ROS Introduction

WeGo & WeCAR

WeGo

목 차

WeGo

01 ROS 소개

Introduction

01 ROS 소개

- ROS(Robot Operating System)는 로봇용 오픈 소스 *메타 운영 체제
- 일반 운영체제에서 제공하는 Hardware abstraction, Low-level device control 등의 기능과 프로세스 사이의 메시지 전달, 패키지 관리 기능을 제공
- 여러 컴퓨터 시스템의 코드를 빌드, 작성, 실행하기 위한 도구와 라이브러리를 제공

- *메타 운영 체제: Windows, MacOS와 같은 일반적인 운영체제가 아닌, 기존 운영체제 위해 추가 설치하여, 로봇 및 센서를 제어하고 개발하기 위한 시스템

WeGo

01 ROS 소개

- ROS 사용 목적
- 로봇의 연구 개발 시, 모든 사람이 협업하고 공유하기 위함
- 모든 로봇을 제어할 수 있는 다양한 기능의 프레임워크를 지원하지는 않지만,

  약간의 모듈을 추가하여 편리하게 코드를 재사용할 수 있음
- 디버깅 및 시각화, 시뮬레이션 등의 다양한 도구를 통해 개발의 속도를 향상
- 운영체제 및 프로그래밍 언어와 상관 없이, 각 프로세스(노드) 사이의 통신이 가능
- 독립적인 프로세스로 동작

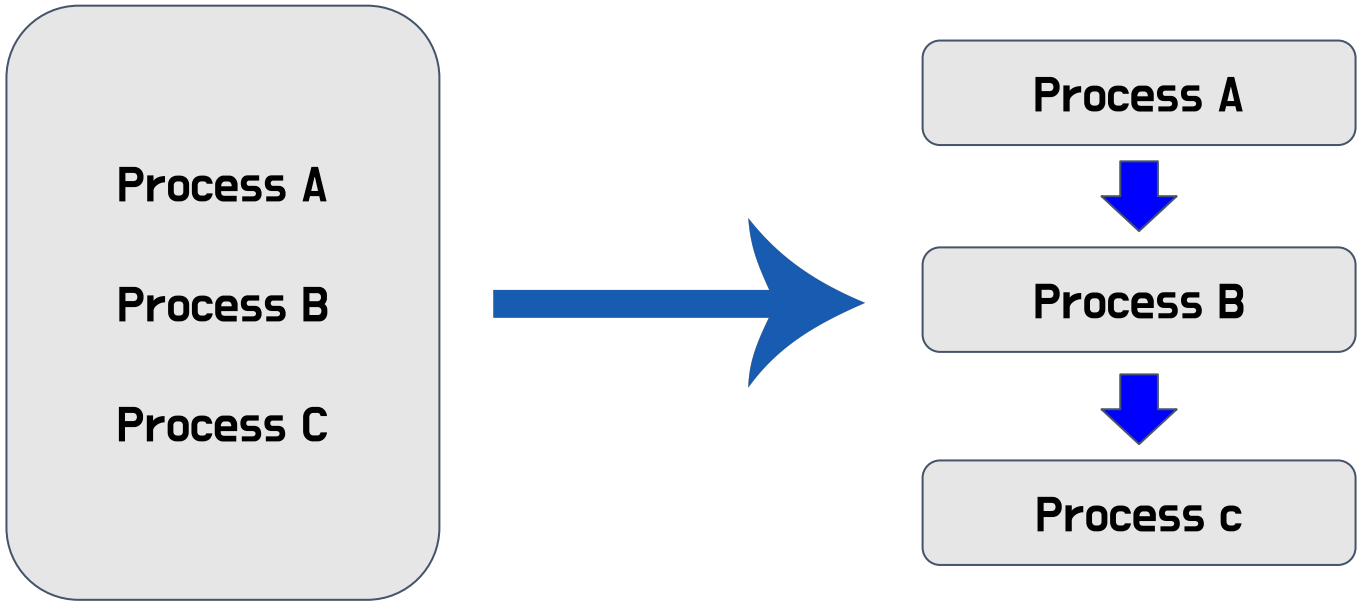
  → 하나의 노드에서 오류가 발생하여도 시스템이 지속적으로 작동할 수 있음

01 ROS 소개

- ROS 사용 목적
- 로봇의 연구 개발 시, 모든 사람이 협업하고 공유하기 위함
- 모든 로봇을 제어할 수 있는 다양한 기능의 프레임워크를 지원하지는 않지만,
  약간의 모듈을 추가하여 편리하게 코드를 재사용할 수 있음
- 디버깅 및 시각화, 시뮬레이션 등의 다양한 도구를 통해 개발의 속도를 향상
- ~~운영체제 및 프로그래밍 언어와 상관 없이,~~
  Ubuntu에서 Python과 C++를 기반으로, 프로세스(노드) 사이의 통신이 가능
- 독립적인 프로세스로 동작
  → 하나의 노드에서 오류가 발생하여도 ~~시스템이 지속적으로 작동할 수 있음~~
  → 마스터가 아닌 일반적인 노드에서 오류가 발생하여도, 이에 대해 보험적으로 코드를
  작성하였을 때, 시스템이 지속적으로 작동할 수 있음

01 ROS 소개

- ROS가 그래서 뭔데?
- 여러 프로세스 사이의 통신 부분을 담당하는 중간 다리 역할을 수행
- 중간 통신 과정을 담당하여, Process를 분리하여 처리를 효율적으로 할 수 있음



without ROS

with ROS

WeGo

02

ROS 개발환경 구축

Environment Setting

WeGo

02 ROS 개발환경 구축

- ROS는 MacOS, Windows, Linux 등의 다양한 운영체제를 지원하지만, 공식적으로 지원하는 운영체제는 Linux Ubuntu임.
- Ubuntu Bionic Beaver(18.04 LTS)를 기준으로 사용 가능한 ROS Melodic Morenia를 설치
- 최소 시스템 요구 사항
  - 하드웨어 : AMD, Intel 및 ARM64 기반의 Chip
  - 운영체제 : Ubuntu Bionic Beaver (18.04 LTS)
  - ROS 버전 : ROS Melodic Morenia

02 ROS 개발환경 구축

1. sources.list에 packages.ros.org의 소프트웨어를 설치를 위해 등록
   - $ sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release -sc) main" > /etc/apt/sources.list.d/ros-latest.list'
2. ROS 저장소에서 패키지를 받기 위해, 키를 추가합니다. 정상적으로 등록되지 않을 경우, http://wiki.ros.org/ 를 확인하여, 키를 등록합니다.
   - $ sudo apt-key adv --keyserver 'hkp://keyserver.ubuntu.com:80' --recv-key C1CF6E31E6BADE8868B172B4F42ED6FBAB17C654

