

# 자율주행 데브코스

## ROS 기초

---

(주)자이트론

---

허성민

---

smher@xytron.co.kr

---



# Contents

ROS의 구조와 기능 소개  
ROS 명령어





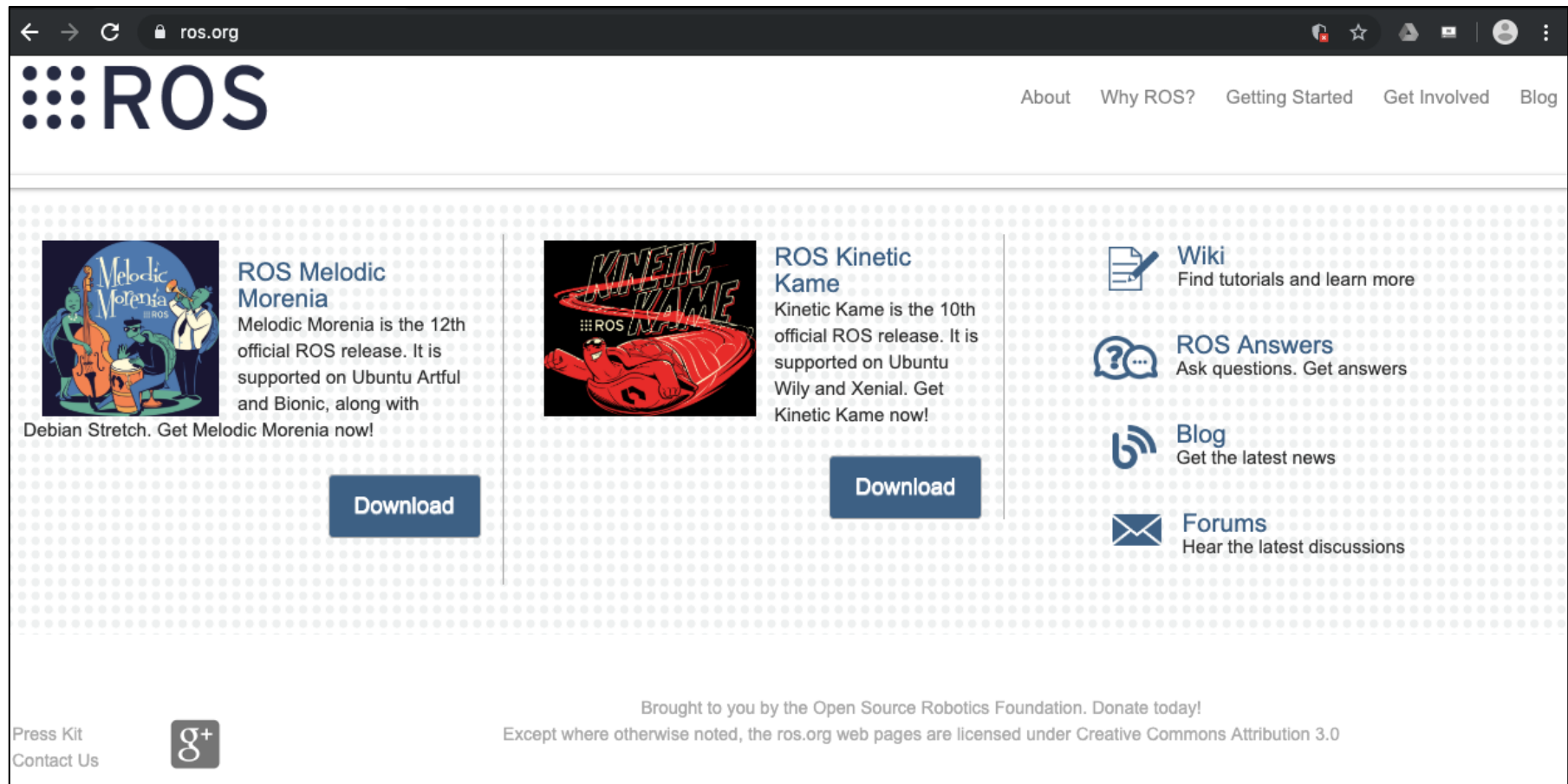
# ROS의 기초

ROS의 구조와 기능 소개



# ROS (Robot Operating System)

- <https://www.ros.org>



The screenshot shows the ROS website homepage. At the top, there's a navigation bar with links: About, Why ROS?, Getting Started, Get Involved, and Blog. The main content area features two large promotional banners. The left banner is for ROS Melodic Morenia, the 12th official ROS release, supported on Ubuntu Artful and Bionic, along with Debian Stretch. It includes a 'Download' button. The right banner is for ROS Kinetic Kame, the 10th official ROS release, supported on Ubuntu Wily and Xenial. It also includes a 'Download' button. To the right of these banners, there are four links with icons: Wiki (Find tutorials and learn more), ROS Answers (Ask questions. Get answers), Blog (Get the latest news), and Forums (Hear the latest discussions). At the bottom, there's a footer with a Google+ logo, a 'Press Kit' and 'Contact Us' link, and a statement: 'Brought to you by the Open Source Robotics Foundation. Donate today! Except where otherwise noted, the ros.org web pages are licensed under Creative Commons Attribution 3.0'.

ros.org

# ROS

About Why ROS? Getting Started Get Involved Blog

**ROS Melodic Morenia**  
Melodic Morenia is the 12th official ROS release. It is supported on Ubuntu Artful and Bionic, along with Debian Stretch. Get Melodic Morenia now!

**Download**

**ROS Kinetic Kame**  
Kinetic Kame is the 10th official ROS release. It is supported on Ubuntu Wily and Xenial. Get Kinetic Kame now!

**Download**

**Wiki**  
Find tutorials and learn more

**ROS Answers**  
Ask questions. Get answers

**Blog**  
Get the latest news

**Forums**  
Hear the latest discussions

Press Kit  
Contact Us

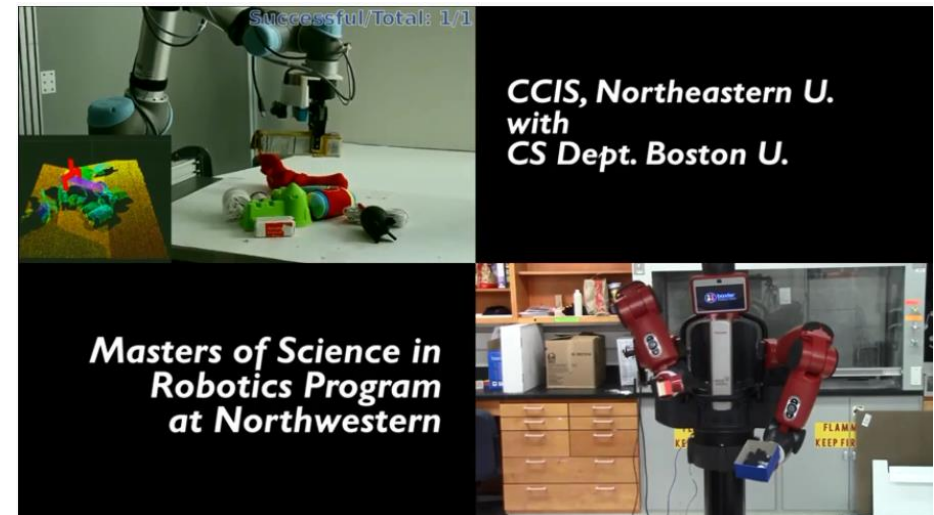
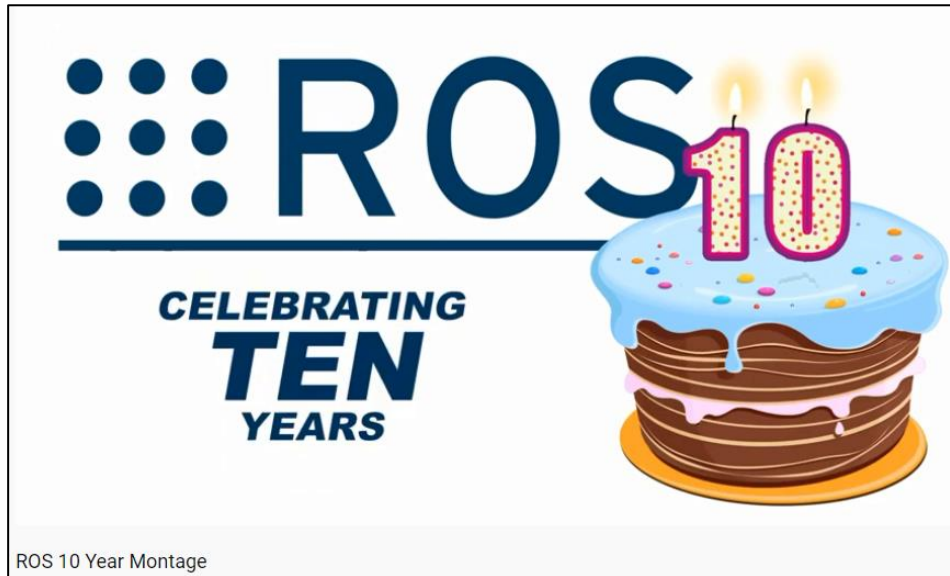
g+

Brought to you by the Open Source Robotics Foundation. Donate today!  
Except where otherwise noted, the ros.org web pages are licensed under Creative Commons Attribution 3.0



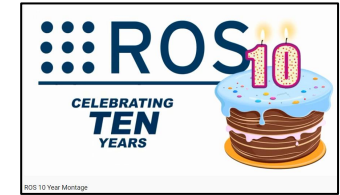
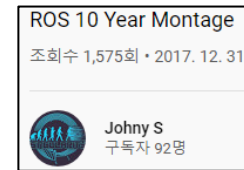
# ROS 소개

- ROS가 적용된 제품들
  - <https://youtu.be/mDwZ21Zia8s>
  - <https://vimeo.com/256842087>



- ROS가 적용된 제품들

- <https://youtu.be/mDwZ21Zia8s>



# ROS (Robot Operating System)

- 오픈소스 로봇 운영체제

- 소스 무료 공개
- 개방형 구조
- 활발한 커뮤니티 활동



- 로봇 소프트웨어를 개발하는 데 필요한 소프트웨어의 집합체

- 소프트웨어 프레임워크 (Software Framework)

