Week_5

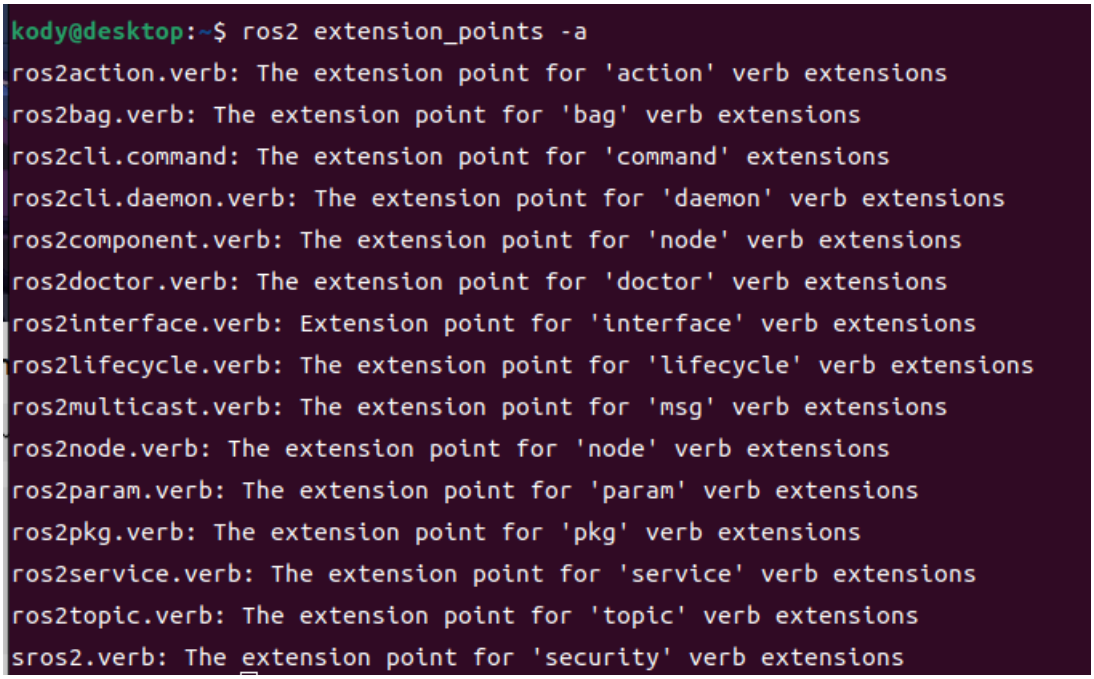
17.1 ROS 도구

- ROS의 도구들은 **ROS 커뮤니티**에서 **다년간의 의견과 경험**이 녹아져 만들어 졌으며 이를 이용하여 **자신의 로봇 프로젝트**의 코드들을 쉽게 빌드하고 패키징하여 관리할 수 있다.
- 데이터들을 기록,재생,관리하고 CLI나 GUI 형태로 프로그램을 만들거나, 3D simulator 안에서 디버깅하거나 테스트할 수 있다.

- ROS 도구는 **4가지 형태**로 분류할 수 있다.
  - CLI형태
    - Command-Line Tools
  - GUI형태
    - RQt
  - 3D 시각화 툴
    - Rviz
  - 3D simulator
    - Gazebo

17.1.1 CLI기반 Command-Line Tools

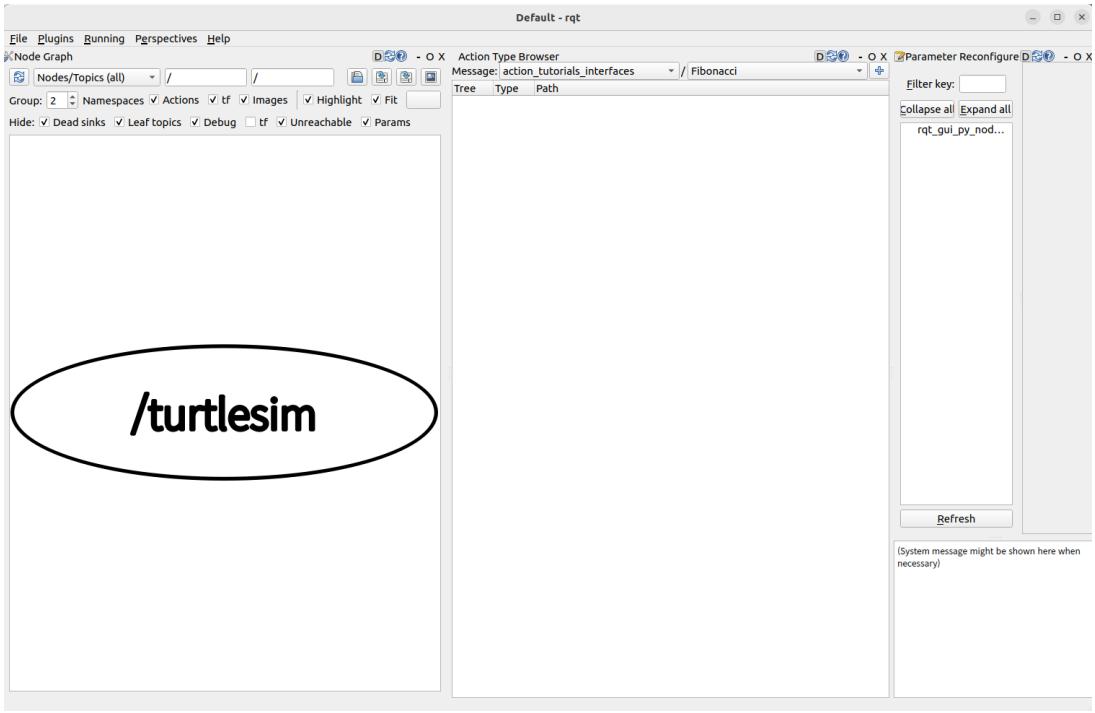
- 거의 모든 ROS기능을 다룬다
- 개발환경 / 빌드 / 테스트 툴(colcon)
- ros2bag등 데이터를 기록, 재생, 관리하는 툴
  - 등등 20가지



여러 가지가 있다.

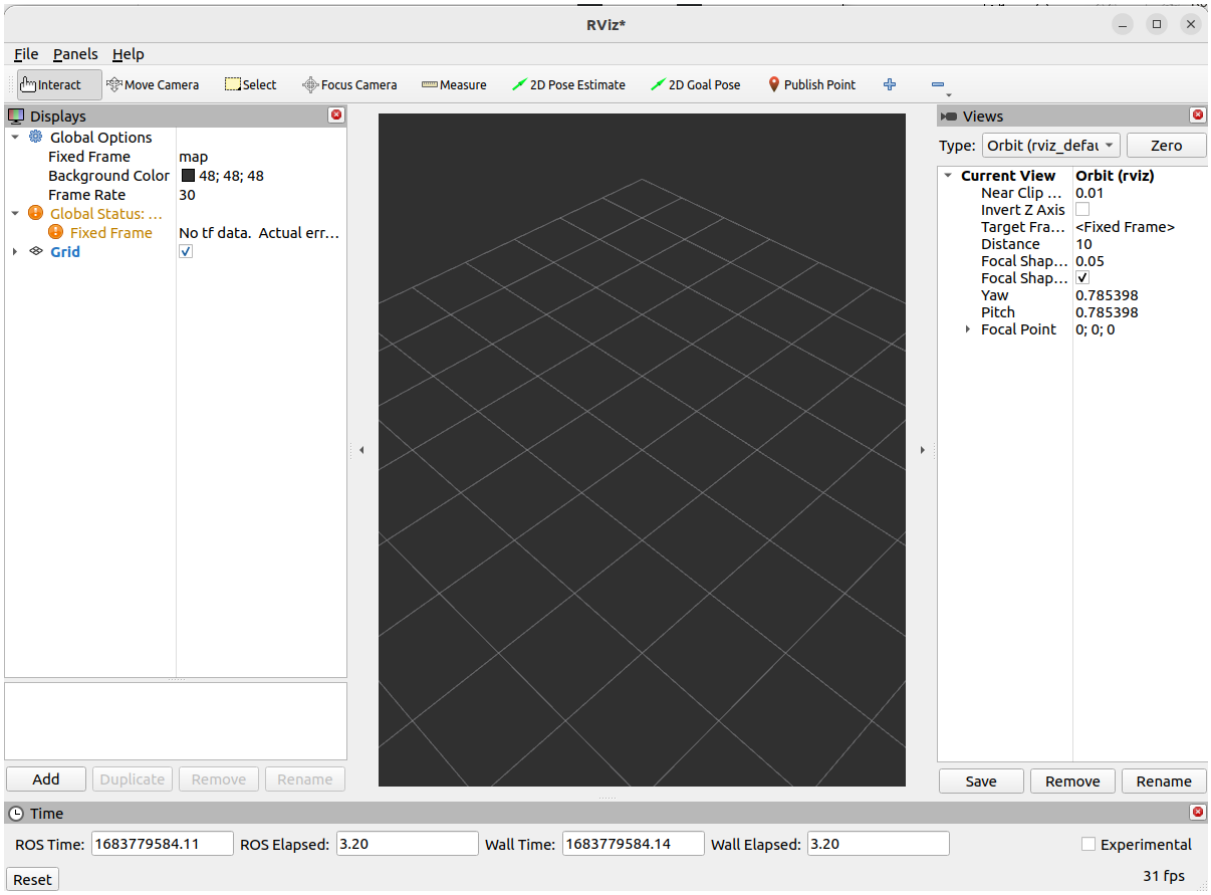
17.1.2 GUI기반 RQt

- GUI 개발을 위한 Qt프레임워크 제공
- 노드와 노드 사이의 연결 정보 표시(rqt_graph)
- 시간에 따라 변화하는 데이터(속도,전압 등)를 플로팅(rqt_plot)
    - 그 외 30가지



17.1.3 RViz

- 3D 시각화 툴이다
- 센서 데이터를 시각화하는 도구이다(레이저, 카메라 등을 다룬다.)
- 로봇 외형 / 계획된 동작을 표현할 수 있다.



