

ROS(Robot Operating System) 기초

- ROS란? : 로봇 소프트웨어를 개발하는데 필요한 소프트웨어의 집합체

메타 운영체제(Meta OS), 미들웨어 (소프트웨어 모듈 + 라이브러리 집합 + 도구 집합)

자율주행차에는 ROS , 스마트폰에는 안드로이드!

- ROS 구성도

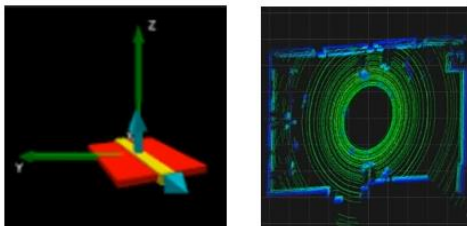


- ROS 특징

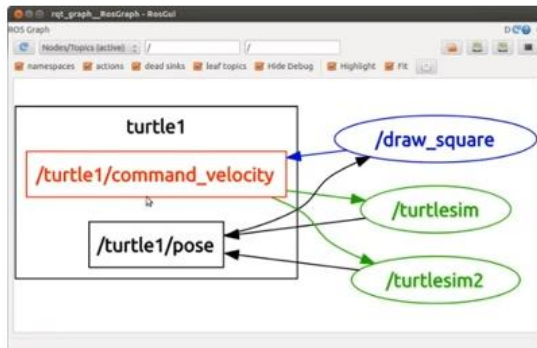
1. 로봇 SW를 만들기 위한 코드의 재사용이 용이한 환경제공이 목표
2. 다양한 도구들을 함께 제공
3. 다양한 OS 환경에서 통일된 방법으로 상호작용을 구현하는 것이 가능

- ROS가 제공하는 도구들

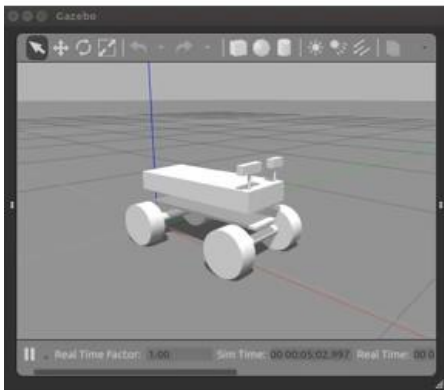
- RVIZ : 시각화 도구, 센서데이터를 비롯한 주변환경 변화를 시각화



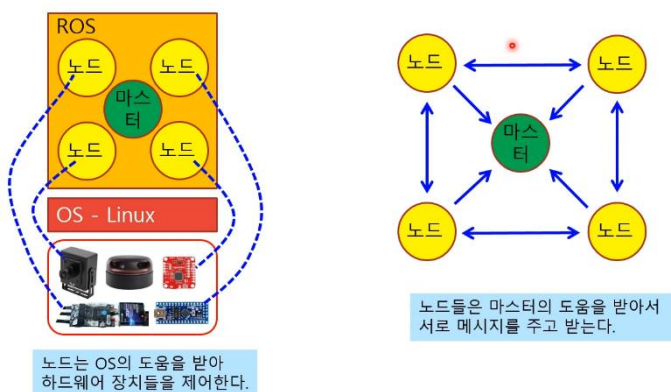
- RQT : Qt기반의 GUI 응용 개발도구, 노드 연결 정보를 그래프로 표현
사용자 상호작용을 UI를 갖춘 응용 개발에 이용



