



ROS 1. Introduction



Outline



- ▶ ROS 개요
- ▶ ROS 프로그램의 구성 및 특징
- ▶ ROS 설치



What is ROS?



- ▶ ROS – Robot Operating System
 - ▶ 일반적인 OS(운영체제) – Windows, Linux, macOS, ...
 - ▶ 컴퓨터 하드웨어와 응용소프트웨어 사이를 중재하여, 하드웨어를 제어하고 응용소프트웨어를 위한 기반 환경을 제공
 - ▶ ROS is not an OS.






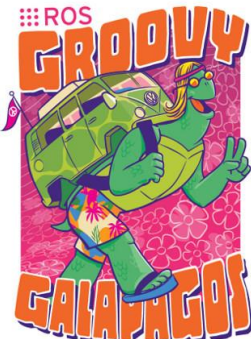

The Robot Operating System (ROS) is a flexible framework for writing robot software. It is a collection of tools, libraries, and conventions that aim to simplify the task of creating complex and robust robot behavior across a wide variety of robotic platforms.







(<http://www.ros.org>)



History – ROS₁



 ROS Box Turtle	 ROS C Turtle	 ROS DIAMONDBACK LAUNCH!	 ELECTRIC EMTS ROS	 ROS FUERTE	 ROS GROOVY GALAPAGOS	 ROS HYDRO MEDUSA
Box Turtle	C Turtle	Diamondback	Electric Emys	Fuerte Turtle	Gloovy Galapagos	Hydro Medusa
Mar. 2, 2010	Aug. 2, 2010	Mar. 2, 2011	Aug. 30, 2011	Apr. 23, 2012	Dec. 31, 2012	Sep. 4, 2013

 INDIGO IGLOO ROS	 JADE TURTLE ROS	 KINETIC KAME ROS	 ROS LUNAR+LOGGERHEAD	 Melodic Morenia ROS	 ROS NOETIC NINJEMYS
Indigo Igloo	Jade Turtle	Kinetic Kame	Lunar Loggerhead	Melodic Morenia	Noetic Ninjemys
Jul. 22, 2014	May 23, 2015	May 23, 2016	May 23, 2017	May 23, 2018	May 23, 2020



History – ROS₂



			
Ardent Apalone	Bouncy Bolson	Crystal Clemmys	Dashing Diademata
Dec. 8, 2017	Jul. 2, 2018	Dec. 14, 2018	May 31, 2019
			
Eloquent Elusor	Foxy Fitzroy	Galactic Geochelone	Humble Hawksbill
Nov. 22, 2019	Jun. 5, 2020	May 23, 2021	May 23, 2022

Supported Platforms

	Distribution	End of Life	Supported Platforms
ROS1	Noetic Ninjemys	May 2025	Ubuntu 20.04, Debian 10
	Melodic Morenia	May 2023	Ubuntu 18.04, Ubuntu 17.10 Debian 9, Windows 10
	Lunar Loggerhead	May 2019	Ubuntu 17.04, Ubuntu 16.10, Ubuntu 16.04 Debian 9, Fedora 26
ROS2	Humble Hawksbill	May 2027	Tier1: Ubuntu 22.04, Windows 10 Tier2: RHEL 8 Tier3: Ubuntu 20.04, macOS, Debian Bullseye
	Galactic Geochelone	Dec. 2022	Tier1: Ubuntu 20.04, Windows 10 Tier2: RHEL 8 Tier3: Debian 11, OpenEmbedded 2.6, macOS 10.14
	Foxy Fitzroy	May 2023	Tier1: Ubuntu 20.04, macOS 10.14, Windows 10 Tier3: Debian 10, OpenEmbedded 2.6



이 강의에서는



- ▶ ROS1-Noetic Ninjemys을 사용
 - ▶ ROS2에 비해 훨씬 더 많은 문서, 참고 자료, 예제, 튜토리얼이 인터넷에 존재
 - ▶ ROS2가 아직 널리 사용되고 있다고 보기 어려움
 - ▶ ROS1을 배운 후 ROS2가 많이 사용되면 그 때 ROS2로 바꾸는 것을 추천