17.1.4 Gazebo

- 3D 시뮬레이터이다
- 물리 엔진을 탑재했기 때문에 로봇, 센서, 환경 모델을 지원한다.
- ROS와 높은 호환성을 보인다.

17.2 ROS2 CLI 명령어

- ROS2 CLI는 터미널 창에 입력하여 사용하는 명령어를 말하고
- ROS2 CLI 명령어는 ros2cli라는 이름으로 20개가 제공되고 있다.

- ros2cli개발 작업은 ROS2 TSC멤버인 Cananical과 AWS RoboMaker 개발자들이 참여하고 있다.

17.3 ROS2 CLI사용법

- ROS2 CLI는 터미널 창에서 각 명령어가 하나씩 수행되는 방식으로 동작하고
- 이는 다음 형식으로 실행시킬 수 있다.

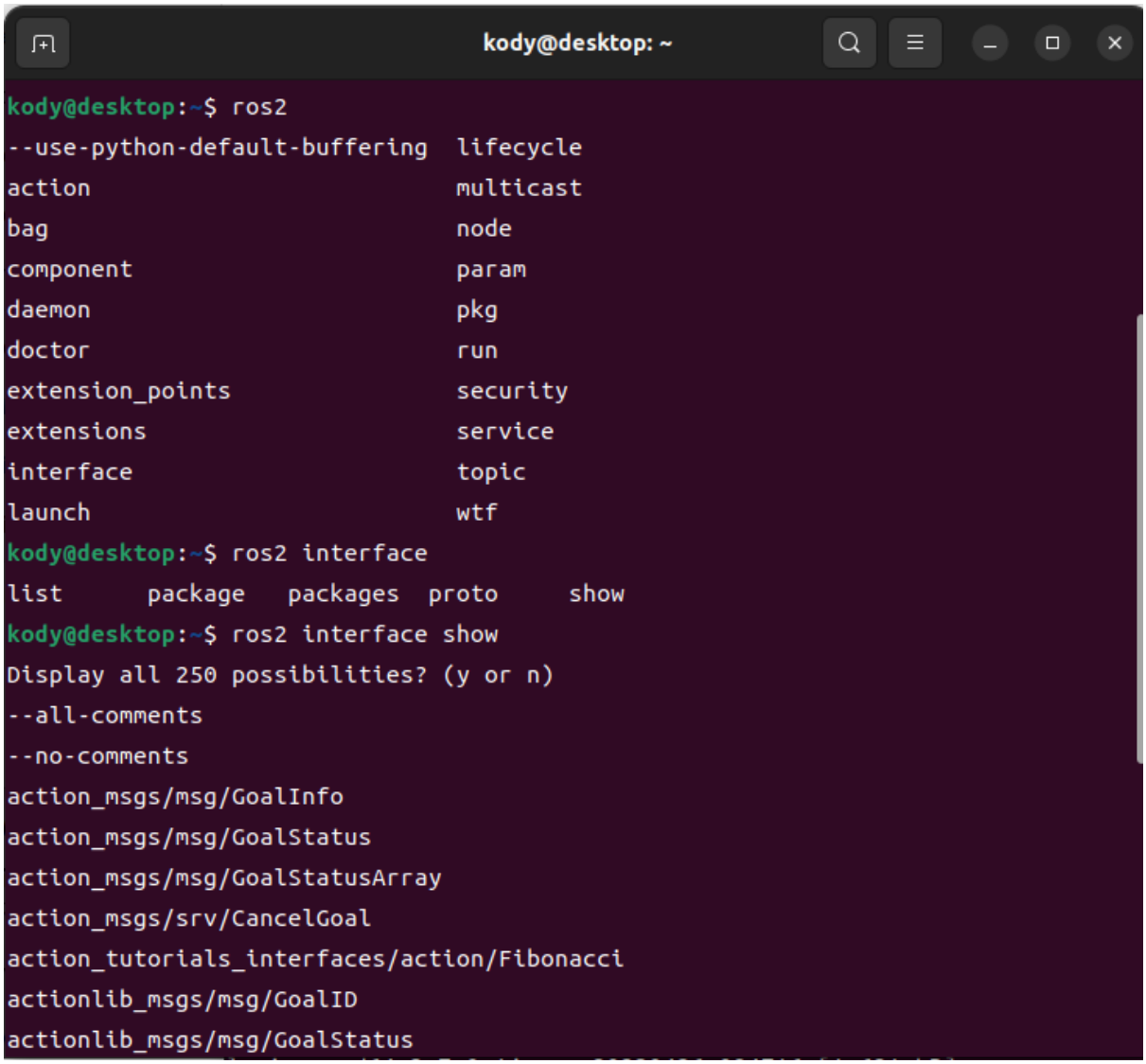
> 💡 $ ros2 [verbs] [sub-verbs] [options] [arguments]

- **ros2**는 고유한 entry-point로 **리눅스의 다른 명령어와 구분**해주기 위해서 존재하는 명령어이다.
- [verbs]는 run, launch, node와 같은 툴을 선택하는 옵션이다.
- [sub-verbs]는 툴에 대한 하위 옵션이다, (node의 경우 info, list등이 sub-verb가 된다)

- 터미널에서 **auto-completion**을 지원하기 때문에 **탭(Tab)키**를 이용하여 **외우지 않고도** 명령어들을 쉽고 빠르게 실행할 수 있다.

→ 이를 이용하면 **학습 또는 개발 과정을 크게 단축시킬 수 있다**

- Tab키 사용 예시



형식

> 💡 $ ros2 [tab] [tab] [tab] [tab] ….

- help옵션 사용(-h)

형식

```
$ ros2 -h
$ ros2 node -h
$ ros2 node info -h
```

17.4.1 ROS2 실행 명령어

- 형식: ros2cli + [verbs] / [arguments] / 기능
- ros2 run / <package> <executable> / 특정 패키지의 **노드 실행**(복수 노드실행가능)
- ros2 launch / <package> <launch-file> / 특정 패키지의 특정 **런치 파일 실행**

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

17.4.2 ROS2 정보 명령어

→정리해야 할 양이 너무 많으므로 터미널 + Tab키로 대체

```
kody@desktop:~$ ros2 pkg
create       executables  list         prefix       xml
kody@desktop:~$ ros2 node
info  list
```

```
kody@desktop:~$ ros2 topic
--include-hidden-topics  find                    pub
bw                       hz                      type
delay                    info
echo                     list
kody@desktop:~$ ros2 service
--include-hidden-services  list
call                       type
find
kody@desktop:~$ ros2 action
info       list       send_goal
kody@desktop:~$ ros2 interface
list      package   packages  proto     show
kody@desktop:~$ ros2 param
delete    describe  dump      get       list      load      set
kody@desktop:~$ ros2 bag
convert  info     list     play     record   reindex
```

17.4.3 ROS2 기능 보조 명령어

- extensions, extension_points, daemon, multicast, doctor, wtf, lifecycle, component, security등의 보조 명령어가 있다.

```
$ ros2 [기능 보조 명령어]
```

```
형식으로 사용한다.
```

18장 ROS2 GUI 개발 툴인 RQt

18.1 종합 GUI툴 RQt

RQt는 다양한 목적의 GUI툴을 모아둔 ROS의 통합 툴박스이다.

- ROS1 개발 초기에는
  - rxbag(토픽을 저장)
  - rxplot(시간축 계열의 데이터를 2차 플롯 창에 표시)
  - rxgraph(노드 간의 상관관계를 그래프로 표시)

    와 같은 GUI 툴을 개발해서 사용했으나

편의성을 높이고자 RQt를 공개하였다.

이러한 기능들은 rqt_bag, rqt_plot, rqt_graph와 같이 **플러그인 형태로 RQt에 포함되었다.**

- ROS2 Foxy 기준으로(지금 버전은 humble이다) RQt플러그인은 디폴트로 14개가 설치되는데,

  2가지를 추가 설치할 수 있다.

18.2 RQt 프레임워크

- RQt는 프레임워크이기 때문에 GUI개발에서 필요한 부분들을 API 형태로 제공한다.
- RQt는 ROS + Qt의 합성어로 Qt기반이기 때문에 크로스 플랫폼과 다양한 프로그래밍 언어를 지원한다.
    - Qt 구글 검색 결과

Qt

컴퓨터 프로그램

Qt는 컴퓨터 프로그래밍에서 GUI 프로그램 개발에 널리 쓰이는 크로스 플랫폼 프레임워크이다. 서버용 콘솔과 명령줄 도구와 같은 비GUI 프로그램 개발에도 사용된다. 그래픽 사용자 인터페이스를 사용하는 경우에는 Qt를 위젯 툴킷으로 분류한다. 위키백과

플랫폼: 크로스 플랫폼

프로그래밍 언어: C++, C++17

개발: 더 Qt 컴퍼니, 노키아, Digia, Qt Project

라이선스: GPL, LGPL, 상용 버전

안정화 버전: 6.5 / 2023년 4월 3일

종류: 개발 라이브러리

관련 검색어

10개 이상 항목 더보기

Qt Creator

PyQt

Qt 디자이너

CMake

Qt는 GUI 개발에 쓰이는 크로스 플랫폼 프레임워크이고 개발 언어는 C++를 사용한다.

