ECE 4703
Mobile Autonomous Robots

Jenny Zhen Yu
zhenyu@cpp.edu
Department of Electrical and Computer Engineering
California State Polytechnic University, Pomona

Lecture 1: Mobile Autonomous Robots
Introduction

Outline

- **Mobile Autonomous Robots Introduction**
  - Robot software platform
  - Need and future
  - About ROS
  - Features of ROS
  - Configuring the ROS development environment
  - ROS installation
  - ROS operation test

What do These Two Things Have in Common

Personal Computer

Personal Phone

https://pixabay.com/en/computer-desktop-modern-device-154114/ , https://commons.wikimedia.org/wiki/File:Samsung-Galaxy-S7-Edge-Smartphone.jpg

Hardware Modules Enable Various Combinations

Personal Computer

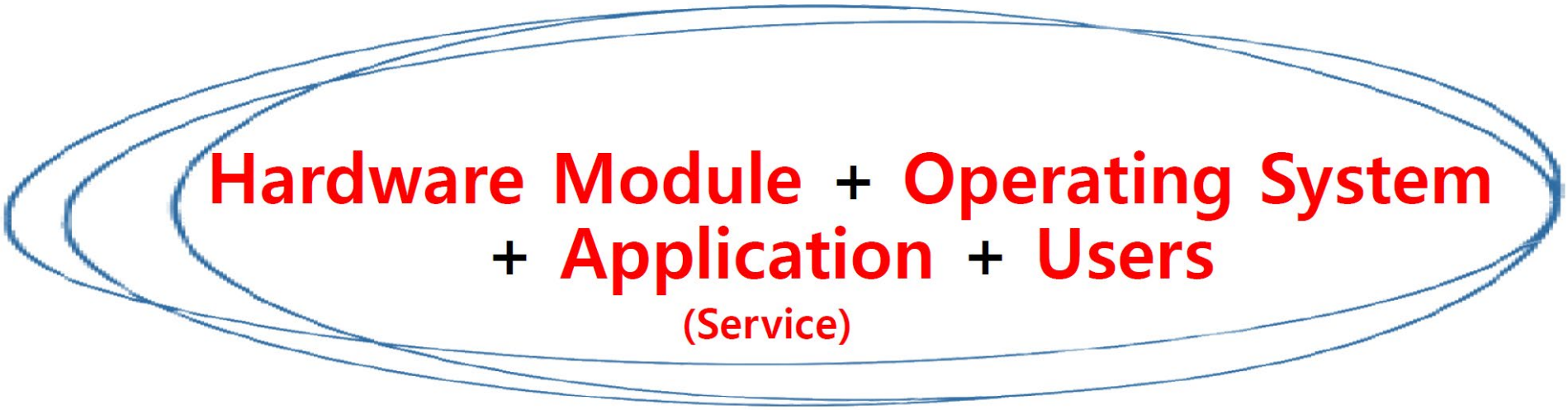
Personal Phone

http://www.techpowerup.com/, http://global.samsungtomorrow.com/

Ecosystems

	PC (Personal Computer)	PP (Personal Phone)	PR (Personal Robot)
User			
Service			
Operating System			
Hardware			

Division of Labor in Product Ecosystem



Operating System and Application

Personal Computer

Personal Phone

http://en.wikipedia.org/wiki/Linux, https://www.apple.com/, http://windows.microsoft.com/, The icons are trade mark of each company

Changes Brought by the Software Platform

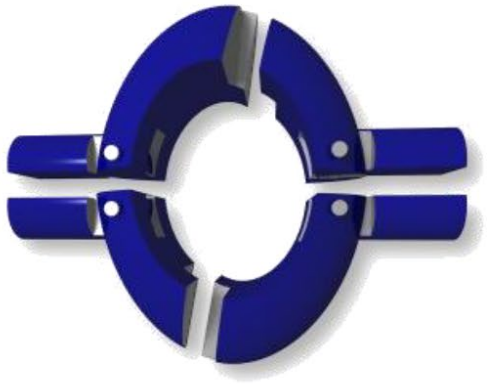
[The first commercially available mobile phone (?) In 1983 Motorola DynaTAC 8000 and developer Martin Cooper, and the evolution of mobile phone]

http://blog.cartoys.com/date/2013/04/, https://line.do/the-evolution-of-cell-phones/r5q4ui/vertical

