# 로봇 운영체제 ROS



주아다하고 기기기당하다 지식-지능해 메카트로닉스 실험실 신승EH 교수

PPT 자료 출처: 표윤석 외 3명, 『ROS 로봇 프로그래밍 기초 개념부터 프로그래밍 학습, 실제 로봇에 적용까지』, 루비페이퍼(2017)



# 다양한 하드웨어의 결합이 가능한 하드웨어 모듈



**Personal Computer** 



**Personal Phone** 

<sup>[1] &</sup>lt;a href="http://www.techpowerup.com/">http://www.techpowerup.com/</a>[2] <a href="http://global.samsungtomorrow.com/">http://global.samsungtomorrow.com/</a>



## 운영 체제 (Operating System) + 애플리케이션(App)







**Personal Phone** 

<sup>[1]</sup> http://en.wikipedia.org/wiki/Linux

<sup>2]</sup> https://www.apple.com/

B] <a href="http://windows.microsoft.com/">http://windows.microsoft.com/</a> The icons are trade mark of each company



### 소프트웨어 플랫폼이 가져온 변화

- 하드웨어 인터페이스 통합
- 하드웨어 추상화 규격화 모듈화
- 가격 ↓, 성능 ↑
- <u>하드웨어</u> <u>운영체제</u> <u>애플리케이션 분리</u>
- 사용자 수요에 맞는 서비스에 집중!
- <u>유저</u>증가! 구매와 피드백, 새로운 생태계의 선환 구조 형성









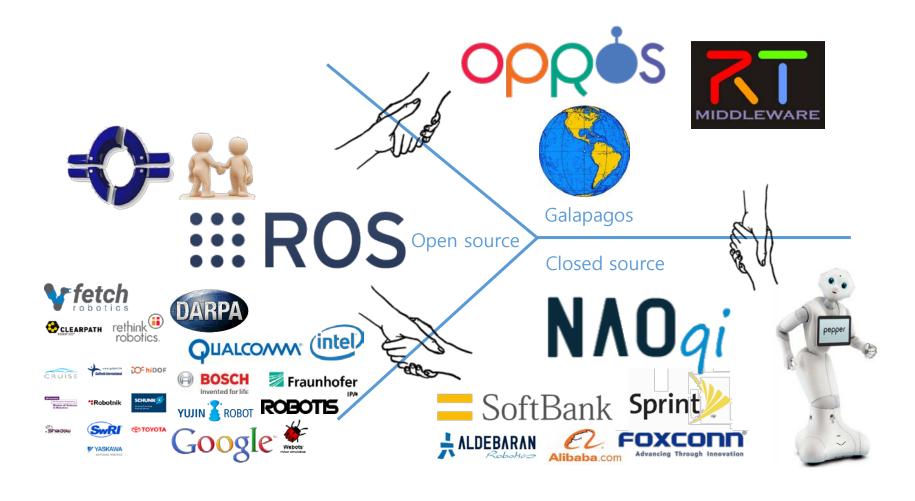




# र्रंभ रेल्स्रासा



# - 주요 로봇 운영체제



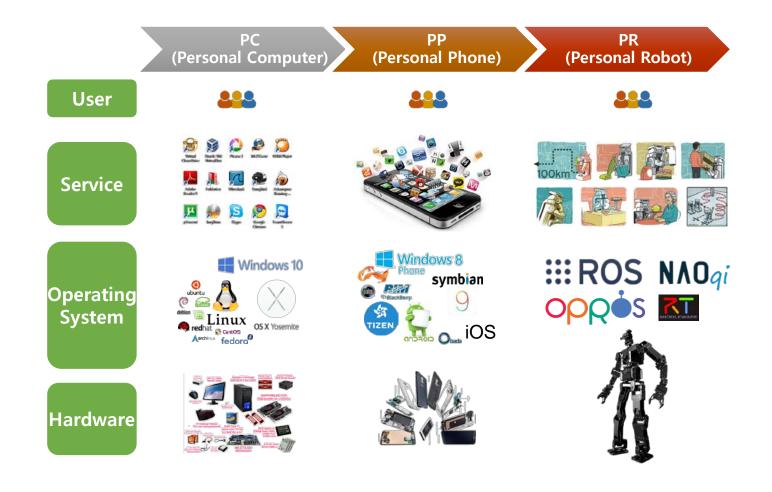
- [1] http://ros.org
- [2] kr.forwallpaper.com
- [3] http://www.softbank.jp/en/corp/group/sbr/
- [4] http://www.opros.or.kr/
- [5] http://www.openrtm.org/
- [6] http://schools-wikipedia.org/