18.3 RQt 실행 방법

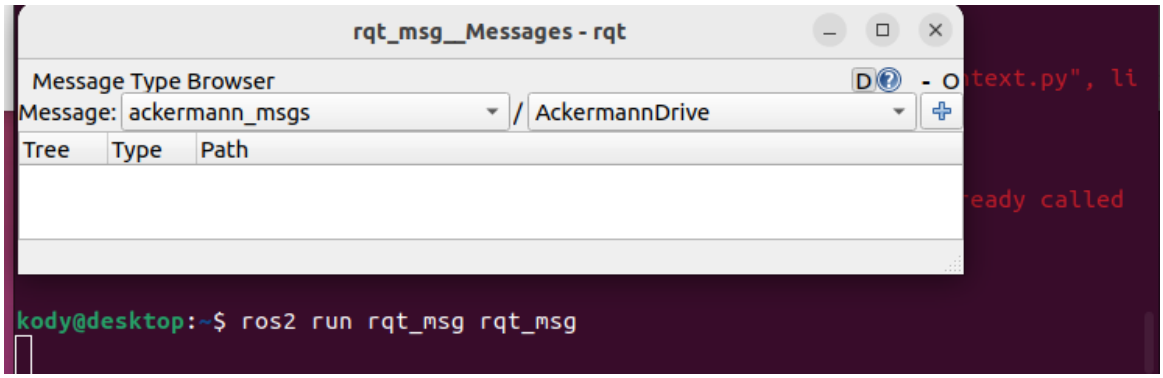
RQt를 실행하는 방법은 세 가지가 있다.

- rqt를 입력하여 실행

```
$ rqt
```

- **ros2 run 명령어**를 이용하여 rqt관련 패키지의 노드를 실행

```
$ ros2 run rqt_mag rqt_msg
```



- 지정된 단축 명령어를 사용하여 실행

```
$ rqt_graph
$ rqt_topic
```

18.4 RQt 플러그인의 종류

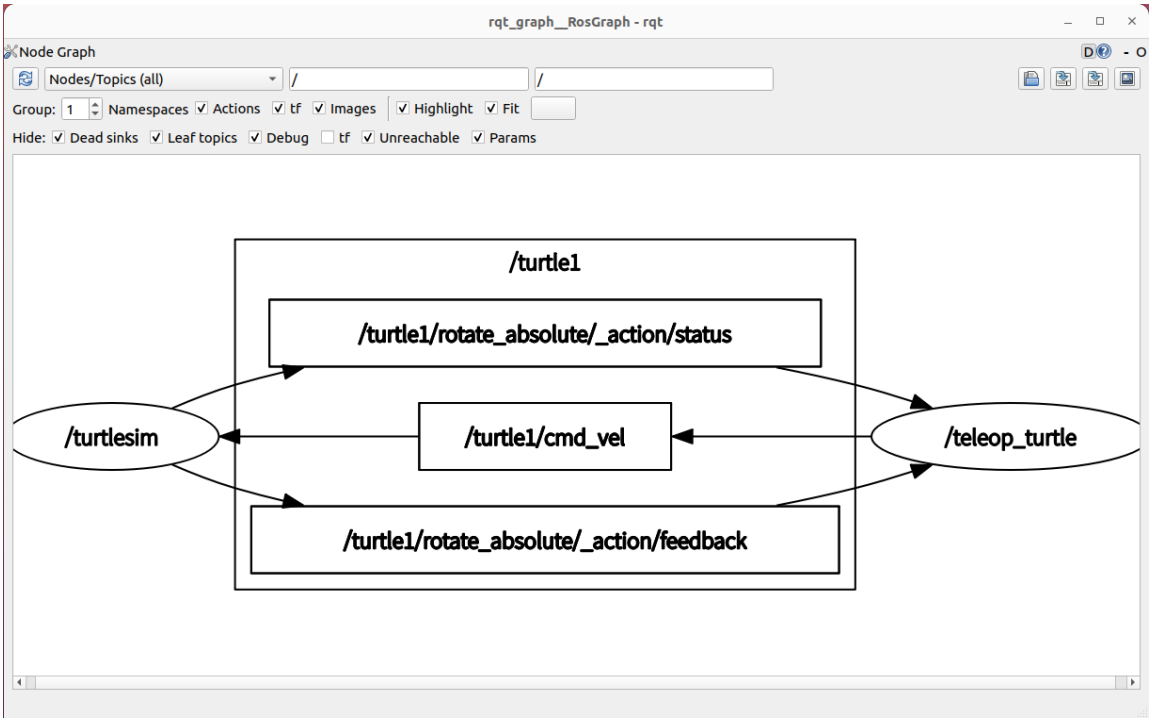
- ROS2 기준으로 RQt플러그인은 16개 사용 가능하다(Foxy기준)

  책을 참고하자

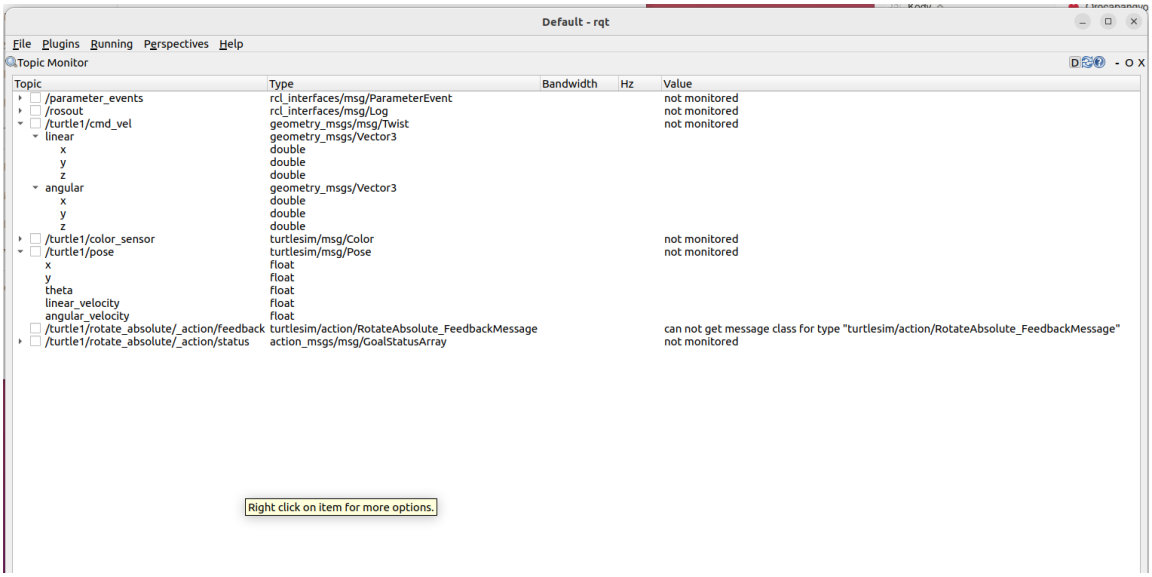
18.5 RQt사용 예시

- $ rqt 로 실행후 좌측 상단의 plugins으로 접근한다

- Node Graph



- Topic Monitor

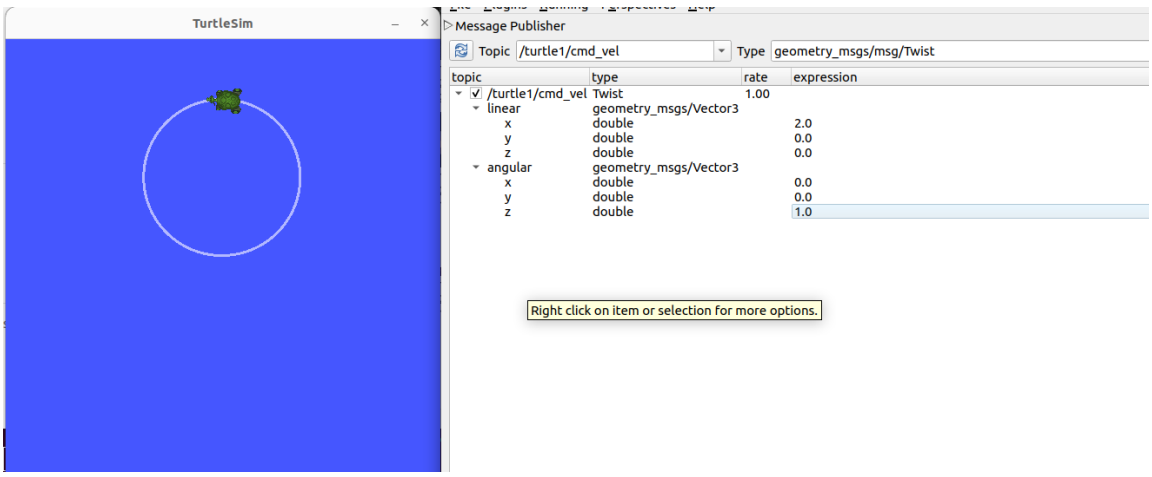


- Message Publisher

  ros topic pub 명령어의 GUI버전이다.

  plugins에서 Message Publisher으로 접근한 뒤에

  - Topic을 /turtle1/cmd_vel으로
  - Type를 geometry_msgs/msg/Twist로 설정하고
  - 주기(Freq)는 1Hz로 설정하고
  - +를 눌러주자