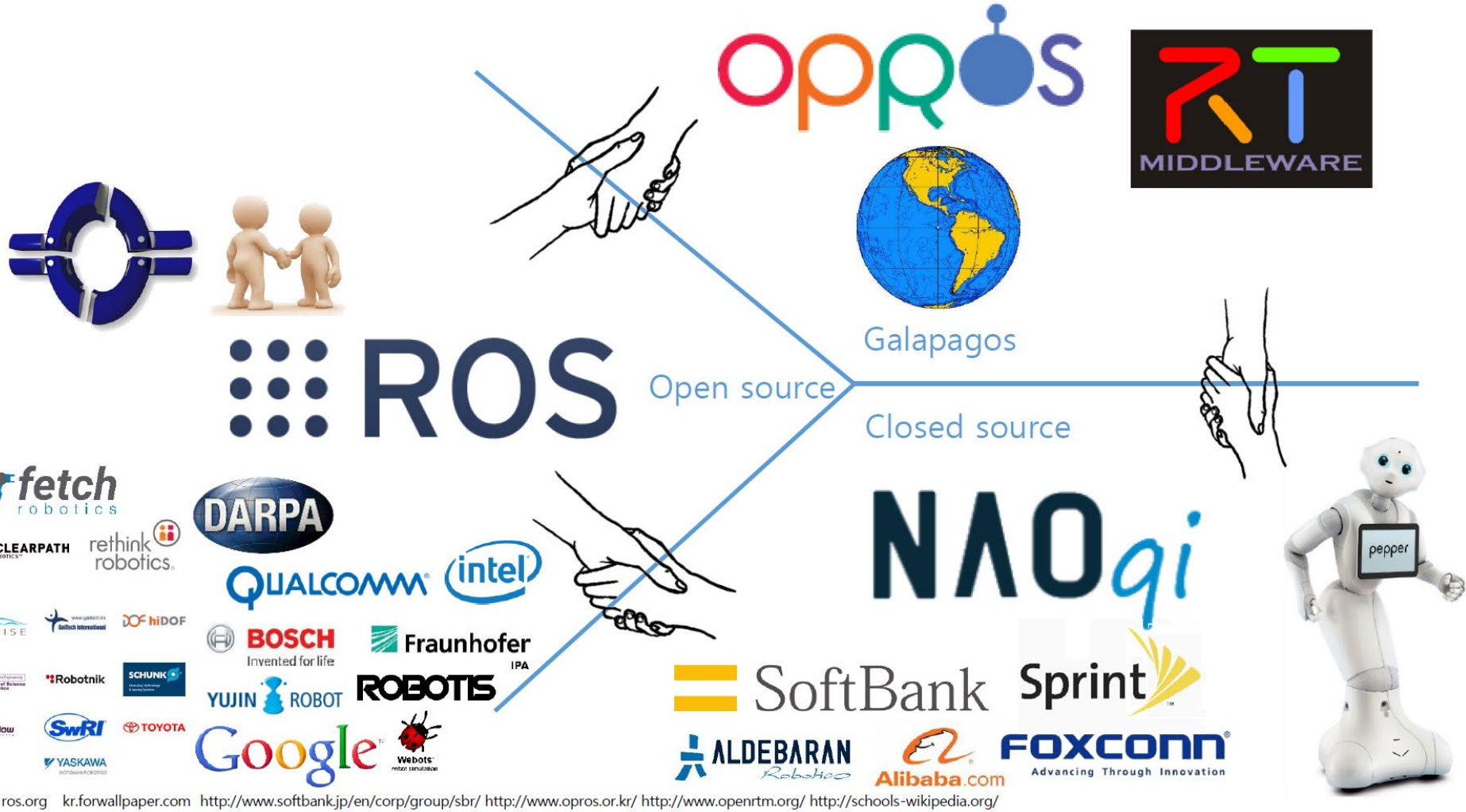
Changes Brought by the Software Platform

- Hardware interface integration
- Hardware abstraction, standardization, and modularization
- Price ↓, Performance ↑
- Separation of hardware, operating system and application
- Focus on services that meet your needs!
- User increases! Purchase and feedback, forming a new ecosystem

Various Robot OS



Major Robot Operating System



http:// ros.org kr.forwallpaper.com http://www.softbank.jp/en/corp/group/sbr/ http://www.opros.or.kr/ http://www.openrtm.org/ http://schools-wikipedia.org/