WeGo

02 ROS 개발환경 구축

3. 패키지 인덱스를 업데이트합니다.
    - $ sudo apt update
4. ROS Melodic을 설치합니다. 필요에 따라 다른 버전을 설치할 수도 있지만, desktop-full을 기준으로 설치하도록 하겠습니다.
    - $ sudo apt install ros-melodic-desktop-full
5. ROS 사용을 위해, rosdep을 초기화합니다.
    - $ sudo rosdep init
    - $ rosdep update

02 ROS 개발환경 구축

6. 새로운 터미널이 실행될 때마다, ROS 환경 변수가 자동으로 추가되도록 환경 설정합니다.
   - $ echo "source /opt/ros/melodic/setup.bash" >> ~/.bashrc
   - $ source ~/.bashrc
7. ROS 패키지 이외에 필요한 다른 패키지를 설치합니다.
   - $ sudo apt install python-rosinstall python-rosinstall-generator python-wstool build-essential
8. 빌드 툴로 사용하는 catkin을 이용하여, 작업 폴더를 초기화하고 초기 빌드를 진행 (일반적으로 catkin_ws를 사용하지만 원하는 이름 사용 가능)
   - $ mkdir -p ~/catkin_ws/src && cd ~/catkin_ws
   - $ catkin_make

02 ROS 개발환경 구축

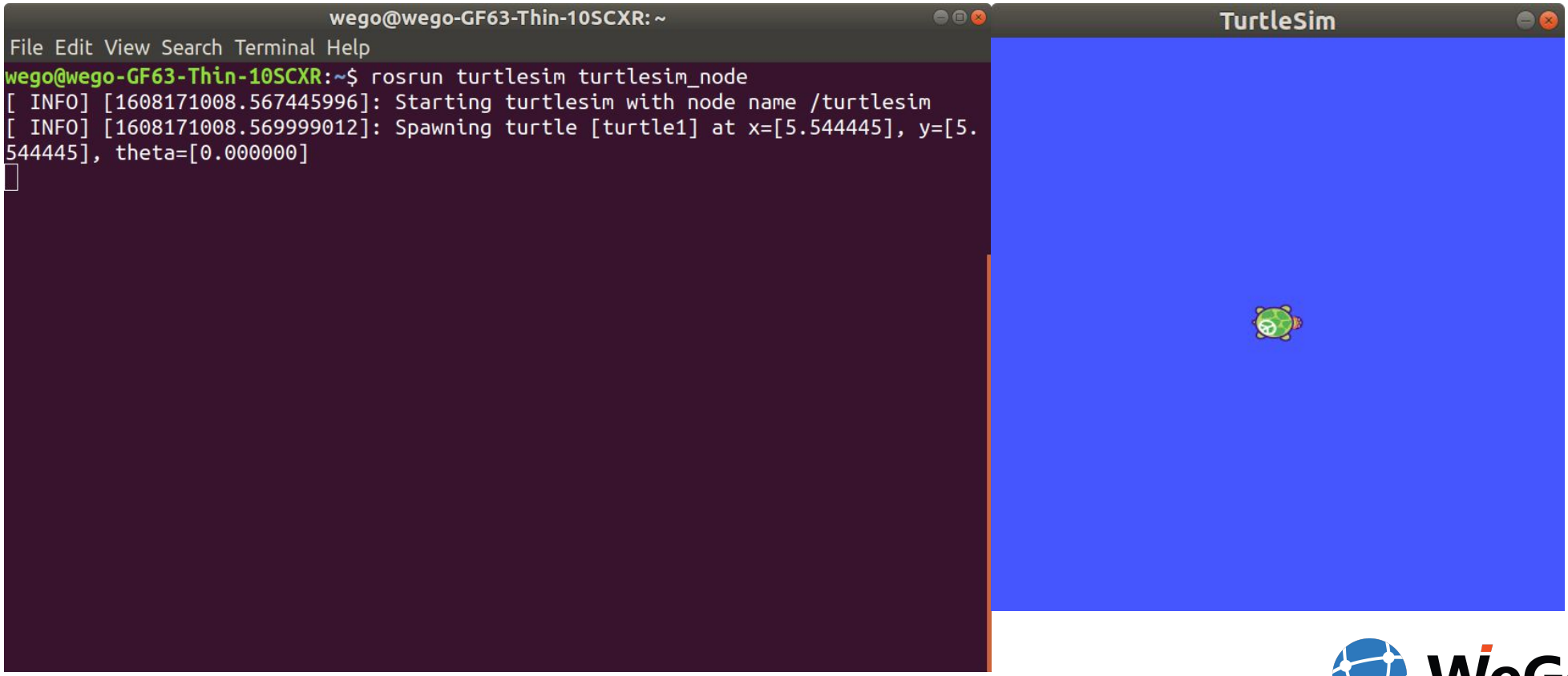
9. 빌드가 진행되면 build와 devel 폴더가 되고, setup.bash 파일을 불러옵니다.
   - $ source devel/setup.bash
10. ROS가 정상적으로 설치되었는지 확인하기 위해, Master를 실행해봅니다
   - $ roscore

```
wego@wego-GF63-Thin-10SCXR:~$ roscore
... logging to /home/wego/.ros/log/95e18916-400c-11eb-bb45-d83bbf196793/roslaunc
h-wego-GF63-Thin-10SCXR-3754.log
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://wego-GF63-Thin-10SCXR:46143/
ros_comm version 1.14.10


SUMMARY
========

PARAMETERS
 * /rosdistro: melodic
 * /rosversion: 1.14.10

NODES

auto-starting new master
process[master]: started with pid [3765]
ROS_MASTER_URI=http://wego-GF63-Thin-10SCXR:11311/

setting /run_id to 95e18916-400c-11eb-bb45-d83bbf196793
process[rosout-1]: started with pid [3791]
started core service [/rosout]
```