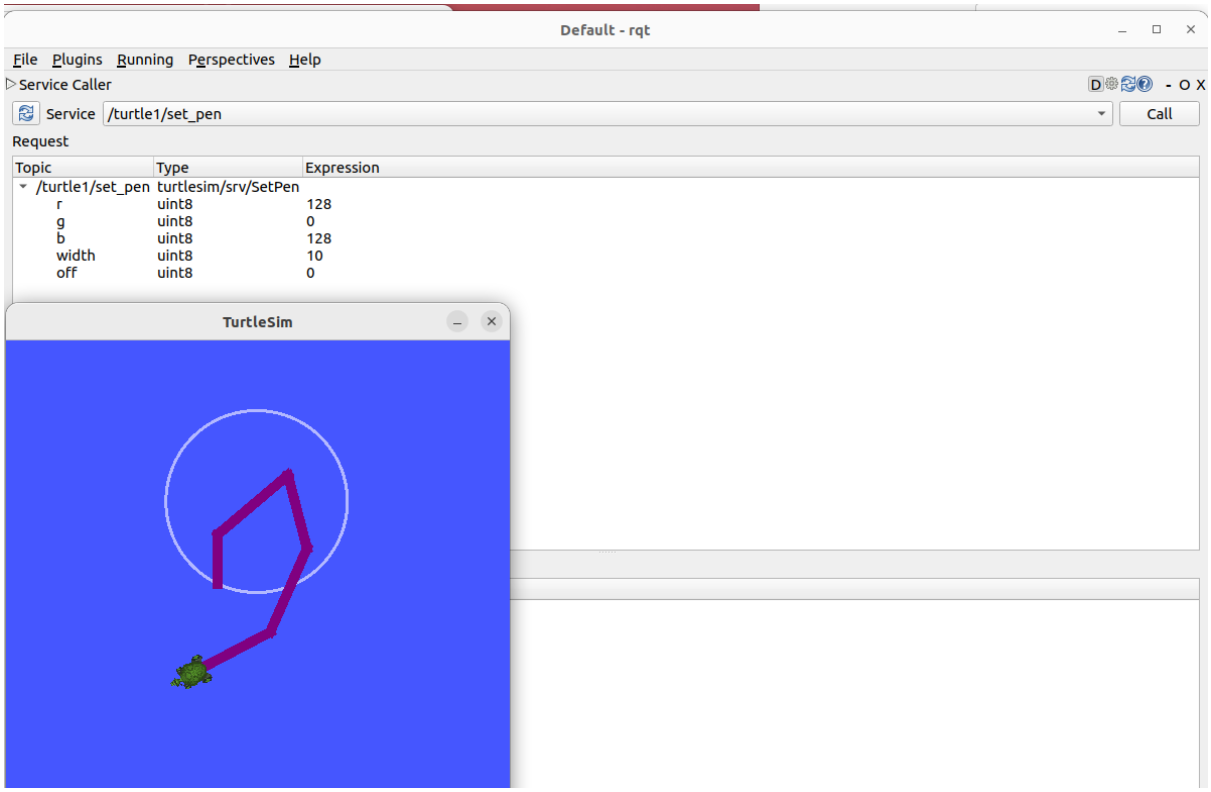
나타난 Topic의 창을 열어

    - linear.x를 2.0으로
    - angular.z를 1.0으로 설정하고
    - 체크 박스를 선택하면 거북이가 움직인다.
    - 이는 병진속도 2.0m/s , 회전속도 1.0rad/s로 거북이를 움직이게 한 것이다.

- 이외에도 Message Type Browser도 plugin에서 사용할 수 있다.

18.5.5 Service Caller

- Plugins에서 Service Caller를 실행한다.(Plugins → Services →Service Caller)
- Service를 /turtle/set_pen으로 설정한 후에 Expression값을 r:128, b:128, width:10으로 설정하고 Call을 클릭한다.

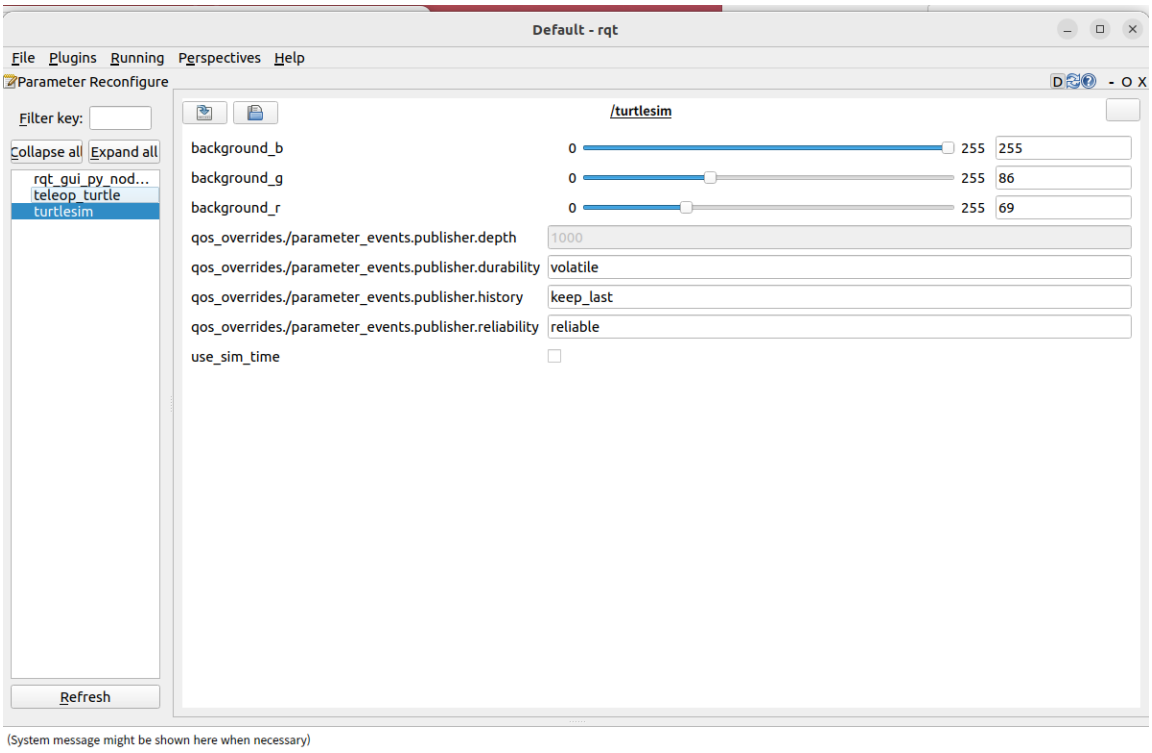


- 선이 보라색, width가 10으로 설정된 것을 볼 수 있다.

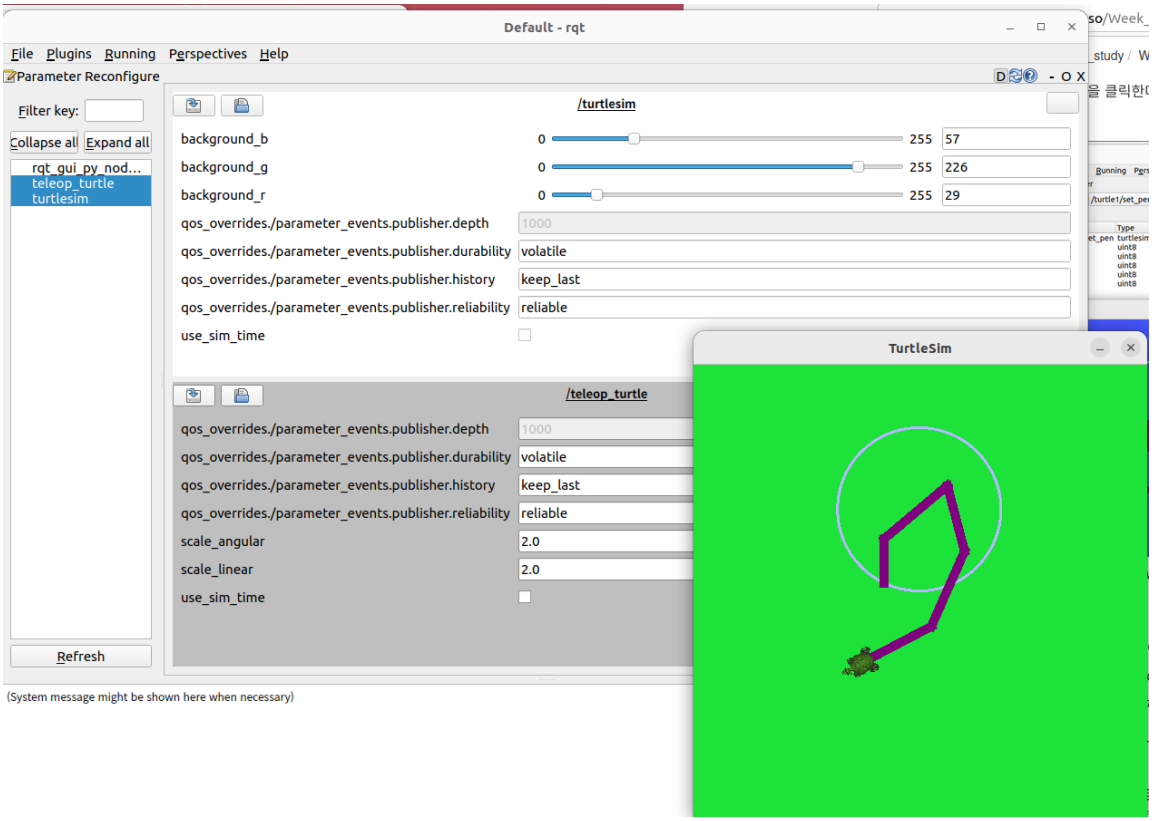
18.5.6 Parameter Reconfigure

- Plaugins →Configuration → Dynamic Reconfigure
- 노드에서 제공하는 파라미터 값을 확인하고 변경하는 플로그인으로
- ros2 param명령어의 GUI버전이다.

- 우측에서 turtlesim노드를 선택한다.



- 바를 움직이면 파라미터를 변경할 수 있고 다른 노드들의 파라미터도 변경할 수 있다.