- GAZEBO : 물리 엔진 기반의 3차원 시뮬레이터, 시뮬레이터 제작 및 모델링에 이용



- ROS에서의 통신



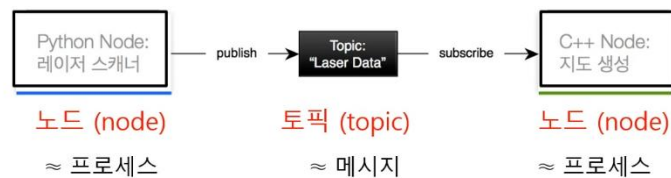
- ROS에서 사용하는 기본 용어들

1. Master : 서로 다른 노드들 사이의 통신을 총괄 관리.('ROS Core' 라고 부름)
2. Nodes : 실행 가능한 최소의 단위, 프로세스로 이해 할 수 있음.

ROS에서 발생하는 통신(메시지 송/수신)의 주체

3. Topics : ROS 노드들에게 메시지를 전달 (Topic안에 메시지가 담겨져서 전달)
4. Publishers : 특정 토픽에 메시지를 담아 외부로 송신하는 노드
5. Subscribers : 특정 토픽에 담겨진 메시지를 수신하는 노드
6. Packages : 하나 이상의 노드와 노드의 실행을 위한 정보 등을 묶어 놓은 단위

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



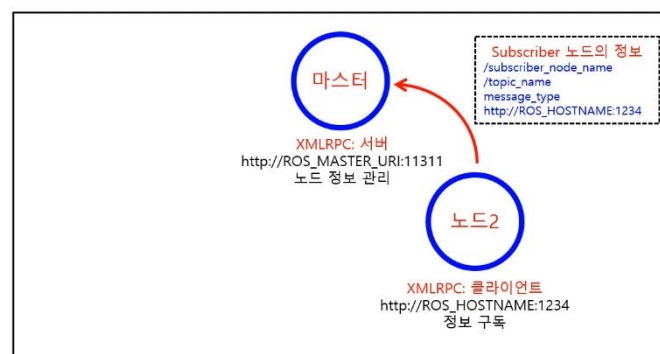
ROS는 노드간 통신을 기반으로 전체 시스템을 구동 시킴

- 하드웨어 부품 또는 소프트웨어 모듈에 노드가 하나씩 할당됨

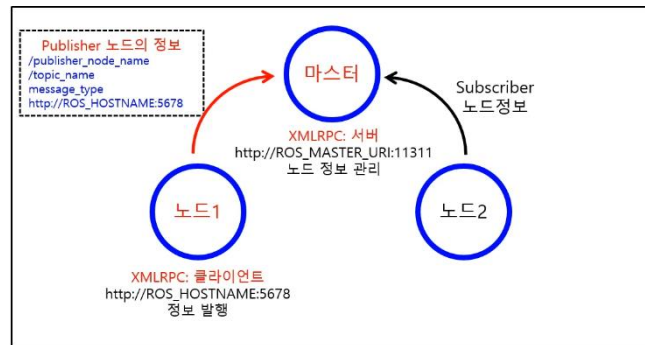
네트워크를 통한 원격 연동도 가능(물리적으로 멀리 있어도 인터넷만 연결되면 동작)

- ROS 노드간 통신 시나리오

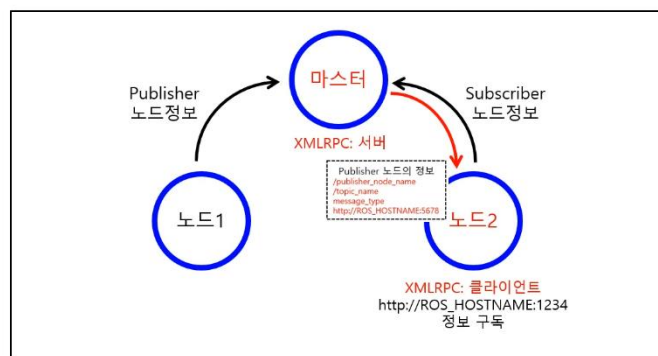
- (1) 마스터 시동 : 통신이 이루어 지려면 우선 roscore가 실행되고 있어야 함
- (2) 구독자 노드 구동 : 마스터로 자신의 노드 정보 및 자신이 받고싶은 토픽 정보도 전송



