



Week 5

1. RQT

1.1. RQT 플러그인의 종류

Topics

- Message Publisher : 메시지 퍼블리셔
- Message Type Browser : 메시지 타입의 데이터 구조 확인
- Topic Monitor : 토픽 목록 확인 및 사용자가 선택한 토픽의 정보 확인

Services

- Service Caller : 실행 중인 서비스 서버에 접속하여 서비스를 요청
- Service Type Browser : 서비스 타입의 데이터 구조를 확인

Actions

- Action Type Browser : Action 타입의 데이터 구조를 확인

Introspection

- Node Graph : 실행 중인 노드들의 관계 및 토픽을 확인 가능한 그래프 뷰
- Process Monitor : 실행 중인 노드들의 CPU 사용률, 메모리 사용률, 스레드 수 등을 확인

Configuration

- Dynamic Reconfigure : 노드들에서 제공하는 파라미터 값 확인 및 변경

Logging

- Console : 노드들에서 발생하는 경고, 에러 등의 메시지를 확인

Miscellaneous Tools

- Python Console : 파이썬 콘솔 화면
- Shell : 셸을 구동

Visualization

- Image View : 카메라의 영상 데이터를 확인
- Plot : 2차원 데이터 plot GUI 플러그인

Robot Tools

- Diagnostics Viewer : 로봇 디바이스의 경고 및 에러 확인
- Robot Steering : 로봇에게 병진 속도와 회전 속도를 토픽으로 퍼블리시하는 GUI 툴
- Controller Manager : 컨트롤러 관리에 필요한 플러그인

1.2. RQT Framework

- ROS환경에서 사용할 수 있는 GUI 개발에 있어서 공통으로 필요한 부분들을 API 형태로 제공
- ROS+QT의 합성어로 QT가 이용됨.

- Qt의 특징인 크로스 플랫폼과 다양한 프로그래밍 언어를 지원함

2. ROS2의 표준 단위 & 좌표 표현

Base Units

These are the base units which are commonly used

| Quantity | Unit |
|----------|----------|
| length | meter |
| mass | kilogram |
| time | second |
| current | ampere |

Derived Units

SI defines seven base units and many derived units. If you are not using SI base units, you should use SI-derived units.

Good documentation can be found on Wikipedia about [SI derived units](#) [3]

Commonly used SI-derived units in ROS are:

| Quantity | Unit |
|-------------|---------|
| angle | radian |
| frequency | hertz |
| force | newton |
| power | watt |
| voltage | volt |
| temperature | celsius |
| magnetism | tesla |

Axis Orientation

In relation to a body the standard is:

- x forward
- y left
- z up

For short-range Cartesian representations of geographic locations, use the [east north up \[5\]](#) (ENU) convention:

- X east
- Y north
- Z up

To avoid precision problems with large float32 values, it is recommended to choose a nearby origin such as your system's starting position.

Suffix Frames

In the case of cameras, there is often a second frame defined with a "_optical" suffix. This uses a slightly different convention:

- z forward
- x right
- y down

For outdoor systems where it is desirable to work under the [north east down \[6\]](#) (NED) convention, define an appropriately transformed secondary frame with the "_ned" suffix:

- X north
- Y east
- Z down

- 모든 좌표계는 오른손 법칙에 따라 표현
- 회전각은 라디안
- 쿼터니언, 회전행렬, 고정축 사용

3. 중간프로젝트

https://github.com/Road-Balance/gcamp_ros2_basic



Project - Spawn Cloned Robot

Use ROS 2 service to bring a robot into the simulation.

- ROS Gazebo Service
- Service Client
- URDF



Results