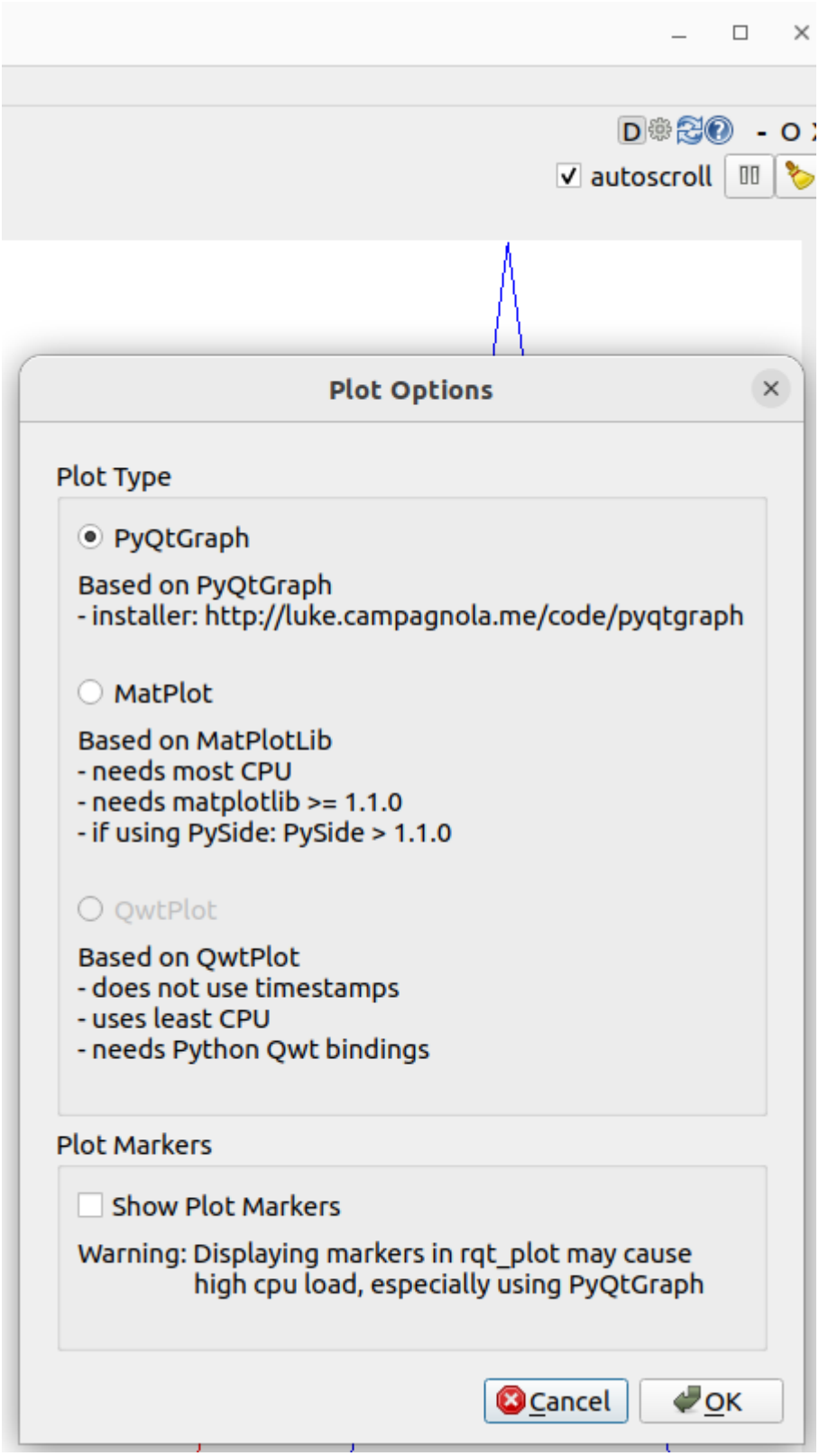
18.5.7 Plot

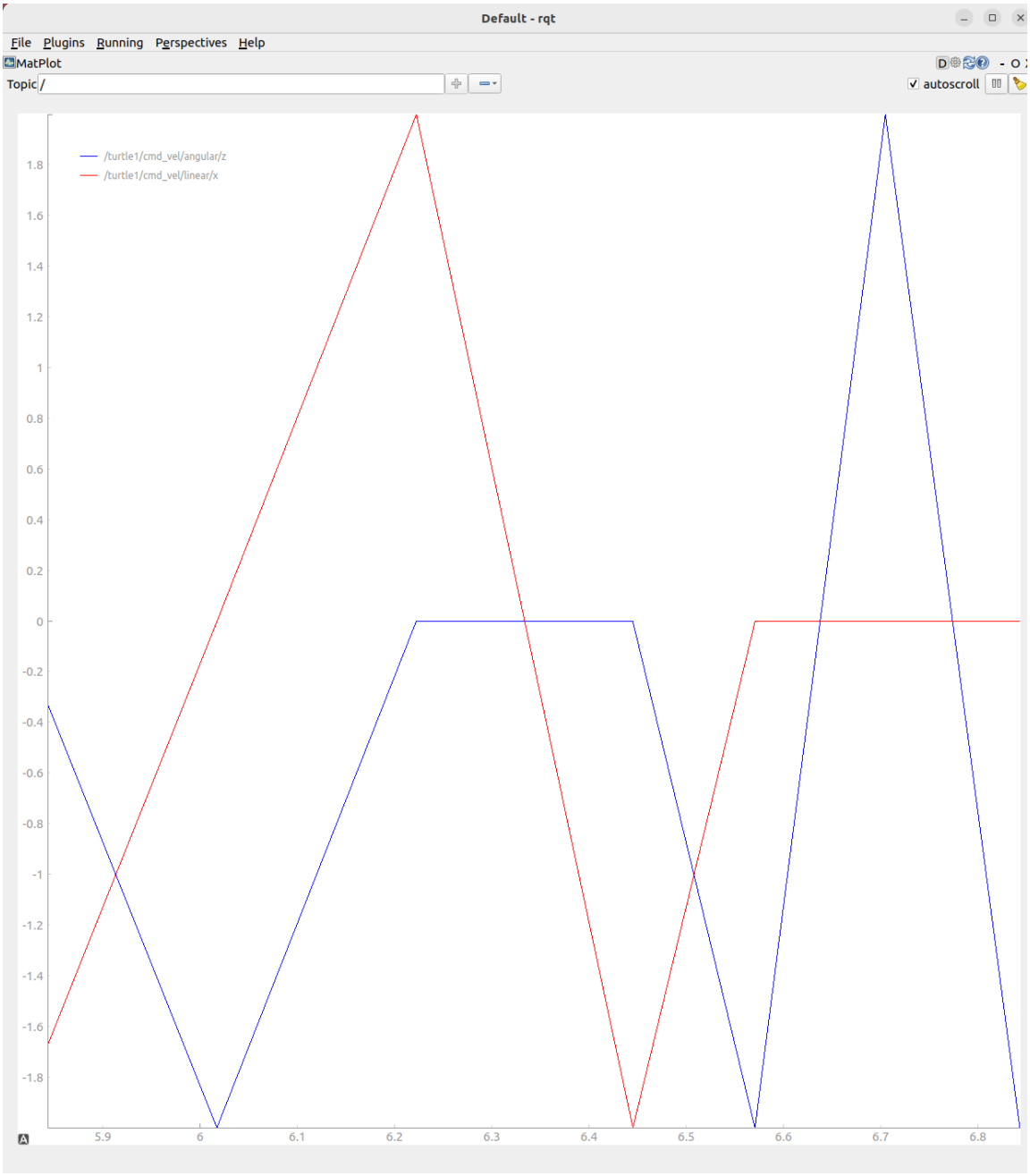
- Plugins → Visualization → Plot

  Plot플러그인은 2차원 데이터를 도식화하는 2차원 데이터 플롯 기능을 가진 플로그인이다.

- Topic에 /turtle1/cmd_vel/linear/x를 입력한 후 +를 클릭한다
- Topic에 /turtle1/cmd_vel/angular/z를 입력한 후 +를 클릭한다.
  - 우측 위 작은 톱니 선택 → 기본이 MatPlot으로 선택되어 있다

    PyQtGraph를 설치한 후에 정상 작동하였다.