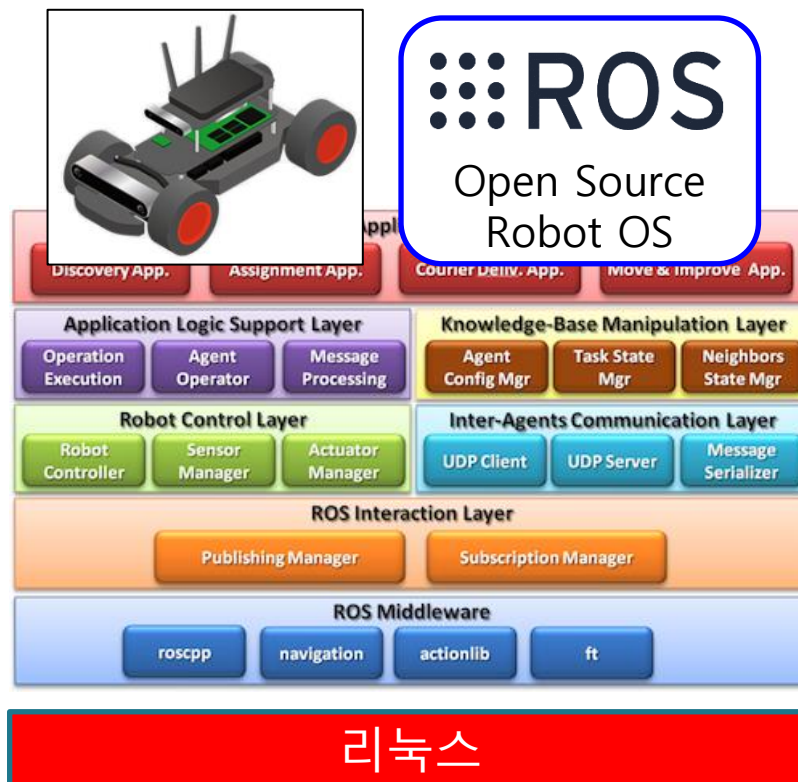
- 메타 운영체제 (Meta OS), 미들웨어 (Middleware)

- 소프트웨어 모듈 + 라이브러리 집합 + 도구 집합



# ROS는 자동차(로봇의 일종) 제어를 위한 미들웨어

- 각종 센서와 모터를 프로그래머가 편하게 사용할 수 있도록 지원
  - 스마트폰에는 Android가... 자율주행차에는 ROS가...







리눅스 / 윈도우 / 맥OS



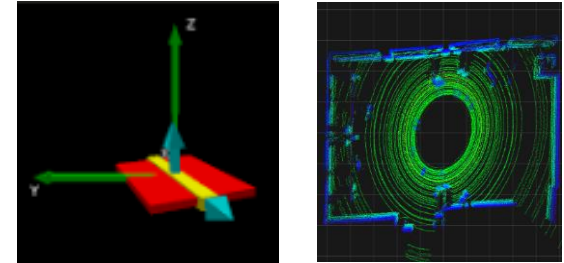
# ROS의 특징

- 로봇 SW를 만들기 위한 코드의 재사용이 용이한 환경제공이 목표
  - 다양한 프로그래밍 언어를 지원: (대표적으로) C++, Python
  - 표준화된 ROS 인터페이스를 따르는 HW와 SW를 편하게 엮을 수 있음
  - 하드웨어 부품과 소프트웨어 부품을 조립하여 여러 응용 구성 가능
  - 대규모 실행 시스템 및 프로세스에도 적용 가능
- 다양한 도구들을 함께 제공
  - RVIZ, RQT, Gazebo, ...
- 다양한 OS 환경에서 통일된 방법으로 상호작용을 구현하는 것이 가능
  - Linux, OS X, Windows, Raspbian, QNX, RTOS, Android, iOS 등
  - 표준화된 통신 프로토콜을 따르는 기기종간의 메시지 교환이 가능

# ROS가 제공하는 도구들

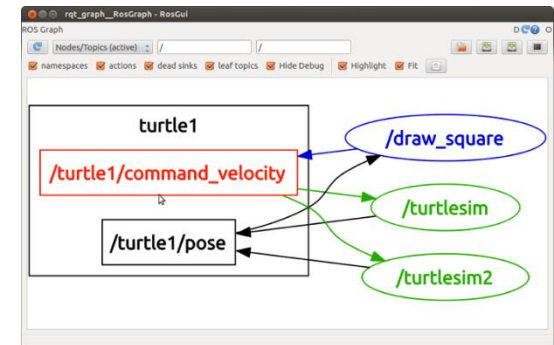
- RVIZ

- 시각화 도구
- 센서데이터를 비롯한 주변환경 변화를 시각화



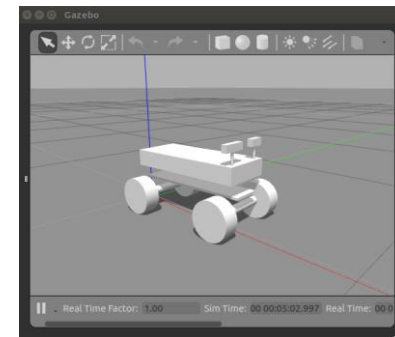
- RQT

- Qt 기반의 GUI 응용 개발 도구
- 노드연결 정보를 그래프로 표현
- 사용자 상호작용을 UI를 갖춘 응용 개발에 이용



- GAZEBO

- 물리 엔진 기반의 3차원 시뮬레이터
- 시뮬레이터 제작 및 모델링에 이용



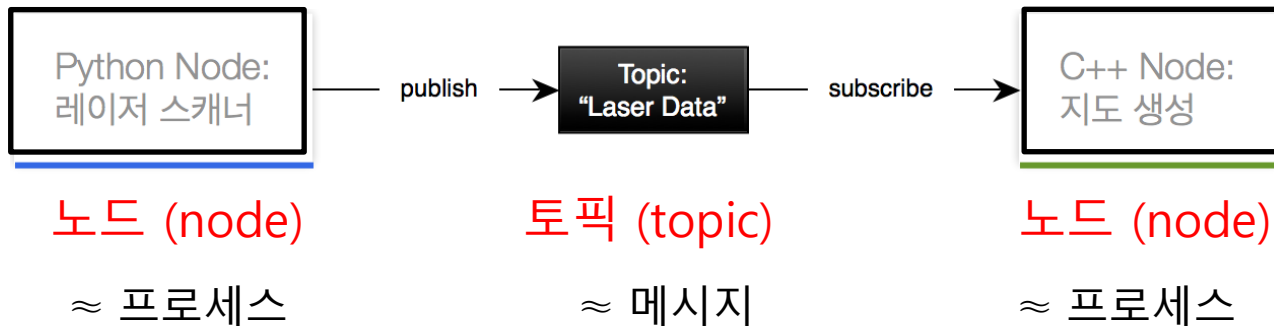
## ROS 버전

- ROS Melodic Morenia
  - May 23rd, 2018 – May 2023
- ROS Lunar Loggerhead
  - May 23rd, 2017 – May 2019
- ROS Kinetic Kame
  - May 23rd, 2016 – April 2021
- ROS Indigo Igloo
  - July 22nd, 2014 – April 2019

