

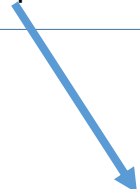
17장 ROS2 도구와 CLI 명령어

목차

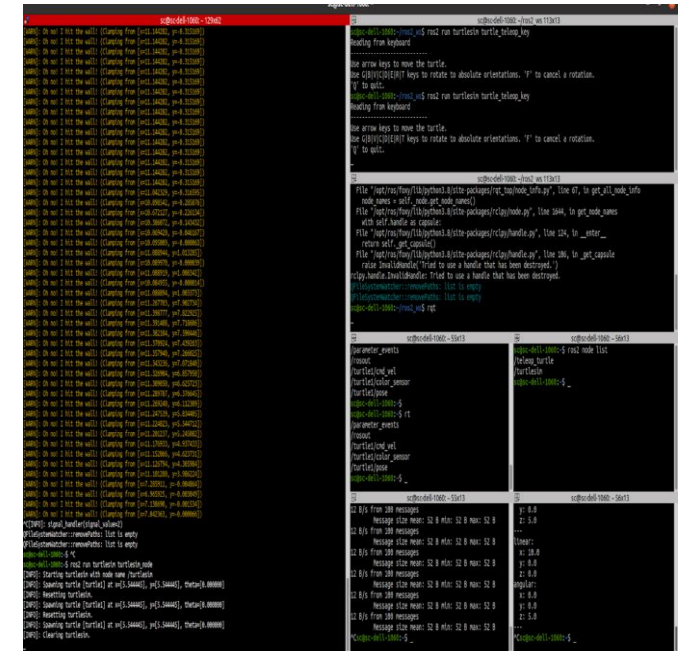
1. CLI 기반 command-line tools
2. GUI 기반 RQT
3. RVIZ

CLI 기반 command-line tools

- 명령어 기반의 툴로 로봇 액세스 및 거의 모든 ROS 기능을 다룬다
- 개발환경 및 빌드, 테스트 툴(colcon)
- 데이터를 기록, 재생, 관리하는 툴(rosbag)
- 패키지/노드/토픽 확인 및 관리 등 20여가지
- CLI : Command Line Interface의 약자로, 터미널 창에 입력하여 사용하는 명령어를 말함.
- 사용법
 - `$ros2 [verbs] [sub-verbs] [options] [arguments]`
 - `ros2` : ros2cli만의 고유 entry-point로 다른 명령어들과 구분
 - `[verbs]` : `run` , `launch` , `node` , `topic` .. 특정 툴을 선택
 - `[sub-verbs]` : 기능을 지정 (`info`, `list`..)
 - Tab 키를 눌러 auto-completion 사용 가능



```
sc@sc-dell-1060:~$ ros2 run turtlesim
draw_square      mimic      turtlesim_node      turtle_teleop_key
```



CLI 기반 command-line tools

ros2 pkg

create	새로운 ROS2 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

```
sc@sc-dell-1060: ~ 120x10
angles
bev_image_cpp_pkg
bev_image_pkg
builtin_interfaces
class_loader
common_interfaces
composition
composition_interfaces
console_bridge_vendor
custom_msg
cv_bridge
```

```
sc@sc-dell-1060: ~ 120x11
sc@sc-dell-1060:~$ ros2 pkg executables bev_image_cpp_pkg
bev_image_cpp_pkg bev_image_publish_node
sc@sc-dell-1060:~$ ros2 pkg executables turtlesim
turtlesim draw_square
turtlesim mimic
turtlesim turtle_teleop_key
turtlesim turtlesim_node
sc@sc-dell-1060:~$ _
```

```
sc@sc-dell-1060: ~ 59x24
sc@sc-dell-1060:~$ ros2 pkg xml bev_image_cpp_pkg
<package format="3">
  <name>bev_image_cpp_pkg</name>
  <version>0.0.0</version>
  <description>TODO: Package description</description>
  <maintainer email="sc@todo.todo">sc</maintainer>
  <license>TODO: License declaration</license>

  <buildtool_depend>ament_cmake</buildtool_depend>

  <depend>rclcpp</depend>
  <depend>std_msgs</depend>
  <depend>sensor_msgs</depend>
  <depend>pcl_conversions</depend>
  <depend>cv_bridge</depend>

  <test_depend>ament_lint_auto</test_depend>
  <test_depend>ament_lint_common</test_depend>

  <export>
    <build_type>ament_cmake</build_type>
  </export>
</package>
sc@sc-dell-1060:~$ _
```

```
sc@sc-dell-1060: ~ 59x24
sc@sc-dell-1060:~$ ros2 pkg prefix bev_image_cpp_pkg
/home/sc/ros2_ws/install/bev_image_cpp_pkg
sc@sc-dell-1060:~$ ros2 pkg prefix turtlesim
/opt/ros/foxy
sc@sc-dell-1060:~$ _
```

CLI 기반 command-line tools

ros2 node

info
list

실행 중인 노드 중 지정한 노드의 정보 출력
실행 중인 모든 노드의 목록 출력

```
sc@sc-dell-1060:~$ ros2 node list
/rqt_gui_py_node_61114
/teleop_turtle
/turtlesim
sc@sc-dell-1060:~$ _
```

```
sc@sc-dell-1060: ~ 118x45
sc@sc-dell-1060:~$ ros2 node info /turtlesim
/turtlesim
Subscribers:
  /parameter_events: rcl_interfaces/msg/ParameterEvent
  /turtle1/cmd_vel: geometry_msgs/msg/Twist
Publishers:
  /parameter_events: rcl_interfaces/msg/ParameterEvent
  /rosout: rcl_interfaces/msg/Log
  /turtle1/color_sensor: turtlesim/msg/Color
  /turtle1/pose: turtlesim/msg/Pose
Service Servers:
  /clear: std_srvs/srv/Empty
  /kill: turtlesim/srv/Kill
  /reset: std_srvs/srv/Empty
  /spawn: turtlesim/srv/Spawn
  /turtle1/set_pen: turtlesim/srv/SetPen
  /turtle1/teleport_absolute: turtlesim/srv/TeleportAbsolute
  /turtle1/teleport_relative: turtlesim/srv/TeleportRelative
  /turtlesim/describe_parameters: rcl_interfaces/srv/DescribeParameters
  /turtlesim/get_parameter_types: rcl_interfaces/srv/GetParameterTypes
  /turtlesim/get_parameters: rcl_interfaces/srv/GetParameters
  /turtlesim/list_parameters: rcl_interfaces/srv/ListParameters
  /turtlesim/set_parameters: rcl_interfaces/srv/SetParameters
  /turtlesim/set_parameters_atomically: rcl_interfaces/srv/SetParametersAtomically
Service Clients:

Action Servers:
  /turtle1/rotate_absolute: turtlesim/action/RotateAbsolute
Action Clients:
```

