



# ORC-PG-ROS2-WEEK4-5

## CH17 ROS 2 도구와 CLI 명령어

### 17.1 ROS 도구

“ROS 1에서 ROS 2에 이르기까지 ROS에서는 로봇 개발에 필요한 다양한 개발 도구를 제공하고 있고, 이를 이용하여 자신의 로봇 프로젝트의 코드들을 쉽게 빌드하고 패키징하여 관리할 수 있다. 또한 데이터들을 기록, 재생, 관리하고 CLI 형태, GUI 형태의 프로그램을 만들고 3차원 시뮬레이터 안에서 자유자재로 디버깅하고 테스트할 수 있다. 필요하다면 이 도구들을 이용하는 것뿐만이 아니 라 이 도구를 개선하거나 신규 도구 작성이 가능하여 로봇 개발의 효율성 향상에 크게 도움이 된다.”

*ROS 도구에는 그 형태에 따라 4가지로 분류할 수 있음*

- CLI 형태의 Command-Line Tools
- GUI 형태의 RQt
- 3차원 시각화 툴인 RViz
- 3차원 시뮬레이터 Gazebo

### CLI (Command Line Interface)

*명령어 기반의 툴, 로봇 액세스 및 거의 모든 ROS 기능을 다룬다.*

*개발환경 및 빌드, 테스트 툴(colcon)*

*데이터를 기록, 재생, 관리하는 툴(ros2bag)*

### GUI 기반 RQt

그래픽 인터페이스 개발을 위한 Qt 기반 프레임워크 제공

노드와 그들 사이의 연결 정보 표시(rqt\_graph)

속도, 전압 등 시간이 지남에 따라 변화하는 데이터를 플로팅(rqt\_plot)

### RViz

3차원 시각화툴

레이저, 카메라 등의 센서 데이터를 시각화

로봇 외형과 계획된 동작을 표현

### Gazebo

### 3차원 시뮬레이터

물리 엔진을 탑재, 로봇, 센서, 환경 모델 등을 지원

타 시뮬레이터 대비 ROS와의 높은 호환성

## CLI

ROS2 CLI 사용법은 기본적으로 터미널 창에서 각 명령어가 하나씩 수행되는 방식.

```
$ros2 [verbs] [sub-verbs] [options] [arguments]
```

*\$ros2 는 ros2cli만의 고유한 entry-point*

### ROS 2 실행 명령어

표 17-1 ROS 2 실행 명령어

ros2cli + [verbs]	[arguments]	기능
ros2 run	⟨package⟩ ⟨executable⟩	특정 패키지의 특정 노드 실행 * executable에 따라 복수 노드도 실행 가능
ros2 launch	⟨package⟩ ⟨launch-file⟩	특정 패키지의 특정 런치 파일 실행

```
$ ros2 run turtlesim turtlesim_node  
$ ros2 launch demo_nodes_cpp talker_listener.launch.py
```

### ROS 2 정보 명령어

ros2cli + [verbs]	[sub-verbs]	가능
ros2 pkg	create	새로운 ROS 2 패키지 생성
	executables	지정 패키지의 실행 파일 목록 출력
	list	사용 가능한 패키지 목록 출력
	prefix	지정 패키지의 저장 위치 출력
	xml	지정 패키지의 패키지 정보 파일(xml) 출력
ros2 node	info	실행 중인 노드 중 지정한 노드의 정보 출력
	list	실행 중인 모든 노드의 목록 출력
ros2 topic	bw	지정 토픽의 대역폭 측정
	delay	지정 토픽의 지연시간 측정
	echo	지정 토픽의 데이터 출력
	find	지정 타입을 사용하는 토픽 이름 출력

	hz	지정 토픽의 주기 측정
	info	지정 토픽의 정보 출력
	list	사용 가능한 토픽 목록 출력
	pub	지정 토픽의 토픽 퍼블리시
	type	지정 토픽의 토픽 타입 출력
ros2 service	call	지정 서비스의 서비스 요청 전달
	find	지정 서비스 타입의 서비스 출력
	list	사용 가능한 서비스 목록 출력
	type	지정 서비스의 타입 출력
ros2 action	info	지정 액션의 정보 출력
	list	사용 가능한 액션 목록 출력
	send_goal	지정 액션의 액션 목표 전송
ros2 interface	list	사용 가능한 모든 인터페이스 목록 출력
	package	특정 패키지에서 사용 가능한 인터페이스 목록 출력
	packages	인터페이스 패키지들의 목록 출력
	proto	지정 패키지의 프로토타입 출력
	show	지정 인터페이스의 데이터 형태 출력
ros2 param	delete	지정 파라미터 삭제
	describe	지정 파라미터 정보 출력
	dump	지정 파라미터 저장
	get	지정 파라미터 읽기
	list	사용 가능한 파라미터 목록 출력
	set	지정 파라미터 쓰기
ros2 bag	info	저장된 rosbag 정보 출력
	play	rosbag 기록
	record	rosbag 재생

