

ROS 실습

Contents

I. ROS 설치 II. Turtlesim 실습 III. ROS CMD IV.거북이 돌리기

Docker 이미지 Pull

- 명령어 : docker pull ros:humble

이 명령어는 다음 위치에서 이미지를 다운로드합니다:

⊗ 공식 Docker Hub 저장소:

https://hub.docker.com/_/ros

- 이 저장소는 Open Robotics 팀이 관리하는 **공식 ROS Docker 이미지**입니다.
- ros:humble은 그 중 Humble Hawksbill 버전에 해당하는 태그입니다.

docker run -it --network=host -d -v /dev:/dev \

- -v /tmp/.X11-unix:/tmp/.X11-unix \
- --name humble -e DISPLAY=\$DISPLAY --restart=always ros:humble /bin/bash

옵션	설명
-it	터미널에서 상호작용 가능하게
network=host	네트워크를 호스트와 공유
-d	백그라운드 실행
-v=/dev:/dev	장치 접근 가능하게 (예: 센서)
-v /tmp/.X11-unix:/tmp/.X11-unix	GUI 출력용 X11 공유
name humble	컨테이너 이름 지정
-e DISPLAY=\${DISPLAY}	GUI 출력을 위해 디스플레이 환경변수 설정
restart=always	컨테이너가 꺼져도 자동 재시작
ros:humble	사용할 이미지 이름
/bin/bash	실행 시 bash 쉘로 시작

명령어 입력:

xhost + 누구나 내 X 서버(GUI 환경)에 접근할 수 있게 허용하라는 뜻입니다.

```
pi@raspberrypi:~ $ xhost +
access control disabled, clients can connect from any host
```

docker ps

```
pi@raspberrypi:~ $ docker ps

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

4c475289a7c8 ros:humble "/ros_entrypoint.sh ..." 3 minutes ago Up 3 minutes humble

pi@raspberrypi:~ $ docker exec -it 4c47 /bin/bash
```

도커 컨테이너에 접속해 봅시다. 👉 docker exec -it 4c47 /bin/bash

컨테이너 ID는 고유하게 식별될 수 있는 한 줄여서 입력해도 됩니다

useradd -m -s /bin/bash ubuntu 👉 ubuntu라는 새 사용자 계정 생성

passwd ubuntu 👉 ubuntu 계정의 비밀번호 설정

usermod -aG sudo ubuntu 👉 ubuntu 계정을 sudo 그룹에 추가

마지막으로 sudo apt-get update -y && sudo apt-get upgrade -y

```
pi@raspberrypi:~ $ docker exec -it 4c47 /bin/bash
root@raspberrypi:/# useradd -m -s /bin/bash ubuntu
root@raspberrypi:/# passwd ubuntu
New password:
Retype new password:
passwd: password updated successfully
root@raspberrypi:/# usermod -aG sudo ubuntu
```

sudo apt-get install vim -y sudo apt-get install ros-humble-desktop-full -y 👉 ROS 2 Humble 의 전체 데스크탑 버전을 설치

👉 ubuntu라는 새 사용자 계정 생성

설치하면 가능한 작업들

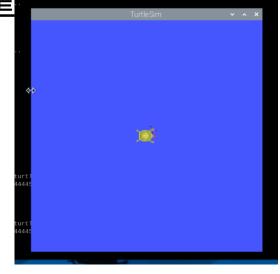
기능	가능 여부
ROS2 노드 실행 및 통신 테스트	☑ 가능
GUI 기반 시각화 (RViz2)	☑ 가능
기본 시뮬레이션 예제 실행 (TurtleSim 등)	☑ 가능
토픽/서비스/액션 구조 테스트	☑ 가능
ROS 기반 센서 데이터 처리	☑ 가능
SLAM 및 Navigation 개발	☑ 가능 (Nav2 포함)

Test ROS2 Environment

ROS 2 환경을 터미널을 열 때마다 자동으로 불러오도록 설정 source /opt/ros/humble/setup.bash echo "source /opt/ros/humble/setup.bash" >> ~/.bashrc source ~/.bashrc

root@raspberrypi:/# echo "source /opt/ros/humble/setup.bash" >> ~/.bashrc root@raspberrypi:/# source ~/.bashrc

ROS 2에서 TurtleSim(거북이 시뮬레이터)로 GUI 출력과 ROS 노드 실행 테스트 ros2 run turtlesim turtlesim_node