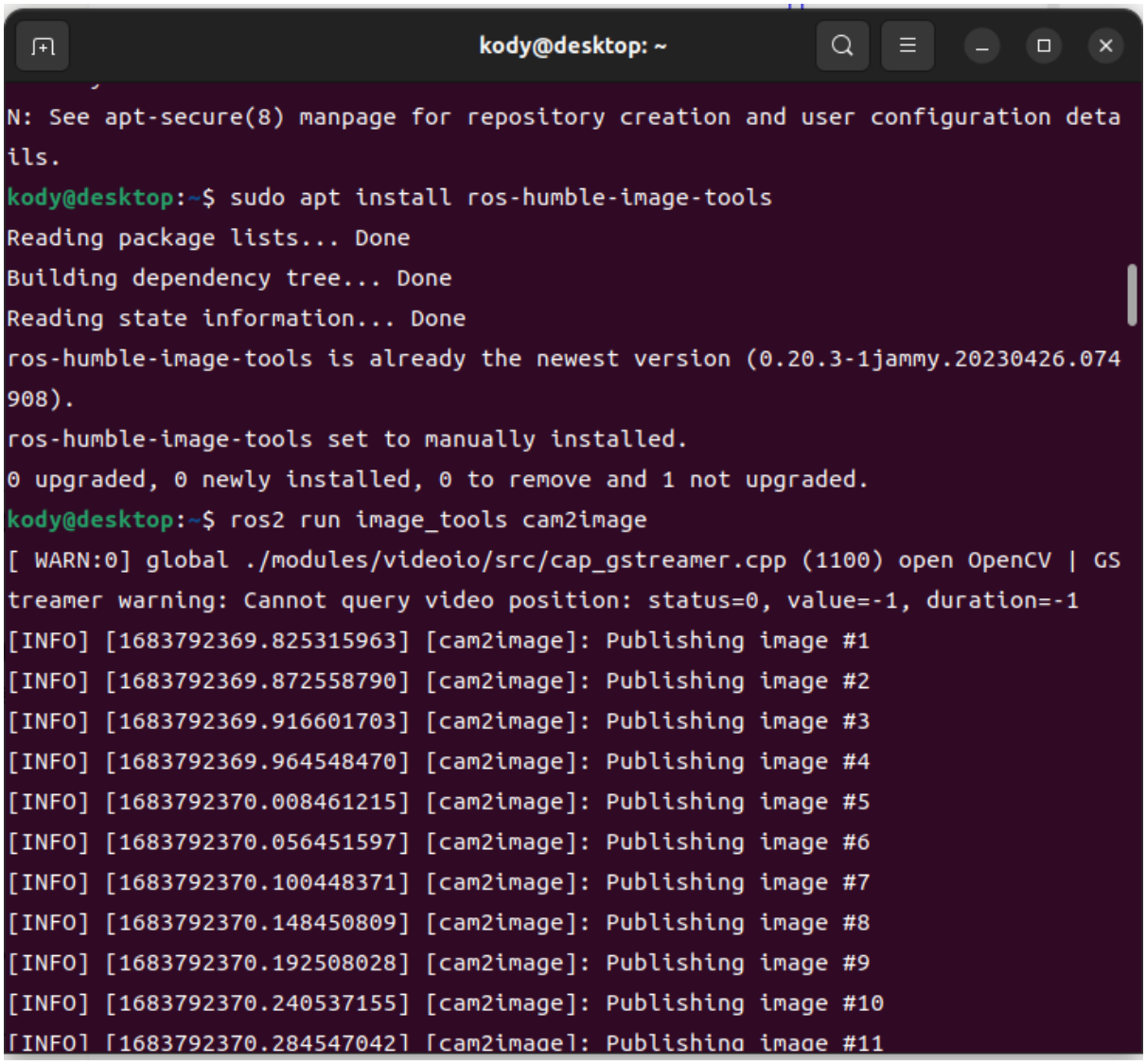

autoscroll
Plot Options
Plot Type
PyQtGraph
Based on PyQtGraph
- installer: http://luke.campagnola.me/code/pyqtgraph
MatPlot
Based on MatPlotLib
- needs most CPU
- needs matplotlib >= 1.1.0
- if using PySide: PySide > 1.1.0
QwtPlot
Based on QwtPlot
- does not use timestamps
- uses least CPU
- needs Python Qwt bindings
Plot Markers
Show Plot Markers
Warning: Displaying markers in rqt_plot may cause
high cpu load, especially using PyQtGraph
Cancel
OK



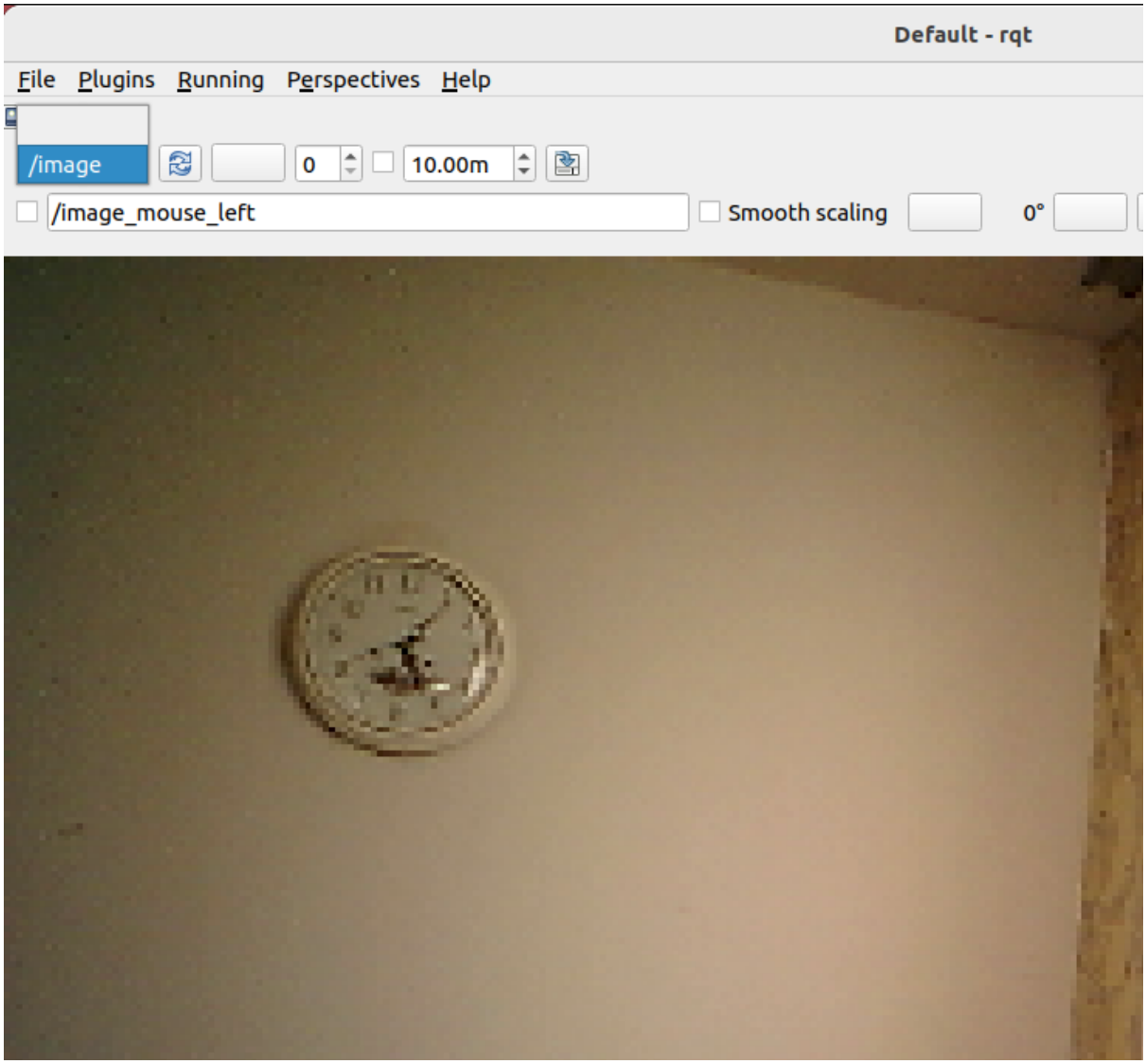
18.5.8 Image View

- Plugins → Visualization → Image View
- 터미널로 image_tools패키지의 cam2image 노드를 실행시키자

```
$ ros2 run image_tools cam2image
```



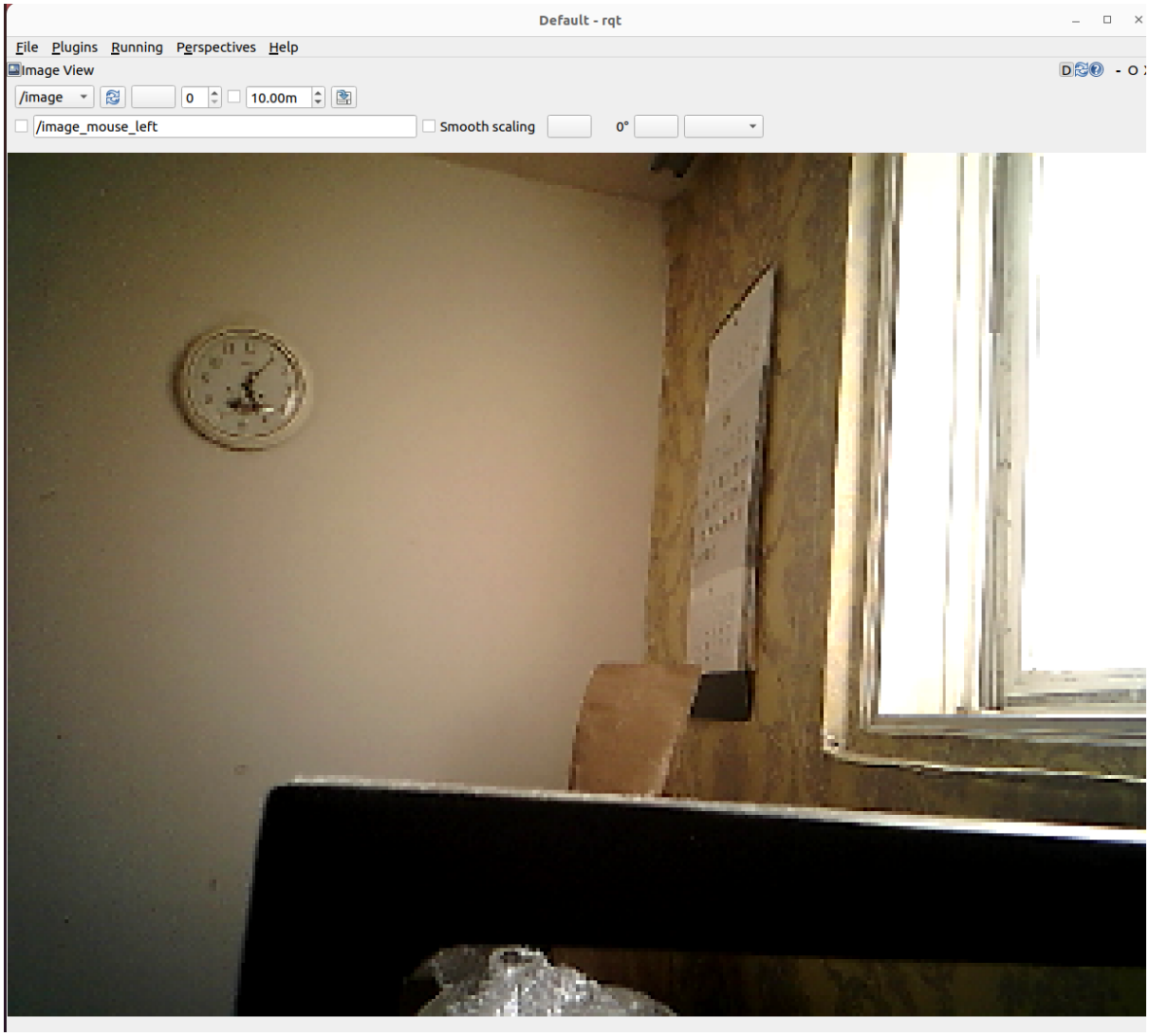
- image View의 박스를 클릭하면 /image가 추가된 것을 볼수 있다. 선택하자



- 카메라가 있다면 영상이 나타난다.
- 카메라가 없는 경우

```
$ ros2 run image_tools cam2image --ros-args -p burger_mode:=true
```