Future Robot Software Platform

- Establish **an interface** between the hardware platform and the software platform
- **Modular hardware** platform proliferation
- Applications can be written without knowledge of hardware
- More **software personnel** can enter the robotics field and participate in the robotics product
- Focus on **services** to be provided to users
- Formation of **User** layer and **feedback** by delivering demanded service
- An opportunity for the rapid advancement of robot development

About ROS

ROS is an open-source, meta-operating system for your robot. It provides the services you would expect from an operating system, including hardware abstraction, low-level device control, implementation of commonly-used functionality, message-passing between processes, and package management. It also provides tools and libraries for obtaining, building, writing, and running code across multiple computers.

http://www.ros.org/wiki/

Software Framework

- Software framework for developing robot software
  - It is possible to jointly develop complex programs by finely dividing them with message exchanging method between nodes.
  - Supports command tool, visualization tool Rviz, GUI toolbar rqt, 3D simulator Gazebo
  - Supports modeling, sensing, recognition, navigation, and manipulation functions commonly used in robotics
  - Create Robotics Ecosystem!

http://www.ros.org/about-ros/

Purpose ROS

Building an ecosystem that enables robotics software development to be collaborated on a global level!

http://imgfave.com/

Is ROS a New OS

- **Operating System**
  - **General purpose computer**
    - Windows(Windows XP, 7, 8 ...)
    - Linux(Ubuntu, Redhat, Fedora, Mint, Gentoo ...)
    - MAC(OS X ...) etc
  - **Smart phone**
    - Android, iOS, Windows Phone, Symbian, RiMO, Tizen etc
- **ROS** = Robot Operating System
- **ROS** is Meta-Operating System

Meta- Operating System

- Meta-Operating System (Meta-Operating System): It is not a precisely defined term but it can be regarded as a system that performs scheduling, loading, monitoring, error handling and utilizing distributed computing resources as a virtualization layer between applications and distributed computing resources.

- It is not a traditional operating system like Windows, Linux, and Android. Rather, ROS uses the traditional operating system (Linux, Windows, OS-X, and Android).

- It uses the existing operating system's process management system, file system, user interface, program utilities (compiler, thread model, etc.). In addition, it provides essential functions for developing robot application software such as data transmission / reception, scheduling and error handling among many different types of hardware in a library form.

- In addition, it develops, manages and provides various application programs based on the robot software framework, and has an ecosystem that distributes packages developed by users.

Meta- Operating System

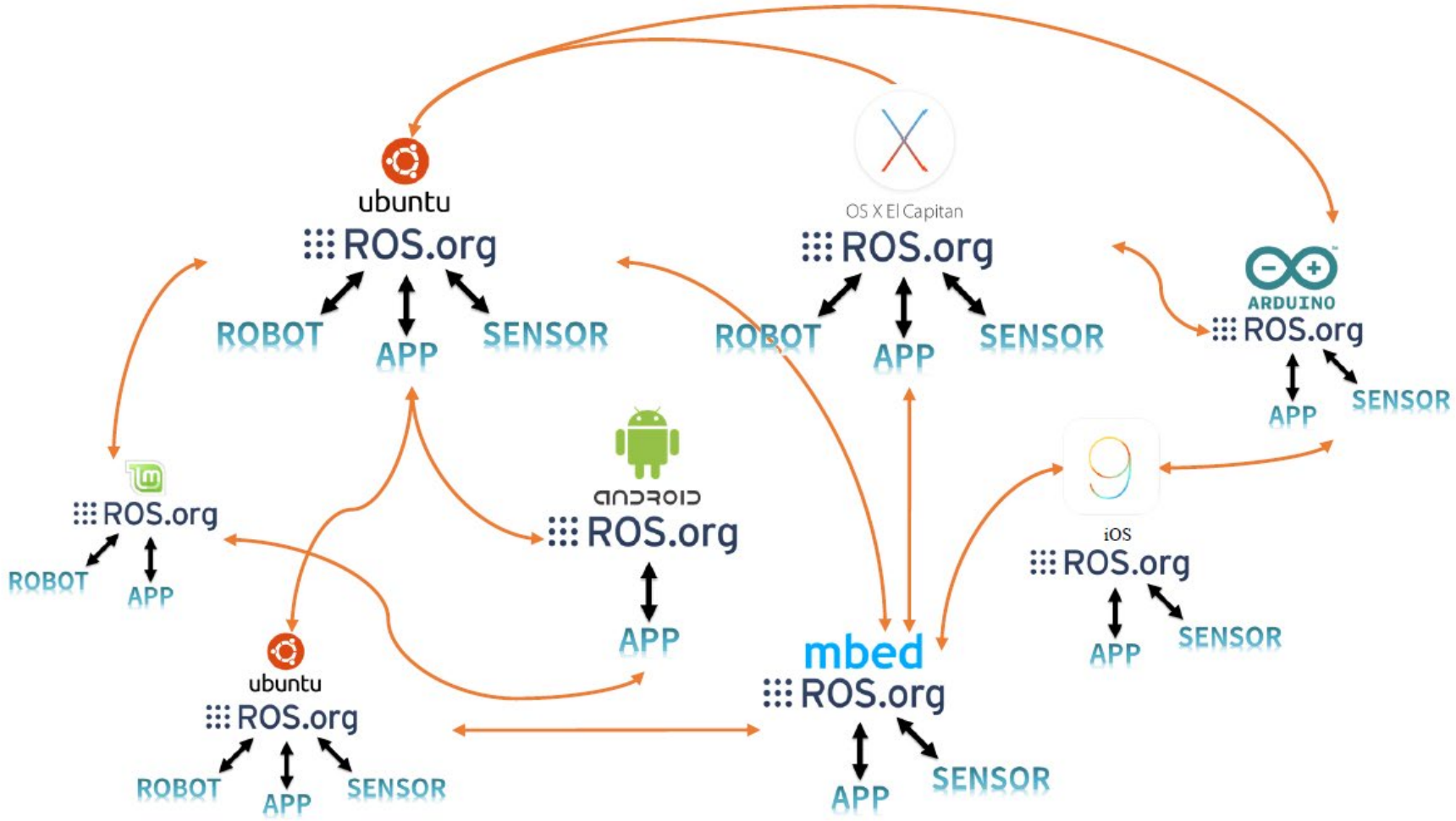
Device drivers, libraries, debug tools, message communication
Drivers, compilation tools, installers, package creation and release