Terminator 로 여러개의 Docker 환경을 준비하기|

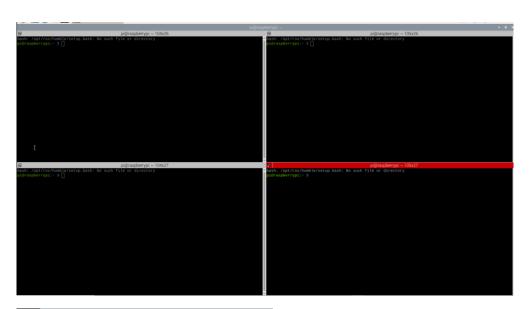
수직 분할: Ctrl + Shift + E

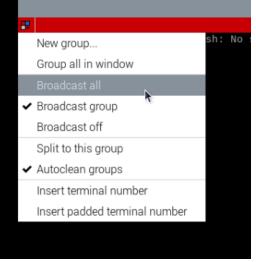
수평 분할: Ctrl + Shift + O

Docker 컨테이너 접속 docker exec -it humble /bin/bash

사소한 팁.

4번을 각각의 터미널에서 입력해도 되지만 Broadcast All 기능으로 한번에 입력도 가능 기능을 On 하면, 같은 TAP 안의 모든 터미널에게 동일한 키보드 입력을 할 수 있다.





turtlesim GUI 실행

ros2 run turtlesim turtlesim_node

- 📌 실행 결과:
- 파란색 배경에 거북이 1마리가 그려진 GUI 창이 뜹니다.
- 이 GUI는 ROS 노드이며, /turtle1/pose, /turtle1/cmd_vel 등 다양한 토픽을 발행/구독합니다.

```
andardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-root
   [1748202696.879282388] [turtlesim]: Starting turtlesim with node name /turtlesim
```

실행 중인 노드 목록 확인

ros2 node list

토픽 목록 확인

ros2 topic list

```
pi@raspberrypi:~ $ docker exec -it humble /bin/bash
root@raspberrypi:/# ros2 node list
/teleop_turtle
/turtlesim
root@raspberrypi:/#
```

```
pi@raspberrypi:~ $ docker exec -it humble /bin/bash
root@raspberrypi:/# ros2 topic list
/parameter_events
/rosout
/turtle1/cmd_vel
/turtle1/color_sensor
/turtle1/pose
root@raspberrypi:/#
```