WeGo

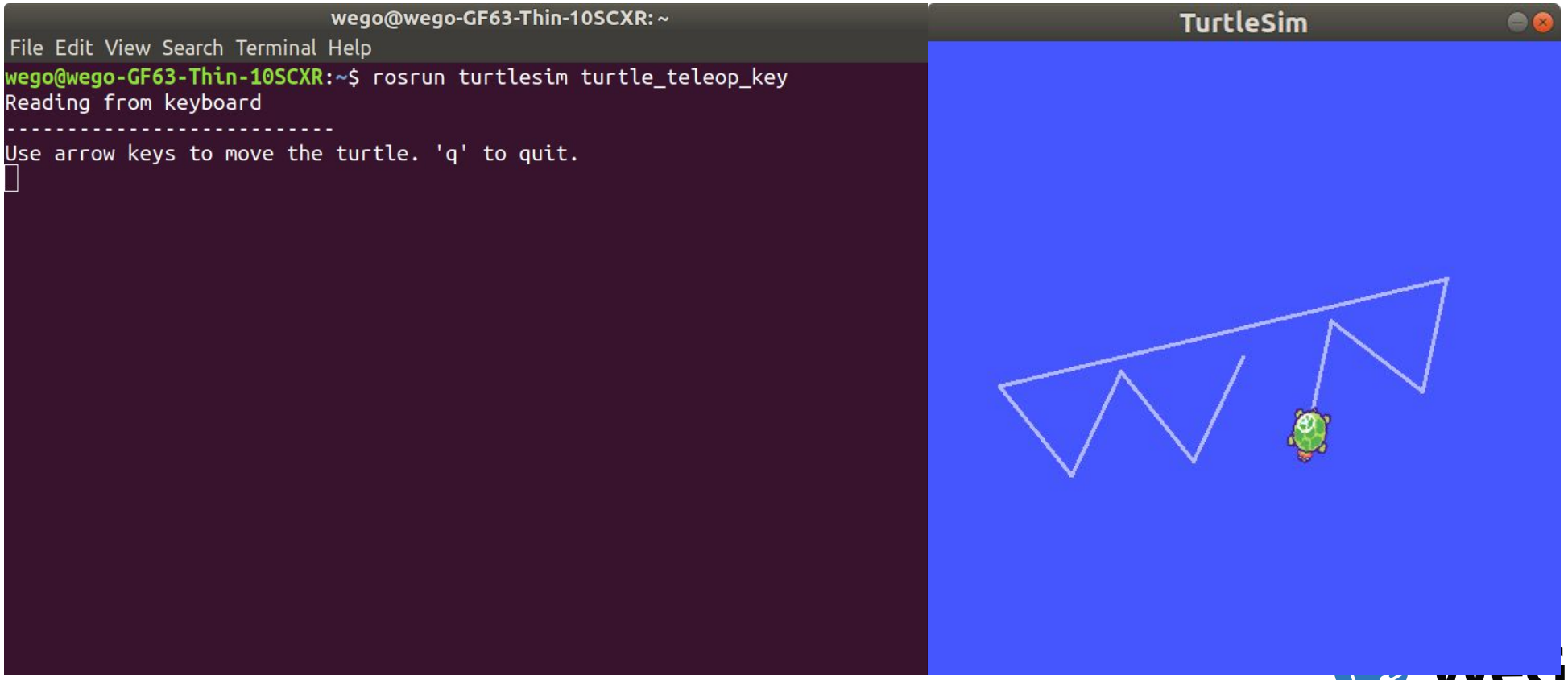
11. Master 터미널을 유지한 채로, 새로운 터미널을 열어서 기본 패키지를 테스트합니다.

- $ rosrun turtlesim turtlesim_node



12. 마스터와 Turtlesim이 실행된 터미널을 유지한 채로, 원격 이동을 위한 새로운 노드를 실행합니다. 실행 후, 키보드 방향키를 눌러서 Turtlesim을 이동합니다.