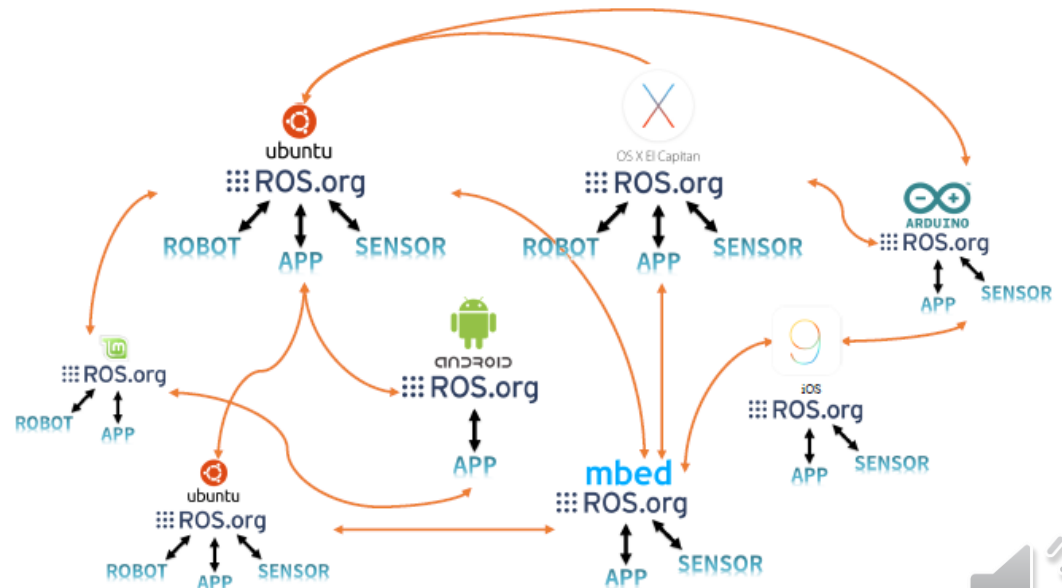


# ROS에서의 통신

- 토픽의 발행&구독 (publish & subscribe) 기반의 메시지 교환

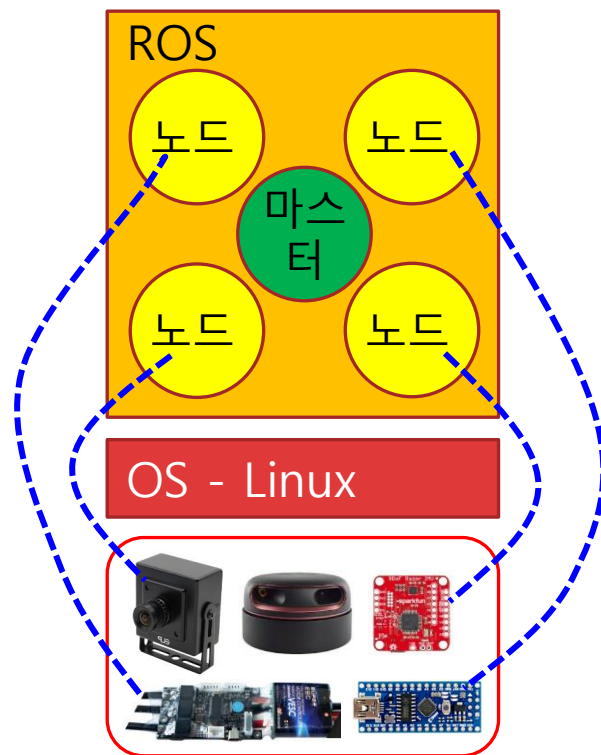


- 이기종간의 메시지 교환의 예

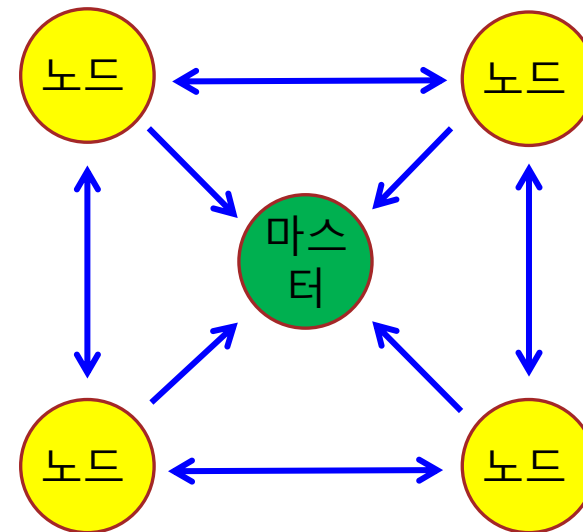


# ROS 핵심 기능

- 노드간 통신을 기반으로 전체 시스템을 구동시킴
  - 하드웨어 부품 또는 소프트웨어 모듈에 노드가 하나씩 할당됨



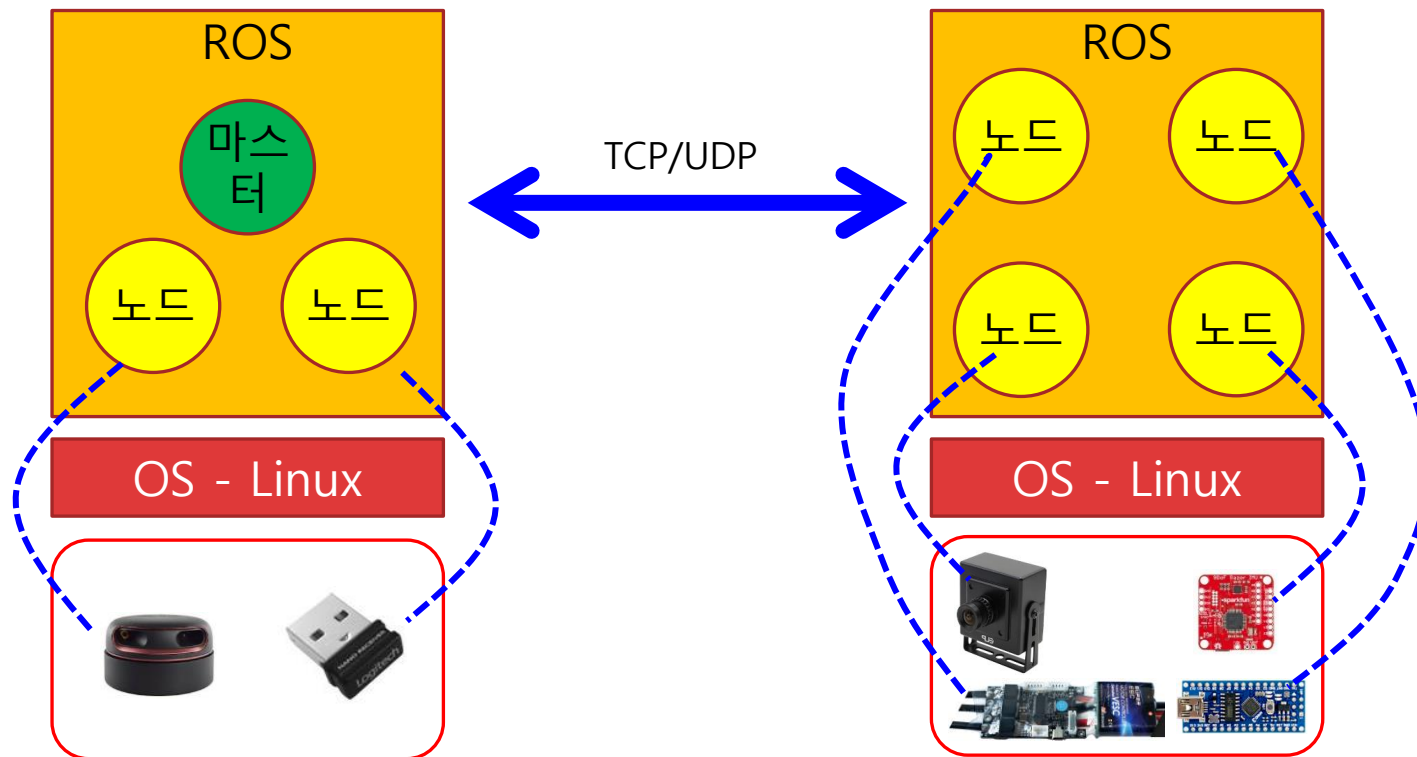
노드는 OS의 도움을 받아 하드웨어 장치들을 제어한다.



노드들은 마스터의 도움을 받아서 서로 메시지를 주고 받는다.

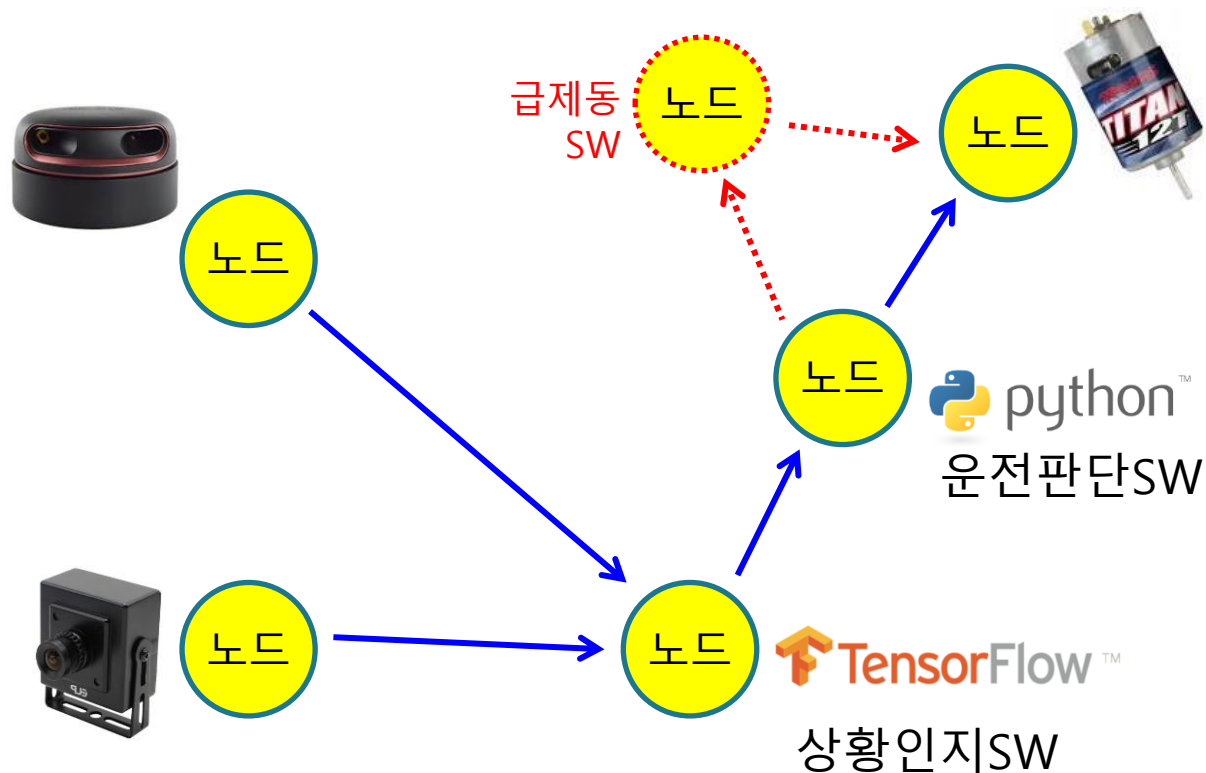
# 네트워크를 통한 원격 연동도 가능

서로 분리된 하드웨어 장치 안에 있는 노드들이  
네트워크 연결을 통해 서로 통신하면서  
하나의 단일 시스템으로서 동작하는 것이 가능



# ROS 기반의 구현사례

- 라이다 + 카메라 + 모터 + SW모듈1 + SW모듈2
  - 하드웨어 장치 3개와 소프트웨어 모듈 2개를 함께 엮어서 원하는 기능을 구현
  - 아래 설계된 시스템의 기능은,
    - ▶ 라이다와 카메라 정보를 상황인지SW가 분석한 후 결과를 운전판단SW로 보내고
    - ▶ 운전판단SW가 제어명령을 생성해서 모터로 보내 차량을 움직이게 함





# ROS에서 사용하는 기본 용어들

- 마스터 (Master)
  - 서로 다른 노드들 사이의 통신을 총괄 관리
  - 통상 'ROS Core' 라고 부름
- 노드 (Nodes)
  - 실행가능한 최소의 단위, 프로세스로 이해할 수 있음
  - ROS에서 발생하는 통신(메시지 송/수신)의 주체
  - HW장치에 대해 하나씩의 노드, SW모듈에 대해 하나씩의 노드 할당
- 토픽 (Topics)
  - ROS 노드들이 관심을 가지고 있는 이야기거리
  - 그 안에 들어 있는 실제 데이터를 메시지라고 부름
  - 예: 센서데이터, 카메라 이미지, 액츄에이터 제어명령, ...



# ROS에서 사용하는 기본 용어들

- 발행자 (Publishers)

- 특정 토픽에 메시지를 담아 외부로 송신하는 노드
- 예: 센서, 카메라, 모터제어 알고리즘, ...



Publisher



Subscriber

- 구독자 (Subscribers)

- 특정 토픽에 담겨진 메시지를 수신하는 노드
- 예: 액추에이터 제어기, 데이터 시각화 도구, ...

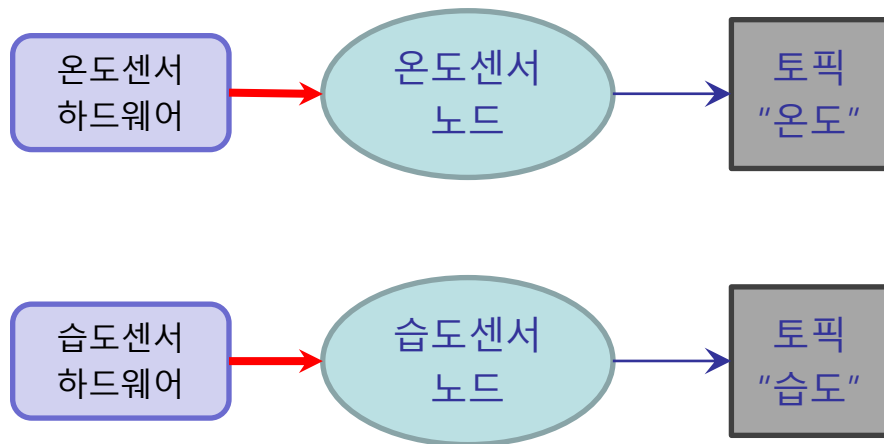
- 패키지 (Packages)