ROBOT ⟷ Meta-Operating System ⟷ SENSOR

↕

APP

Support Communication Between Different Devices



ROS-enabled Operating System

- **Traditional operating system**
  - Ubuntu, OS X, Windows, Fedora, Gentoo, OpenSUSE, Debian, Raspbian, Arch, and QNX Realtime OS. (There can be functional limitations in some OS)
  - Partially available for Android and iOS, smartphone operating systems
  - In case of microcontroller unit (MCU) which can not be equipped with OS, it provides a library to communicate via serial communication, Bluetooth, and LAN
  - Basically it is recommended to run on **Ubuntu**, OS X!

ROS 2.0 supports three major operating systems

ROS Configuration

http://wiki.ros.org/APIs

ROS Versions

Distro	Release date	Poster	*Turtle*, turtle in tutorial	EOL date
ROS Noetic Ninjemys (**Recommended**)	May 23rd, 2020			May, 2025 (Focal EOL)
ROS Melodic Morenia	May 23rd, 2018			June 27, 2023 (Bionic EOL)
ROS Lunar Loggerhead	May 23rd, 2017			May, 2019
ROS Kinetic Kame	May 23rd, 2016			April, 2021 (Xenial EOL)
ROS Jade Turtle	May 23rd, 2015			May, 2017
ROS Indigo Igloo	July 22nd, 2014			April, 2019 (Trusty EOL)
ROS Hydro Medusa	September 4th, 2013			May, 2015
ROS Groovy Galapagos	December 31, 2012			July, 2014
ROS Fuerte Turtle	April 23, 2012			--
ROS Electric Emys	August 30, 2011			--
ROS Diamondback	March 2, 2011			--
ROS C Turtle	August 2, 2010			--
ROS Box Turtle	March 2, 2010	 Box Turtle		--