    - $ rosrun turtlesim turtle_teleop_key

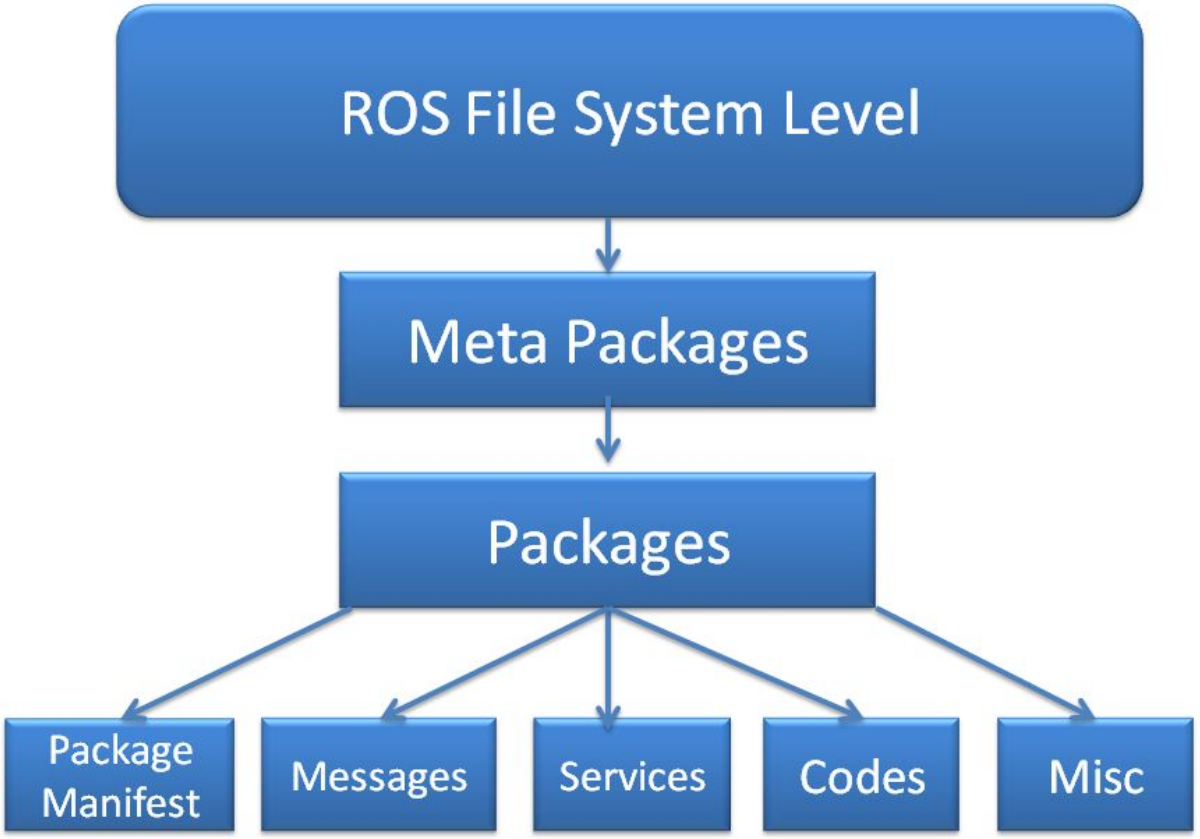


03 ROS 중요 개념

03 ROS 중요 개념

- ROS Filesystem
- Repository - Meta-Package - Package

  - Package Manifest, Message, Service, Source Codes, Misc



WeGo

03 ROS 중요 개념

- Repository : ROS에서 사용하는 다양한 패키지를 저장하는 저장소

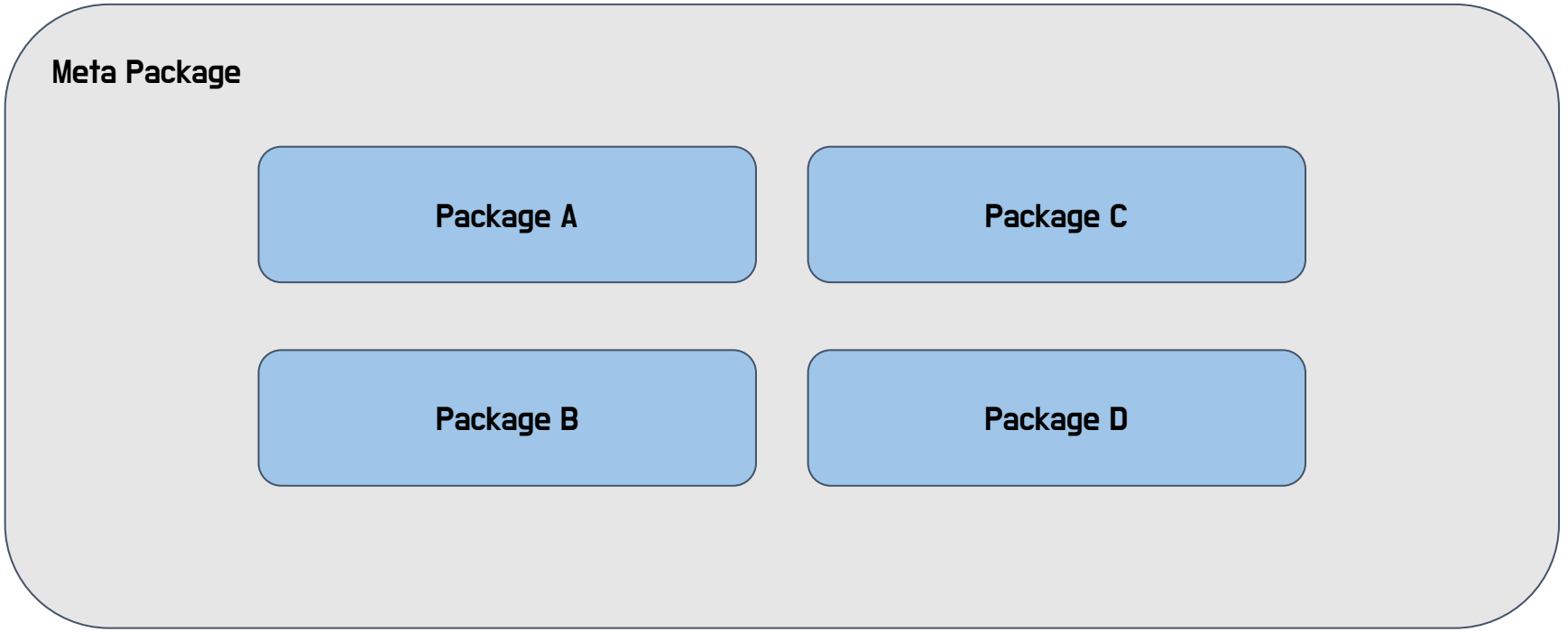
  가장 넓은 개념으로 사용되며, ROS를 설치하는 설치 파일도 포함된 저장소이다.

03 ROS 중요 개념

- Meta Package : 특수화된 패키지로서, 관련된 패키지들을 모아놓은 것

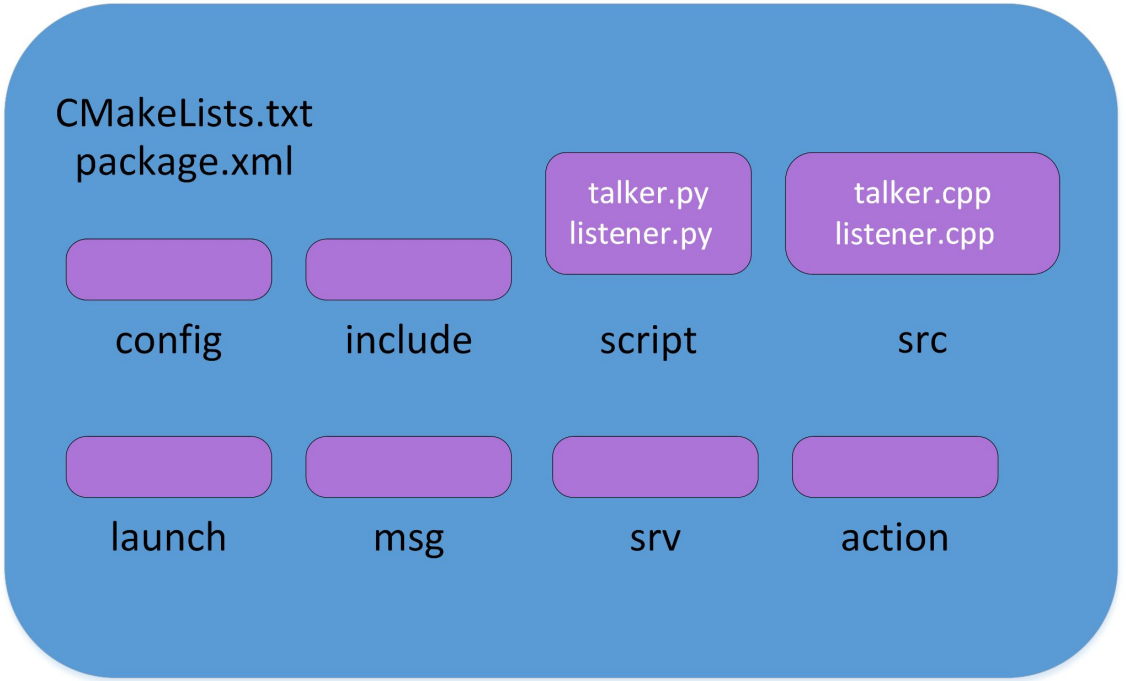
  ex) 자율주행 메타패키지 → 인지, 판단, 제어 패키지 포함

  ex) 튜토리얼 메타패키지 → Topic Pub/Sub, Service Server/Client, Turtlesim



03 ROS 중요 개념

- Package : ROS 소프트웨어를 구성하는 주요 단위. ROS노드. ROS독립 라이브러리. 데이터세트. 구성파일. 3rd Party 소프트웨어 또는 유용한 모듈로 구성되어 있음
- Package Manifest : package.xml 파일을 의미하며. 패키지의 이름. 버전. 설명. 라이선스의 정보를 가지고 있는 xml파일