- 하나 이상의 노드와 노드의 실행을 위한 정보 등을 묶어 놓은 단위
- 노드, 라이브러리, 데이터, 파라미터 등을 포함

# 간단한 ROS 응용 예

- 하드웨어 제조사가 제공하는 ROS 노드들

## 센서



각각 어느 토픽에  
어떤 형태의 메시지를 발행하는지가  
정해져 있음

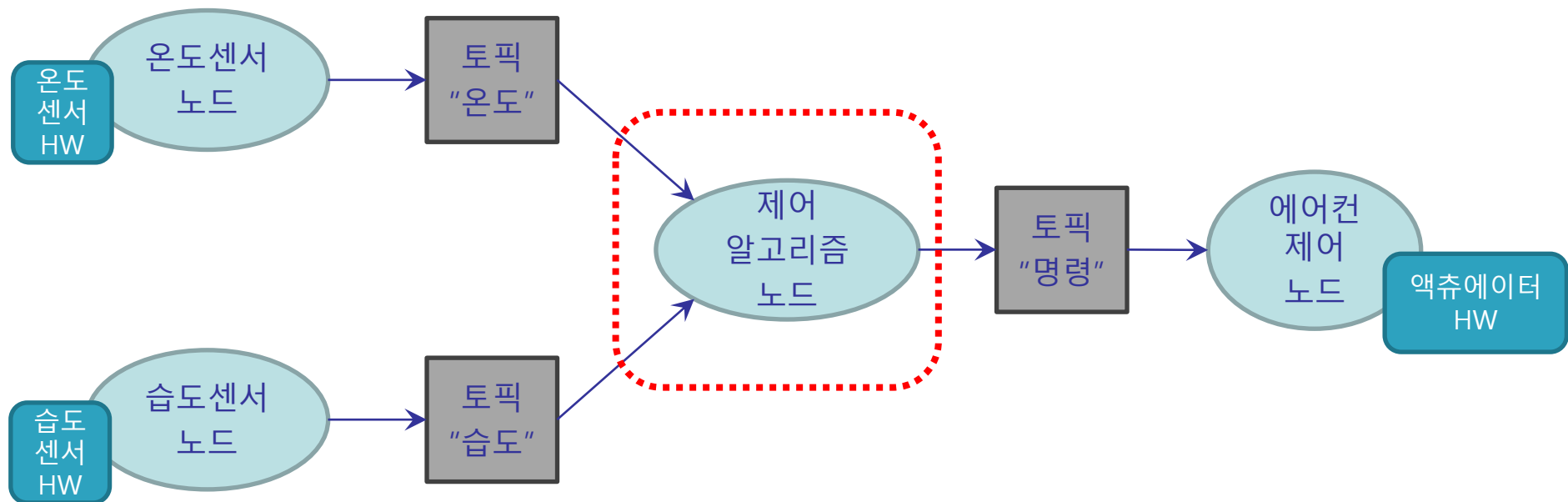
## 액추에이터



어느 토픽에  
어떤 메시지를 발행하면  
어떻게 동작하는지가  
정해져 있음

# 간단한 ROS 응용 예

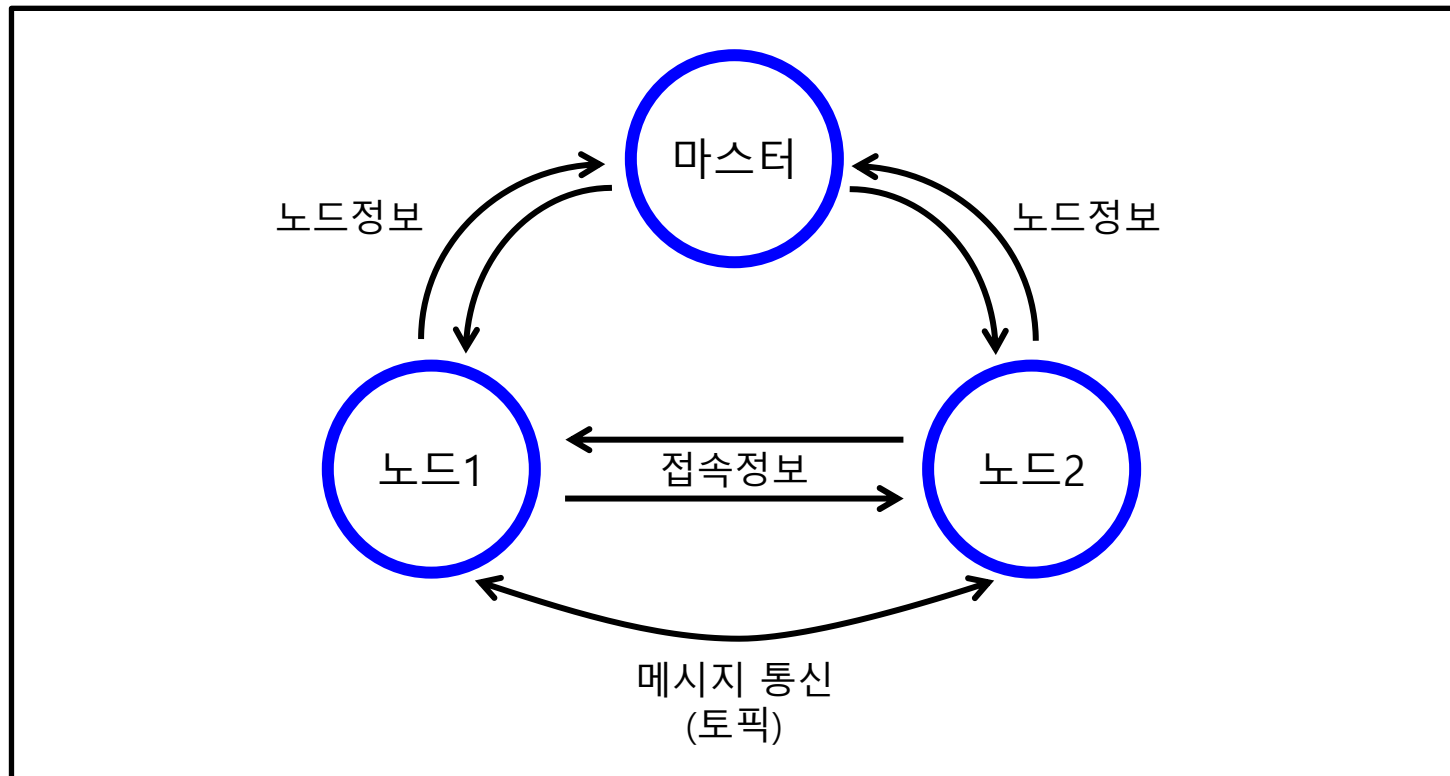
- ROS Core(마스터)가 이 노드들 사이의 메시지 통신을 관장



- 온도센서 노드와 습도센서 노드는 주기적으로 각 토픽에 센싱데이터를 넣어서 발행
- 제어알고리즘 노드는 수집된 데이터에 따른 제어결정을 내리고 명령을 발행
- 에어컨제어 노드는 명령 토픽을 구독하고 하드웨어(에어컨) 제어신호를 생성

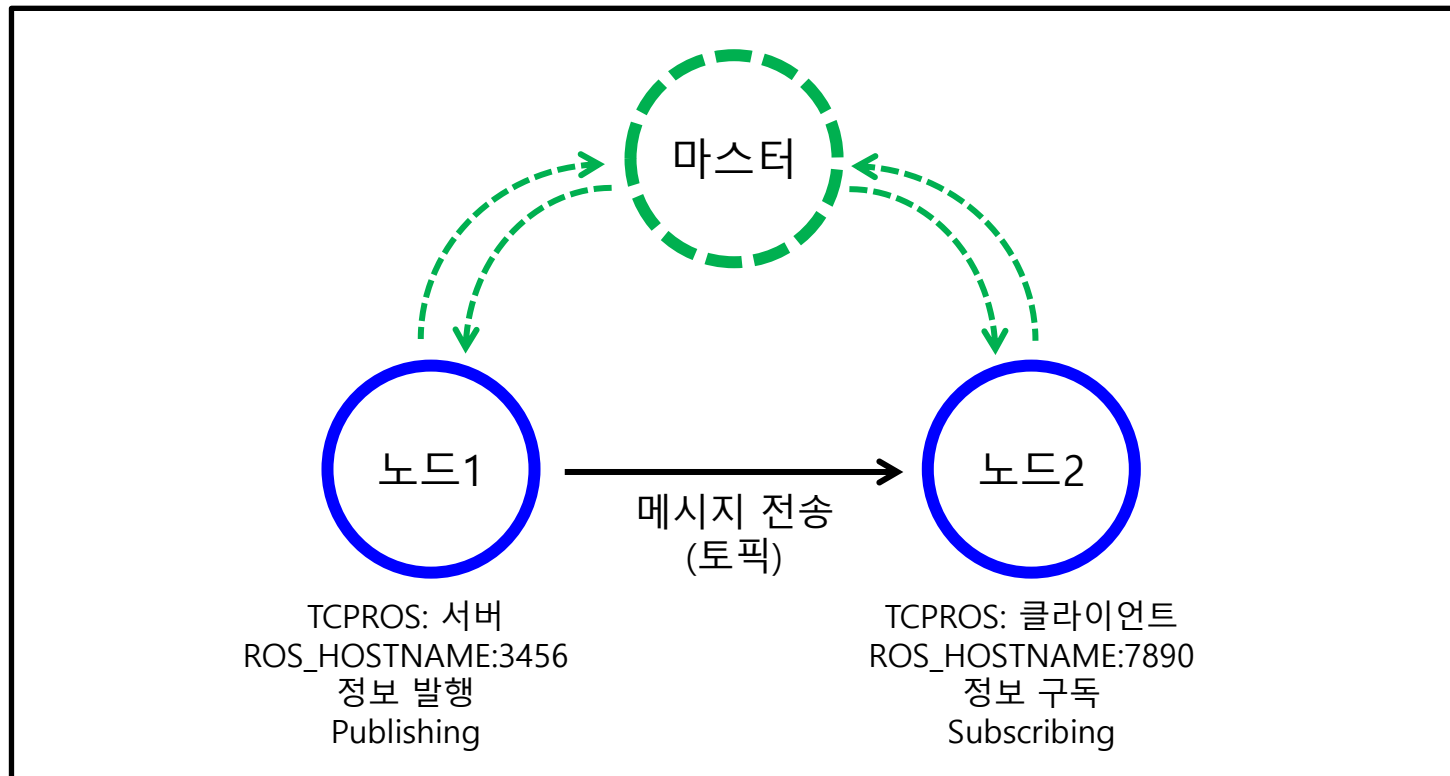
# ROS 노드간 통신 기본 과정 (1)

- 통신이 이루어지기 이전에 통신을 원하는 노드는 마스터에 의뢰하여
- 연결해야 하는 노드의 정보(주소)를 얻어오고, 서로 접속정보를 교환



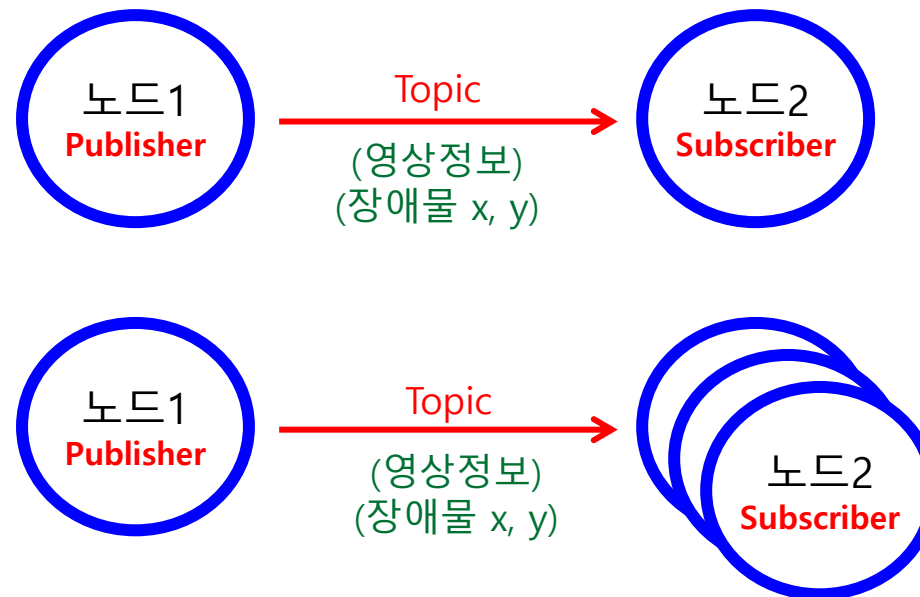
## ROS 노드간 통신 기본 과정 (2)