Current ROS Ecosystem

Developer, User

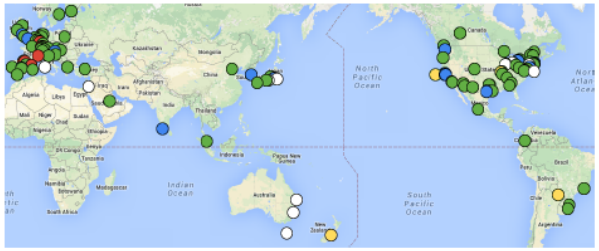
APP

ROS

ROBOT, SENSOR

Robot, Sensor Company

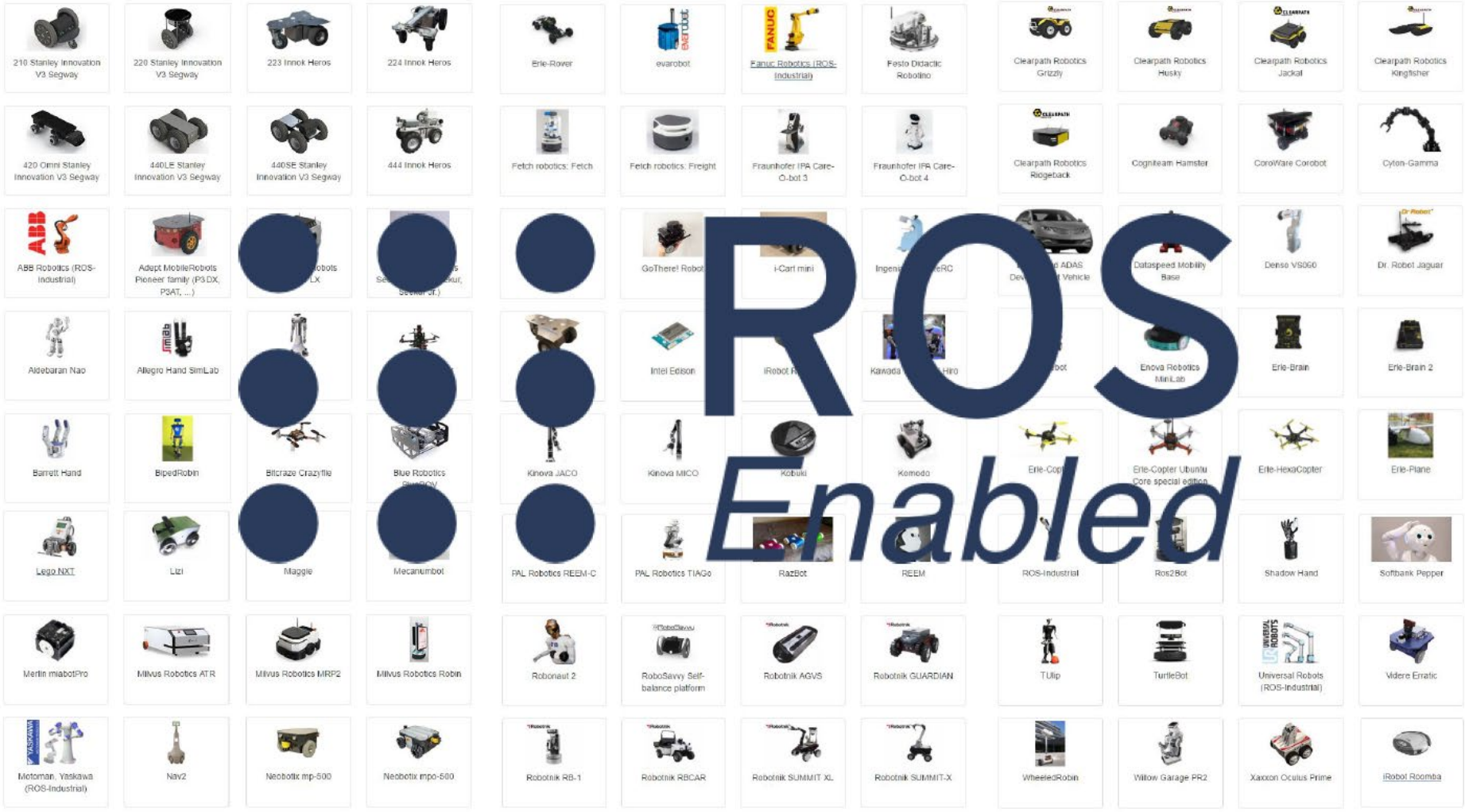
- More than 5,000 packages
- Provides 2,818 official packages (Indigo / March, 2017)
- 13,441,711 .deb packages downloads (July 2017)
- 18,839 Wiki pages (July 2017)



- Application — fetch beer, elevator ···
- Simulator — gazebo, player/stage, STDR Simulator ···
- Intelligence module — navigation, action, grasping ···
- Library — tf, PCL, OpenCV, OpenRave ···
- Device driver — camera_drivers, urg_node ···
- Debug tool — rviz, rqt_graph, rosbag, rostopic ···
- Message communication — rosmaster, rosmsg, rosservice ···
- Execution tool — rosrun, roslaunch ···
- Compile tool — catkin_make, rosbuild ···
- File system — roscd, rosls ···
- Install tool — rosinstall ···
- Programing Language — C++, Python, Lisp, Java, Ruby, MATLAB등