### - 로봇 소프트웨어 플랫폼이 가져온 미래

- 하드웨어 플랫폼과의 소프트웨어 플랫폼간의 인터페이스 확립
- 모듈형 하드웨어 플랫폼 확산
- 하드웨어에 대한 지식이 없어도 응용 프로그램 작성 가능
- 더 많은 소프트웨어 인력물이 로보틱스 분야로 진입, 로봇 제품에 참여 가능
- 유저에게 제공할 <mark>서비스</mark>에 집중
- 실수요가 있는 서비스 제공으로 유저계층 형성 및 피드백
- 로봇 개발이 급속도로 발전 할 수 있는 계기

## 로봇 운영체제의 미래





# र्यु देन्त्रिमासा ROS 1714



# ROS 2t?

ROS is an open-source, meta-operating sy stem for your robot. It provides the services y ou would expect from an operating system, in cluding hardware abstraction, low-level device control, implementation of commonly-used fu nctionality, message-passing between process es, and package management. It also provides tools and libraries for obtaining, building, writ ing, and running code across multiple comput ers.

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





### 소프트웨어 프레임위크



- 로봇 소프트웨어를 개발하기 위한 소프트웨어 프레임워크(로보틱스 생태계 생성)
- 노드간에 메시지 교환 방법으로 복잡한 프로그램을 잘게 나눠 공동 개발이 가능
- 명령어 도구, 시각화 도구 Rviz, GUI 도구 모음 rqt, 3차원 시뮬레이터 Gazebo 지원
- 로보틱스에서 많이 사용되는 모델링, 센싱, 인식, 내비게이션, 매니퓰레이션 기능 지원

# ROS는 새로운 운영 체제(OS)인가?

- 운영 체제 (Operating System)
  - 범용 컴퓨터
    - ◆ Windows(Windows XP, 7, 8 ...)
    - ◆ Linux(Ubuntu, Redhat, Fedora, Mint, Gentoo ...)
    - ◆ MAC(OS X ...) 등
  - 스마트폰
    - ◆ Android, iOS, Windows Phone, Symbian, RiMO, Tizen등
- ROS = Robot Operating System
- **ROS**는 메타운영체제(Meta-Operating System)이다.



# 메타운영체제(Meta-Operating System)

- ▶ 메타운영체제(Meta-Operating System) 딱히 정확히 정의된 용어는 아니지만, 어플리케이션과 분산 컴퓨팅 자원간의 가상화 레이어로 분산 컴퓨팅 자원을 활용하여, 스케쥴링 및 로드, 감시, 에러 처리 등을 실행하는 시스템이라고 볼 수 있다.
- ▶ 즉, 윈도우, 리눅스, 안드로이드와 같은 **전통적인 운영체제는 아니다.** 오히려, ROS 는 기존의 **전통적인 운영체제(리눅스, 윈도우즈, OS-X, 안드로이드)를 이용**하고 있다.
- ▶ 기존 운영체제의 프로세스 관리 시스템, 파일 시스템, 유저 인터페이스, 프로그램 유틸(컴파일러, 스레드 모델 등)등을 사용하고 있다. 이에 추가적으로 다수의 이기종 하드웨어간의 데이터 송수신, 스케쥴링, 에러 처리 등 로봇 응용 소프트웨어 개발을 위한 필수 기능들을 라이브러리 형태로 제공하고 있다.
- ➤ 또한, 이러한 **로봇 소프트웨어 프레임워크**를 기반으로 다양한 목적의 응용 프로그램을 개발, 관리, 제공하고 있으며 유저들이 개발한 패키지 또한 유통하는 생태계 (ecosystem)를 갖추고 있다.