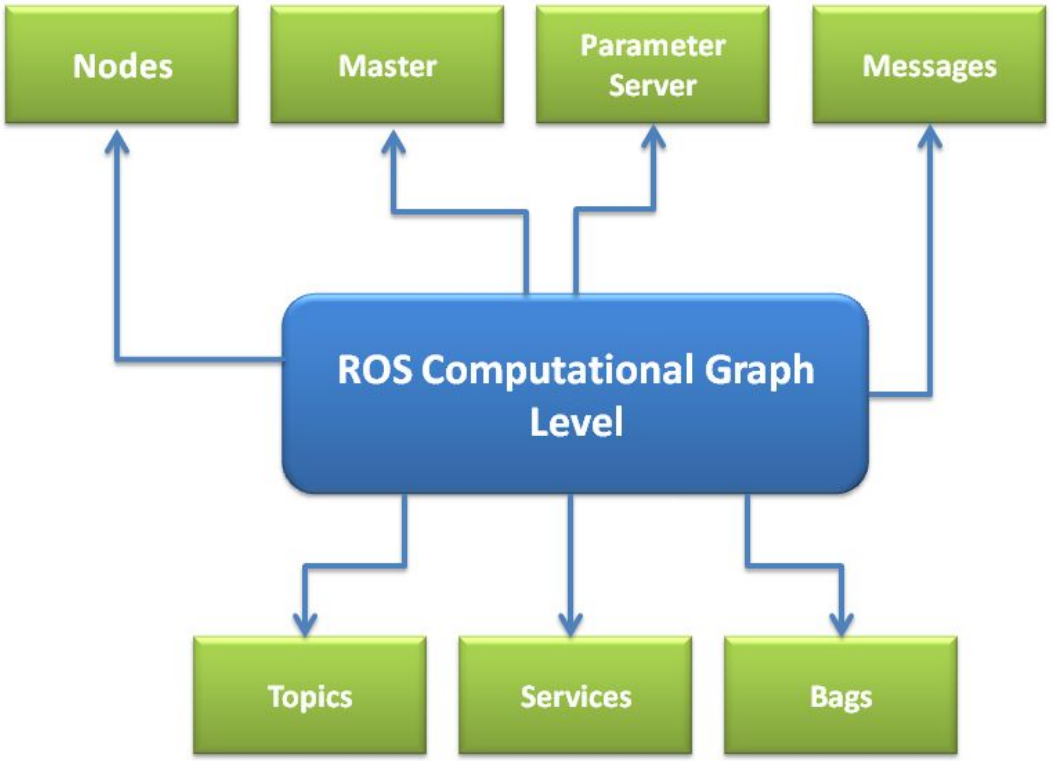
WeGo

03 ROS 중요 개념

- ROS Package의 일반적인 구조
  - include : 패키지에서 사용하는 헤더 파일이 있는 폴더 (.h file)
  - msg : 새로운 메시지를 정의할 때. 메시지 유형이 포함된 폴더 (.msg file)
  - src/package_name : 소스 코드를 포함하는 폴더 (.c .cpp file)
  - srv : 새로운 서비스를 정의할 때. 서비스 유형이 포함된 폴더 (.srv file)
  - scripts : 실행 가능한 Python 스크립트를 포함하는 폴더 (.py .bash .sh file)
  - CMakeLists.txt : CMake 빌드 설정 파일
  - package.xml : 패키지 매니페스트 파일

03 ROS 중요 개념

- ROS Computation Graph Level
- ROS Master는 Parameter Server, Topic 및 Service, Message, Node에 대한 관리를 모두 수행합니다.
- Node는 기본 프로세스, Message는 송수신하는 데이터를 의미

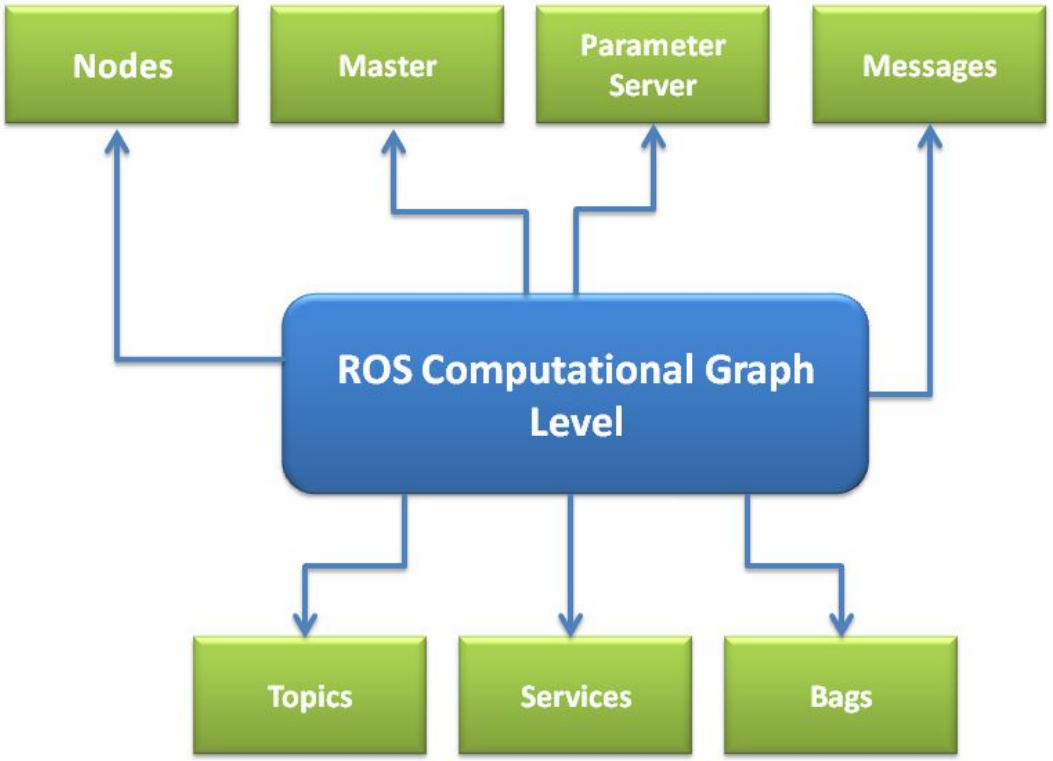


WeGo

03 ROS 중요 개념

- Master : 마스터를 통해, 노드의 이름을 등록하고, 검색하여 정보를 얻을 수 있음.
- Node : ROS의 최소 단위의 프로세스, 실제 하나의 역할만 하도록 권장되어 있음.

