

# ROS2란 무엇인가?

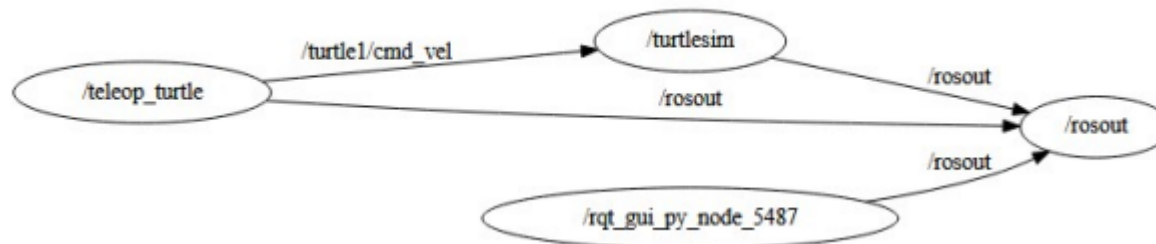
- ▶ ROS2는 여러가지 프로세스로 구성된 다양한 유형의 통신을 사용하는 로봇용 Middleware
  - 기본적으로 Node (프로세스) 와 Message (msg, Data Structure)로 구성되어 있음.

**Node**는 로봇을 위한 연산을 수행하는 프로세스

- 1) 하나의 프로세스로 로봇을 동작시키는 것 보다는 각 기능 별 프로세스를 실행하는 것이 로봇 구동 신뢰성을 높일 수 있음.
- 2) 예시) 제어 Node, 카메라를 위한 인식 Node, Lidar를 위한 Slam Node 등

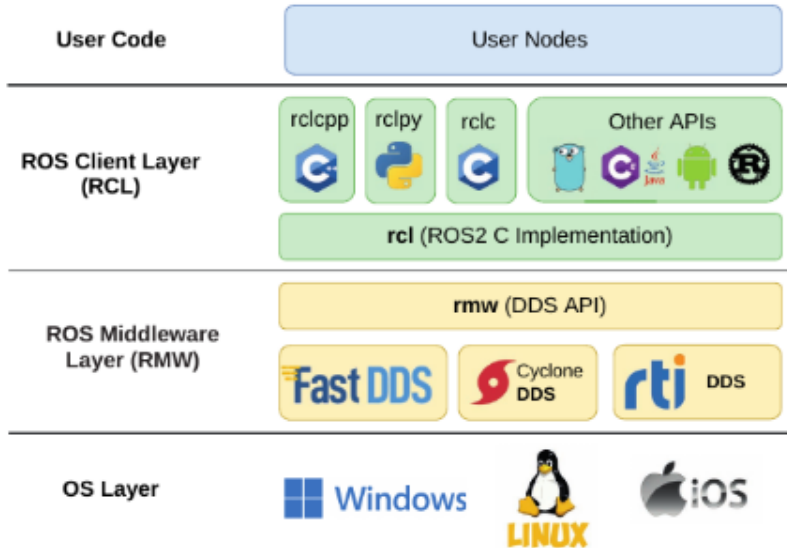
**Message**는 노드 간 통신에 전송되는 Data Structure

- 1) 즉 각 노드는 메시지를 사용하여 서로 통신함.
- 2) 표준 기본 유형이 지원됨 (integer, floating point, Boolean, string, ...)



# ROS2란 무엇인가?

## ➤ ROS2의 구조



**OS Layer:** 좀 더 다양한 운영체제와의 상호 통신이 가능

**ROS Middleware Layer (RMW):** ROS1은 UDP 통신을 지원한데 반에, ROS2는 Data Distribution Service (DDS)를 기반으로 통신을 함.

(선택: Opensource를 쓰거나, 돈을 내고 쓰거나 (보안 Issue))

**ROS Client Library (RCL):** 모든 ROS 2 요소의 기본 기능은 rcl이라는 단일 C 라이브러리에서 구현되었음. 그런 다음 rclcpp 및 rclpy 라이브러리는 이 기능을 각 언어, C++ 및 Python의 특수성에 각각 적용.

