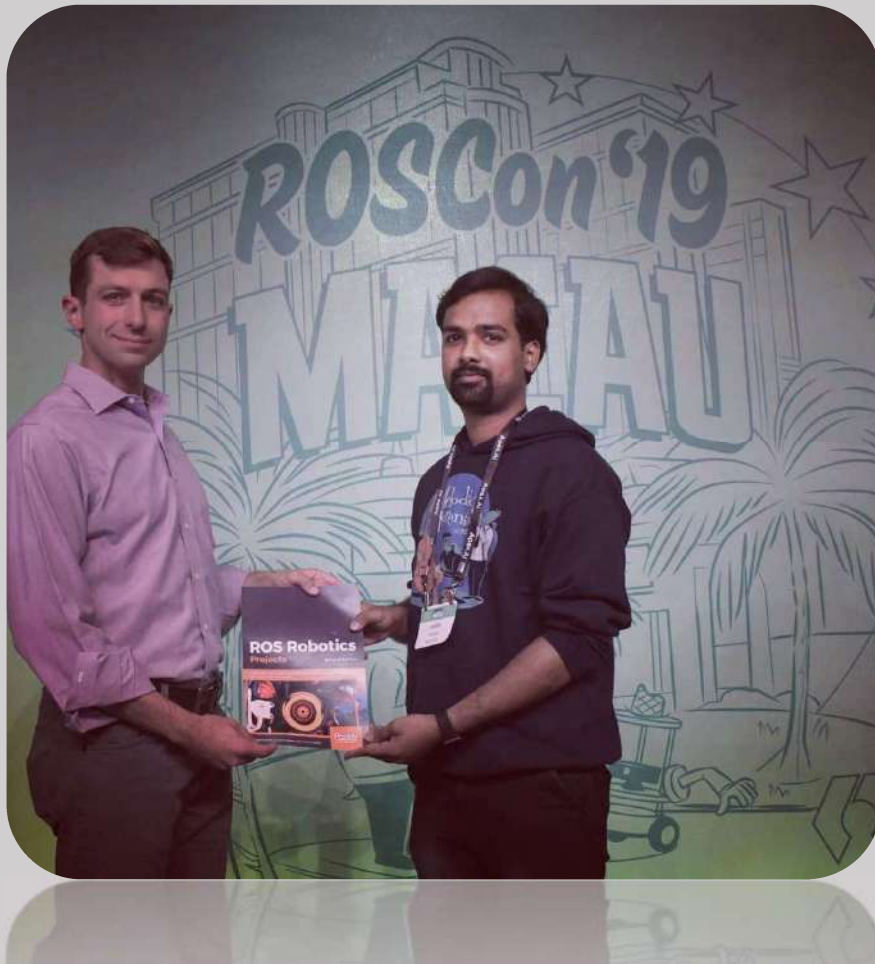
- Lot of opportunities coming for Robotics developers across the globe in which ROS is an important skillset.
- Other than Robotics companies, almost all the universities working in Robotics using ROS in someway.
- ROS is becoming a standard in robotics programming

# ROSCon 2019



<https://roscon.ros.org/2019/>

# ROSCon 2019



<https://roscon.ros.org/2019/>

# ROSCon 2016 & ROSCon 2015 Videos

- <https://roscon.ros.org/2016/>
- <https://roscon.ros.org/2015/>



# Getting ROS support

- ROS Mailing list: [ros-users@ mailing list](mailto:ros-users@ mailing list)
- ROS Wiki: <http://wiki.ros.org/>
- ROS Answers: <http://answers.ros.org/questions/>
- Issue trackers: <http://wiki.ros.org/Tickets>
- ROS Discourse Forums:  
<https://discourse.ros.org/>

# Getting ROS news

- <http://www.ros.org/news/>
- Latest updates of ROS release, new jobs in ROS, new packages in ROS