CLI 기반 command-line tools

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 발행
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

```
sc@sc-dell-1060:~$ ros2 topic list
/parameter_events
/rosout
/turtle1/cmd_vel
/turtle1/color_sensor
/turtle1/pose
sc@sc-dell-1060:~$
```

```
sc@sc-dell-1060:~$ ros2 topic info /turtle1/pose
Type: turtlesim/msg/Pose
Publisher count: 1
Subscription count: 2
sc@sc-dell-1060:~$ ros2 topic info /turtle1/cmd_vel
Type: geometry_msgs/msg/Twist
Publisher count: 1
Subscription count: 1
sc@sc-dell-1060:~$ ros2 topic info /turtle1/color_sensor
Type: turtlesim/msg/Color
Publisher count: 1
Subscription count: 0
sc@sc-dell-1060:~$ ros2 topic info /turtle1/cmd_vel
Type: geometry_msgs/msg/Twist
Publisher count: 1
Subscription count: 1
sc@sc-dell-1060:~$ ros2 topic info /turtle1/cmd_vel
Type: geometry_msgs/msg/Twist
Publisher count: 1
Subscription count: 1
sc@sc-dell-1060:~$ ros2 topic info /turtle1/pose
Type: turtlesim/msg/Pose
Publisher count: 1
Subscription count: 2
sc@sc-dell-1060:~$
```

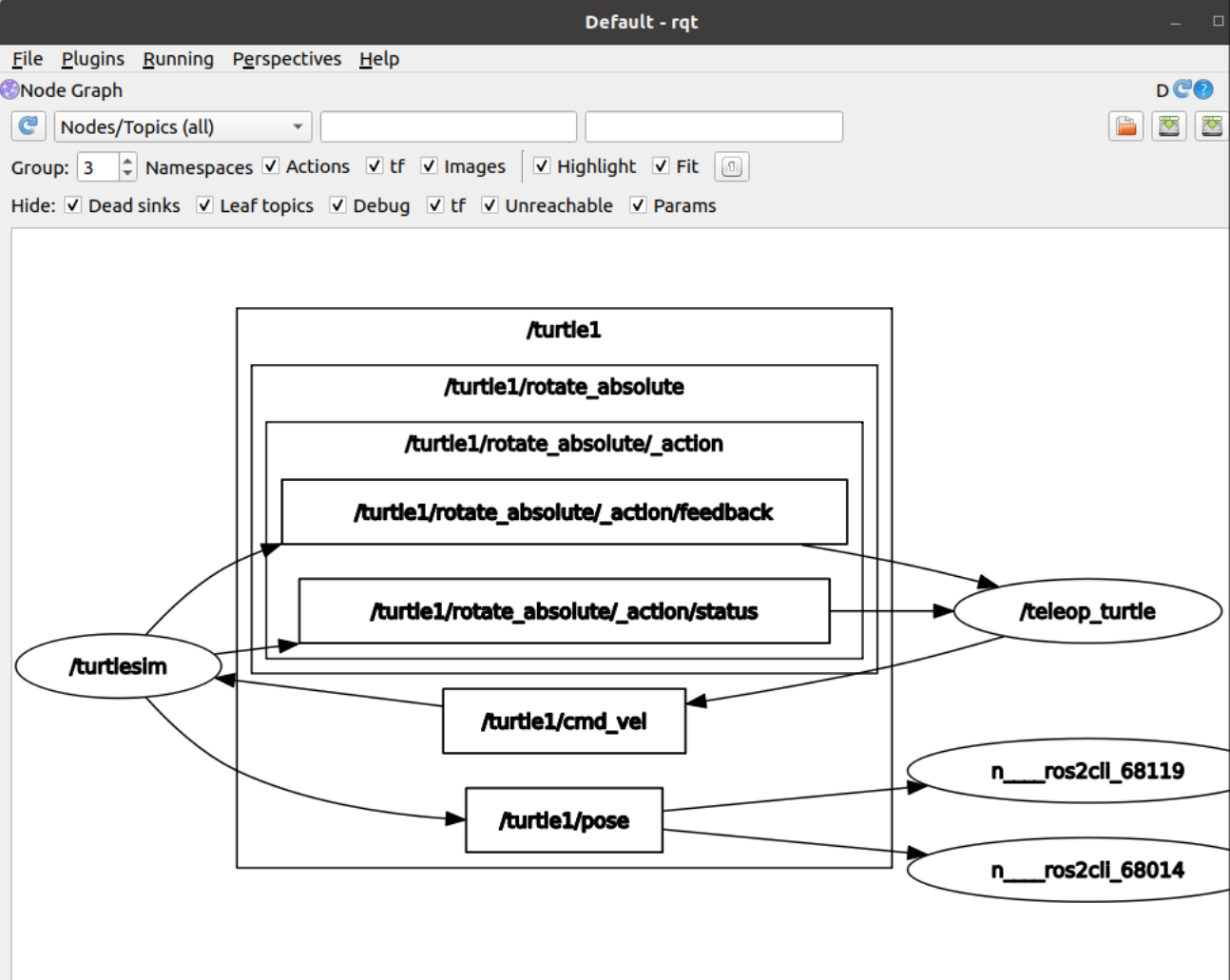
sc@sc-dell-1060: ~ 45x29

```
theta: 1.726447582244873  
linear_velocity: 0.0  
angular_velocity: 0.0  
----  
x: 4.758136749267578  
y: 3.8371784687042236  
theta: 1.726447582244873  
linear_velocity: 0.0  
angular_velocity: 0.0  
----  
x: 4.758136749267578  
y: 3.8371784687042236  
theta: 1.726447582244873  
linear_velocity: 0.0  
angular_velocity: 0.0  
----  
x: 4.758136749267578  
y: 3.8371784687042236  
theta: 1.726447582244873  
linear_velocity: 0.0  
angular_velocity: 0.0  
----  
x: 4.758136749267578  
y: 3.8371784687042236  
theta: 1.726447582244873  
linear_velocity: 0.0  
angular_velocity: 0.0  
----
```

sc@sc-dell-1060: ~ 71x29

[illegible]