- Supports more than 90 kinds of robots, more than 80 kinds of sensors

Current ROS Ecosystem



ROS Enabled

Feature of ROS

❑ Communication infrastructure

- Provides data communication between nodes
- Support for message transfer interface, which is commonly referred as the middleware

- Message parsing function
  - Provides communication system frequently used in robot development
  - Message transfer interface between nodes facilitating encapsulation and code reuse

- Message Record and Play
  - Messages that are transmitted/received between nodes can be stored and reused as needed
  - It is possible to repeat an experiment based on stored messages, and it is easy to develop algorithm

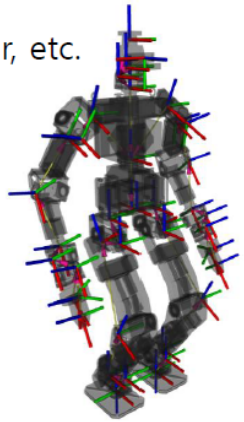
- Use of various programming languages due to the use of messages
  - Since data exchange between nodes use messages, each node can be written in different languages
  - Client libraries: roscpp, rospy, roslisp, rosjava, roslua, roscs, roseus, PhaROS, rosR

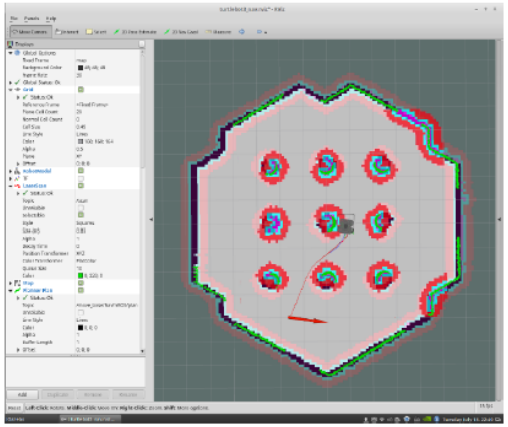
- Distributed parameter system
  - Variables used in the system are created as global key values so they can be shared, modified and applied in real-time

Feature of ROS

❑ Various functions related to robots

- Define a standard message for a robot
  - Modularization by defining standard message such as camera, IMU, laser sensor / odometry, navigation data such as route and map, inducing collaborative work, and improving efficiency
- Robot geometry library
  - Provides TF to calculate the relative coordinates of robot, sensor, etc.
- Robot description language
  - XML document describing physical characteristics of the robot
- Diagnostic system
  - Provides diagnostic system to grasp the state of the robot
- Sensing / recognition
  - Sensor drivers, libraries for sensing / recognition
- Navigation
  - Estimation of poses (position / posture) of robots commonly used in robots, provision of self position estimation in the map
  - SLAM required for map creation, and Navigation library for navigating to destinations within the created map
- Manipulation
  - Provides various Manipulation libraries to support IK and FK used in robot arm as well as pick and place of application
  - Provides GUI manipulation tools (MoveIt!)

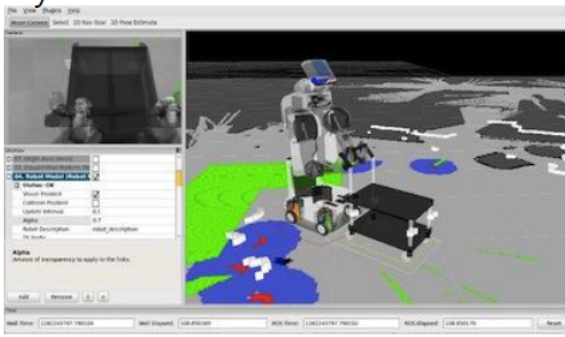


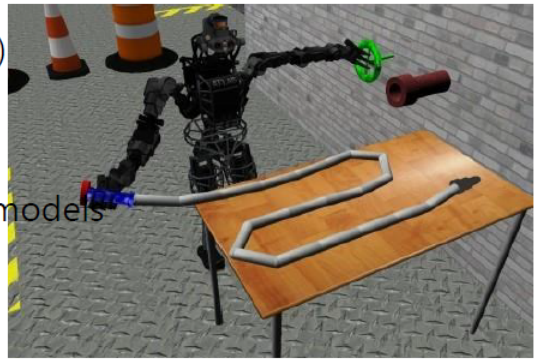


Feature of ROS

❑ Various development tools

- Provides various development tools needed for robot development
- Improving the efficiency of robot development
- Command-Line Tools
  - Access to the robot and use ROS functions only with commands provided by ROS without GUI
- RViz
  - Provide powerful 3D visualization tool
  - Visualize sensor data such as laser, camera, etc.
  - Represent robot outline and planned motion
- RQT
  - Provides Qt-based framework for developing graphic interface
  - Displays connection information among nodes (rqt_graph)
  - Values such as encoder, voltage, numbers that change over time (rqt_plot)
  - Records and plays data in the form of message (rqt_bag)
- Gazebo
  - 3D simulator with physics engine. Supports robot, sensor, environmental models
  - Highly Compatible with ROS