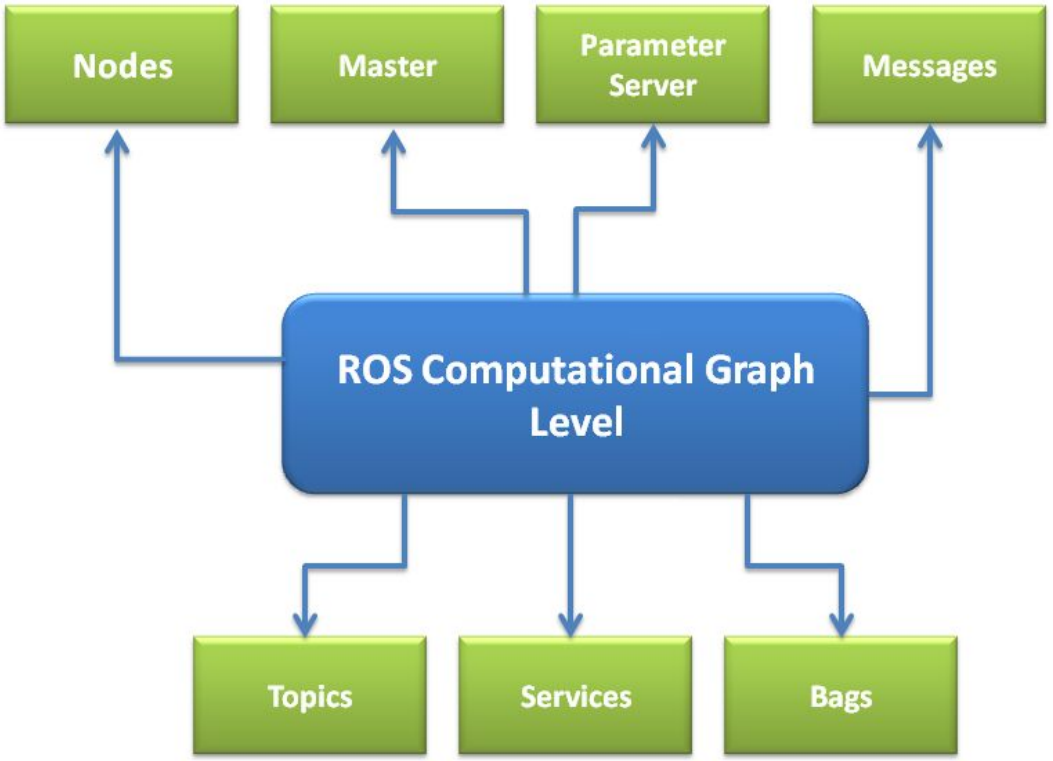
  Node와 Node 사이의 통신은 메시지, 서비스, 파라미터를 통해 통신 가능



WeGo

03 ROS 중요 개념

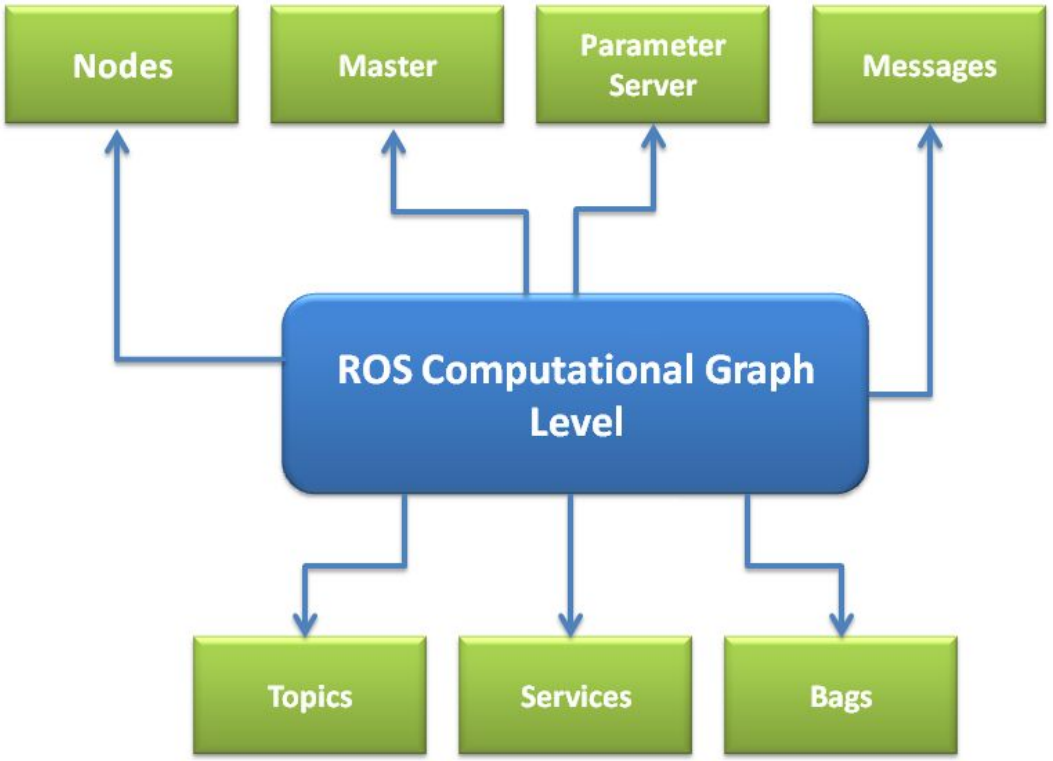
- Parameter Server : Master에 값을 전달하여, 코드 실행 중에 값을 변경 가능
- Bags : ROS에서 데이터를 저장 및 불러올 때 사용하며, Master에서 확인 가능한 Topic들을 모두 저장 가능하며, 이를 다시 재생하는 기능 제공



WeGo

03 ROS 중요 개념

- Messages : 노드 사이의 전달하는 데이터
- Topics : 메시지를 식별하기 이름을 붙여놓은 것. Topic을 보내는 것을 Publish, 받는 것을 Subscribe라고 함
- Services : 입력이 있을 때만 동작하는 형태의 데이터 송수신을 의미



WeGo

03 ROS 중요 개념

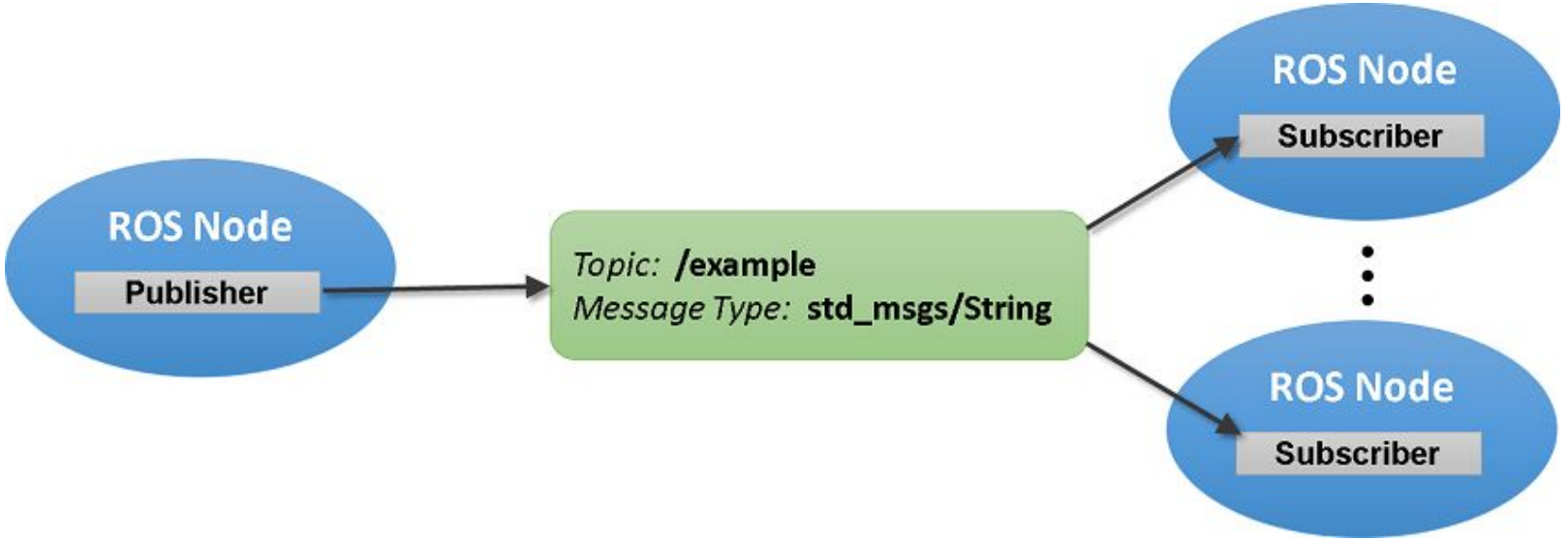
- ROS는 코드를 재사용하기 위해, 각 프로그램을 노드 단위로 세분화 되어 있으며, 각 노드들은 서로 간의 통신을 통해 하나의 응용 프로그램으로 개발됩니다.
- 노드 간의 통신 방법은 사용 용도에 따라 세 가지가 있습니다.
  - Topic : 일방향 통신, 비동기 통신, 지속적으로 통신
  - Service : 양방향 통신, 동기 통신, 일시적으로 통신
  - Action : 양방향 통신, 비동기 통신, 일시적, 지속적으로 모두 통신