## ROS 2 기능 보조 명령어

ros2cli + [verbs]	[sub-verbs](options)	기능
ros2 extensions	(-a)	ros2cli의 extension 목록 출력
	(-v)	
ros2 extension_points	(-a)	ros2cli의 extension point 목록 출력
	(-v)	
ros2 daemon	start	daemon 시작
	status	daemon 상태 보기
	stop	daemon 정지
ros2 multicast	receive	multicast 수신
	send	multicast 전송
ros2 doctor	hello	ROS 설정 및 네트워크, 패키지 버전, rmw 미들웨어 등과 같은 잠재적 문제를 확인하는 도구
	(-r)	
	(-rf)	
	(-iw)	
ros2 wtf	hello	doctor와 동일함
	(-r)	(ros2 doctor의 alias)
	(-rf)	(WTF: Where's The Fire)
	(-iw)	
ros2 lifecycle	get	라이프사이클 정보 출력
	list	지정 노드의 사용 가능한 상태전이 목록 출력
	nodes	라이프사이클을 사용하는 노드 목록 출력
	set	라이프사이클 상태 전환 트리거
ros2 component	list	실행 중인 컨테이너와 컴포넌트 목록 출력
	load	지정 컨테이너 노드의 특정 컴포넌트 실행
	standalone	표준 컨테이너 노드로 특정 컴포넌트 실행
	types	사용 가능한 컴포넌트들의 목록 출력
	unload	지정 컴포넌트의 실행 중지
ros2 security	create_key	보안키 생성
	create_keystore	보안키 저장소 생성
	create_permission	보안 허가 파일 생성
	generate_artifacts	보안 정책 파일을 이용하여 보안키 및 보안 허가 파일 생성
	generate_policy	보안 정책 파일(policy.xml) 생성
	list_keys	보안키 목록 출력

현재까지 개발된 ROS 2 CLI 관련 정보 참고 리포지토리

<https://github.com/ros2/ros2cli>

ros2\_cheats\_sheet/cli\_cheats\_sheet.pdf at master · ubuntu-robotics/ros2\_cheats\_sheet

A collection of ROS2-related cheats sheet. Contribute to ubuntu-robotics/ros2\_cheats\_sheet development by creating an account on GitHub.

[https://github.com/ubuntu-robotics/ros2\\_cheats\\_sheet/blob/master/cli/cli\\_cheats\\_sheet.pdf](https://github.com/ubuntu-robotics/ros2_cheats_sheet/blob/master/cli/cli_cheats_sheet.pdf)

ubuntu-robotics/  
**ros2\_cheats\_sheet**

A collection of ROS2-related cheats sheet

2 Contributors 1 Issue 136 Stars 39 Forks

## CH18 GUI 개발을 위한 RQt

### GUI 기반 RQt

그래픽 인터페이스 개발을 위한 Qt 기반 프레임워크 제공

노드와 그들 사이의 연결 정보 표시(rqt\_graph)

속도, 전압 등 시간이 지남에 따라 변화하는 데이터를 플로팅(rqt\_plot)

RQt는 플러그인 형태로 다양한 도구와 인터페이스를 구현할 수 있는 그래픽 사용자 인터페이스(Graphical User Interface) 프레임워크이며 다양한 목적의 GUI 툴을 모아둔 ROS의 종합 GUI 툴박스