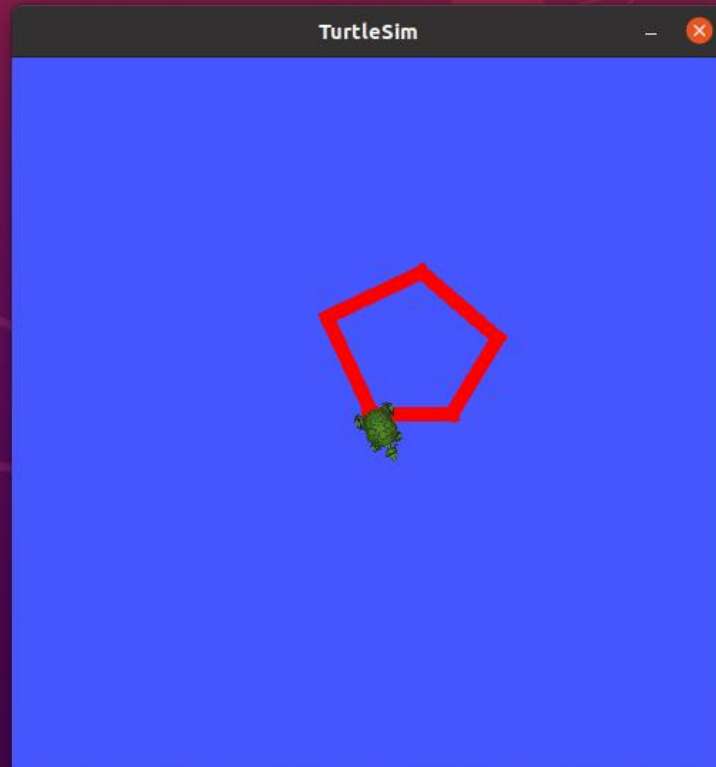
```
sc@sc-dell-1060:~$ cd ../../media/sc/HardDisk/
sc@sc-dell-1060:/media/sc/HardDisk$ cd rosbag2_2021_06_08-20_46_28/
sc@sc-dell-1060:/media/sc/HardDisk/rosbag2_2021_06_08-20_46_28$ ros2 bag play rosbag2_2021_06_08-20_46_28_0.db3 -l
```



```

/turtlesim/get_parameters: rcl_interfaces/srv/GetParameters
/turtlesim/list_parameters: rcl_interfaces/srv/ListParameters
/turtlesim/set_parameters: rcl_interfaces/srv/SetParameters
/turtlesim/set_parameters_atomically: rcl_interfaces/srv/SetParametersAtomically
Service Clients:

Action Servers:
/turtle1/rotate_absolute: turtlesim/action/RotateAbsolute
Action Clients:
```

```

sy@sy-home: ~ 51x12
sy@sy-home:~$ ros2 run turtlesim turtlesim_node
[INFO]: Starting turtlesim with node name /turtlesim
[INFO]: Spawning turtle [turtle1] at x=[5.544445],
y=[5.544445], theta=[0.000000]
[INFO]: Resetting turtlesim.
[INFO]: Spawning turtle [turtle1] at x=[5.544445],
y=[5.544445], theta=[0.000000]
[INFO]: Resetting turtlesim.
[INFO]: Spawning turtle [turtle1] at x=[5.544445],
y=[5.544445], theta=[0.000000]

sy@sy-home:~$ ros2 run turtlesim turtle_teleop_key
Reading from keyboard
-----
Use arrow keys to move the turtle.
Use G|B|V|C|D|E|R|T keys to rotate to absolute orie
ntations. 'F' to cancel a rotation.
'Q' to quit.
|

sy@sy-home: ~ 105x12
^Csy@sy-home:~$ ^C
sy@sy-home:~$ ^C
sy@sy-home:~$ ros2 service call /turtle1/set_pen turtlesim/srv/SetPen "{r:255, g:0, b:0}"
Failed to populate field: 'SetPen_Request' object has no attribute 'r:255'
sy@sy-home:~$ ros2 service call /turtle1/set_pen turtlesim/srv/SetPen "{r: 255, g: 0, b: 0, width: 10}"
waiting for service to become available...
requester: making request: turtlesim.srv.SetPen_Request(r=255, g=0, b=0, width=10, off=0)

response:
turtlesim.srv.SetPen_Response()

sy@sy-home:~$ |

sy@sy-home: ~ 51x26
st()

response:
std_srvs.srv.Empty_Response()

sy@sy-home:~$ ros2 service list
/clear
/kill
/reset
/spawn
/teleop_turtle/describe_parameters
/teleop_turtle/get_parameter_types
/teleop_turtle/get_parameters
/teleop_turtle/list_parameters
/teleop_turtle/set_parameters
/teleop_turtle/set_parameters_atomically
/turtle1/set_pen
/turtle1/teleport_absolute
/turtle1/teleport_relative
/turtlesim/describe_parameters
/turtlesim/get_parameter_types
/turtlesim/get_parameters
/turtlesim/list_parameters
/turtlesim/set_parameters
/turtlesim/set_parameters_atomically
sy@sy-home:~$ |

sy@sy-home: ~ 51x26
lifecycle_msgs/srv/GetState
logging_demo/srv/ConfigLogger
map_msgs/srv/GetMapROI
map_msgs/srv/GetPointMap
map_msgs/srv/GetPointMapROI
map_msgs/srv/ProjectedMapsInfo
map_msgs/srv/SaveMap
map_msgs/srv/SetMapProjections
nav_msgs/srv/GetMap
nav_msgs/srv/GetPlan
--More--^C
sy@sy-home:~$ ^C
sy@sy-home:~$ ^C
sy@sy-home:~$ ^C
sy@sy-home:~$ ros2 service type /reset
std_srvs/srv/Empty
sy@sy-home:~$ ros2 service call /reset std_srvs/srv
/Empty
waiting for service to become available...
requester: making request: std_srvs.srv.Empty_Reque
st()

response:
std_srvs.srv.Empty_Response()

sy@sy-home:~$ |

```


TurtleSim



sy@sy-home: ~

sy@sy-home: ~ 51x12

```
[INFO]: Resetting turtlesim.  
[INFO]: Spawning turtle [turtle1] at x=[5.544445],  
y=[5.544445], theta=[0.000000]  
[INFO]: Resetting turtlesim.  
[INFO]: Spawning turtle [turtle1] at x=[5.544445],  
y=[5.544445], theta=[0.000000]  
[INFO]: Resetting turtlesim.  
[INFO]: Spawning turtle [turtle1] at x=[5.544445],  
y=[5.544445], theta=[0.000000]  
[INFO]: Rotation goal completed successfully  
[INFO]: Rotation goal completed successfully
```

sy@sy-home: ~ 51x12

```
sy@sy-home:~$ ros2 run turtlesim turtle_teleop_key  
Reading from keyboard  
-----  
Use arrow keys to move the turtle.  
Use G|B|V|C|D|E|R|T keys to rotate to absolute orie  
ntations. 'F' to cancel a rotation.  
'Q' to quit.
```

sy@sy-home: ~ 105x12

```
708}"  
Waiting for an action server to become available...  
Sending goal:  
  theta: 1.5708  
  
Goal accepted with ID: 86d19d601840489da3ed6765c2fc4709  
  
Result:  
  delta: -1.5520002841949463  
  
Goal finished with status: SUCCEEDED  
sy@sy-home:~$ ros2 action send_goal /turtle1/rotate_absolute turtlesim/action/RotateAbsolute "{theta: 1.5  
708}"
```

sy@sy-home: ~ 51x26

```
sy@sy-home:~$ ros2 action list  
/turtle1/rotate_absolute  
sy@sy-home:~$ |
```

sy@sy-home: ~ 51x26

```
sy@sy-home:~$ ros2 action info /turtle1/rotate_abso  
lute  
Action: /turtle1/rotate_absolute  
Action clients: 1  
  /teleop_turtle  
Action servers: 1  
  /turtlesim  
sy@sy-home:~$ |
```

CLI 기반 command-line tools

ros2 param

delete	지정 파라미터 삭제
describe	지정 파라미터 정보출력
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	-l Loop 재생
record	Rosbag 재생	-a 모든 토픽 기록 -b 크기를 지정하여 끊어서 저장

ros2 interface

list	사용가능한 모든 인터페이스 목록 출력
Package	특정패키지에서 사용가능한 인터페이스 목록 출력
Packages	인터페이스 패키지들의 목록 출력
Proto	지정 패키지의 프로토타입 출력
show	지정 인터페이스의 데이터 형태 출력

GUI 기반 RQT

- 그래픽 인터페이스 개발을 위한 Qt 기반 프레임워크 제공
- 노드와 그들 사이의 연결 정보 표시
- 속도, 전압 등 시간이 지남에 따라 변화하는 데이터 플로팅
- 그 외 30여가지

RVIZ

- 3차원 시각화툴
- 레이저, 카메라 등의 센서 데이터를 시각화
- 로봇 외형과 계획된 동작을 표현

SIMULATOR

- 3차원 시뮬레이터
- 물리엔진을 탑재, 로봇, 센서, 환경 모델 등을 지원
- Gazebo, Igsvl, webot 과 같은 시뮬레이터 존재

RVIZ

18장 ROS2 GUI 개발을 위한 RQt

Ros의 종합 GUI 툴 RQt

- RQt는 플러그인 형태로 다양한 도구와 인터페이스를 구현할 수 있는 GUI(Graphical User Interface) 프레임 워크이며 다양한 목적의 GUI 툴을 모아둔 ROS의 종합 GUI 툴박스이다.
- 참고로 RQt의 플러그인[10]은 ROS 1 Noetic 기준으로 30개가 디폴트로 설치되는데 ROS 2 Foxy 기준으로는 14개가 디폴트로 설치되고 Robot Tools의 2가지를 추가로 설치 가능하다.
- ROS 2 포팅 작업[11]은 현재 진행형(PickNik사를 시작으로 현재는 Amazon이 주도함)으로 ROS 2 Foxy 기준으로 사용 가능한 RQt 플러그인 종류는 다음과 같다. 이전에 다룬 ROS의 CI 툴과 비슷한 기능이 꽤 많은데 CI툴과 GUI 툴은 각기 목적에 따라 편한 것을 사용하면 된다.