http://www.ros.org/core-components/

ROS Application Development Environment

- Hardware: Desktop or laptop using Intel or AMD processor
- Operating System: Ubuntu 20.04.6 LTS
- ROS: Noetic Ninjemys

ROS Installation

- **ROS Installation**
  - http://wiki.ros.org/noetic/Installation/Ubuntu
  - **Appendix A**
  - **Appendix B**
- **ROS environment setting**

ROS Operation Test

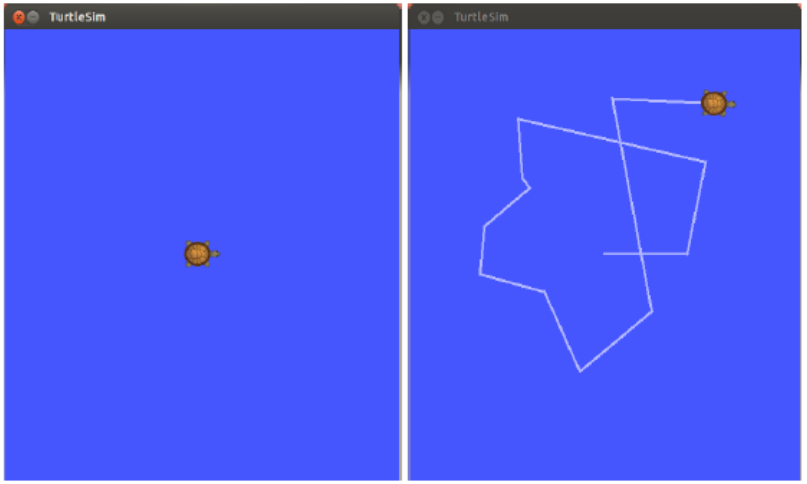
- **turtlesim package**

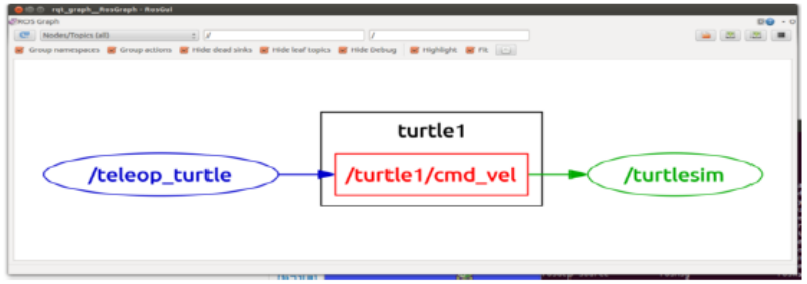
  - roscore

  - rosrun turtlesim turtlesim_node

  - rosrun turtlesim turtle_teleop_key

  - rosrun rqt_graph rqt_graph





Integrated Development Environment (IDE) Available on ROS

- http://wiki.ros.org/IDEs

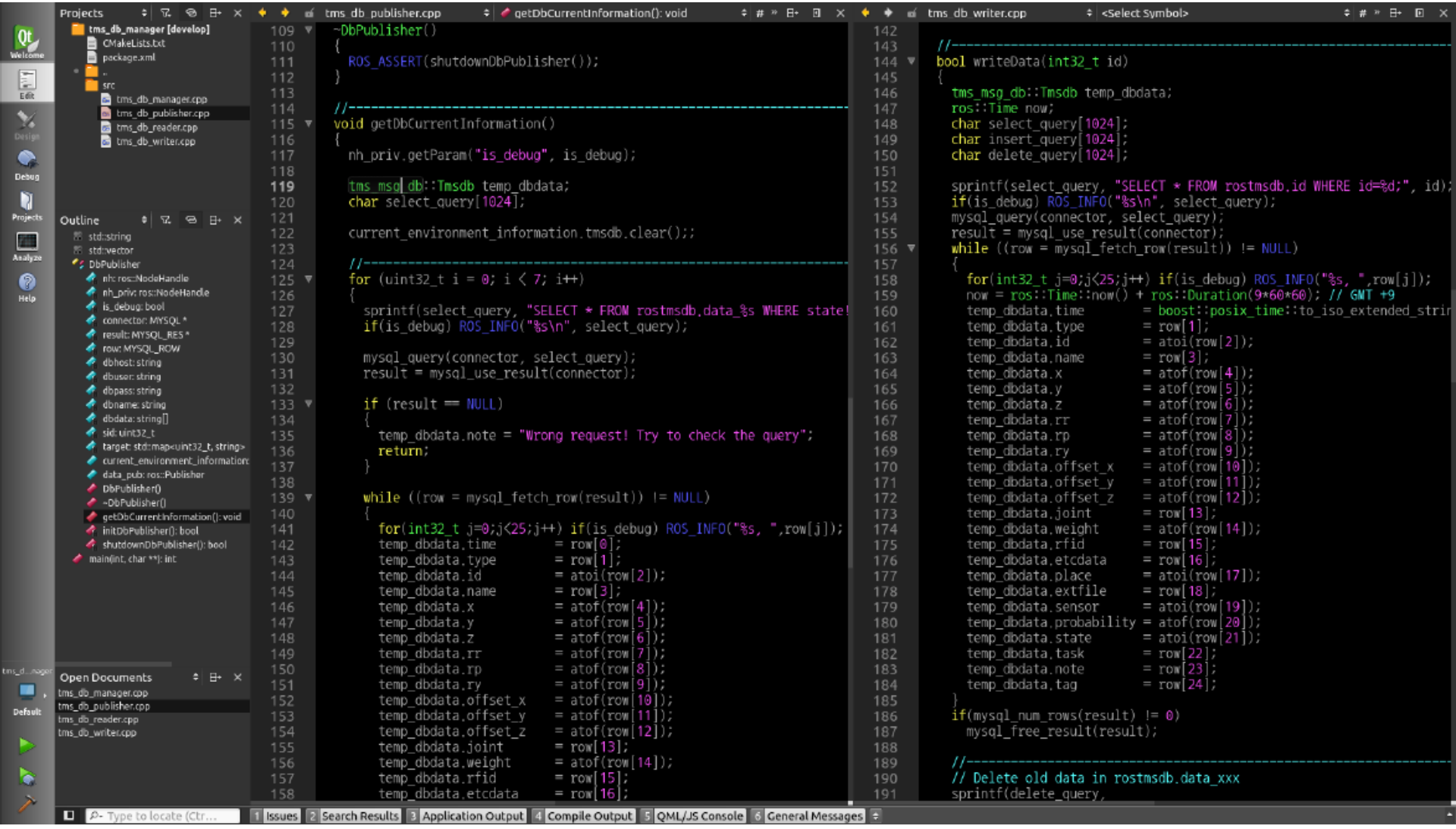
- Recommendation 1 : **Qtcreator** + Qt Creator Plugin for ROS
  - Install: sudo apt-get install qtcreator
  - Advantage: Can be Used as 'CmakeLists.txt', Easy to develop 'rqt plug-in' & 'GUI'

- Recommendation 2 : **Visual Studio Code** + ROS Extension
  - Install: https://code.visualstudio.com/
  - Advantage: A light text editor oriented, Fast
  - Similar to 'Atom', 'Sublime Text', 'Clion' etc.

- Recommendation 3 : **Eclipse**
  - Install: http://www.eclipse.org/
  - Advantage: A familiar IDE that many people use (but, Heavy!)

http://wiki.ros.org/IDEs

Integrated Development Environment (IDE) Available on ROS



Reference

✓ Download link

✓ Language:

English, Chinese, Japanese, Korean

"ROS Robot Programming"

A Handbook is written by TurtleBot3 Developers

Reference

- R. Siegwart, I. R. Nourbakhsh, D. Scaramuzza. Introduction to Autonomous Mobile Robots. MIT Press, 2nd Edition, 2011, ISBN-10: 0262015358.
- Y. Pyo, H. Cho, R. Jung, and T. Lim, ROS Robot Programming, ROBOTIS Co., Ltd., 2017, ISBN 979-11-962307-1-5
- J. O'Kane, A Gentle Introduction to ROS, CreateSpace Independent Publishing Platform, 2013, ISBN-13: 978-1492143239