위치 정보 메시지 확인

ros2 topic echo /turtle1/pose

출력 예시

x: 5.544444561004639

y: 5.544444561004639

theta: 0.0

linear_velocity: 0.0

angular_velocity: 0.0

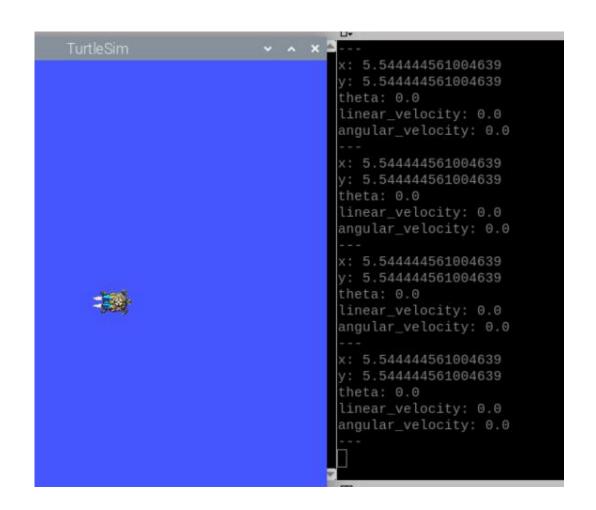
x: 5.544444561004639

y: 5.544444561004639

theta: 0.05

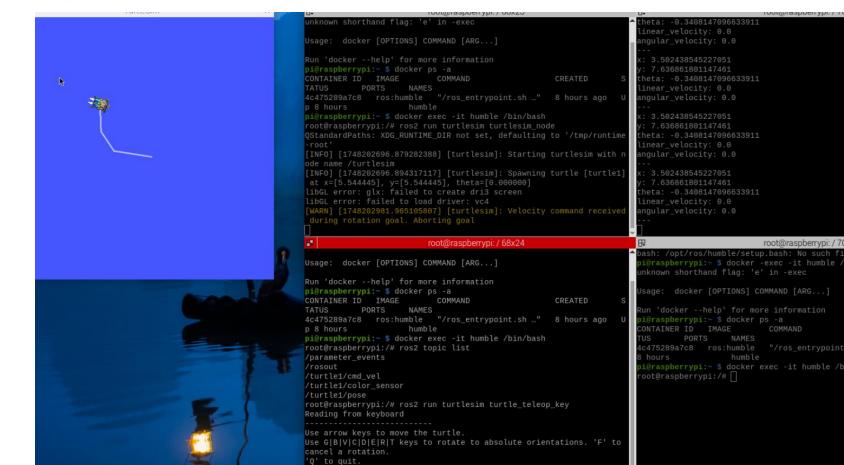
linear_velocity: 2.0

angular_velocity: 0.0



키보드로 거북이 조종

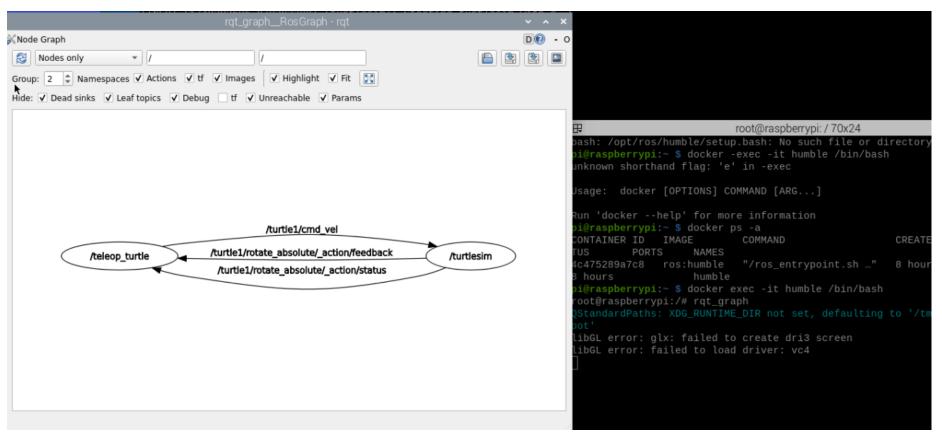
ros2 run turtlesim turtle_teleop_key



RQT 그래프로 ROS 노드 정보 확인하기

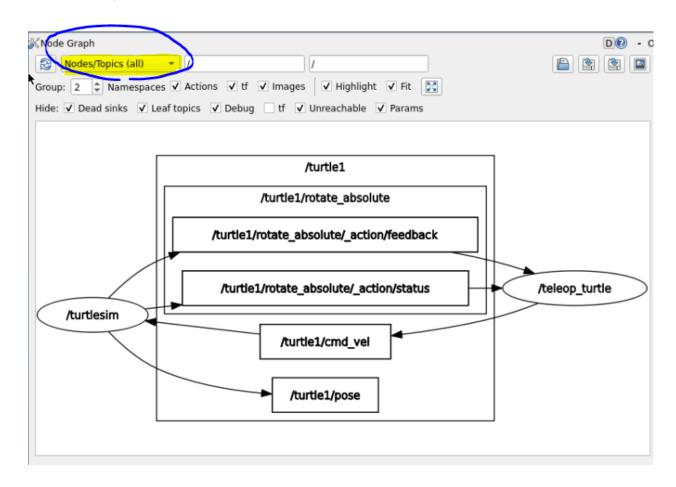
rqt_graph는 ROS (특히 ROS2)에서 노드와 토픽 간의 통신 구조를 시각화해주는 매우 유용한 GUI 툴

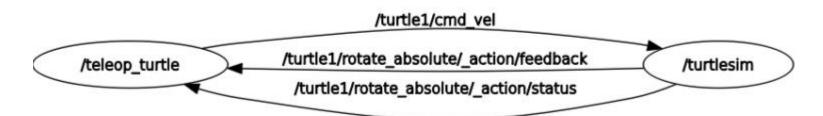
터미널에서 명령어 입력 rqt_qraph



RQT 그래프로 ROS 노드 정보 확인하기

Node/topics 정보를 ALL로 하면, 전체 그래프를 볼 수 있다.





📘 노드 (타원형으로 표시)

- /teleop_turtle
 → 키보드 입력 등을 통해 거북이를
 제어하는 퍼블리셔 노드
 - → 사용자가 조작하는 입력 장치 역할
- /turtlesim
 - → 거북이를 화면에 보여주고 제어 명 령에 반응하는 **시뮬레이션 노드**
 - → 실제 로봇처럼 동작

■ 토픽 (화살표로 표시)

- /turtle1/cmd_vel
 - o 메시지 타입: geometry_msgs/msg/Twist
 - o 속도 제어 명령을 전달하는 토픽
 - /teleop_turtle → 퍼블리시
 /turtlesim → 서브스크라이브
- /turtle1/rotate_absolute/_action/feedback
 - 액션 피드백용 토픽
 - 거북이가 목표 각도로 회전하는 도중의 **중간 상태**를 퍼블리시
 - /turtlesim → 퍼블리셔 /teleop_turtle → 서브스크라이버
- /turtle1/rotate_absolute/_action/status
 - 액션의 상태 정보를 포함하는 토픽 (예: 활성화됨, 완료됨 등)
 - /turtlesim → 퍼블리셔
 /teleop_turtle → 서브스크라이버