- 통신환경 구축이 완료되고 나면
- 노드간 통신은 마스터를 거치지 않고 직접 이루어짐



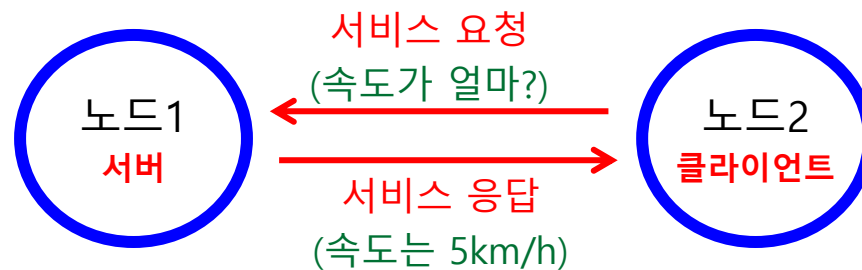
# ROS 노드간 통신의 두 가지 방식 (1)

- 토픽 (topic) 방식의 통신
  - 일방적이고 지속적인 메시지 전송
  - 1:1 뿐만 아니라 1:N 및 N:N 통신도 가능



## ROS 노드간 통신의 두 가지 방식 (2)

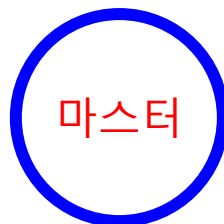
- 서비스 (service) 방식의 통신
  - 서버가 제공하는 서비스에 클라이언트가 요청을 보내고 응답을 받는 방식
  - 양방향 통신, 일회성 메시지 송수신