### RQt 플러그인의 종류

Topics	Introspection	Visualization
<ul style="list-style-type: none"><li>▪ Message Publisher</li><li>▪ Message Type Browser</li><li>▪ Topic Monitor</li></ul>	<ul style="list-style-type: none"><li>▪ Node Graph</li><li>▪ Process Monitor</li></ul>	<ul style="list-style-type: none"><li>▪ Image View</li><li>▪ Plot</li></ul>
Services	Configuration	Robot Tools
<ul style="list-style-type: none"><li>▪ Service Caller</li><li>▪ Service Type Browser</li></ul>	<ul style="list-style-type: none"><li>▪ Dynamic Reconfigure</li></ul>	<ul style="list-style-type: none"><li>▪ Diagnostics Viewer</li><li>▪ Robot Steering</li></ul>
Actions	Logging	
<ul style="list-style-type: none"><li>▪ Action Type Browser</li></ul>	<ul style="list-style-type: none"><li>▪ Console</li></ul>	
	Miscellaneous Tools	
	<ul style="list-style-type: none"><li>▪ Python Console</li><li>▪ Shell</li></ul>	

### RQt 프레임워크

ROS 환경에서 사용할 수 있는 GUI 개발에 있어서 공통으로 필요한 부분들을 API 형태로 제공한다. 이를 통해 ROS와 연동되는 GUI 툴을 RQt 플러그인 형태로 개발할 수 있어 개발된 각 플러그인은 RQt에서 통합하여 사용할 수 있다.

## RQt 설치 및 실행

RO2 최소 설치 등 다른 방법으로 설치했다면 추가적으로 설치해야 한다.

특히 robot-monitor / robot-steering / 디버그 심볼 패키지(dbgsym) 등은 기본 설치에 없기 때문에 추가적으로 설치해야 한다.

```
$ sudo apt install ros-foxy-rqt*
ros-foxy-rqt
ros-foxy-rqt-action
ros-foxy-rqt-common-plugins
ros-foxy-rqt-console
ros-foxy-rqt-graph
ros-foxy-rqt-gui
ros-foxy-rqt-gui-cpp
ros-foxy-rqt-gui-cpp-dbgsym
ros-foxy-rqt-gui-py
ros-foxy-rqt-image-view
ros-foxy-rqt-image-view-dbgsym
ros-foxy-rqt-msg
ros-foxy-rqt-plot
ros-foxy-rqt-publisher
ros-foxy-rqt-py-common
ros-foxy-rqt-py-common-dbgsym
ros-foxy-rqt-py-console
ros-foxy-rqt-reconfigure
ros-foxy-rqt-robot-monitor
ros-foxy-rqt-robot-steering
ros-foxy-rqt-service-caller
ros-foxy-rqt-shell
ros-foxy-rqt-srv
ros-foxy-rqt-top
ros-foxy-rqt-topic
```

## RQt 실행 방법 (기본)

```
$ rqt
```

## RQt 실행 방법 2 (ros2 run 명령어 사용)

```
$ ros2 run rqt_msg rqt_msg
```

## RQt 실행 방법 3 (단축 명령어 사용)

```
$ rqt_grapgh
$ rqt_topic
```

## RQt 플러그인의 종류 (ROS 2 16개만 사용 가능)

### 액션 (Actions)

- Action Type Browser : Action 타입의 데이터 구조를 확인

## 구성 (Configuration)

- Dynamic Reconfigure : 노드들에서 제공하는 파라미터 값 확인 및 변경
- Launch : roslaunch의 GUI 버전

## 내성 (Introspection)

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

## 로깅 (Logging)

- Bag : ROS 데이터 로깅
- Console : 노드들에서 발생하는 경고(Warning), 에러(Error) 등의 메시지를 확인
- Logger Level : ROS의 Debug, Info, Warn, Error, Fatal 로거 정보를 선택하여 표시

## 다양한 툴(Miscellaneous Tools)

- Python Console : 파이썬 콘솔 화면
- Shell : 셸(shell)을 구동
- Web : 웹 브라우저를 구동

## 로봇(Robot)

- 사용하는 로봇에 따라 계기판(Dashboard) 등의 플러그인을 이곳에 추가

## 로봇툴(Robot Tools)

- Controller Manager : 컨트롤러 관리에 필요한 플러그인
- Diagnostic Viewer : 로봇 디바이스의 경고 및 에러 확인
- Moveit! Monitor : 로봇 매니퓰레이터 툴인 Moveit! 데이터 확인
- Robot Steering : 로봇에게 병진 속도와 회전 속도를 토픽으로 퍼블리시하는 GUI 툴
- Runtime Monitor : 실시간으로 노드들에서 발생하는 에러 및 경고를 확인