참고 교재



- ▶ ROS 로봇 프로그래밍, 표윤석 등 지음, 루비페이퍼, 2017년 8월
- ▶ **ROS로 배우는 로봇 프로그래밍**, 모건 퀴글리 등 지음, 제이펍, 2017년 3월
- ▶ ROS 로보틱스 프로그래밍, 렌틴 조셉 등 지음, 에이콘출판, 2017년 2월
- ▶ ROS로 효과적인 로봇 프로그래밍하기, Anil mahtani 등 지음, 홍릉과학출판사, 2018년 1월



Outline



- ▶ ROS 개요
- ▶ ROS 프로그램의 구성 및 특징
- ▶ ROS 설치

○○○ 로봇 소프트웨어 작성의 어려움 ○○○

▶ 기존 구성

```
while(1) {  
    라이다 신호 입력  
    카메라 신호 입력  
    판단  
    로봇 제어  
}
```

○○○ 로봇 소프트웨어 작성의 어려움 ○○○

▶ 기존 구성

```
while(1) {  
    라이다 신호 입력  
    카메라 신호 입력  
    판단  
    로봇 제어  
}
```

▶ 문제점

- ▶ 각 부분의 실행 순서?
 - ▶ 어느 부분을 먼저 수행할 것인가?
 - ▶ 데이터의 의존성을 고려해야 함
- ▶ 반복 주기?
- ▶ 각 부분 사이의 데이터 전달을 위한 데이터 형식

○○○ 로봇 소프트웨어 작성의 어려움 ○○○

▶ 기존 구성

```
while(1) {  
    라이다 신호 입력  
    카메라 신호 입력  
    판단  
    로봇 제어  
}
```

▶ 문제점

- ▶ 각 부분의 실행 순서?
- ▶ 반복 주기?
 - ▶ 각 부분마다 실행되는 주기가 다름
- ▶ 각 부분 사이의 데이터 전달을 위한 데이터 형식

○○○ 로봇 소프트웨어 작성의 어려움 ○○○

▶ 기존 구성

```
while(1) {  
    라이다 신호 입력  
    카메라 신호 입력  
    판단  
    로봇 제어  
}
```

▶ 문제점

- ▶ 각 부분의 실행 순서?
- ▶ 반복 주기?
- ▶ 각 부분 사이의 데이터 전달을 위한 데이터 형식

○○○ 로봇 소프트웨어 작성의 어려움 ○○○

▶ 기존 구성

```
while(1) {  
    라이다 신호 입력  
    카메라 신호 입력  
    판단  
    로봇 제어  
}
```