# ROS 노드간 통신 시나리오

- (1) 마스터(roscore) 시동:
  - 통신이 이루어지려면 우선은 roscore가 실행되고 있어야 함

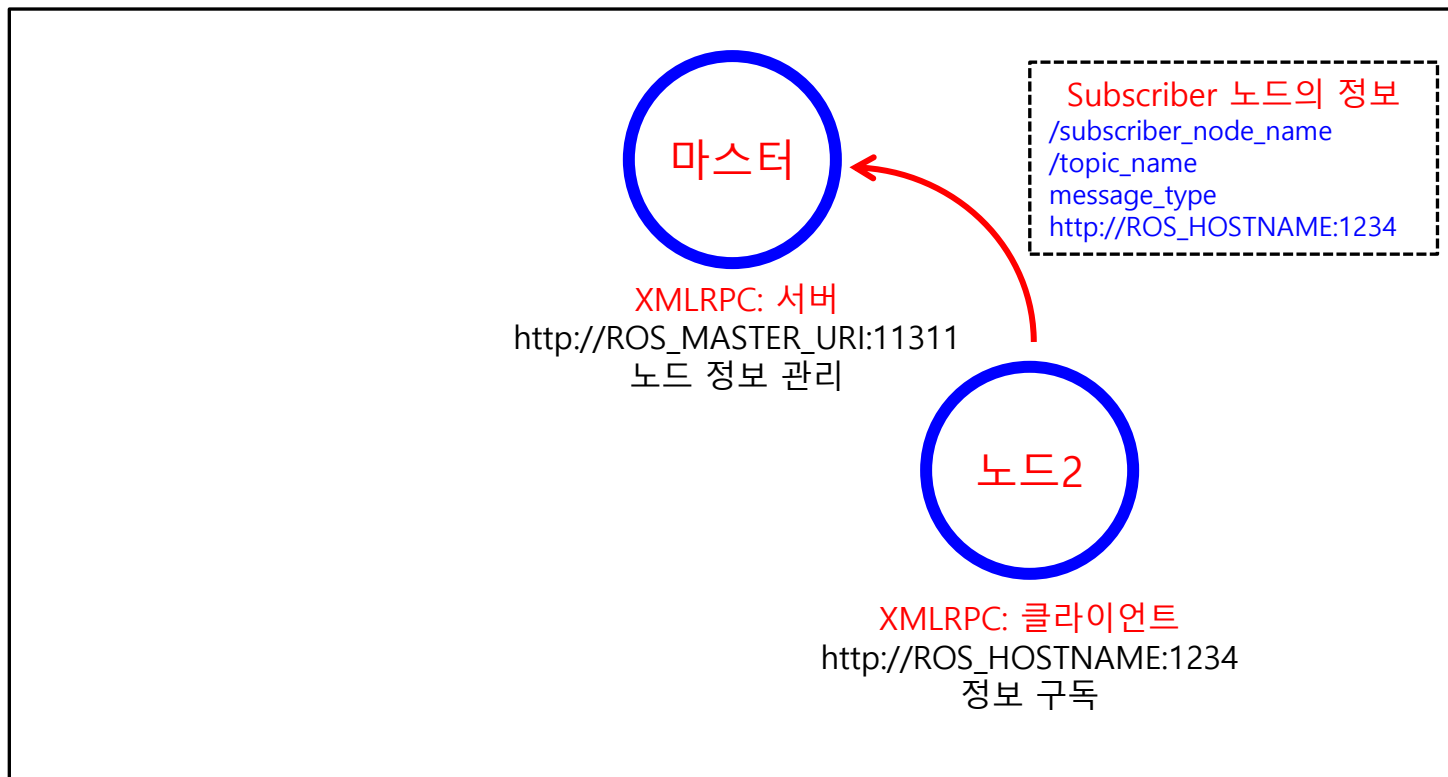


XMLRPC: 서버  
http://ROS\_MASTER\_URI:11311  
노드 정보 관리



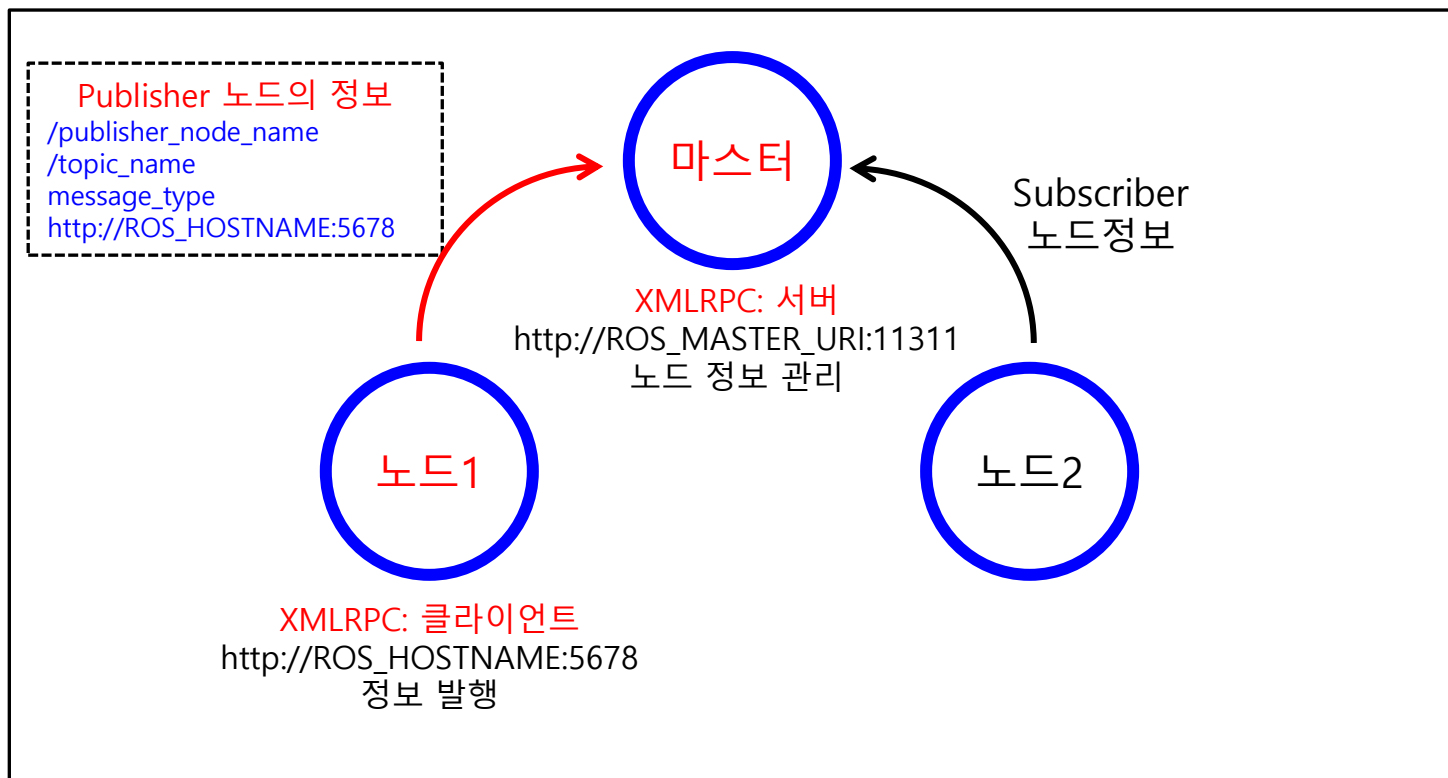
# ROS 노드간 통신 시나리오

- (2) 구독자(subscriber) 노드 구동:
  - 특정 토픽(topic)에 발행되는 메시지를 수신하기를 요청



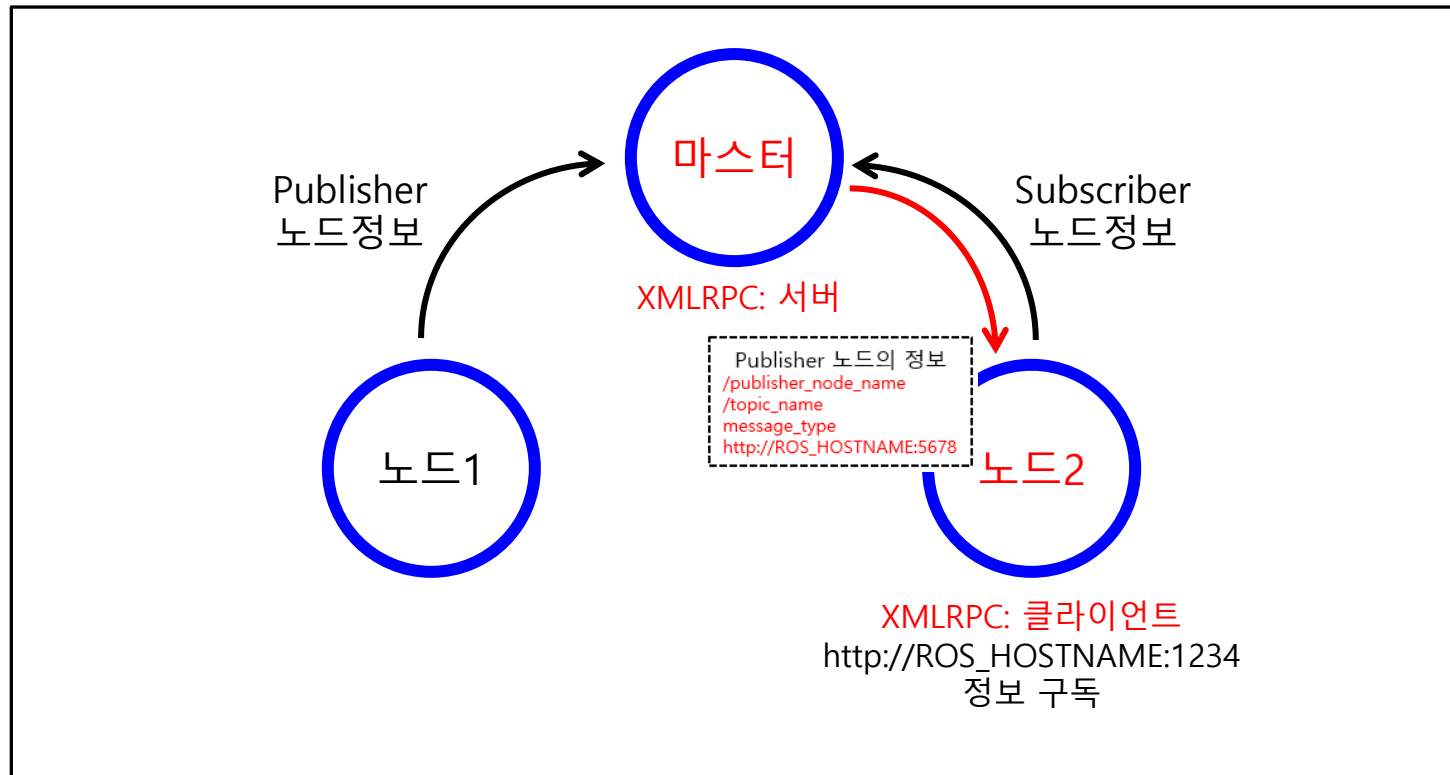
# ROS 노드간 통신 시나리오

- (3) 발행자(publisher) 노드 구동:
  - 특정 토픽(topic) 메시지를 발행하겠다는 의사를 전달



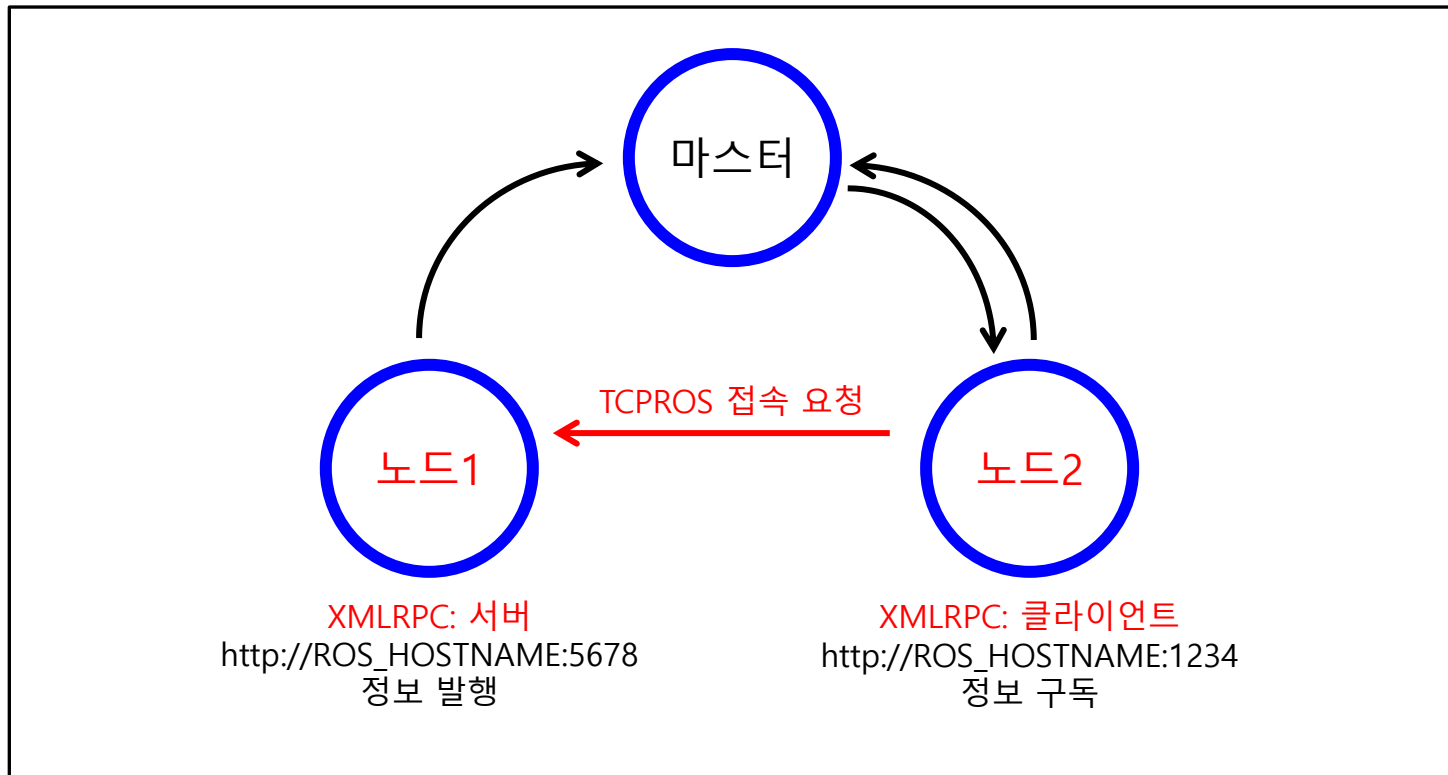
# ROS 노드간 통신 시나리오

- (4) 노드 정보 전달:
  - 마스터가 발행자 정보를 구독자에게 전달



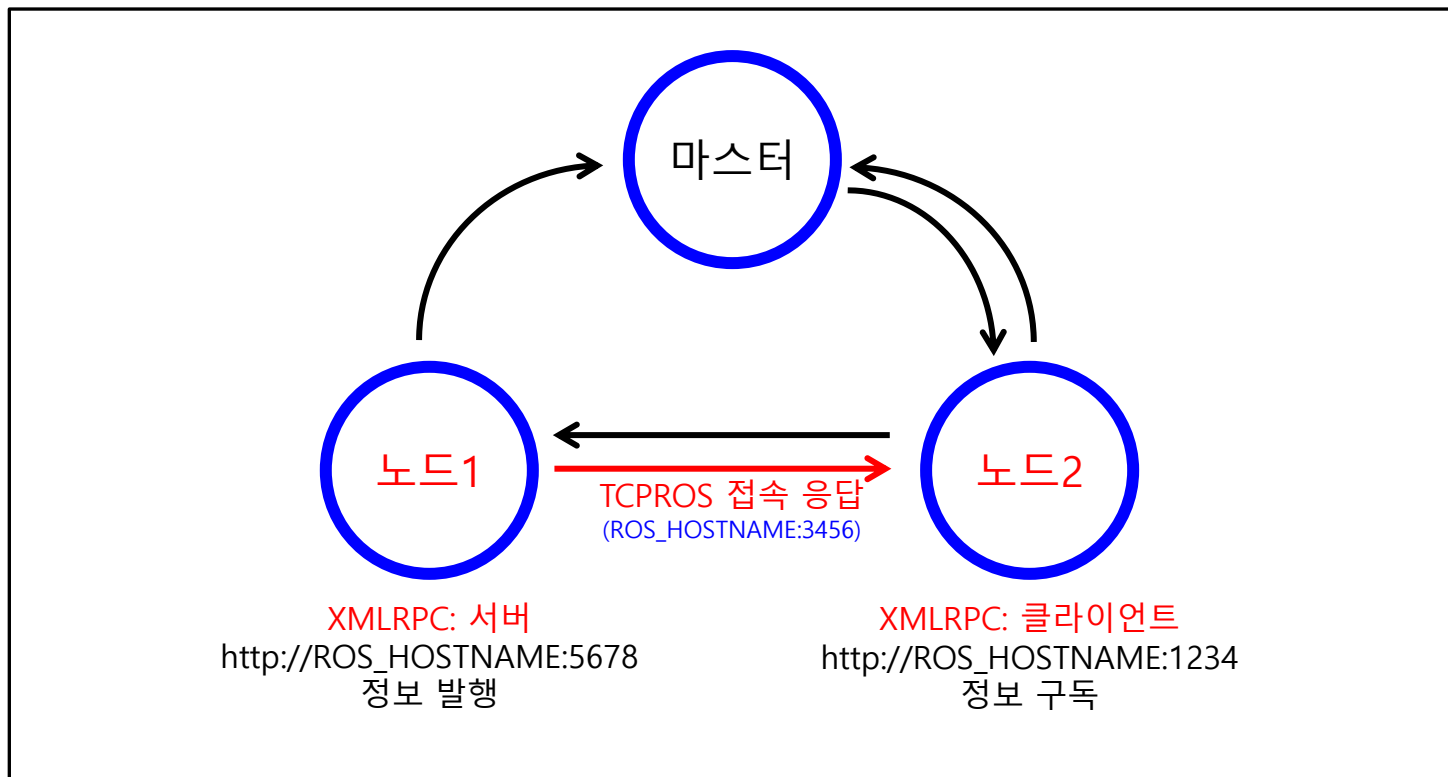
# ROS 노드간 통신 시나리오

- (5) 노드간 접속 요청:
  - 구독자 노드가 발행자 노드에 TCPROS 접속을 요청



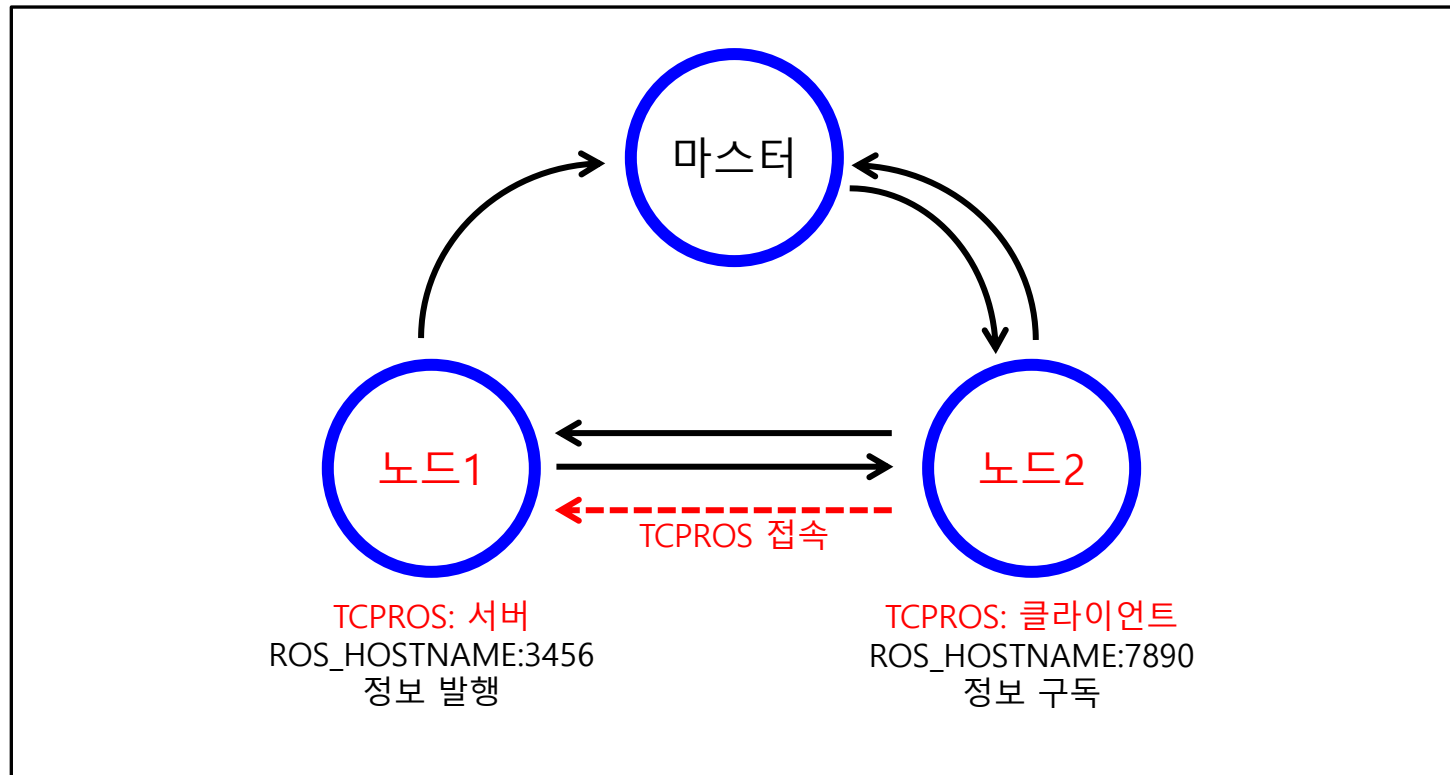
# ROS 노드간 통신 시나리오

- (6) 노드간 접속 요청에 대한 응답:
  - 발행자 노드가 자신의 TCPROS URI(포트 포함)를 전송하여 응답



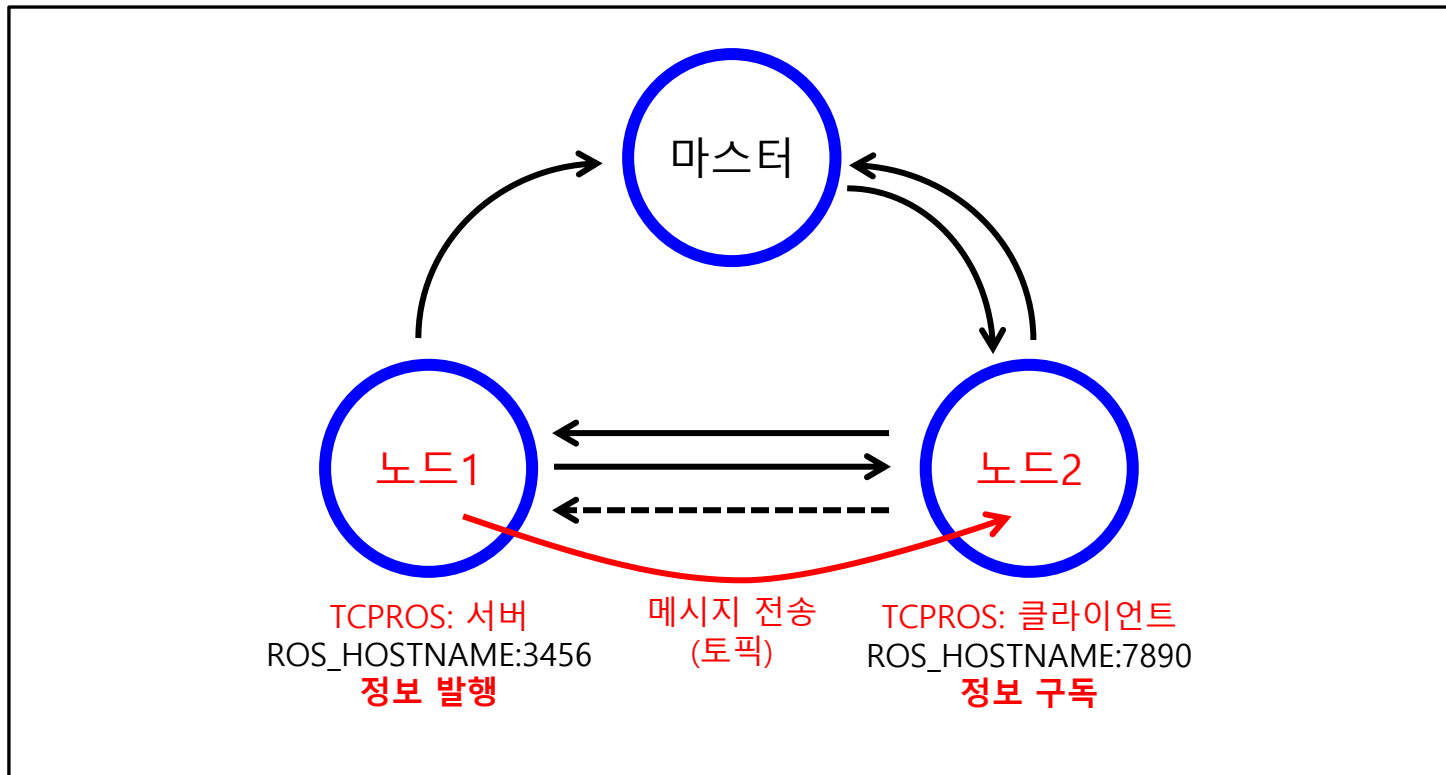
# ROS 노드간 통신 시나리오

- (7) TCPROS 접속:
  - 발행자 노드와 구독자 노드 사이에 소켓(socket) 연결이 이루어짐



# ROS 노드간 통신 시나리오

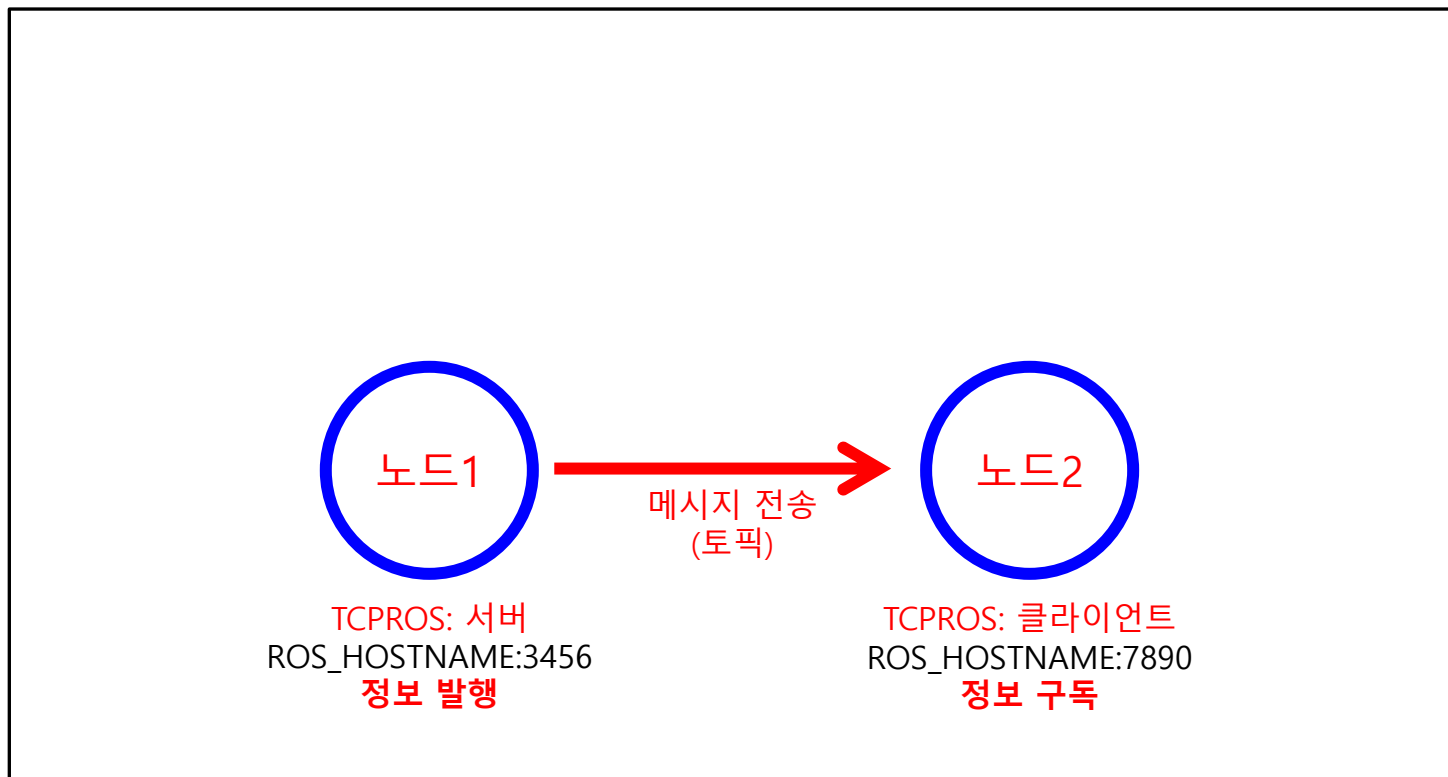
- (8) 메시지 전송
  - 발행자 노드가 구독자 노드에게 메시지 전송 (토픽)