## 서비스(Services)

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

## 토픽 (Topics)

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

## 시각화(Visualization)

- Image View : 카메라의 영상 데이터를 확인
- Navigation Viewer : 로봇 네비게이션의 위치 및 목표지점 확인
- Plot : 2차원 데이터 플롯 GUI 플러그인, 2차원 데이터의 도식화
- Pose View : 현재 TF의 위치 및 모델의 위치 표시
- RViz : 3차원 시각화 툴인 RViz 플러그인
- TF Tree : tf 관계를 트리로 나타내는 그래프 뷰

## RQt 사용 예시

PASS



## CH19 ROS2의 표준 단위

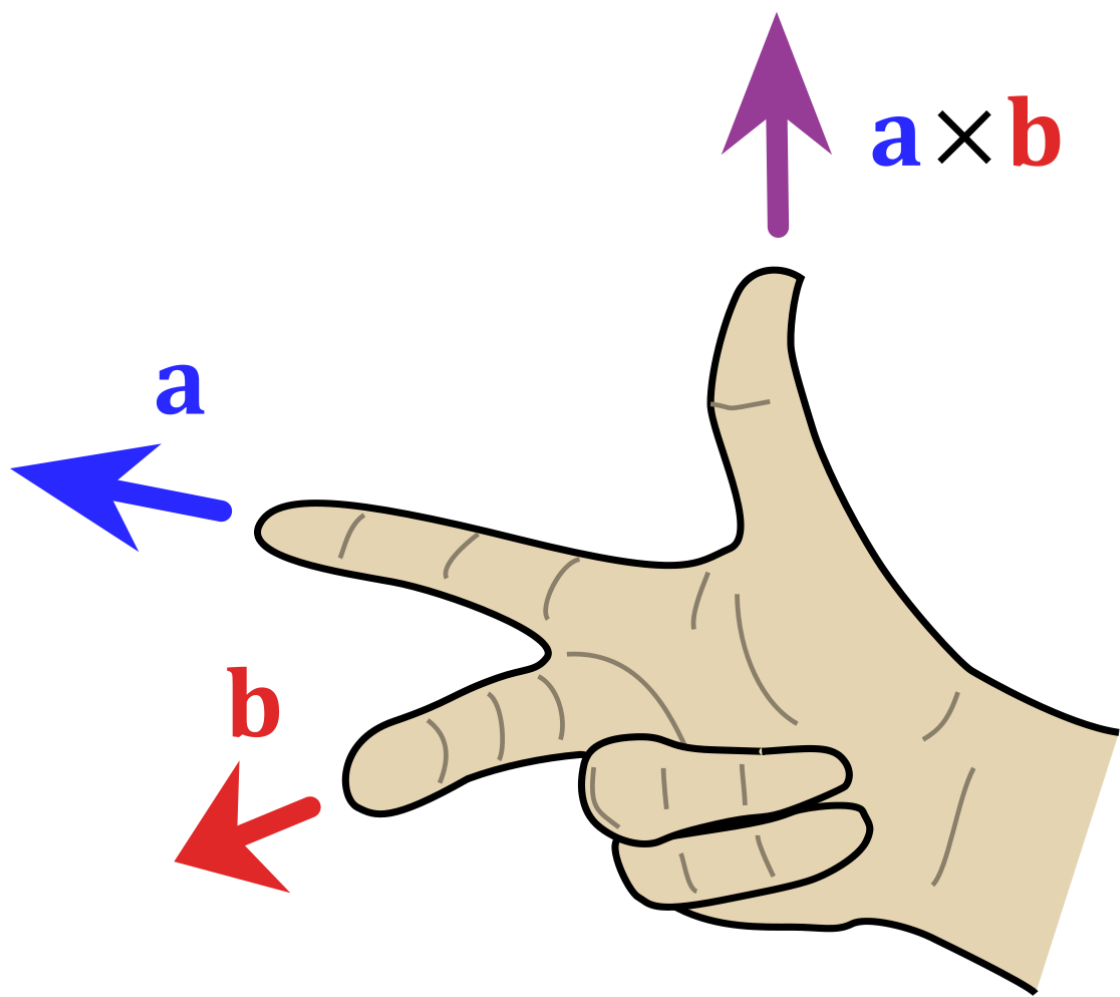
ROS 프로그래밍에 사용하는 표준 단위로 세계적으로 가장 널리 사용되고 있는 국제단위계인 SI 단위(SI unit)와 국제단위계의 7개 기본 단위를 조합해 만들어진 SI 유도 단위(SI derived unit)를 표준 단위로 정하였다.