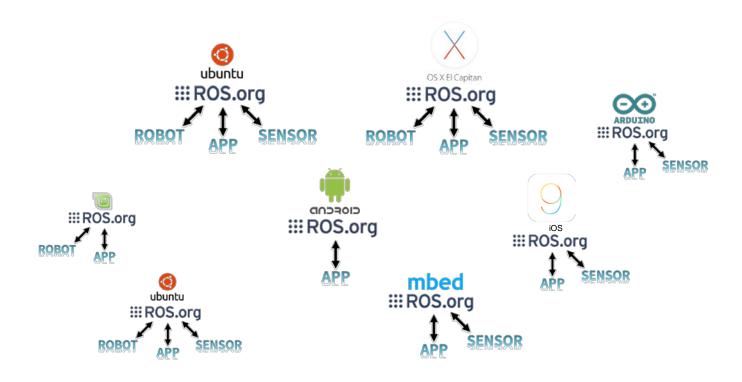
# 메타운영체제(Meta-Operating System)



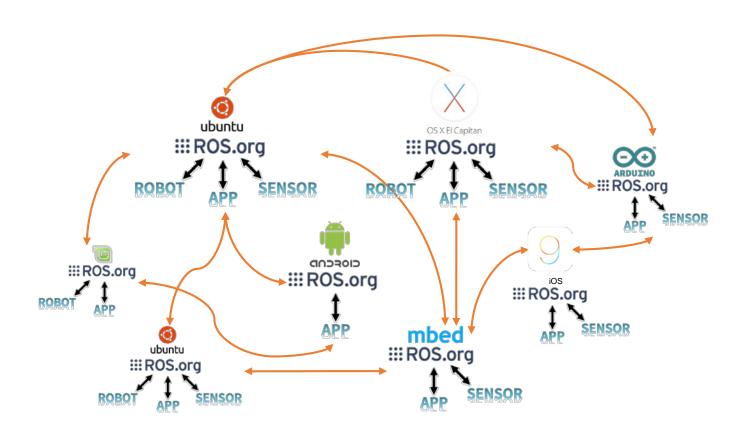
디바이스 드라이버, 라이브러리, 디버그 도구, 메시지 통신구동 도구, 컴파일 도구, 인스톨러, 패키지 생성 및 릴리즈



## 이기종 디바이스 간의 통신 지원



## 이기종 디바이스 간의 통신 지원



# ROS를 사용 가능한 운영체제

### ■ 기존 전통적인 운영체제

- ROS를 사용 가능한 운영체제(OS)로는 Ubuntu, OS X, Windows, Fedora, Gentoo, OpenSUSE, Debian, Raspbian, Arch, QNX Realtime OS 등이 있으나 기능 제한사항이 있을 수 있다.
- 스마트폰 운영체제인 Android, iOS 의 경우, 부분적 사용 가능
- OS를 탑재할 수 없는 마이크로 컨트롤러 유닛(MCU)의 경우, 시리얼 통신, 블루투스, LAN 경유로 통신할 수 있는 라이브러리 제공
- 기본적으로는 **Ubuntu**, OS X 에서 구동하는 것을 추천!

지식-지능형 메카트로닉스 실험실 - 신승태

ROS 2.0 은 3대 OS 모두

지원

# R05의 구성

Client Layer	roscpp	rospy	roslisp	rosjava	roslibjs		
Robotics Application	Movelt!	navigatioin	executive smach	descartes	rospeex		
	teleop pkgs	rocon	mapviz	people	ar track		
Robotics Application Framework	dynamic reconfigure	robot localization	robot pose ekf	Industrial core	robot web tools	ros realtime	mavros
	tf	robot state publisher	robot model	ros control	calibration	octomap mapping	
	vision opencv	image pipeline	laser pipeline	perception pcl	laser filters	ecto	
Communication Layer	common msgs	rosbag	actionlib	pluginlib	rostopic	rosservice	
	rosnode	roslaunch	rosparam	rosmaster	rosout	ros console	
Hardware Interface Layer	camera drivers	GPS/IMU drivers	joystick drivers	range finder drivers	3d sensor drivers	diagnostics	
	audio common	force/torque sensor drivers	power supply drivers	rosserial	ethercat drivers	ros canopen	
Software Development Tools	RViz	rqt	wstool	rospack	catkin	rosdep	
Simulation	gazebo ros pkgs	stage ros					http://wiki.ros.org/API