# ROS2란 무엇인가?

## ▶ ROS2는 여러가지 프로세스로 구성된 다양한 유형의 통신을 사용하는 로봇용 Middleware

- 로봇 중심의 특징들을 내포 : 대부분의 로봇 연구자들이 ROS를 사용하는 이유

### Robot에 대한 표준 메시지 정의 (2차시 ROS2 개발툴)

- 1) Pose, Transform, vector 등 로봇 제어에 필요한 message
- 2) IMU, PointCloud, Image 등 로봇 센서리 시스템에 필요한 message

### Robot 기하학 라이브러리 제공 (7차시 ROS2 Tools)

- 1) TF 라이브러리는 로봇의 좌표 변환을 관리함. (예: 베이스에서 바라보는 물체의 위치)

### Robot Description 제공 (8차시 ROS2 시뮬레이션)

- 1) URDF (Unified Robot Description Format)을 사용하여, XML 방식으로 로봇의 형태, 바퀴의 형태 등을 관리
- 2) 로봇의 물리적 특성 (바퀴의 크기, 마찰력, 무게 등)과 센서 들의 정보 (카메라 해상도 등)을 관리

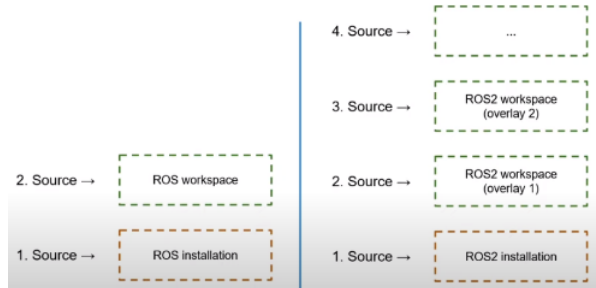
### 원격 관제 시스템 제공 (2차시 ROS2 개발툴)

- 1) Behavior를 관제하거나, 원격으로 로봇의 동작 상태 등을 관제할 수 있음.

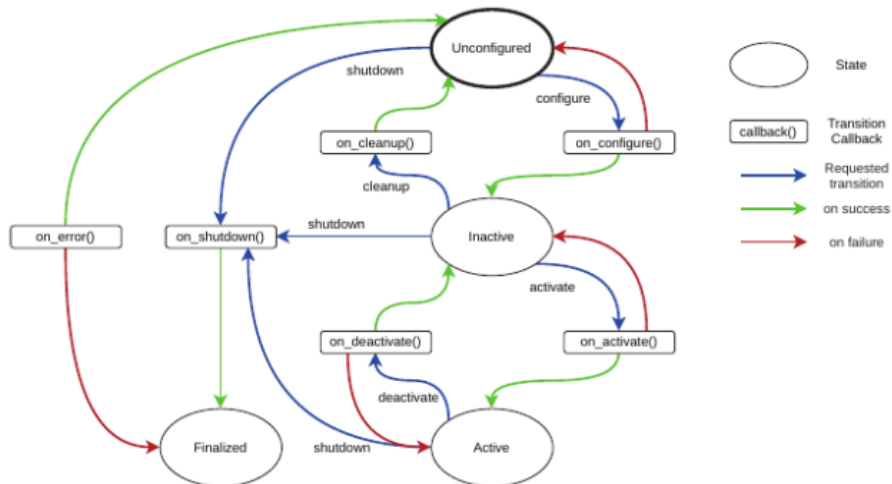
# ROS2란 무엇인가?

## ➤ ROS2 vs ROS1

- 분산형 아키텍처 (roscore의 삭제)
- 통신 방식의 다양화 (Topic, Service, Action)
- 통신 서비스 품질을 선택할 수 있음 (QoS)
- 빌드 환경이 catkin에서 colcon으로 전환
- 작업공간의 다양화 (Overlay)



## - 라이프사이클 노드의 탄생



# ROS2란 무엇인가?

## ▶ ROS2는 여러가지 프로세스로 구성된 다양한 유형의 통신을 사용하는 로봇용 Middleware

– Low level에서 ROS 는 다음의 기능을 제공

### Publish/Subscribe Message 전달 (3차시 ROS 통신 – Topic)

- 1) 익명의 Publish/Subscribe 동기/비동기 통신을 통한 분산 노드 간 통신
- 2) 통신을 기록 (recording) 해 두고, 이를 다시 재생 (Playback) 하여, 디버깅을 하기 용이하게 함.