- ☑ ROS 2 파일 시스템 구성 요소 (File System)
- 패키지(Package): ROS 2 소프트웨어의 기본 단위. 노드, 메시지, 설정 파일 등을 포함.
- 매니페스트(package.xml): 패키지의 메타데이터(이름, 버전, 의존성 등)를 정의.

package.xml 파일은 ROS2에서 매우 중요한 역할을 합니다. 패키지에 대한 설명과 관리 정보를 제공하여, ROS2가 패키지 간 의존성을 올바르게 처리하고, 적절하게 빌드하고 실행할 수 있도록 도와줍니다.

☑ ROS 2 기본 용어

용어	설명
DDS	ROS 2의 통신 기반: 고성능, 실시간 통신 지원
Node	실행 가능한 최소 단위 프로세스
Message	int, float 등 기본 자료형 데이터
Topic	노드 간 비동기 메시지 통신 채널
Publisher / Subscriber	토픽 발행 / 구독
Service	요청-응답 방식의 동기 통신
Client / Server	서비스를 요청 / 응답하는 노드

◆ 패키지 관리 명령어 (ros2 pkg)

명령어	설명
create	패키지 생성
list	패키지 목록 보기
executables	실행 가능한 노드 목록
prefix	패키지 경로 확인
xml	package.xml 내용 보기

```
root@raspberrypi:/# ros2 pkg -h
usage: ros2 pkg [-h] Call `ros2 pkg <command> -h` for more detailed usage. ...
Various package related sub-commands
options:
                       show this help message and exit
 -h, --help
Commands:
              Create a new ROS 2 package
  create
 executables Output a list of package specific executables
              Output a list of available packages
  prefix
              Output the prefix path of a package
              Output the XML of the package manifest or a specific tag
  ×ml
 Call `ros2 pkg <command> -h` for more detailed usage.
root@raspberrypi:/#
```

◆ 패키지 관리 명령어 (ros2 pkg)

패키지XML 확인: ros2 pkg xml <패키지명>

EX:

ros2 pkg xml turtlesim

```
oot@raspberrypi:/# ros2 pkg xml turtlesim
package format="3">
 <name>turtlesim</name>
 <version>1.4.2
 <description>
   turtlesim is a tool made for teaching ROS and ROS packages.
 </description>
 <maintainer email="audrow@openrobotics.org">Audrow Nash</maintainer>
 <maintainer email="michael.jeronimo@openrobotics.org">Michael Jeronimo</maintainer>
 cense>BSD</license>
 <url type="website">http://www.ros.org/wiki/turtlesim</url>
 <url type="bugtracker">https://github.com/ros/ros_tutorials/issues</url>
 <url type="repository">https://github.com/ros/ros_tutorials</url>
 <author email="dthomas@osrfoundation.org">Dirk Thomas</author>
 <author>Josh Faust</author>
 <author email="mabel@openrobotics.org">Mabel Zhang</author>
 <author email="sloretz@openrobotics.org">Shane Loretz</author>
 <build_depend>qt5-qmake</build_depend>
 <build_depend>qtbase5-dev</build_depend>
 <buildtool_depend>ament_cmake</buildtool_depend>
 <buildtool depend>rosidl default generators</buildtool depend>
 <exec depend>libqt5-core</exec depend>
 <exec_depend>libqt5-gui</exec_depend>
 <exec_depend>rosidl_default_runtime</exec_depend>
 <depend>ament index cpp</depend>
 <depend>geometry msgs</depend>
 <depend>rclcpp</depend>
 <depend>rclcpp_action</depend>
 <depend>std_msgs</depend>
 <depend>std_srvs</depend>
 <member_of_group>rosidl_interface_packages</member_of_group>
 <export>
   <build_type>ament_cmake
 </export>
</package>
root@raspberrypi:/#
```

◆ 패키지 관리 명령어 (ros2 pkg)

설치된 경로 확인: ros2 pkg prefix <패키지명>

ros2 pkg prefix turtlesim

직접 XML 출력해보기 cat /opt/ros/humble/share/turtlesim/package.xml

```
root@raspberrypi:/# ros2 pkg prefix turtlesim
/opt/ros/humble
root@raspberrypi:/# cat /opt/ros/humble/share/turtlesim/package.xml
<?xml version="1.0"?>
<?xml-model href="http://download.ros.org/schema/package_format3.xsd" schematyp</p>
<package format="3">
 <name>turtlesim</name>
 <version>1.4.2
 <description>
   turtlesim is a tool made for teaching ROS and ROS packages.
 </description>
 <maintainer email="audrow@openrobotics.org">Audrow Nash</maintainer>
 <maintainer email="michael.jeronimo@openrobotics.org">Michael Jeronimo</maint</pre>
```