Topics

- Message Publisher
- Message Type Browser
- Topic Monitor

Services

- Service Caller
- Service Type Browser

Actions

- Action Type Browser

Introspection

- Node Graph
- Process Monitor

Configuration

- Dynamic Reconfigure

Logging

- Console

Miscellaneous Tools

- Python Console
- Shell

Visualization

- Image View
- Plot

Robot Tools

- Diagnostics Viewer
- Robot Steering


Ros의 종합 GUI 툴 RQt

- Ros2 관련해서 패키지를 바이너리 형태로 다운받을 수 있는지 확인.
 - apt-cache search ros-foxy-... 검색

```
sy@sy-home:~$ apt-cache search ros-foxy-rqt*
ros-foxy-rqt - rqt is a Qt-based framework for GUI development for ROS.
ros-foxy-rqt-action - rqt_action provides a feature to introspect all available ROS action types.
ros-foxy-rqt-common-plugins - rqt_common_plugins metapackage provides ROS backend graphical tools suite that can be used on/off of robot runtime.
ros-foxy-rqt-console - rqt_console provides a GUI plugin for displaying and filtering ROS messages.
ros-foxy-rqt-graph - rqt_graph provides a GUI plugin for visualizing the ROS computation graph.
ros-foxy-rqt-gui - rqt_gui provides the main to start an instance of the ROS integrated graphical user interface provided by qt_gui.
ros-foxy-rqt-gui-cpp - rqt_gui_cpp enables GUI plugins to use the C++ client library for ROS.
ros-foxy-rqt-gui-cpp-dbgsym - debug symbols for ros-foxy-rqt-gui-cpp
ros-foxy-rqt-gui-py - rqt_gui_py enables GUI plugins to use the Python client library for ROS.
ros-foxy-rqt-image-view - rqt_image_view provides a GUI plugin for displaying images using image_transport.
ros-foxy-rqt-image-view-dbgsym - debug symbols for ros-foxy-rqt-image-view
ros-foxy-rqt-moveit - An rqt-based tool that assists monitoring tasks for MoveIt! motion planner developers and users.
ros-foxy-rqt-msg - A Python GUI plugin for introspecting available ROS message types.
ros-foxy-rqt-plot - rqt_plot provides a GUI plugin visualizing numeric values in a 2D plot using different plotting backends.
ros-foxy-rqt-publisher - rqt_publisher provides a GUI plugin for publishing arbitrary messages with fixed or computed field values.
ros-foxy-rqt-py-common - rqt_py_common provides common functionality for rqt plugins written in Python.
ros-foxy-rqt-py-console - rqt_py_console is a Python GUI plugin providing an interactive Python console.
ros-foxy-rqt-reconfigure - This rqt plugin provides a way to view and edit parameters on nodes.
ros-foxy-rqt-robot-dashboard - rqt_robot_dashboard provides an infrastructure for building robot dashboard plugins in rqt.
ros-foxy-rqt-robot-monitor - rqt_robot_monitor displays diagnostics_agg topics messages that are published by diagnostic_aggregator.
ros-foxy-rqt-robot-steering - rqt_robot_steering provides a GUI plugin for steering a robot using Twist messages.
ros-foxy-rqt-service-caller - rqt_service_caller provides a GUI plugin for calling arbitrary services.
ros-foxy-rqt-shell - rqt_shell is a Python GUI plugin providing an interactive shell.
ros-foxy-rqt-srv - A Python GUI plugin for introspecting available ROS message types.
ros-foxy-rqt-top - RQT plugin for monitoring ROS processes.
ros-foxy-rqt-topic - rqt_topic provides a GUI plugin for displaying debug information about ROS topics including publishers, subscribers, publishing rate, and ROS Messages.
sy@sy-home:~$ |
```