# रेष्ठ रेल्येसासा ROS इर्र



# 특징 /) 통신 인프라

- 노드 간 데이터 통신을 제공
- 통상적 미들웨어로 지칭되는 메시지 전달 인터페이스 지원
- 메시지 파싱 기능
  - → 로봇 개발 시에 빈번히 사용되는 통신 시스템 제공
  - → 캡슐화 및 코드 재사용을 촉진하는 노드들 간의 메시지 전달 인터페이스
- 메시지의 기록 및 재생
  - → 노드 간 송/수신되는 데이터인 메시지를 저장하고 필요시에 재사용 가능
  - → 저장된 메시지를 기반으로 반복적인 실험 가능, 알고리즘 개발에 용이함
- 메시지 사용으로 인한 다양한 프로그래밍 언어 사용 가능
  - → 노드 간의 데이터 교환이 메시지를 사용하기 때문에 각 노드는 서로 다른 언어로 작성 가능
  - → 클라이언트 라이브러리: roscpp, rospy, roslisp, rosjava, roslua, roscs, roseus, PhaROS, rosR
- 분산 매개 변수 시스템
  - → 시스템에서 사용되는 변수를 글로벌 키값으로 작성하여 공유 및 수정하여 실시간으로 반영



# 특징 2) 로봇 관련 다양한 기능

#### ■ 로봇에 대한 표준 메시지 정의

 카메라, IMU, 레이저 등의 센서 / 오도메트리, 경로 및 지도 등의 내비게이션 데이터 등의 표준 메시지를 정의하여 모듈화, 협업 작업을 유도, 효율성 향상

#### ■ 로봇 기하학 라이브러리

■ 로봇, 센서 등의 상대적 좌표를 트리화 시키는 TF 제공

#### ■ 로봇 기술 언어

■ 로봇의 물리적 특성을 설명하는 XML 문서 기술

#### ■ 진단 시스템

■ 로봇의 상태를 한눈에 파악할 수 있는 진단 시스템 제공

#### ■ 센싱/인식

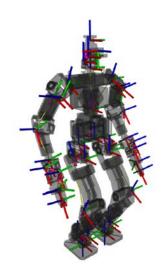
■ 센서 드라이버, 센싱/인식 레벨의 라이브러리 제공

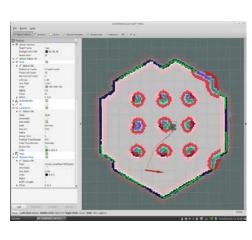
#### ■ 내비게이션

- 로봇에서 많이 사용되는 로봇의 포즈(위치/자세) 추정, 지도내의 자기 위치 추정 제공
- 지도 작성에 필요한 SLAM, 작성된 지도 내에서 목적지를 찾아가는 Navigation 라이브러리를 제공

#### ■ 매니퓰레이션

- 로봇 암에 사용되는 IK, FK 는 물론 응용단의 Pick and Place 를 지원하는 다양한 Manipulation 라이브러리 제공
- GUI 형태의 매니퓰레이션 Tools 제공(Movelt!)







# -특징 3) 다양한 개발 도구

- 로봇 개발에 필요한 다양한 개발 도구를 제공
- 로봇 개발의 효율성 향상
- Command-Line Tools
  - → GUI 없이 ROS에서 제공되는 명령어로만 로봇 억세스 및 거의 모든 ROS 기능 소화

#### RViz

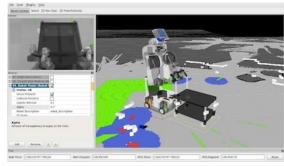
- → 강력한 3D 시각화툴 제공
- → 레이저, 카메라 등의 센서 데이터를 시각화
- → 로봇 외형과 계획된 동작을 표현

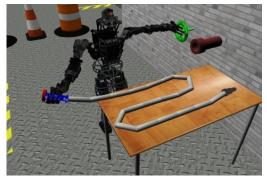
#### RQT

- → 그래픽 인터페이스 개발을 위한 Qt 기반 프레임 워크 제공
- → 노드와 그들 사이의 연결 정보 표시(rqt\_graph)
- → 인코더, 전압, 또는 시간이 지남에 따라 변화하는 숫자를 플로팅(rqt\_plot)
- → 데이터를 메시지 형태로 기록하고 재생(rqt\_bag)

#### Gazebo

- → 물리 엔진을 탑재, 로봇, 센서, 환경 모델 등을 지원, 3차원 시뮬레이터
- → ROS와의 높은 호완성







Virtual Machine + Ubuntu 20.04 + ROS (Noetic)