▶ 문제점

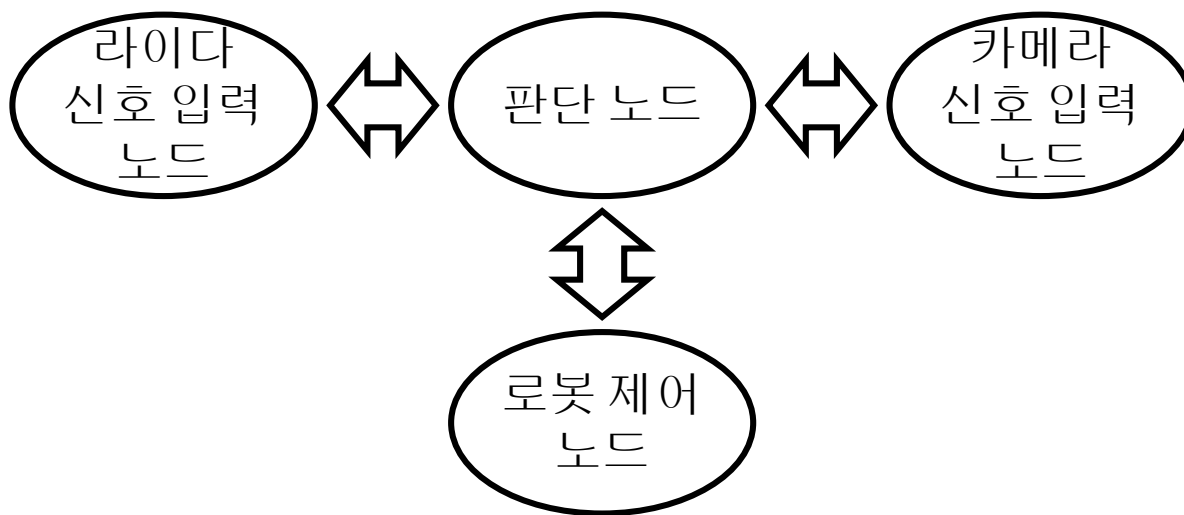
- ▶ 각 부분의 실행 순서?
 - ▶ 반복 주기?
 - ▶ 각 부분 사이의 데이터 전달을 위한 데이터 형식
- ▶ 외부 라이브러리를 이용하면 더 심각해짐
- ▶ 체계적인 프레임워크가 필요함 → ROS



ROS의 구성



- ▶ 노드(node): 기본적인 프로그램 단위
 - ▶ 각 프로그램 단위(Node, 노드)의 병렬적, 독립적 수행
- ▶ 통일적인 노드 인터페이스





ROS를 통한 해결



노드의 독립적 실행

부분적인 실행/중단/교체가 용이함

노드의 병렬적 실행

전체 실행 주기에 대해 고려할 필요 없음

독립적 실행 및 통일적인 인터페이스

프로그램 통합이 용이함

독립적인 개발과 라이브러리화
→ ROS 생태계 확장
→ 로봇 소프트웨어의 빠른 개발



ROS의 특징



- ▶ 각 노드를 서로 다른 언어로 개발 가능
 - ▶ 데이터 인터페이스가 맞으면 통합시 문제 없음
- ▶ 각 노드가 서로 다른 기기에서 동작 가능
- ▶ 로봇 개발을 위한 기능들의 제공
 - ▶ 로봇 기하에 대한 라이브러리
 - ▶ 로봇 기술 언어
 - ▶ rviz와 rqt 같은 다양한 툴



노드(Node)



- ▶ ROS 프로그램은 여러 개의 독립적인 노드로 구성됨



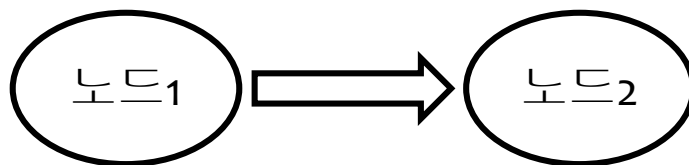
- ▶ 노드들은 서로 정보와 데이터를 교환함
 - ▶ 토픽(Topic)
 - ▶ 서비스(Service)
 - ▶ 액션(Action)



토픽 (Topic)



- ▶ 토픽: 정의된 자료형을 가지는 메시지 스트림
 - ▶ 단방향 전송



- ▶ 토픽 발행(publish): 송신
- ▶ 토픽 구독(subscribe): 수신

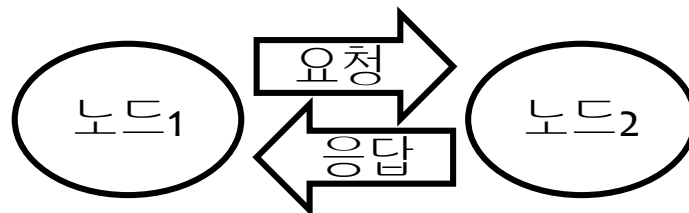


서비스 (Service)