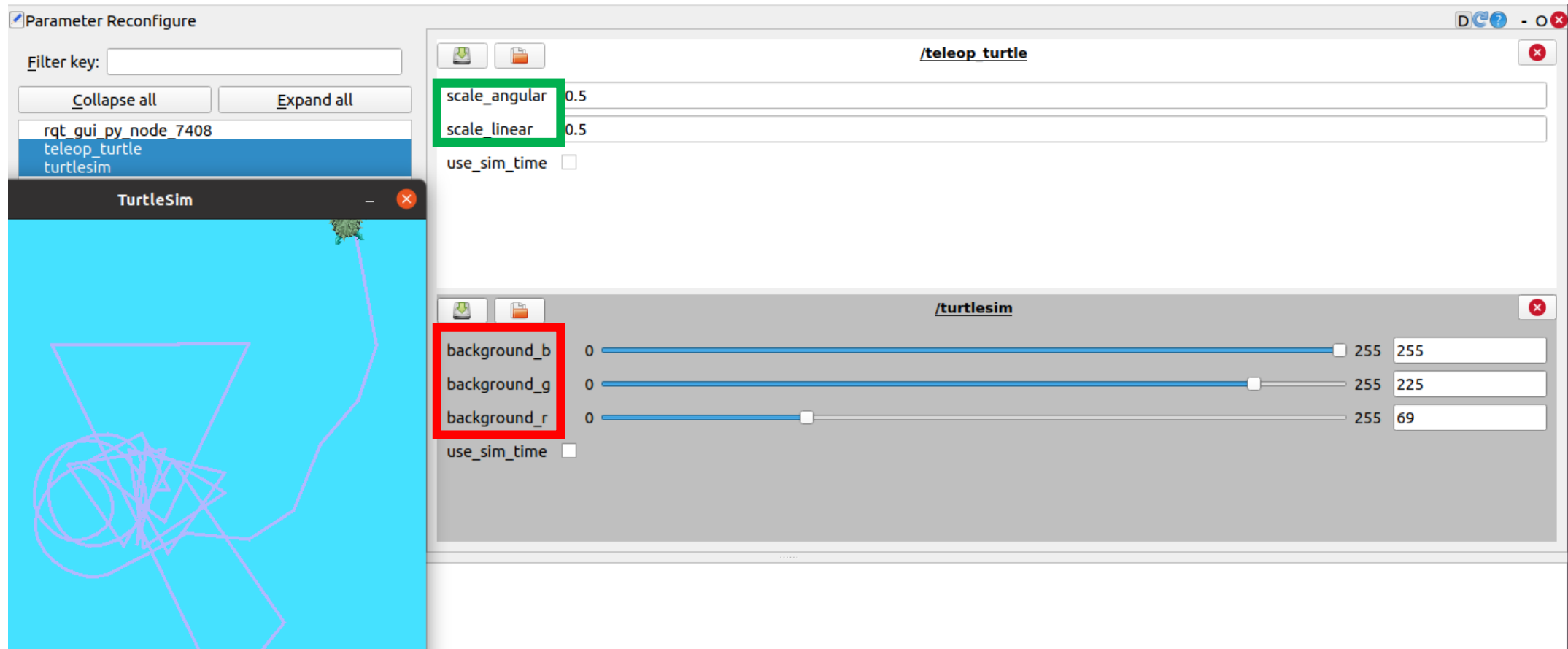
1. Actions

1. Actions Type Browser : Action 타입의 데이터 구조 확인

Action Type Browser		
Message: turtlesim / RotateAbsolute 		
Tree	Type	Path
▼ Action Goal	turtlesim/RotateAbsolute/Goal	turtlesim/RotateAbsolute/Goal
theta	float	turtlesim/RotateAbsolute/Goal/theta
▼ Action Result	turtlesim/RotateAbsolute/Result	turtlesim/RotateAbsolute/Result
delta	float	turtlesim/RotateAbsolute/Result/delta
▼ Action Feedback	turtlesim/RotateAbsolute/Feedback	turtlesim/RotateAbsolute/Feedback
remaining	float	turtlesim/RotateAbsolute/Feedback/remaining

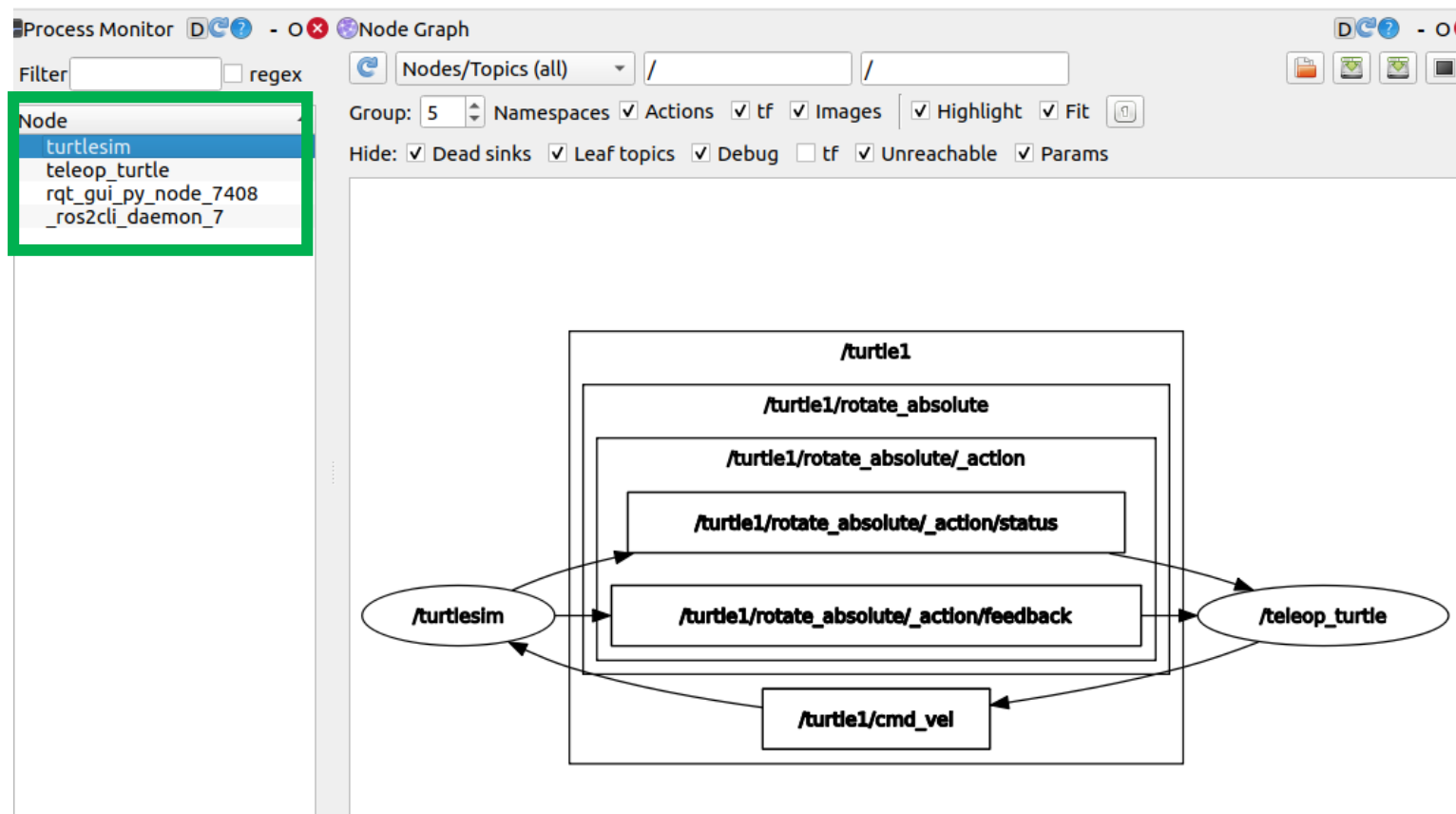
1. Parameter Reconfigure

1. Dynamic Reconfigure : 노드들에서 제공하는 파라미터 값을 확인하고 변경할 수 있는 RQt 플러그인이다.



1. Inspection

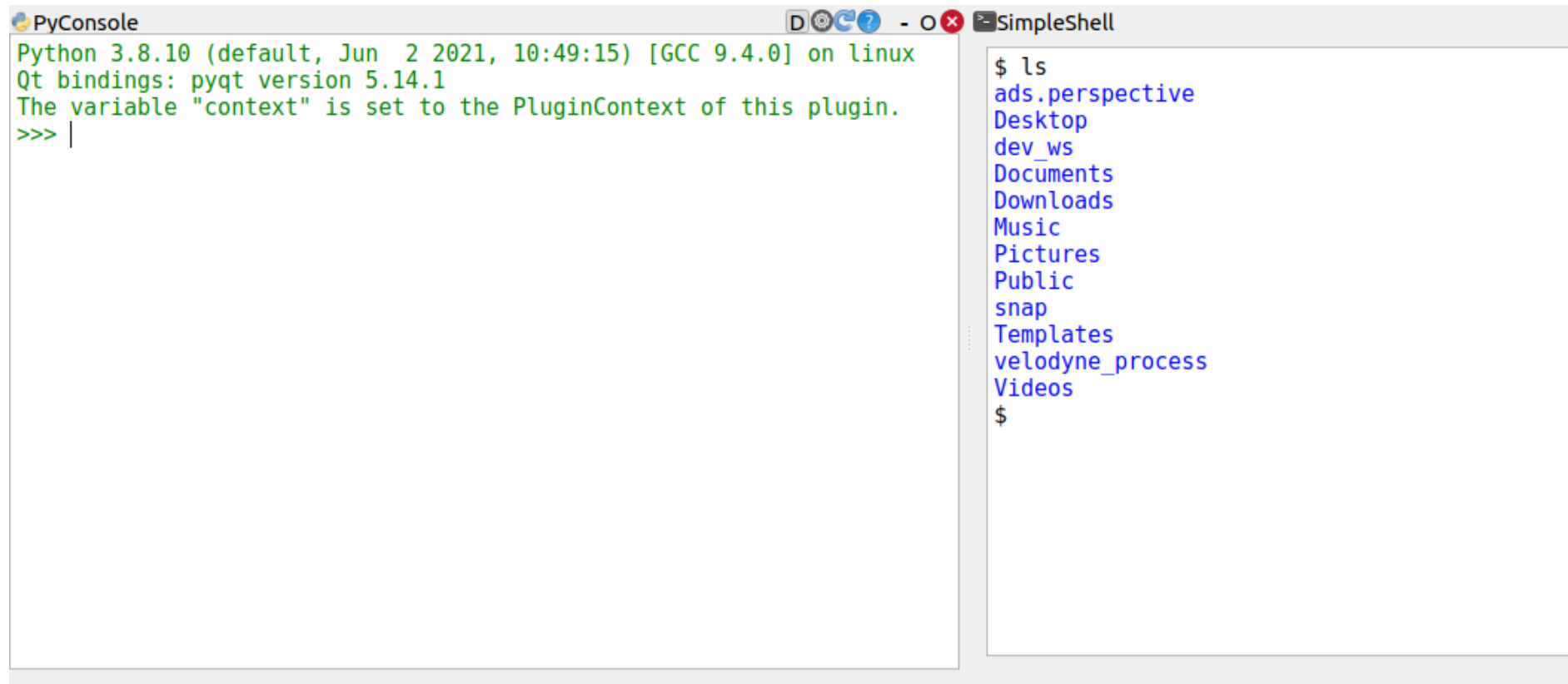
1. Node Graph : 실행중인 노드들의 관계 및 토픽을 확인 가능한 그래프 뷰
2. Process Monitor: 실행 중인 노드들의 CPU 사용률, 메모리 사용률, 스레드 수 등을 확인