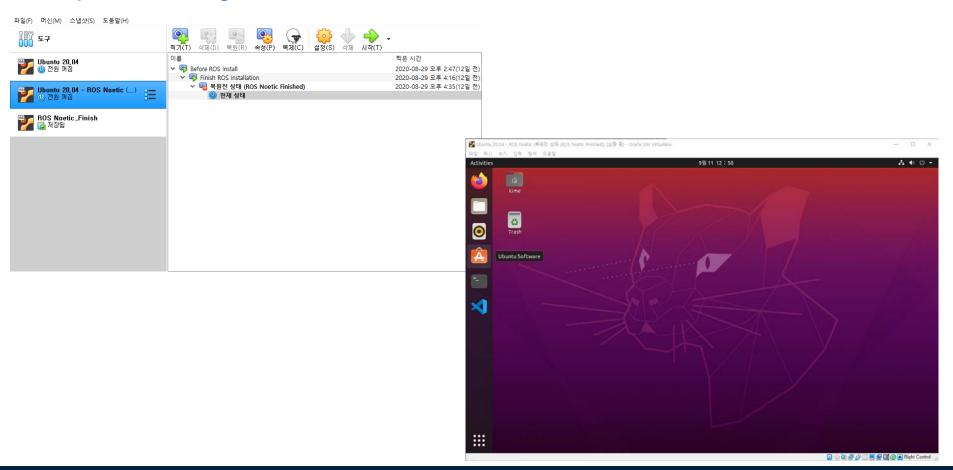
ROS द्वी।





### Virtual Box를 이용한 ROS 및 IDE 설치

- Ubuntu install of ROS Noetic
- http://wiki.ros.org/noetic/Installation/Ubuntu
- Visual Studio Code
- https://code.visualstudio.com/docs/setup/linux#\_installation
- http://wiki.ros.org/IDEs





#### ☞ ROS 설치 스텝

- 1. Sources.list 설정 \$ sudo sh —c 'echo "deb http://packages.ros.org/ros/ubuntu \$(lsb\_release —sc) main" > /etc/apt/sources.list.d/ros latest.list'
- 2. Keys 설정 \$ sudo apt-key adv --keyserver 'hkp://keyserver.ubuntu.com:80' --recv-key C1CF6E31E6BADE8868B172B4F42ED6FBAB17C654
- 3. ROS Noetic 설치 \$sudo apt update \$sudo apt install ros-noetic-desktop-full \$sudo apt install ros-noetic-PACKAGE
- 4. 환경 설정 \$source /opt/ros/noetic/setup.bash \$echo "source /opt/ros/noetic/setup.bash" >> ~/.bashrc \$source ~/.bashrc



# ROS 간면 설치

```
**ROS 간편 설치

$ wget

https://raw.githubusercontent.com/knowledge-

intelligence/ROS_RobotManipulator/master/ros_noe

tic_install.sh && chmod 755

./ros_noetic_install.sh &&

./ros_noetic_install.sh
```



# Code: ros\_noetic\_install.sh

```
#!/bin/bash
# Gen @ 200829
# : Install ROS Noetic in Ubuntu 20.04 with VS Code
name catkinws="catkin ws"
name ros distro="noetic"
version=`lsb release -sc`
echo "[Update & upgrade the package]"
sudo apt-get update
echo "[Installing chrony and setting the ntpdate]"
sudo apt-get install -y chrony ntpdate
echo "[Add the ROS repository]"
if [ ! -e /etc/apt/sources.list.d/ros-latest.list ]; then
c "echo \"deb http://packages.ros.org/ros/ubuntu ${version} m
ain\" > /etc/apt/sources.list.d/ros-latest.list"
fi
echo "[Download the ROS keys]"
roskey=`apt-key list | grep "ROS builder"`
if [ -z "$roskey" ]; then
fi
```

```
echo "[Update & upgrade the package]"
echo "[Installing ROS]"
sudo apt-get install -y ros-$name_ros_distro-desktop-
full ros-$name ros distro-rqt-*
echo "[rosdep init and python-rosinstall]"
sudo sh -c "rosdep init"
. /opt/ros/$name_ros_distro/setup.sh
sudo apt-get install -y python-rosinstall
echo "[Making the catkin workspace and testing the catkin mak
e]"
mkdir -p ~/$name catkinws/src
cd ~/$name catkinws/src
catkin init workspace
cd ~/$name catkinws/
echo "[Setting the ROS evironment]"
c "echo \"source /opt/ros/$name ros distro/setup.bash\" >> ~/
.bashrc"
c "echo \"source ~/$name catkinws/devel/setup.bash\" >> ~/.ba
shrc"
c "echo \"export ROS MASTER URI=http://localhost:11311\" >> ~
/.bashrc"
sh -c "echo \"export ROS HOSTNAME=localhost\" >> ~/.bashrc"
echo "[Complete!!!]"
```