03 ROS 중요 개념

- Topic
  - 단방향 통신으로 연속적으로 데이터를 송수신하는 경우에 사용(ex. 센서 데이터)
  - 한 번 접속 시, 지속적으로 데이터를 송수신 가능하며,

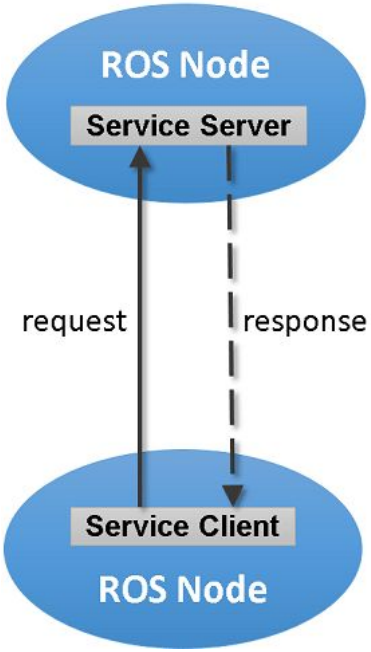
    Publisher는 데이터의 송신 시점을 지정할 수 있습니다.
  - 하나의 Topic에 대해, 여러 개의 Publisher와 Subscriber가 존재할 수 있으나,

    동시에 다 수의 Publisher가 존재하는 것은 권장하지는 않습니다.



03 ROS 중요 개념

- Service
  - 양방향 통신으로 노드 간의 통신 요청과 동시에 응답하여 처리가 필요한 경우 사용
  - 토픽과 달리 한 번 통신 후, 연결이 종료되는 방식으로, 다시 통신이 필요할 경우, 접속부터 새롭게 수행해야합니다.
  - 장기적으로 실행되는 프로세스 또는 예외가 발생할 수 있는 프로세스에서는 사용을 권장하지 않습니다.



Service Name : /example_service

Service Type : roscpp_tutorials/TwoInts

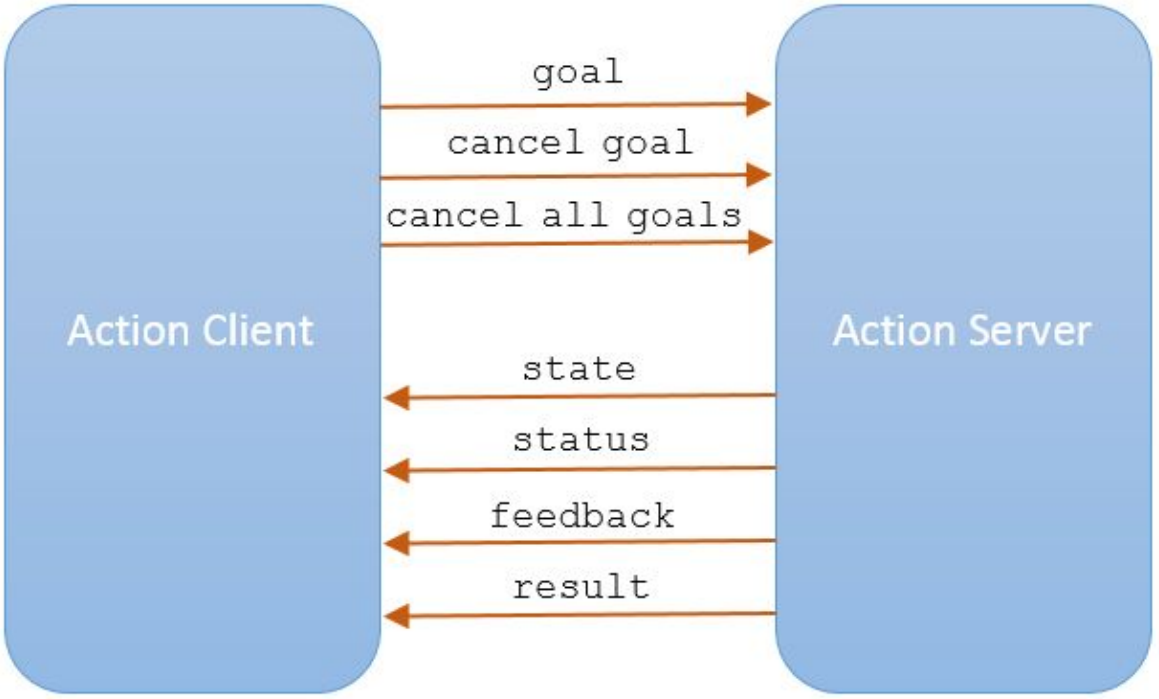
Request Type : roscpp_tutorials/TwoIntsRequest

Response Type : roscpp_tutorials/TwoIntsResponse

WeGo

03 ROS 중요 개념

- Action
  - Service와 유사. 장기적으로 수행되는 프로세스에서 피드백을 제공하는 양방향 통신
  - 최종 응답까지 시간이 오래 걸리는 경우 사용. 메시지 자체는 비동기식으로 전송됩니다.
  - 중간에 취소도 가능하며. 상태. 피드백. 결과를 모두 수신 가능



WeGo

04 ROS 명령어

Command

04 ROS 명령어

- ROS 명령어는 터미널에서 특정 기능이나 프로그램을 실행할 때, 사용하는 인터페이스
- 단축키 기능, 명령 편집 기능 등을 설정할 수 있습니다.
- roscd : 위치한 ros 패키지의 폴더로 이동 ⭐
- rosls : 해당 ros 패키지 내부 폴더의 파일 목록을 확인하는 명령어
- rosed : ros 패키지 내부 파일을 편집하는 명령어 (기본적으로 vim을 사용) ⭐
  - echo "export EDITOR='code'" >> ~/.bashrc 로 기본 실행을 VS code로 변경
- roscp : ros 패키지 내부 파일의 복사가 가능한 명령어
- rospd : 디렉토리 인덱스에 파일을 추가
- rosd : 디렉토리 인덱스 확인