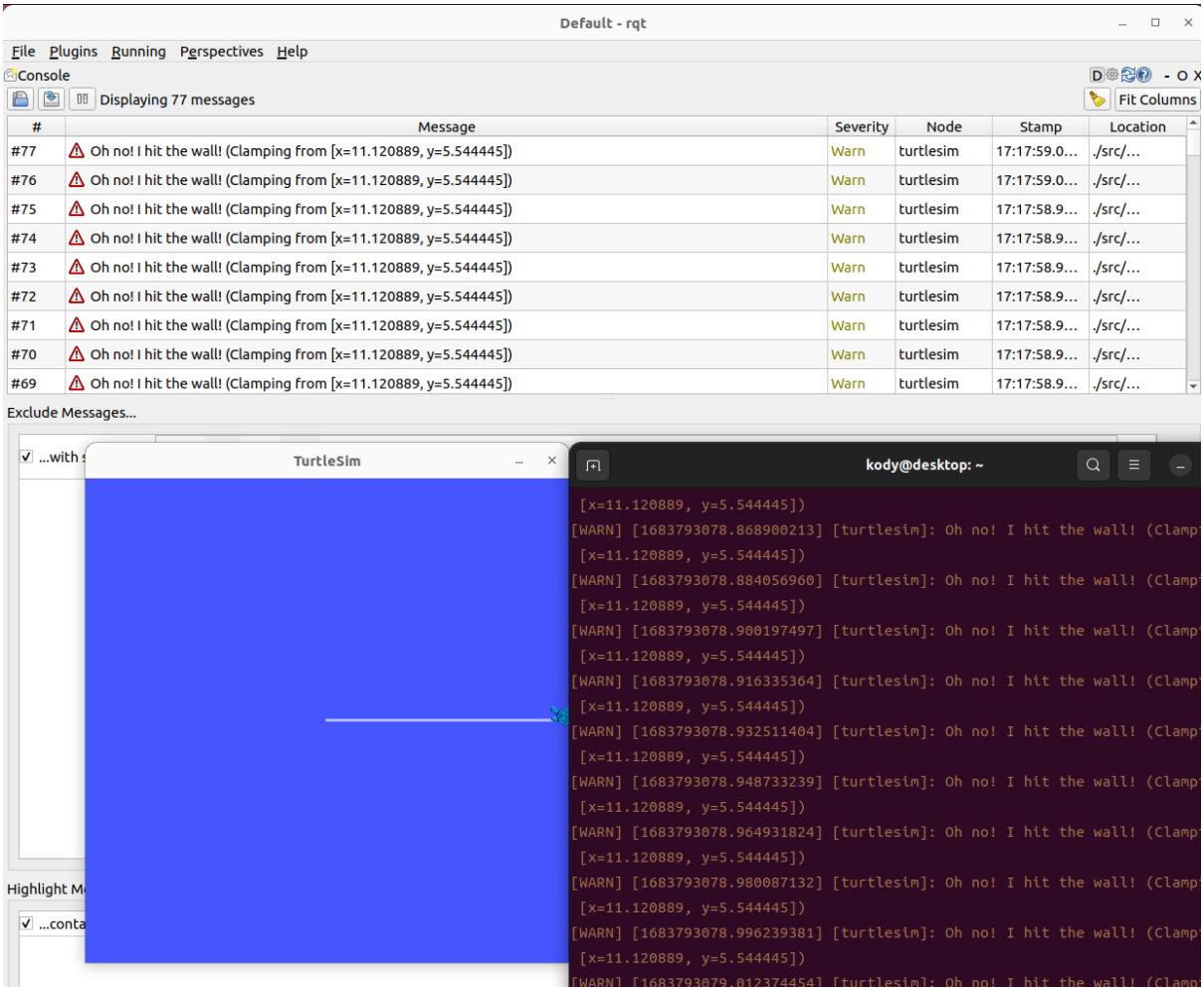
명령어로 움직이는 버거 이미지로 테스트할 수 있다.



18.5.9 Console

- Console플러그인은 노드에서 발생되는
  - 정보(info)
  - 경고(Warning)
  - 에러(Error)
- 등의 rosout데이터를 확인할 수 있는 플러그인이다.
- Plugins → Logging → Console

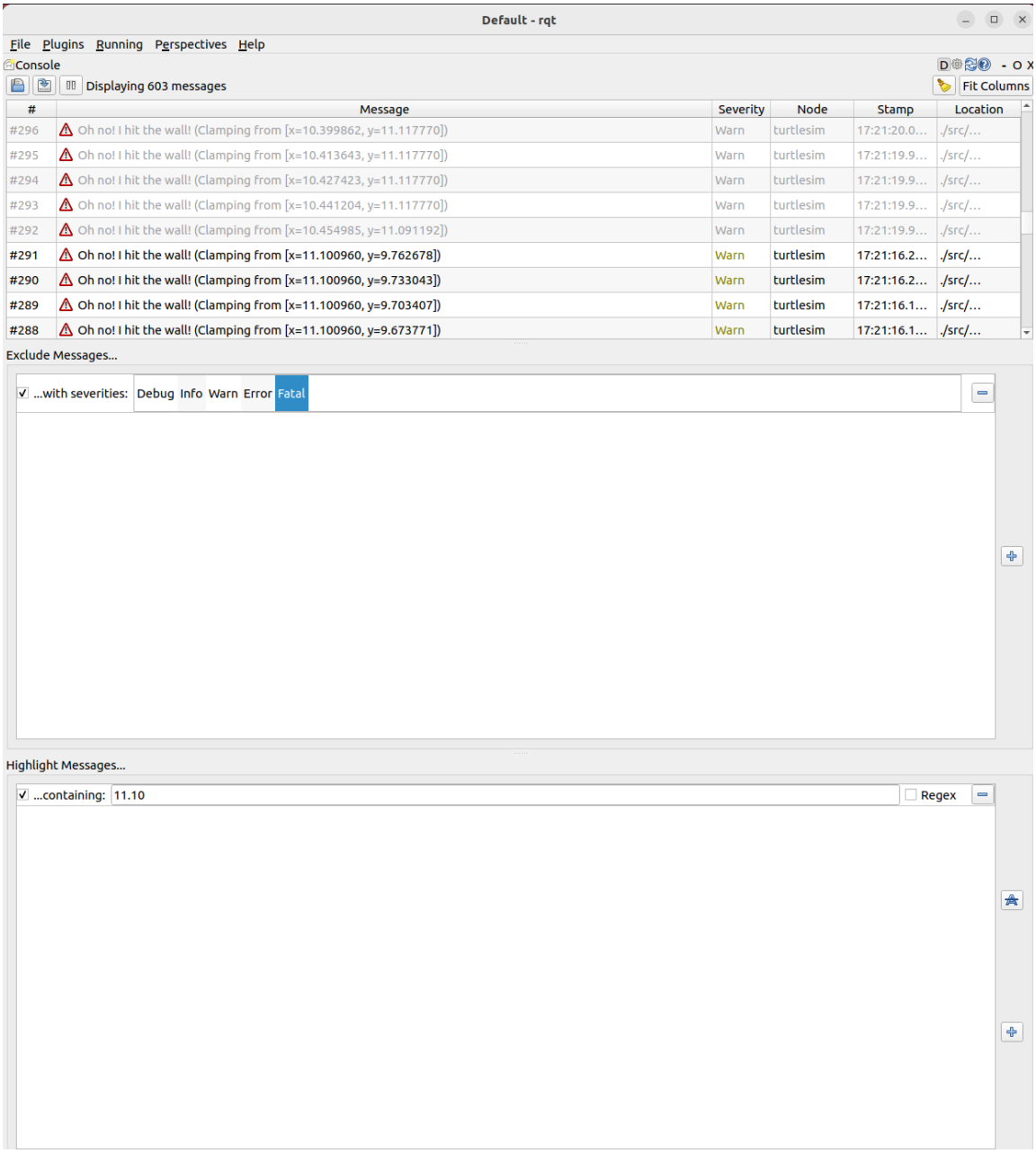
- 플러그인을 실행하고 teleop_turtle을 이용하여 거북이를 벽에 부딪혀보자



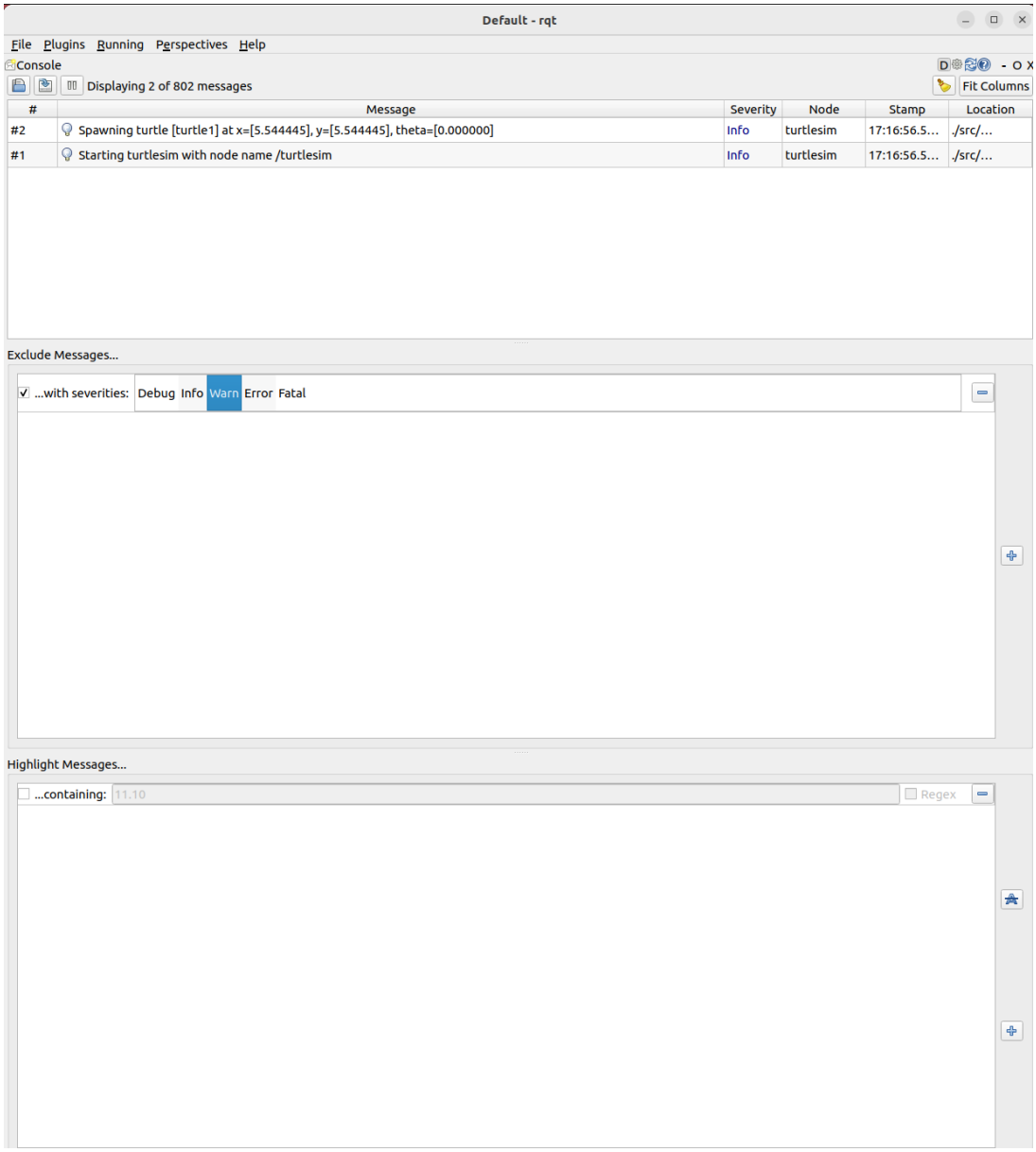
- 터미널의 명령어를 GUI상에서 확인할 수 있다.