◆ 패키지 관리 명령어 (ros2 pkg)

```
<package format="3">
    <name>turtlesim</name>
    <version>1.2.6</version>
    <description>A simple simulator for teaching ROS2</description>
    <maintainer email="dev@example.com">ROS Developer</maintainer>
    license>Apache-2.0</license>
    <depend>rclcpp</depend>
    <depend>std_msgs</depend>
</package></package>
```

- ◆ 노드 실행 (ros2 run)
- ros2 run <패키지명> <노드명>

◆ 노드 정보 조회 (ros2 node)

명령어	설명
list	활성화된 노드 목록
info 특정 노드의 토픽, 서비스 정보 보기	

◆ 토픽 관련 명령어 (ros2 topic)

명령어	설명
list	활성 토픽 목록
info	토픽의 타입, pub/sub 정보
echo	메시지 실시간 출력
pub	명령줄에서 토픽 메시지 발행
hz/bw/delay	메시지 주기, 대역폭, 지연 측정
type/find	토픽 타입 확인 및 검색

```
oot@raspberrypi:/# ros2 topic -h
usage: ros2 topic [-h] [--include-hidden-topics] Call `ros2 topic <command> -h` for more detailed usage. ...
Various topic related sub-commands
  -h, --help
                       show this help message and exit
 --include-hidden-topics
        Display bandwidth used by topic
 delay Display delay of topic from timestamp in header
 echo Output messages from a topic
 find Output a list of available topics of a given type
        Print the average receiving rate to screen
 info Print information about a topic
 list Output a list of available topics
 pub Publish a message to a topic
 type Print a topic's type
 Call `ros2 topic <command> -h` for more detailed usage.
 oot@raspberrypi:/#
```

인터페이스 관련 명령어 (ros2 interface)

명령어	설명
list	사용 가능한 모든 인터페이스 목록
show	특정 메시지/서비스 구조 보기
package, packages, proto	패키지별 인터페이스 보기

```
root@raspberrypi:/# ros2 interface -h
usage: ros2 interface [-h] Call `ros2 interface <command> -h` for more detailed usage. ...
Show information about ROS interfaces
options:
 -h, --help show this help message and exit
Commands:
           List all interface types available
 list
           Output a list of available interface types within one package
 package
          Output a list of packages that provide interfaces
 packages
 proto
           Output an interface prototype
           Output the interface definition
 show
 Call `ros2 interface <command> -h` for more detailed usage.
root@raspberrypi:/#
```

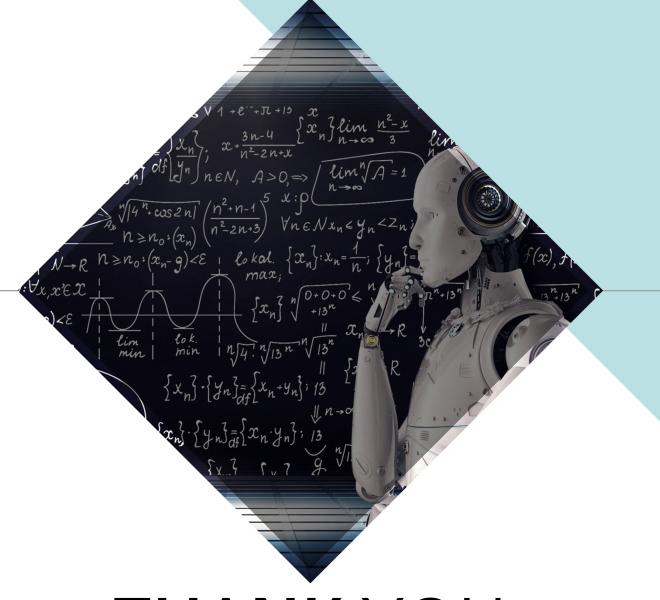
튜토리얼

https://docs.ros.org/en/humble/Tutorials/Beginner-CLI-Tools/Launching-Multiple-Nodes/Launching-Multiple-Nodes.html

이 예제를 실행해 보자!

- 1. 예제를 실행해보기
- 2. 실제 파이썬 파일 경로를 찾아보기
- 3. 원돌기 메시지 분석 해보기

QnA



THANK YOU