▶ 서비스: 동기식 양방향 메시지 통신

▶ 양방향



- ▶ 서비스를 요청(request)하면 그에 대해 응답(response)
- ▶ 서비스 요청 노드: 서비스 클라이언트(client)
- ▶ 서비스 응답 노드: 서비스 서버(server)
- ▶ 상태 질의, 환경 설정 관리와 같은 간단한 읽기/쓰기 상호작용에 적합

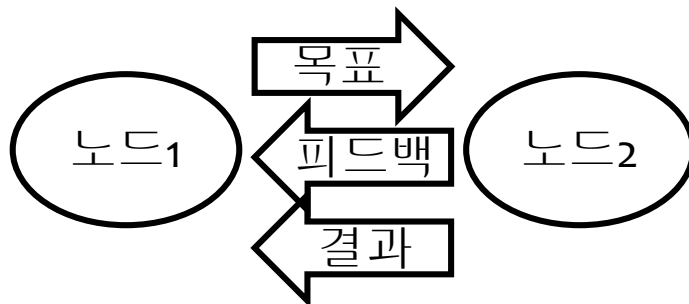


액션 (Action)



▶ 액션: 비동기식 양방향 메시지 통신

▶ 양방향



- ▶ 목표를 보내면 그에 따라 행동한 뒤 결과를 회신함
- ▶ 행동의 중간에 피드백을 제공할 수 있음
- ▶ 목표 요청 노드: 액션 클라이언트(client)
- ▶ 목표 행동 노드: 액션 서버(server)
- ▶ 시간이 오래 걸리는 작업에 적합



Outline



- ▶ ROS 개요
- ▶ ROS 프로그램의 구성
- ▶ ROS 설치



설치 과정



- ▶ VMware or Boot Camp
- ▶ Ubuntu Linux 20.04
- ▶ ROS Noetic Ninjemys



VMware 다운로드 및 설치



▶ 다운로드

- ▶ www.vmware.com
- ▶ Resources → Product Trials
- ▶ 가장 아래에 Download Free Products
- ▶ VMware Workstation Player (Free) 클릭
- ▶ VMware Workstation Player 16.1.2 Go to Download 클릭
- ▶ Windows 64-bit 용으로 다운로드

▶ 다운로드가 끝나면 설치

○○○ Ubuntu Linux 설치 파일 다운로드 ○○○

▶ 다운로드

- ▶ www.ubuntu.com
- ▶ Download → Other ways to download → older releases
- ▶ Ubuntu 20.04.5 LTS 클릭
- ▶ 64-bit PC (AMD64) desktop image 다운로드

VMware 실행 및 Ubuntu Linux 설치

▶ VMware 실행

- ▶ Create a New Virtual Machine 클릭
- ▶ Installer disc image file에서 Ubuntu Linux 설치 파일 선택
- ▶ 계정 정보 입력
 - ▶ User name과 password는 꼭 기억해야 함
- ▶ 가상 머신 이름은 적당히
- ▶ 가상 머신의 디스크 용량은 40GB 정도
- ▶ 설정을 끝내고 Finish

▶ Ubuntu Linux 설치

- ▶ Play virtual machine 버튼 클릭
- ▶ 설치가 진행됨
- ▶ 설치가 끝나면 위의 username과 password로 로그인



▶ Terminal 실행

```
user@hostname$ sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release
-sc) main" > /etc/apt/sources.list.d/ros-latest.list'
user@hostname$ sudo apt install curl
user@hostname$ curl -s https://raw.githubusercontent.com/ros/rosdistro/master/ros.asc
| sudo apt-key add -
user@hostname$ sudo apt update
user@hostname$ sudo apt install ros-noetic-desktop-full
user@hostname$ echo "source /opt/ros/noetic/setup.bash" >> ~/.bashrc
user@hostname$ source ~/.bashrc
user@hostname$ sudo apt install python3-rosdep python3-rosinstall python3-rosinstall-
generator python3-wstool build-essential
user@hostname$ sudo rosdep init
user@hostname$ rosdep update
```