### Request/Response 원격 호출 (5차시 ROS 통신 – Service)

- 1) 각 프로세스 간 동기식 호출/응답 시스템 제공



# ROS2란 무엇인가?

---

- ▶ ROS2는 여러가지 프로세스로 구성된 다양한 유형의 통신을 사용하는 로봇용 Middleware
  - ROS2의 기본적인 Tool들 (Cheat Sheet)

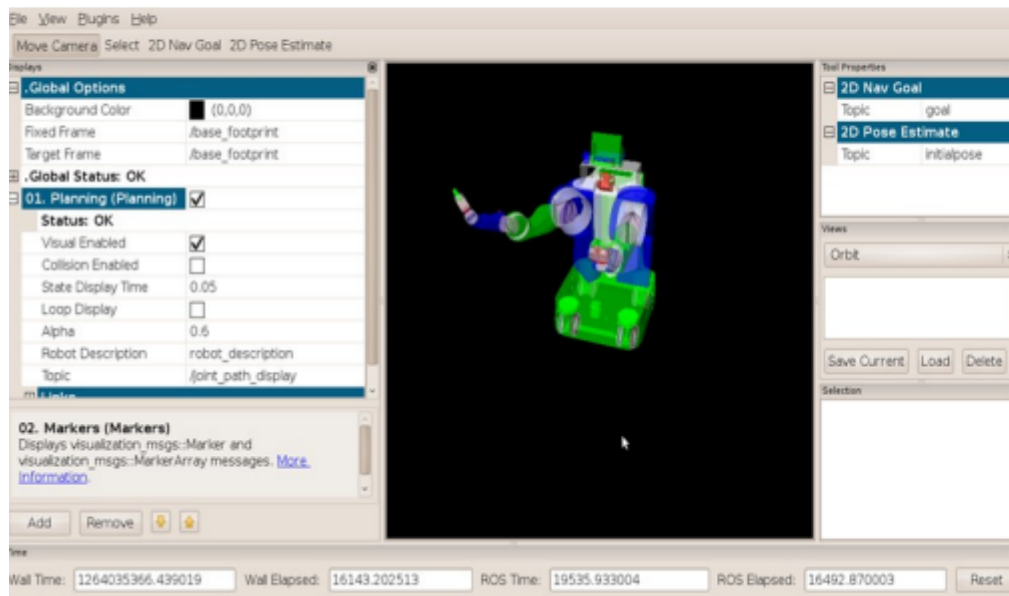
- ros2 daemon: Introspect/configure the ROS 2 daemon
- ros2 launch: Run a launch file
- ros2 lifecycle: Introspect/manage nodes with managed lifecycles
- ros2 interface: Introspect msg/srv types
- ros2 node: Introspect ROS nodes
- ros2 param: Introspect/configure parameters on a node
- ros2 pkg: Introspect ROS packages
- ros2 run: Run ROS nodes
- ros2 security: Configure security settings
- ros2 service: Introspect/call ROS services
- ros2 topic: Introspect/publish ROS topics

# ROS2란 무엇인가?

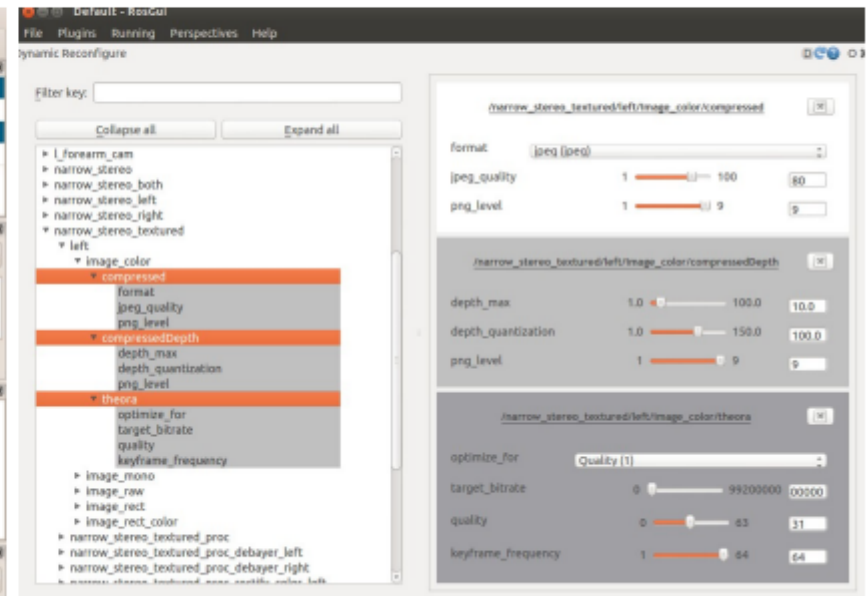
- ▶ ROS2는 여러가지 프로세스로 구성된 다양한 유형의 통신을 사용하는 로봇용 Middleware
  - ROS2의 기본적인 Tool들 (Cheat Sheet)