  터미널 창과 달리 rosout메시지의 총 개수(#),메시지와 종류(Message), Severity, Node, Stamp, Location등의 정보를 확인할 수 있다.

  - Highlight Messages를 이용하여 x좌표가 11.1인 정보만 강조하였다.



- Exclude Message 의 Warn을 선택하여 Severity가 Warn인 충돌 메시지를 제외하고 볼 수 있다.



18.6 ROS의 CLI와 GUI

ROS2도구와 ros2cli를 다루었고 이번 장에서는 GUI도구인 RQt를 다루었다.

- ROS에서는 재사용성과 코드 공유에 대한 철학이 매우 중요하고
  이는 로봇의 연구 개발에 있어서 Re-inventing을 줄이는 역활을 하고 있다.

19장 ROS2의 표준 단위

19.1 ROS2 표준 단위의 필요성

- geometry_msgs/msgs/Twist메시지는
  - float64자료형의
    - linear.x
    - linear.y
    - linear.z
    - angular.x
    - angular.y
    - angular.z를 사용하고 있으며
  - 이는 병진속도 3개와 회전 속도 3개를 의미한다.
  - 이는 m/s 와 rad/s로 정의한 것이지만 cm/s, degree/s로 사용하면 잘 작동하지 않을뿐만 아니라 큰 사고로 이어질 수 있다.
- 그렇기에 ROS커뮤니티에서는 ROS개발 초기부터 단위 불일치로 인한 문제를 줄이기 위해 **표준 단위에 대한 규칙**을 세웠다.

19.2 ROS2의 표준 단위

- ROS커뮤니티에서는 국제단위계인 SI단위와
- 국제단위계의 7개 기본 단위를 조합해 만들어진 SI 유도 단위를 표준 단위로 정하였다.
- SI 단위는 7개이고
- SI 유도 단위는 20개가 있다.

- https://www.ros.org/reps/rep-0103.html 인용(ros2에도 그대로 적용되는 지는 추가로 확인해야 한다.) REP103를 따른다

Units

We have chosen to standardize on SI units. These units are the most consistent international standard. SI units are maintained by Bureau International des Poids et Mesures. [1] There is good documentation on Wikipedia for International System Of Units [2]

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

이외에도 문서에는

Chirality, Axis Orientation, Suffix Frames, Rotation Representation에 대한 내용이 있다.

20장 ROS2의 좌표 표현

20.1 ROS2 좌표 통일의 필요성

- 서로 다른 좌표 표현 방식을 사용할 때에 발생하는 좌표계 불일치(Coordinate system) 불일치를 막기 위한 방법이다.
  - 컴퓨터 비전 분야는
    - z forward
    - x right
    - y down

      을 기본 좌표계로 사용하는데

      로봇은
    - x forward
    - y left
    - z down

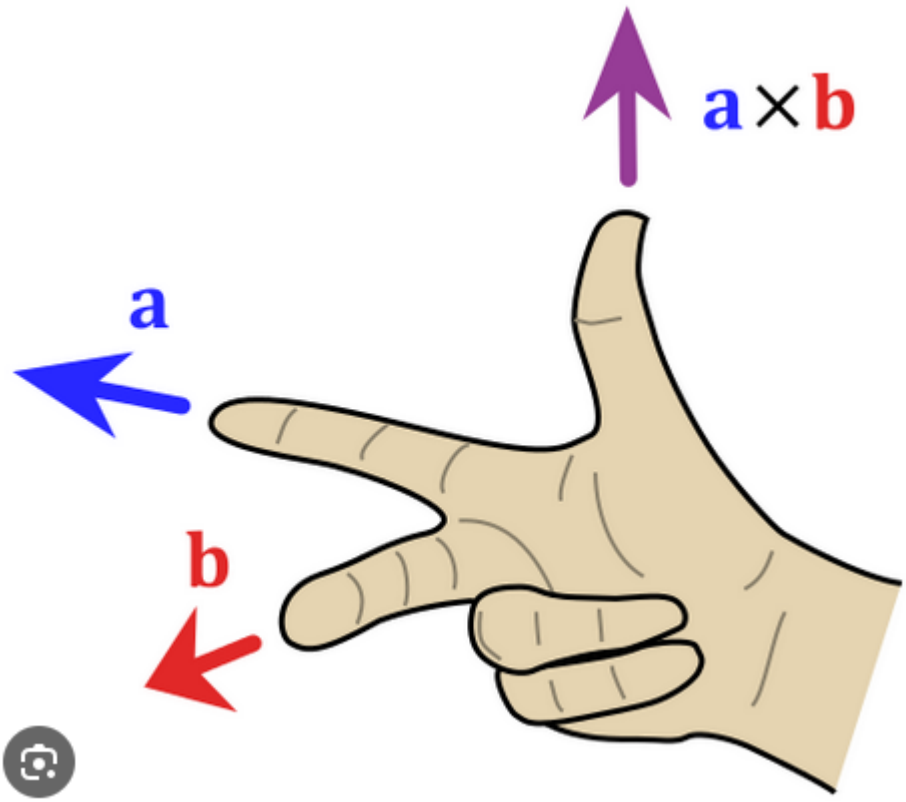
      을 기본 좌표계로 사용하여 불일치가 발생한다.

    그 외에도 LiDAR IMU Torque센서도 제조사별로 다른 좌표계를 사용하는 문제들이 있기 때문에

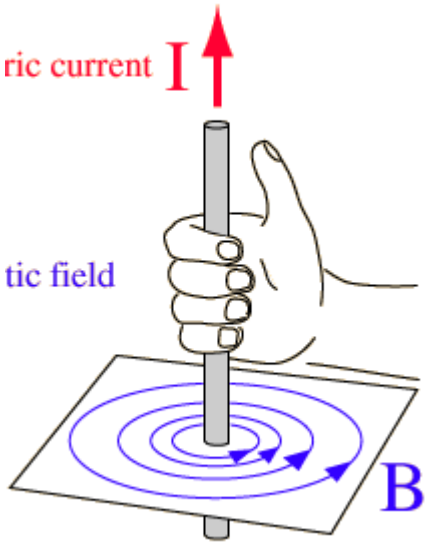
- 좌표 변환 API가 있더라도 기본 출력을 확인해야 좌표 불일치 문제를 미연에 방지할 수 있다.

20.2 좌표 표현의 기본 규칙

- ROS커뮤니티에서는 모든 좌표계를 **오른손 법칙**에 따라서 표현한다.



- 회전축은 위 그림과 같이 검지,중지, 엄지를 사용한다.



- 오른손의 손가락을 감는 반시계 방향으로 정회전 +방향을 나타낸다.

20.3 좌표 표현의 축 방향 규칙

1. 기본 3축
    a. ROS커뮤니티에서는 축 방향으로 x forward, y left, z up을 사용한다.
    b. Rviz나 Gazebo에서는 기본 3축을 RGB색으로 표현하는데
        i. Red:x축
        ii. Green : y축
        iii. Blue : z축
        을 의미한다.

2. ENU좌표
    a. 지리적 위치(Geographic location)의 단거리 데카르트 표현은 ENU(East North Up)규칙을 사용한다.

3. 접미사 프레임 사용(Suffix Frames)

a. 기본3축과 ENU좌표에서 벗어나는 경우 접미사 프레임을 사용하여 기본 좌표계와 구분한다.

b. 자주 사용되는 접미사 프레임은 _optical 접미사와 _ned접미사가 있다

   필요시 좌표변환하여 사용한다.

20.4 좌표 표현의 회전 표현(Rotation Representation)규칙

1. 쿼터니언(Quaternion)
   a. 가장 많이 사용됨(x,y,z,w)
   b. 특이점 없음
2. 회전 매트릭스(Rotation matrix)
   a. 특이점 없음
3. 고정축 roll, pitch, yaw
   a. 각 속도에 사용한다
4. 오일러각도 yaw, pitch, roll
   a. 전역 좌표계에서 회전이 발생해서 → 한 축이 다른 축의 회전과 겹치는 문제가 있으므로(짐벌락) 사용을 권장하지 않는다.