1. Miscellaneous Tools(다양한 툴)

1. Python Console : 파이썬 콘솔
2. Shell : 셸을 구동

활용방안에 대한 의문



```
PyConsole
Python 3.8.10 (default, Jun  2 2021, 10:49:15) [GCC 9.4.0] on linux
Qt bindings: pyqt version 5.14.1
The variable "context" is set to the PluginContext of this plugin.
>>> |

SimpleShell
$ ls
ads.perspective
Desktop
dev_ws
Documents
Downloads
Music
Pictures
Public
snap
Templates
velodyne_process
Videos
$
```


1. 서비스

1. Service caller : 실행 중인 서비스 서버에 접속하여 서비스를 요청
2. Service Type Browser : 서비스 타입의 데이터 구조를 확인.

The screenshot displays the RQt interface with three main components:

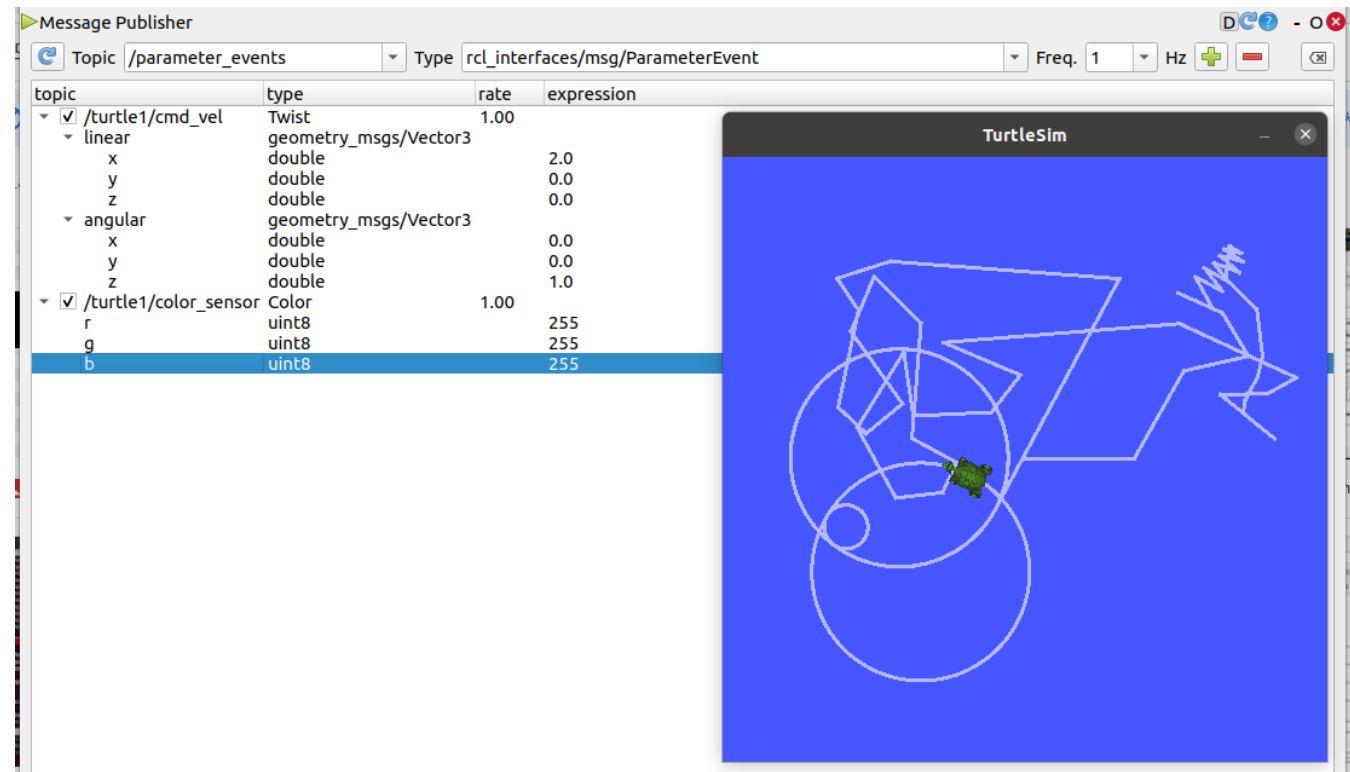
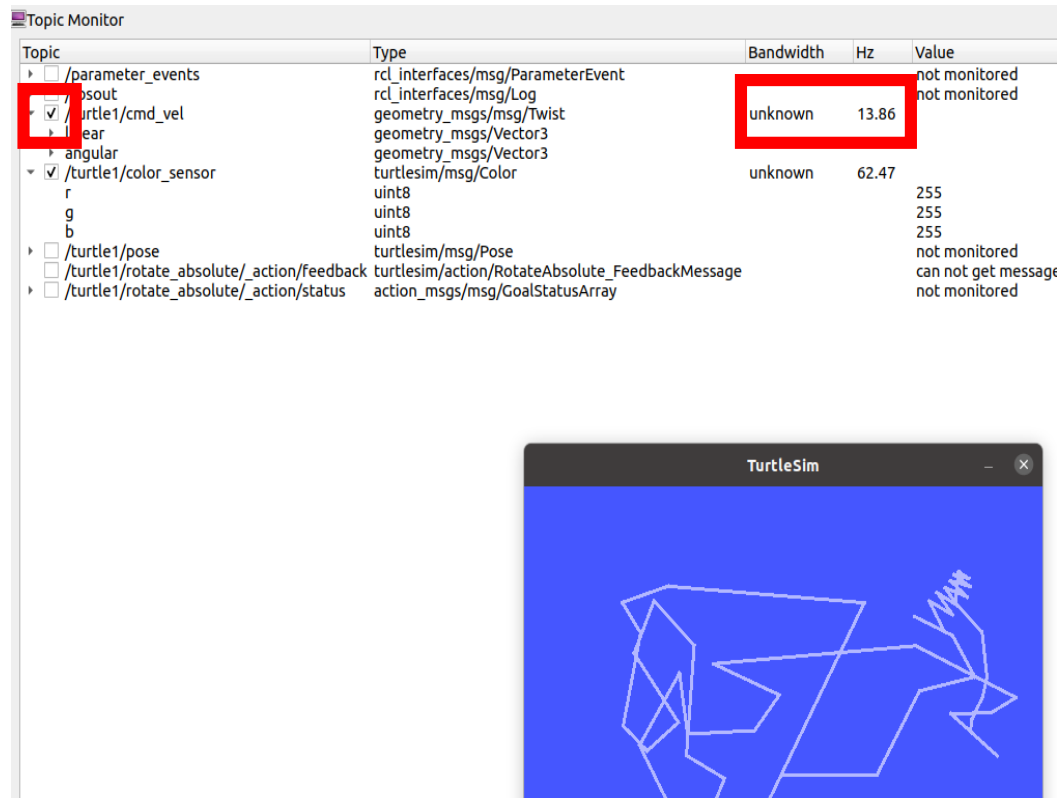
- Service Caller:** Shows a service named `/reset` being called. The request table is empty. The response table shows a single entry: `/` with type `std_srvs/srv/Empty.Response`.
- Service Type Browser:** Displays a tree view of service types. The selected service is `std_srvs/Empty/Request` and `std_srvs/Empty/Response`.
- TurtleSim:** A window showing a turtle in a blue environment.