## Graphical Tools

rviz



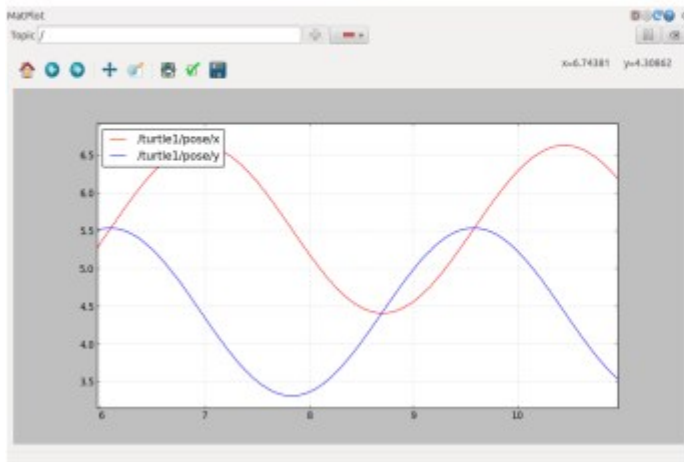
rqt\_reconfigure



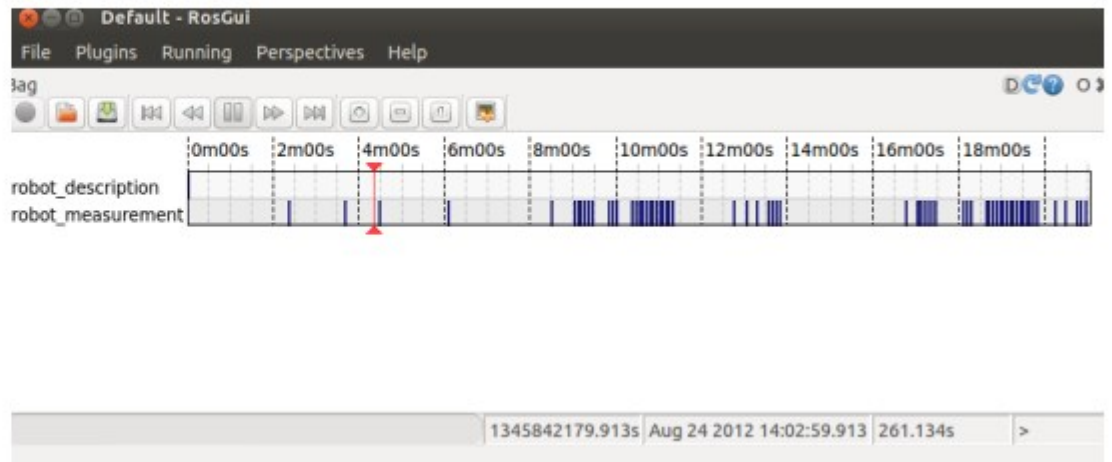
# ROS2란 무엇인가?

- ▶ ROS2는 여러가지 프로세스로 구성된 다양한 유형의 통신을 사용하는 로봇용 Middleware
  - ROS2의 기본적인 Tool들 (Cheat Sheet)