# ROS 노드간 통신 시나리오

- (9) 메시지 전송 반복:
  - 접속이 한번 이루어진 뒤에는 별도 절차 없이 지속적으로 메시지 송수신





# ROS의 기초

ROS 명령어



# ROS 기본 용어

- 마스터 (Master)
  - 노드간 통신을 총괄 관리
  - ROS Core 라 불림
- 노드 (Node)
  - 토픽을 주고 받는 통신 주체
  - 실행가능한 최소단위의 프로세스
  - 하드웨어 장치에 하나씩  
또는 소프트웨어 모듈에 하나씩
- 토픽 (Topics)
  - 주고받는 메시지
  - 예: 센서데이터, 카메라이미지, 명령
- 발행자 노드 (Publisher)
  - 토픽(Topics)을 만들어 보내는 노드
  - 예: 정보수집센서, 카메라
- 구독자 노드 (Subscriber)
  - 토픽(Topics)를 받는 노드
  - 예: 모터제어기, 정보수집SW모듈
- 패키지
  - 하나 이상의 노드와 노드실행을 위한 정보 등을 묶어놓은 것
  - 노드, 라이브러리, 데이터, 설정파일 (configuration) 등을 포함

## ROS 셸 명령어

- roscd : 지정한 ros 패키지 폴더로 이동
- rosls : ros 패키지 파일 목록 확인
- rosed : ros 패키지 파일 편집
- roscp : ros 패키지 파일 복사

## ROS 실행 명령어

- roscore : master + rosout + parameter server
- rosrn : 패키지 노드 실행
- roslaunch : 패키지 노드를 여러 개 실행
- rosclean : ros 로그파일 검사 및 삭제

## ROS 정보 명령어

- rostopic : 토픽 정보 확인
- rosnode : 노드 정보 확인
- rosparm : 파라미터 정보 확인, 수정
- rosbag : 메세지 기록, 재생
- rosmg : 메세지 정보 확인
- rosverion : 패키지 및 배포 버전정보 확인
- roswtf : ros 시스템 검사

## ROS catkin 명령어

- catkin\_create\_pkg : catkin 빌드 시스템으로 패키지 생성
- catkin\_make : catkin 빌드 시스템으로 빌드
- catkin\_eclipse : 패키지를 eclipse에서 사용할 수 있게 변경
- catkin\_prepare\_release : changelog 정리 및 버전 태깅
- catkin\_init\_workspace : catkin 빌드 시스템의 작업 폴더 초기화
- catkin\_find : 검색

## ROS package 명령어

- rospack : 패키지와 관련된 정보 보기
- rosinstall : 추가 패키지 설치
- rosdep : 해당 패키지의 의존성 파일 설치
- roslocate : 패키지 정보 관련 명령어
- rosmake : 패키지 빌드 (구 시스템에서 사용)
- roscreate-pkg : 패키지 자동 생성 (구 시스템에서 사용)

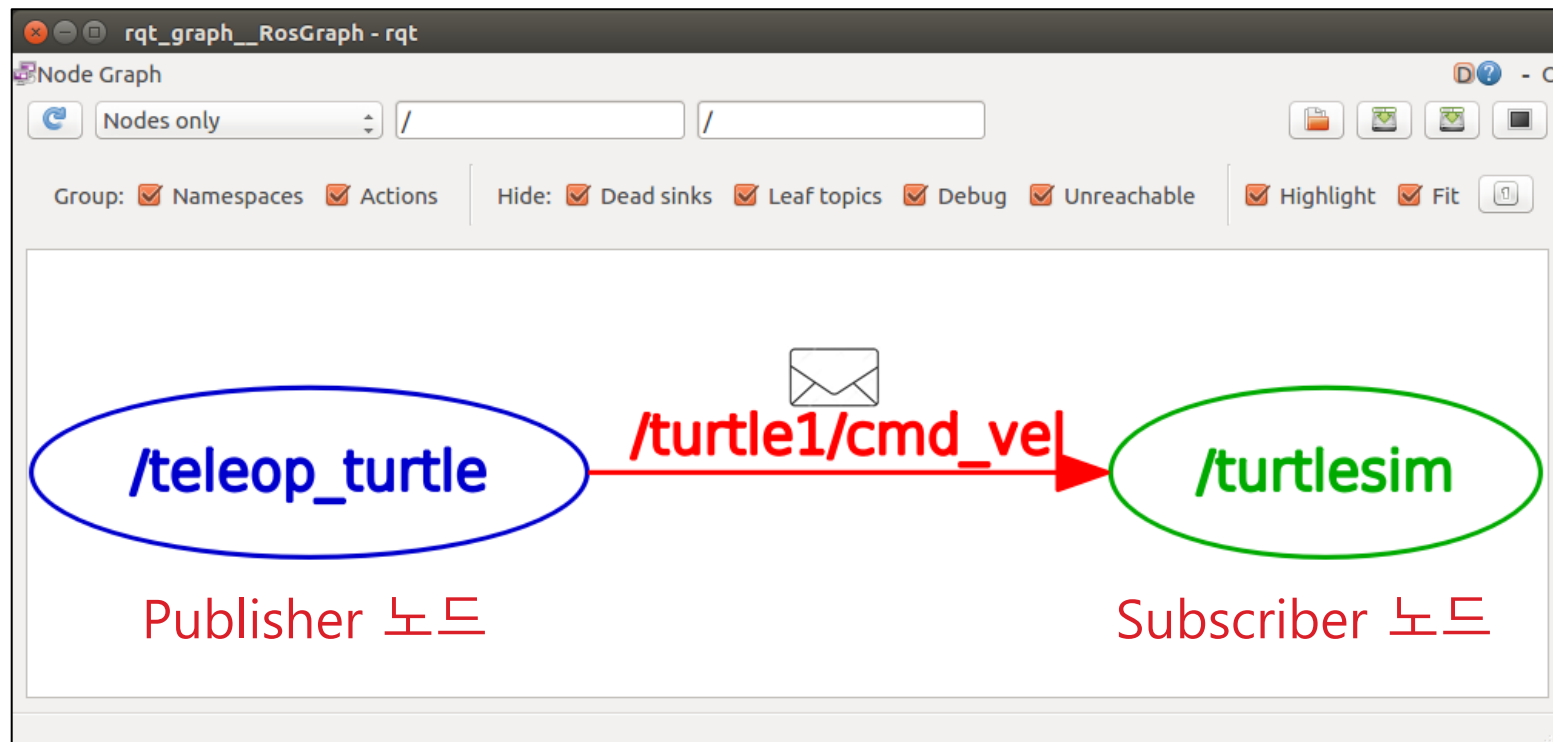
# ROS 주요 명령어

- roscore : ROS 기본 시스템이 구동되기 위해 필요한 프로그램들을 실행
  - `$ roscore`
- rosruntime [package name] [node\_name] : 패키지에 있는 노드를 선택 실행
  - `$ rosruntime turtlesim turtlesim_node`
- rosnodetop [info...] : 노드의 정보를 표시 (발행,구독 정보)
  - `$ rosnodetop info node_name`
- rostopic [option]: 토픽의 정보를 표시
  - `$ rostopic info /imu` (토픽의 정보를 출력 (메시지 타입, 노드 이름 등))
- roslaunch [package\_name] [file.launch]: 파라미터 값과 함께 노드를 실행
  - `$ roslaunch usb_cam usb_cam-test.launch`



# ROS에서 제공하는 쓸만한 도구

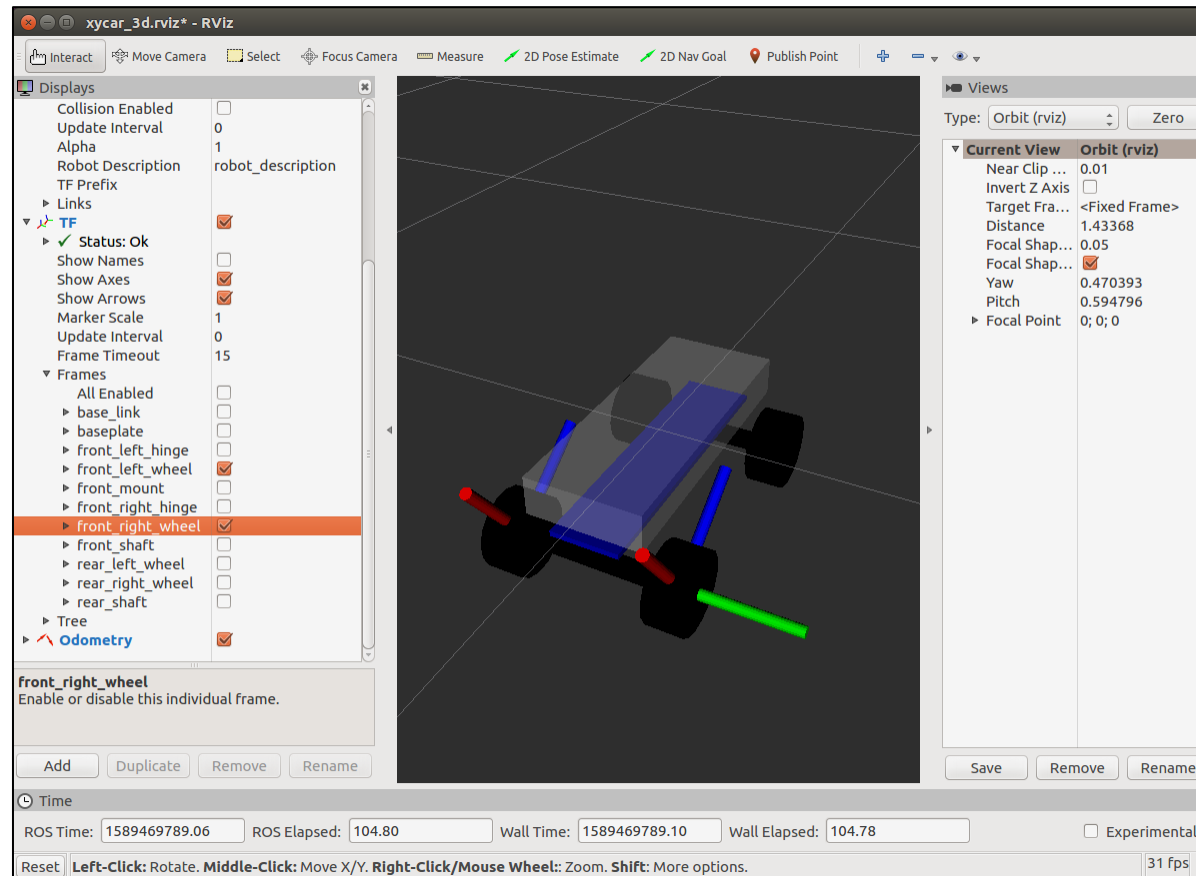
- rqt\_graph
  - 노드와 토픽의 관계 정보를 그래프로 출력
  - \$ rqt\_graph



# ROS에서 제공하는 쓸만한 도구

- RVIZ

- ROS의 3차원 시각화 도구, 각종 데이터를 보기 좋게 인포그래픽 스타일로 표시





# Q&A

---



감사합니다.



(주)자이트론

허성민

smher@xytron.co.kr