The terminal output at the bottom shows the following log messages:

```
sy@sy-home: ~  
sy@sy-home: ~ 126x13  
[INFO]: Resetting turtlesim.  
[INFO]: Spawning turtle [turtle1] at x=[5.544445], y=[5.544445], theta=[0.000000]  
[INFO]: Resetting turtlesim.  
[INFO]: Spawning turtle [turtle1] at x=[5.544445], y=[5.544445], theta=[0.000000]  
[ERROR]: Tried to kill turtle [], which does not exist  
[ERROR]: Tried to kill turtle [], which does not exist  
[ERROR]: Tried to kill turtle [], which does not exist  
[INFO]: Clearing turtlesim.  
[INFO]: Resetting turtlesim.  
[INFO]: Spawning turtle [turtle1] at x=[5.544445], y=[5.544445], theta=[0.000000]  
[INFO]: Resetting turtlesim.  
[INFO]: Spawning turtle [turtle1] at x=[5.544445], y=[5.544445], theta=[0.000000]  
sy@sy-home: ~ 126x10  
sy@sy-home:~$ [INFO]: Result received: 0 1 1 2 3 5 8 13 21 34 55  
[INFO]: command not found  
sy@sy-home:~$  
sy@sy-home:~$ ros2 run turtlesim turtle_teleop_key  
Reading from keyboard  
-----  
Use arrow keys to move the turtle.  
Use G|B|V|C|D|E|R|T keys to rotate to absolute orientations. 'F' to cancel a rotation.  
'Q' to quit.
```

1. 토픽

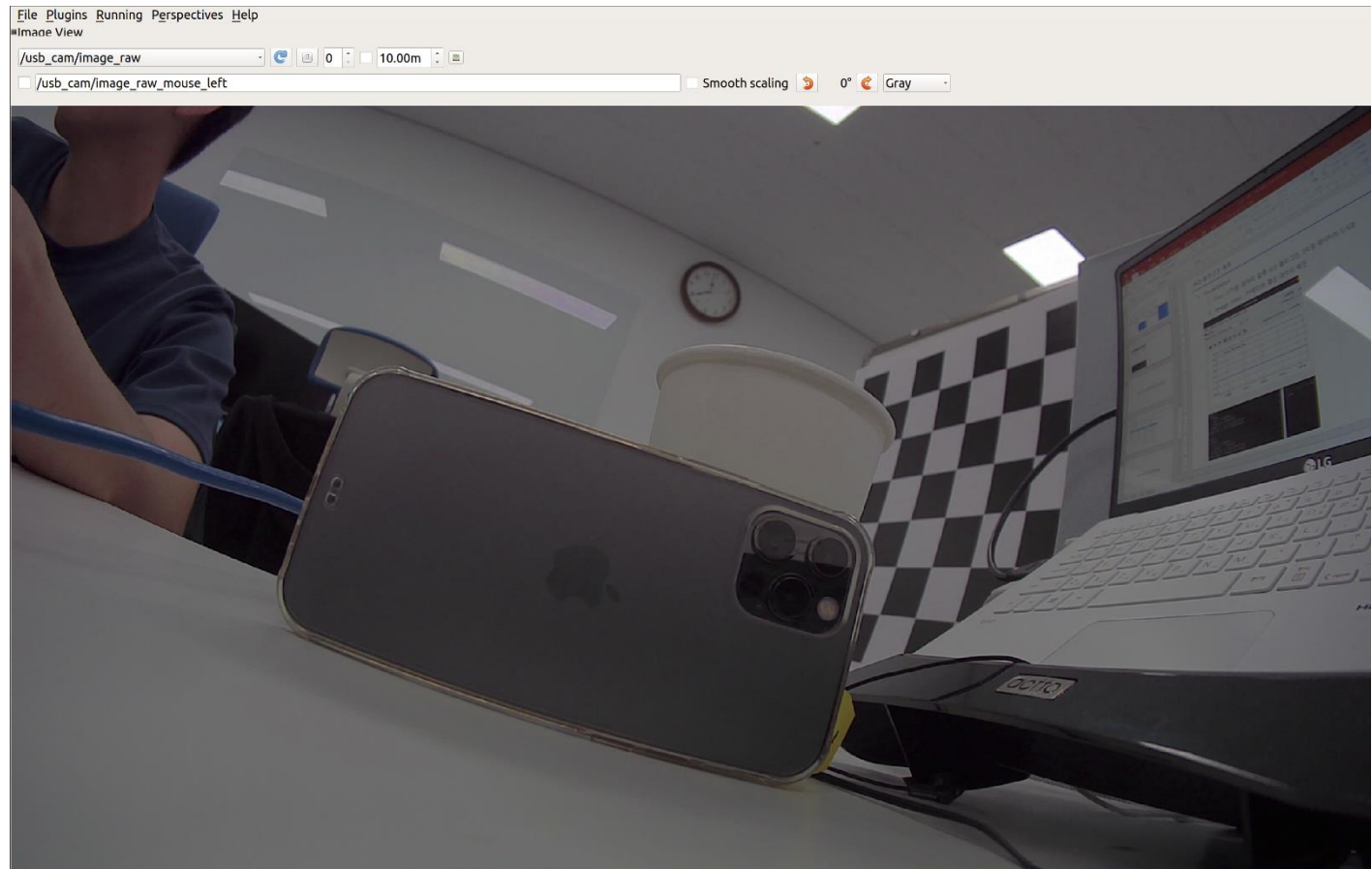
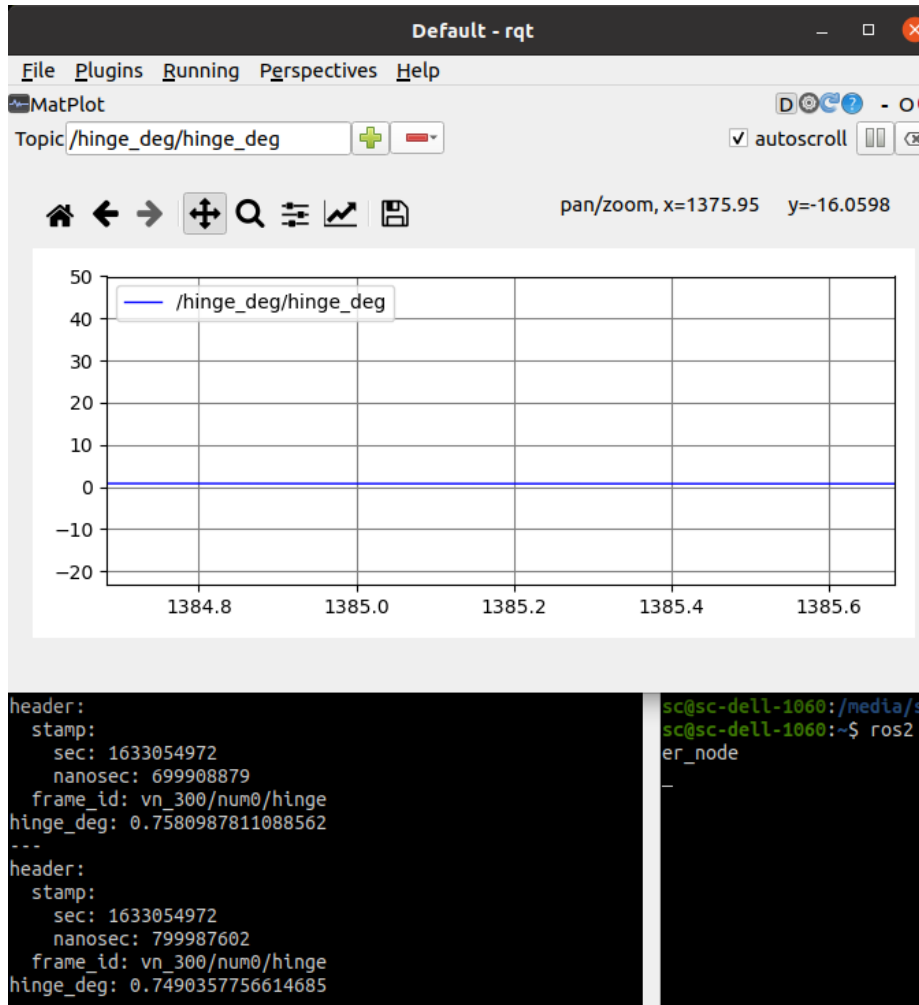
1. Topic Monitor : 토픽 목록 확인 및 사용자가 선택한 토픽의 정보를 확인
2. Message Publisher : 메시지 발행- 'ros2 topic pub' 명령어의 GUI버전
3. Message Type Browser : 메시지 타입의 데이터 구조 확인