04 ROS 명령어

- roscore : master를 실행하는 명령어 ⭐
- rosrun : 패키지의 노드를 실행하는 명령어 ⭐
- roslaunch : 여러 개의 노드 및 master도 함께 실행하는 명령어 ⭐
- rosclean : ROS 로그 파일을 검사하거나, 제거 및 정리하는 명령어
- rostopic : 등록된 Topic을 다루는 명령어 ⭐
- rosservice : 등록된 Service를 다루는 명령어
- rosnode : 등록된 Node를 다루는 명령어 ⭐
- rosparam : 등록된 Parameter를 다루는 명령어 ⭐
- rosbag : 등록된의 Topic들을 저장 및 재생하기 위한 명령어 ⭐
- rosmsg : 등록된 Message 유형을 다루는 명령어 ⭐
- rossrv : 등록된 서비스 유형을 다루는 명령어

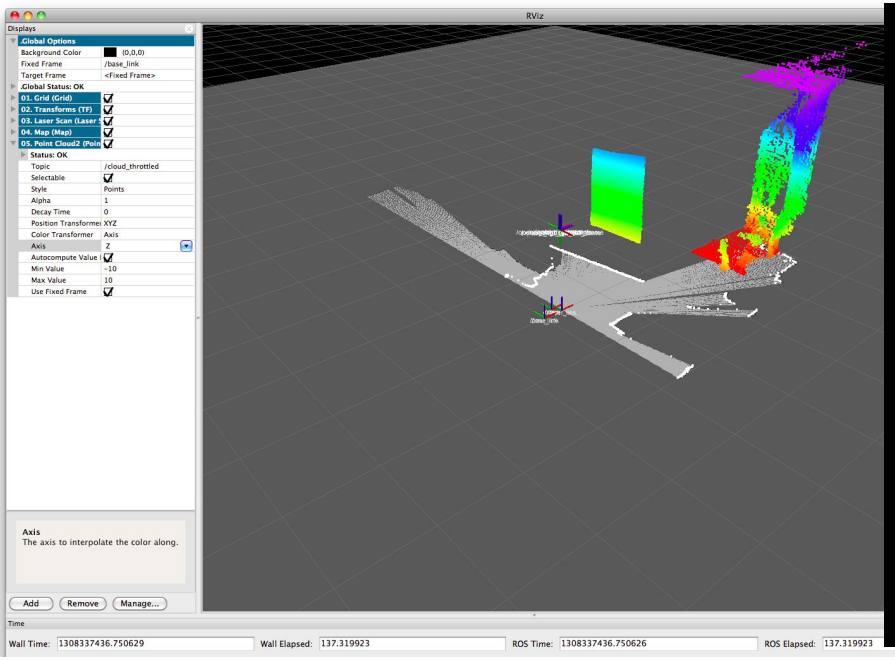
04 ROS 명령어

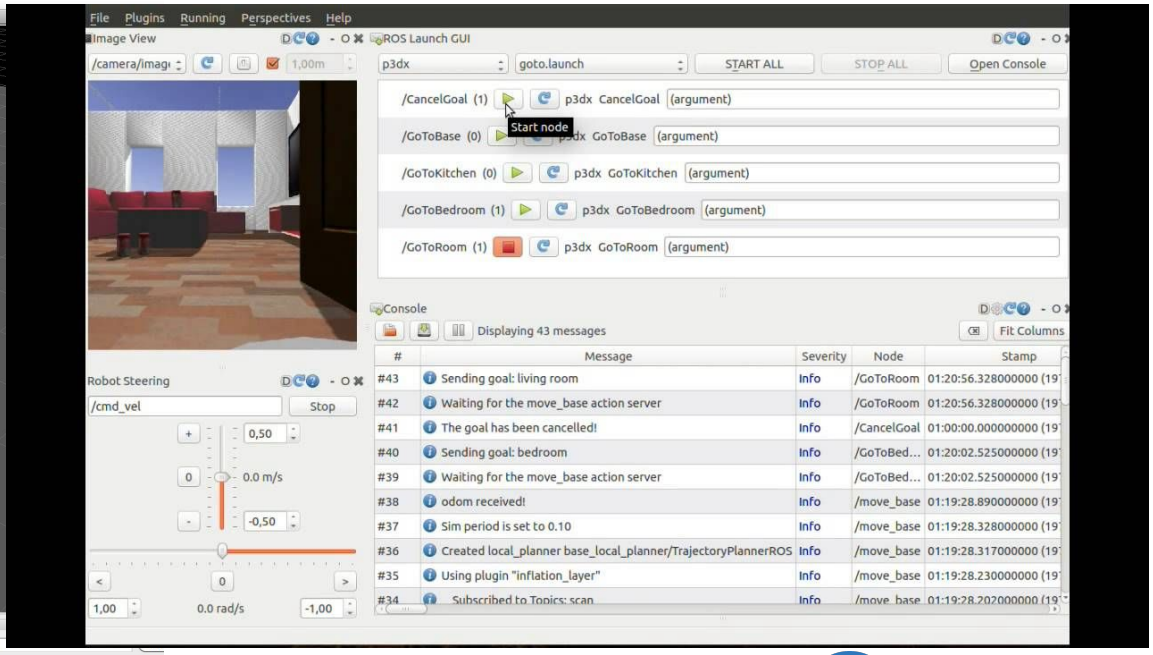
- catkin_create_pkg : 새로운 패키지를 생성하는 명령어 ⭐
- catkin_make : catkin 작업 공간의 C++ 파일을 빌드하는 명령어 ⭐
- rospack : 파일 시스템에서 사용 가능한 패키지의 정보를 검색하기 위한 명령어
- rosdep : 시스템 의존성 파일들을 설치하기 위한 명령어 ⭐
- rviz : 3D 시각화를 위한 rviz를 실행하는 명령어 ⭐
- rqt : 데이터 시각화를 위한 rqt를 실행하는 명령어 ⭐

05 ROS Tools

05 ROS Tools

- ROS 개발 시, 자주 사용하는 유용한 도구
- Rviz : ROS를 위한 3D 시각화 도구이며, 숫자로 확인하는 것을 시각화하여, 확인하기 유용한 도구
- RQT : ROS 정보를 시각화 하기 위한 여러가지 플러그인을 불러와서 확인하는 도구





WeGo

WeGo Robotics

Tel. 031 – 229 – 3553

Fax. 031 – 229 – 3554

제품 문의: go.sales@wego-robotics.com

기술 문의: go.support@wego-robotics.com