rqt\_plot



rqt\_bag



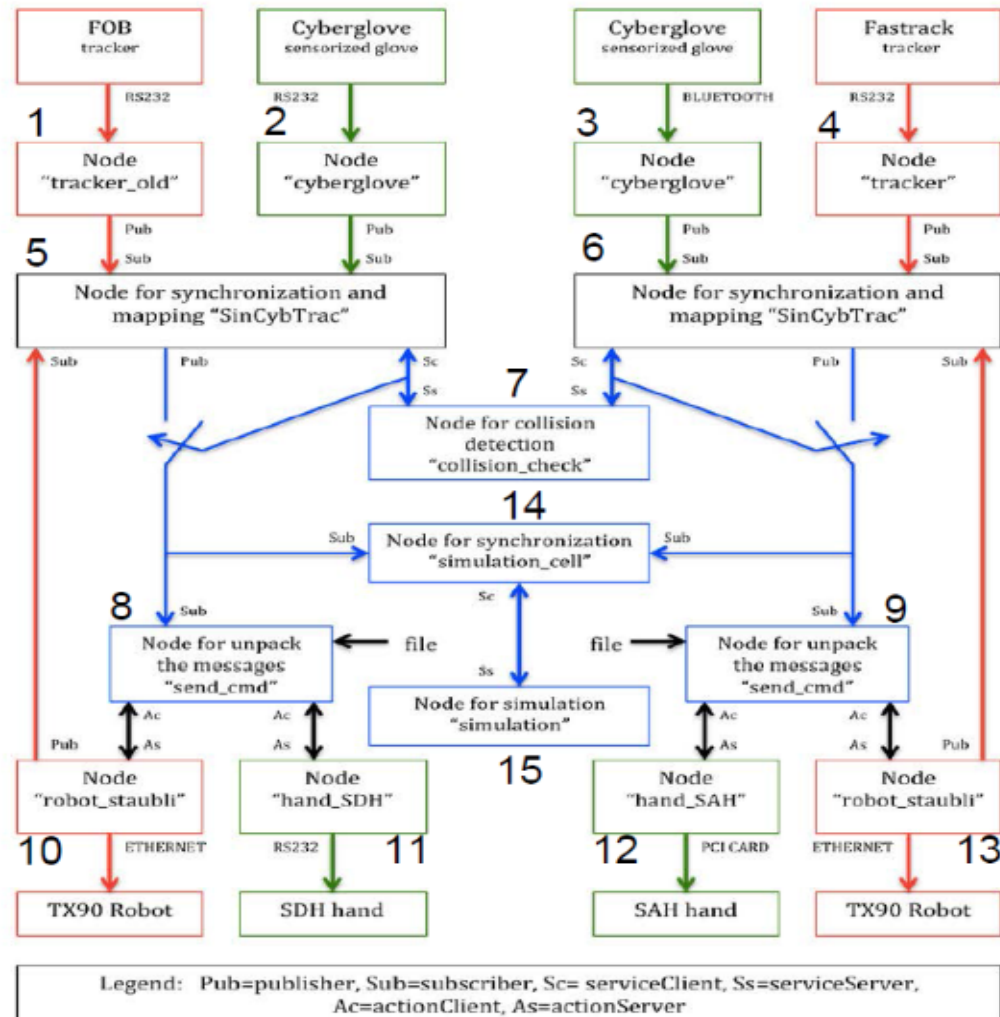
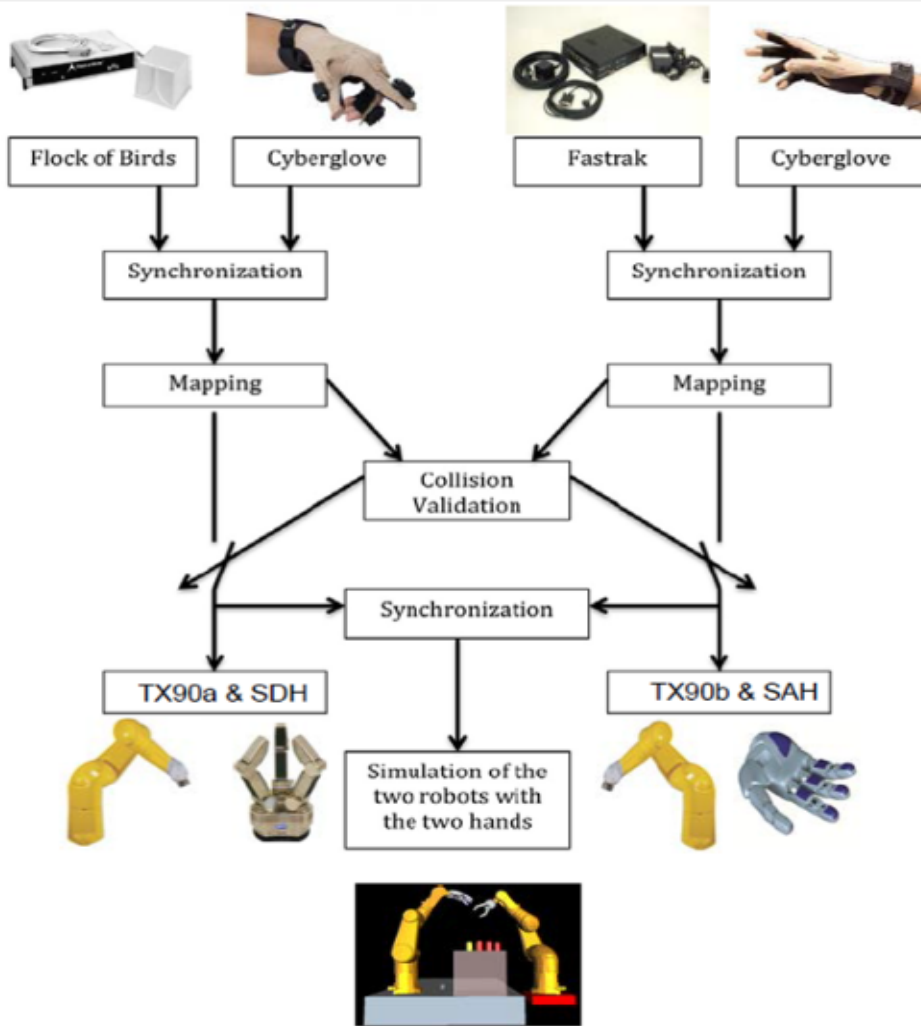