물리량	단위(SI unit)	물리량	단위(SI derived unit)
Length(길이)	Meter(m)	Angle(평면각)	Radian(rad)
Mass(질량)	Kilogram(kg)	Frequency(주파수)	Hertz(Hz)
Time(시간)	Second(s)	Force(힘)	Newton(N)
Current(전류)	Ampere(A)	Power(일률)	Watt(W)
		Voltage(전압)	Volt(V)
		Temperature(온도)	Celsius(°C)
		Magnetism(자기장)	Tesla(T)

## CH20 ROS 2의 좌표 표현

서로 다른 좌표 표현 방식을 사용할 경우 발생하는 좌표계(Coordinate system) 불일치를 사전에 막기 위한 규칙

ROS 커뮤니티에서는 모든 좌표계의 삼차원 벡터 표기 관습을 이해하기 위한 일반적인 기억법인 오른손 법칙에 따라 표현함



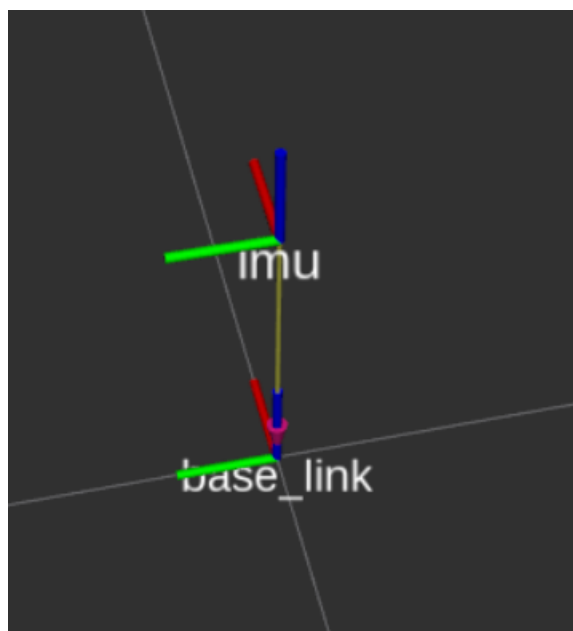
오른손의 손가락을 감는 반시계 방향이 정회선 + 방향



**EX) 회전 각은 라디안(Radian) 단위를 이용하고 있고, 로봇이 제자리에서 시계 12시 방향에서 9시 방향으로 회전하였다면 로봇은 정방향 +1.5708rad만큼 회전하였음**

## 좌표 표현 축 방향(Axis Orientation) 규칙

ROS 커뮤니티에서는 그림 20-2와 같이 축 방향(Axis Orientation)으로 x forward, y left, z up을 사용한다. 시각화 툴 RViz나 3차원 시뮬레이터 Gazebo에서 이러한 기본 3축의 표현을 RGB의 원색으로 표현하는데 순서대로 Red는 x축, Green은 y축, Blue는 z축을 의미한다.



### ENU 좌표

비교적 큰 맵을 다루는 드론, 실외 자율주행 로봇에서 사용하는 좌표

#### 접미사 프레임(Suffix Frames)

x,y,z 및 ENU 좌표에서 벗어나는 경우 사용

- -optical 접미사 : 컴퓨터 비전 분야의 경우 카메라 센서 이미지에 \_optical 접미사를 붙여 구분
- \_ned 접미사 : 실외에서 동작하는 시스템의 경우 NED(North East Down) 좌표계를 사용. 이때 \_ned 접미사를 붙여 구분

## 좌표 표현의 회전 표현(Rotation Representation)규칙

### 쿼터니언(Quaternion)

- 간결한 표현 방식으로 가장 널리 사용됨(x, y, z, w)
- 특이점 없음(No singularities)

#### **회전 매트릭스(Rotation matrix)**

- 특이점 없음(No singularities)

#### **고정축 roll, pitch, yaw(Fixed axis roll, pitch, yaw about X, Y, Z axes respectively)**

- 각 속도에 사용

#### **오일러 각도 yaw, pitch, roll(Euler angles yaw, pitch, and roll about Z, Y, X axes respectively)**

- 전역 좌표계에서 회전이 발생하기 때문에 한 축의 회전이 다른 축의 회전과 겹치는 문제(일명 짐벌락)로 인해 사용을 권장하지 않음