RQt 플러그인 종류

1. Visualization

1. Plot : 2차원 데이터 플롯 GUI 플러그인, 2차원 데이터의 도식화
2. Image View : 카메라의 영상 데이터 확인



19장 ROS2의 물리량 표준 단위

1. 표준 단위의 필요성

1. 단위의 불일치는 치명적인 소프트웨어 버그로 이어질 수 있다.

예) 개발자 : m/s, rad/s 정의하여 개발 → 사용자 : cm/s, deg/s 로 생각하여 사용

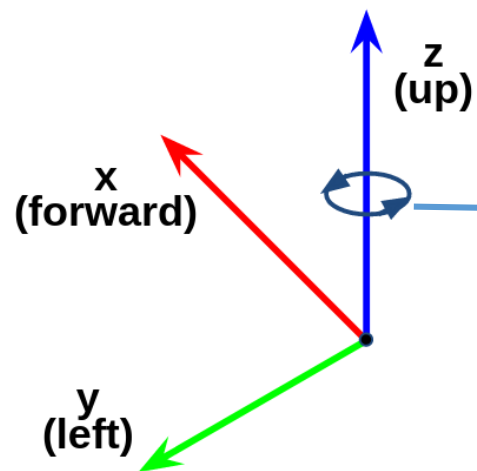
2. ROS2의 표준 단위

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)

[표 1: ROS 2의 표준 단위]

20장 ROS2 의 좌표표현

1. 기본 3축



회전각 : radian

로봇이 제자리에서 12시방향 → 9시방향으로 회전

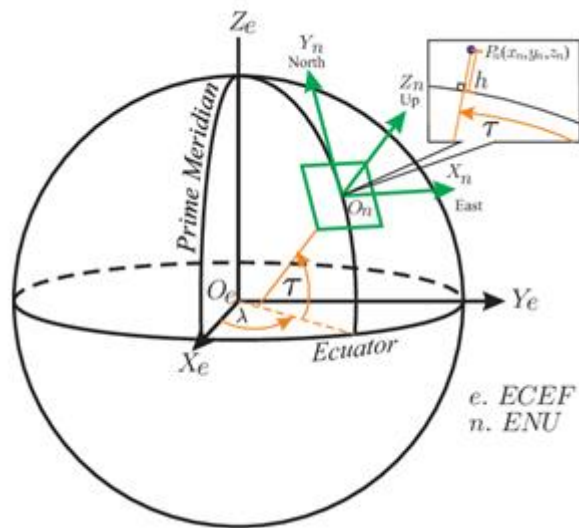
정방향으로 +1.5708 rad회전

2. ENU 좌표

지리적 위치(geographic locations)의 단거리 데카르트 표현의 경우 ENU(east north up, [5]) 규칙을 사용함.

실내 로봇에서는 잘 다루지 않는 좌표

비교적 큰 맵을 다루는 드론, 실외 자율주행 로봇에서 사용하는 좌표라고 생각하면 된다.

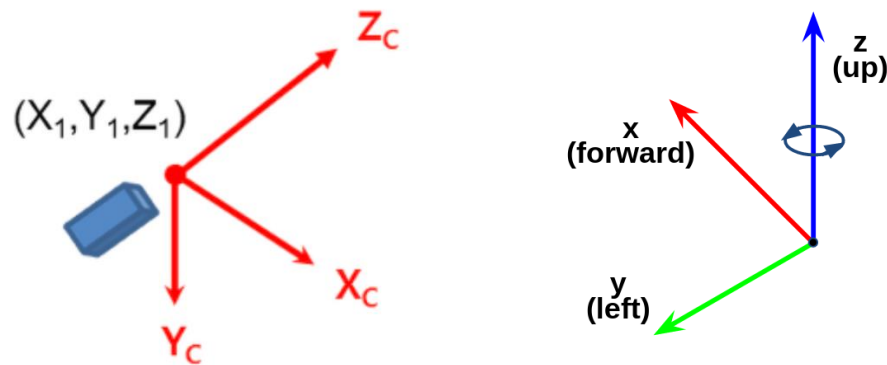


1. North(N) – 경도선과 평행을 이루며, 북쪽을 바라보는 방향
2. East(E) – 위도선과 평행을 이루며, 동쪽을 바라보는 방향
3. Up(U) – 지구 중심과 멀어지는 방향 표현

3. 접미사 프레임 사용

1. _optical 접미사

- 컴퓨터 비전 분야의 경우, 카메라 좌표계로 많이 사용되는 z forward, x right, y down를 사용함
- 카메라 센서 메시지에 _optical 접미사를 붙여 다른 좌표계간 혼란을 일으키지 말아야함.



2. _ned 접미사

- 실외에서 동작하는 시스템 경우, ENU가 아니라 NED(north, east, down) 좌표계를 사용할 때도 있음.
- 이때에는 _ned 접미사를 붙여 구분.