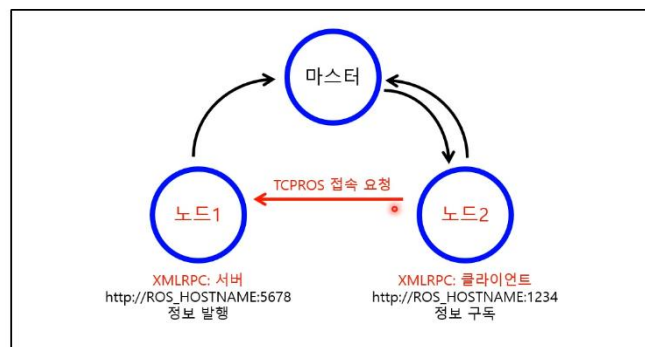
(3) 발행자 노드 구동 : 마스터로 자신의 노드 정보 및 자신이 전송하려고 하는 토픽 정보 전송



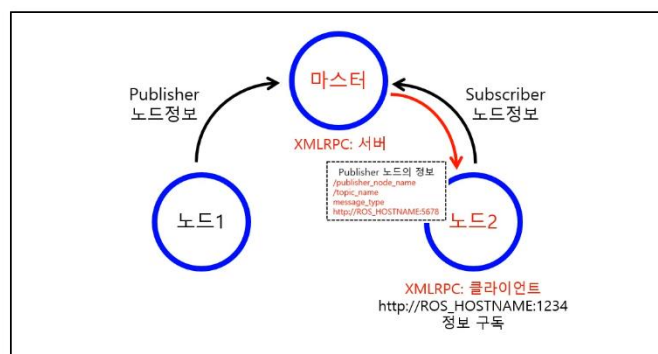
(4) 노드 정보 전달 : 발행자의 정보를 구독자에게 전달



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

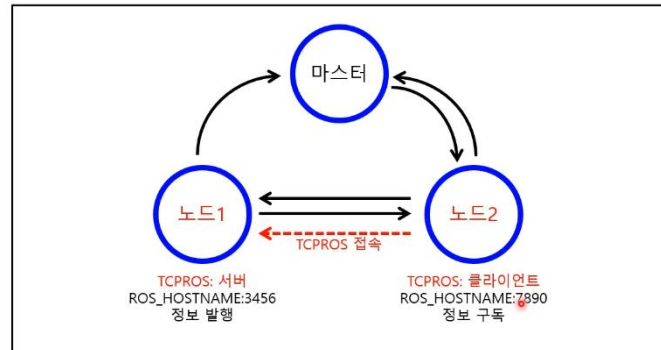


(6) 노드간 접속 요청에 대한 응답 : 발행자 노드가 자신의 TCPROS URI를 전송하여 응답

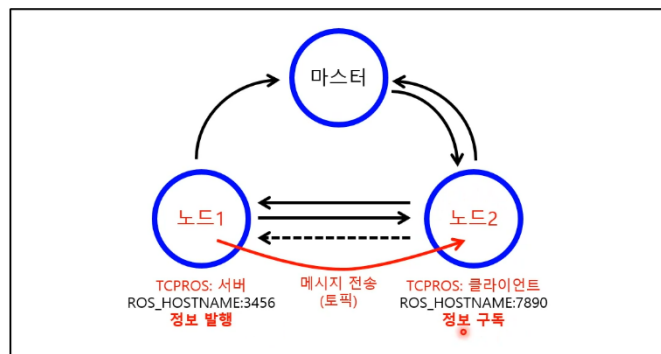


(7) TCPROS 접속 : 발행자 노드와 구독자 노드 사이에 소켓 연결이 이루어짐

두 노드가 1 대 1 통신하기 위한 주소정보를 공유하고 통신



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



(9) 메시지 전송 반복: 접속이 한번 이루어지면 지속적으로 메시지 전송(마스터는 있어야 함)