---

04

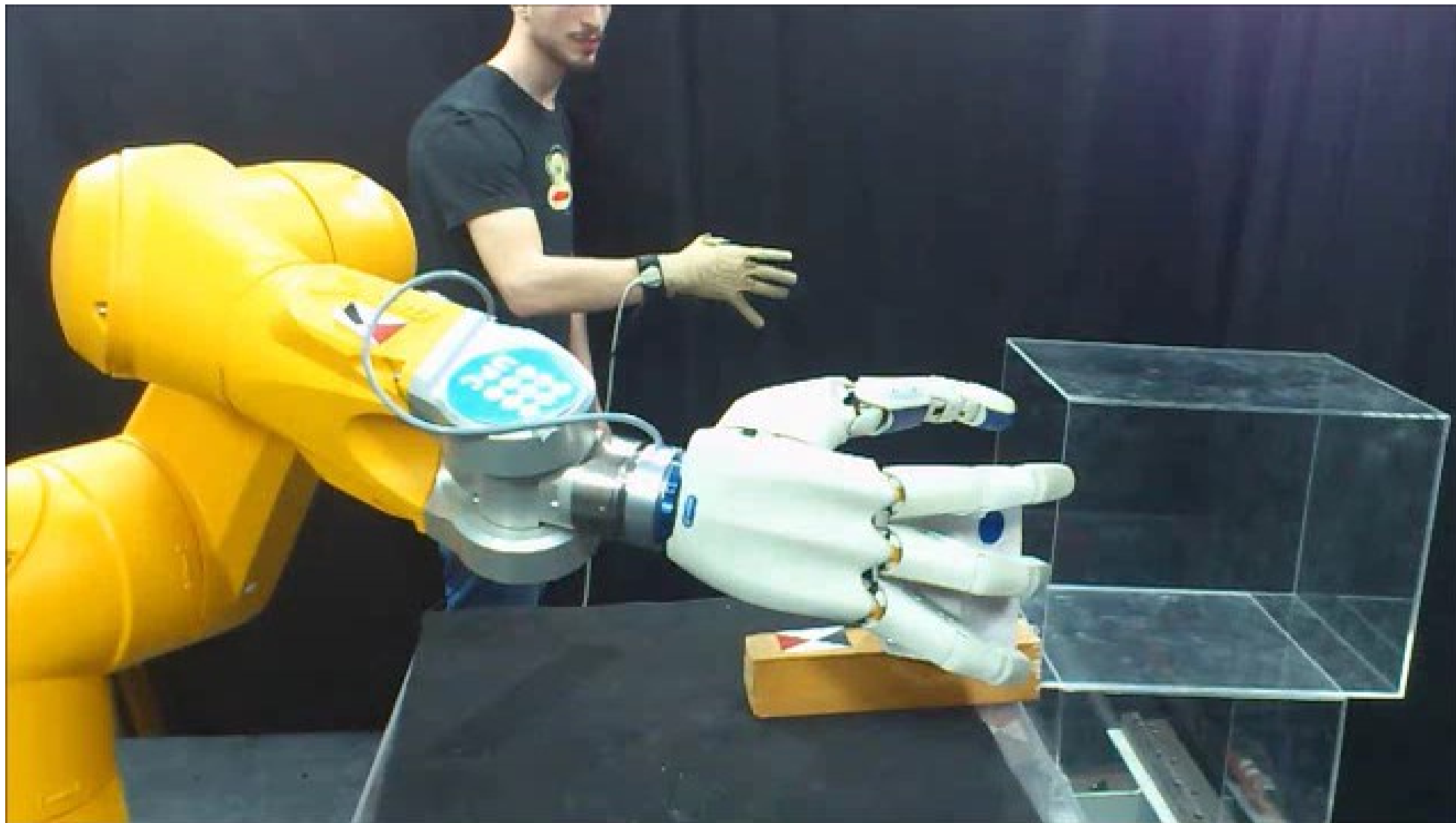
# ROS2의 사용 예

# ROS2의 사용 예



- 노드 1-4: 글로브와 손의 움직임을 트래킹해서 이 위치를 보냄 (Topic Publish)
- 노드 5-6: 노드 1-4의 위치를 획득 (Topic Subscribe), 이를 7번 노드에 충돌 감지를 해주라 요청 (Service Request)
- 노드 8-9: 최적의 위치를 Node 5-6에 전송 받아 (Topic Subscribe), 로봇의 움직임을 부여 (Node 10-13)

# ROS2의 사용 예



---

05

# ROS2의 설치법 (Humble)

# ROS2의 설치법

- ▶ Ubuntu 22.04 기준 ROS2 Humble 설치 (humble/Installation/Ubuntu-Install-Debians - [ros.org](https://www.ros.org) 참고)

## System 설정확인 (UTF-8)

```
$ locale # check for UTF-8
$ sudo apt update && sudo apt install locales
$ sudo locale-gen en_US en_US.UTF-8
$ sudo update-locale LC_ALL=en_US.UTF-8 LANG=en_US.UTF-8
$ export LANG=en_US.UTF-8
$ locale # verify settings
```

## Setup Sources

```
$ sudo apt install software-properties-common
$ sudo add-apt-repository universe
$ sudo apt update && sudo apt install curl -y
```

### Add Ros2 Key

```
$ sudo curl -sSL https://raw.githubusercontent.com/ros/rosdistro/master/ros.key -o /usr/share/keyrings/ros-archive-keyring.gpg
```