# Visual Studio Code 설치

#### ☞IDE: Visual Studio Code 설치

\$sudo snap install --classic code # or code-insiders

```
Setting up ros-noetic-rqt-reconfigure (0.5.3-lfocal.20200724.184244) ...
Setting up ros-noetic-rqt-robot-dashboard (0.5.8-1focal.20200724.185627) ...
Setting up ros-noetic-visualization-tutorials (0.11.0-1focal.20200724.195149) ...
Setting up ros-noetic-ros-core (1.5.0-1focal.20200724.183234) ...
Setting up ros-noetic-geometry (1.13.2-1focal.20200724.190245) ...
Setting up ros-noetic-rqt-srv (0.4.8-1focal.20200724.185044) ...
Setting up ros-noetic-rgt-action (0.4.9-1focal.20200724.185037) ...
Setting up ros-noetic-urdf-sim-tutorial (0.5.1-1focal.20200724.194954) ...
Setting up ros-noetic-gazebo-plugins (2.9.1-1focal.20200724.190229) ...
Setting up ros-noetic-rqt-launch (0.4.8-1focal.20200724.184224) ...
Setting up ros-noetic-gazebo-ros-pkgs (2.9.1-lfocal.20200724.192612) ...
Setting up ros-noetic-laser-pipeline (1.6.4-1focal.20200724.191250) ...
Setting up ros-noetic-ros-base (1.5.0-1focal.20200724.184136) ..
Setting up ros-noetic-rqt-robot-plugins (0.5.8-1focal.20200724.195009) ...
Setting up ros-noetic-robot (1.5.0-1focal.20200724.190846)
Setting up ros-noetic-rqt-common-plugins (0.4.9-1focal.20200724.185831) ...
Setting up ros-noetic-perception (1.5.0-1focal.20200724.193330) ...
Setting up ros-noetic-viz (1.5.0-1focal.20200724.195833) ..
Setting up ros-noetic-desktop (1.5.0-1focal.20200724.200614) ...
Setting up ros-noetic-simulators (1.5.0-lfocal.20200724.195807) ...
Setting up ros-noetic-desktop-full (1.5.0-lfocal.20200724.201236) ...
Processing triggers for libc-bin (2.31-Oubuntu9) ...
kime@kime-VirtualBox: $
                                                                 🚇 🌃 🔊 🔗 👅 🗐 🚳 🕒 Right Con
```

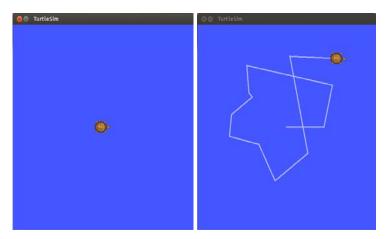
# ROS ENCE

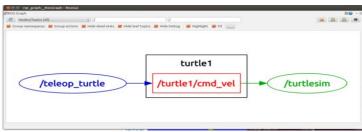


# ROS 동작 테스트

### ➤ turtlesim 패키지

- \$ roscore
- \$ rosrun turtlesim turtlesim\_node
- \$ rosrun turtlesim turtle\_teleop\_key
- \$ rosrun rqt\_graph rqt\_graph



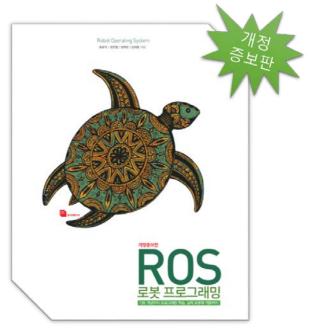




# ROS 동작 테스트 데모



### 참고문헌



표윤석 외 3명, 『ROS 로봇 프로그래밍 기초 개념부터 프로그래밍 학습, 실제 로봇에 적용까지』, 루비페이퍼(2017)



오로카 (www.oroca.org)