### Add Repository

```
$ echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/ros-archive-keyring.gpg]
http://packages.ros.org/ros2/ubuntu $(. /etc/os-release && echo $UBUNTU_CODENAME) main" | sudo tee
/etc/apt/sources.list.d/ros2.list > /dev/null
```

# ROS2의 설치법

- ▶ Ubuntu 22.04 기준 ROS2 Humble 설치 (humble/Installation/Ubuntu-Install-Debian - [ros.org](https://www.ros.org/doc/humble/Installation/Ubuntu-Install-Debian/) 참고)

## ROS2 설치

```
$ sudo apt update & sudo apt upgrade  
$ sudo apt install ros-humble-desktop  
$ sudo apt install ros-dev-tools
```

## 환경설정

```
$ echo "source opt/ros/humble/setup.bash" >> ~/.bashrc
```

- ▶ 추가 패키지들 설치 (Optional)

## ROS Control 관련

```
$ sudo apt install ros-humble-ros2-control ros-humble-ros2-controllers  
$ sudo apt install ros-humble-rqt*  
$ sudo apt install ros-humble-gazebo-ros-pkgs ros-humble-gazebo-ros2-control
```

## ROS TF 관련

```
$ sudo apt-get install ros-humble-tf2 ros-humble-tf2-tools ros-humble-tf-transformations
```

## URDF 관련

```
$ sudo apt install ros-humble-urdfdom
```

감사합니다

