

# ROS DAY1

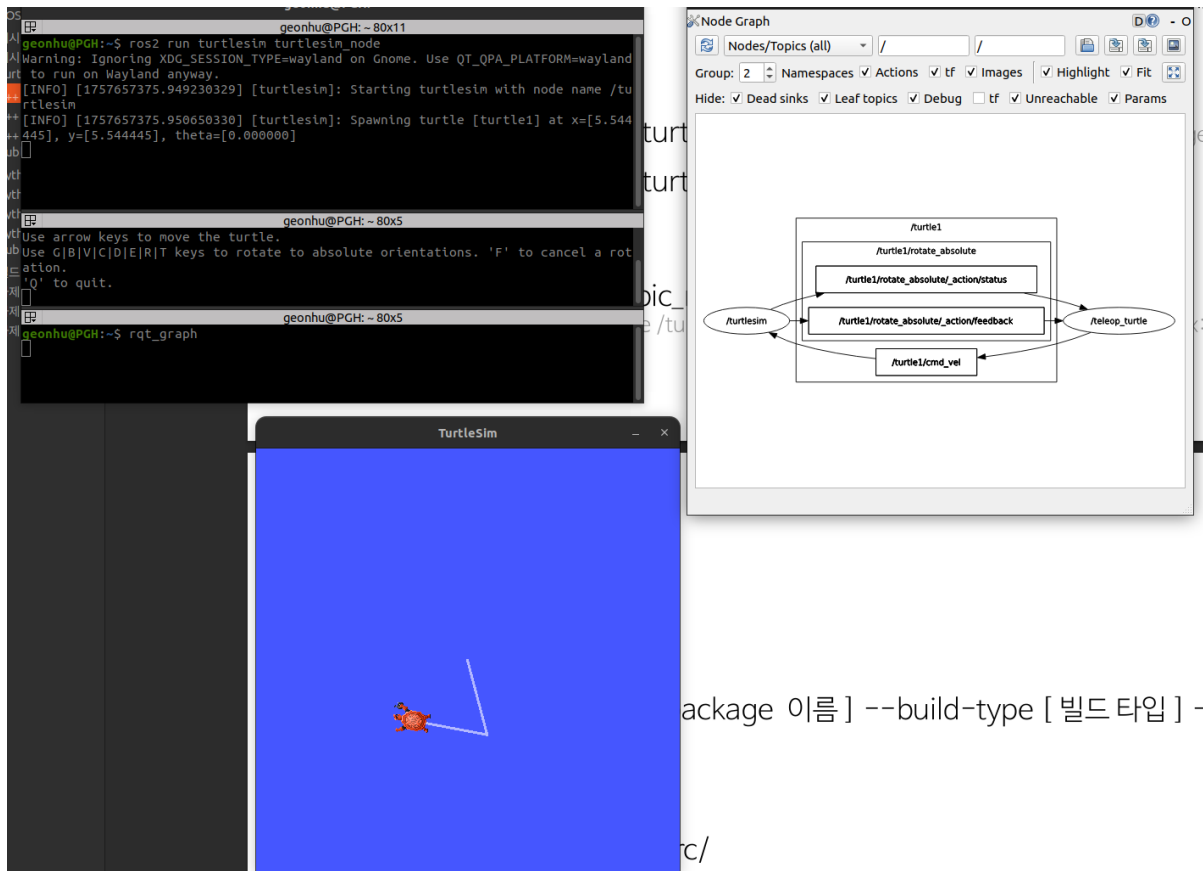
## 1번 과제

2023741024  
로봇학부 박건후

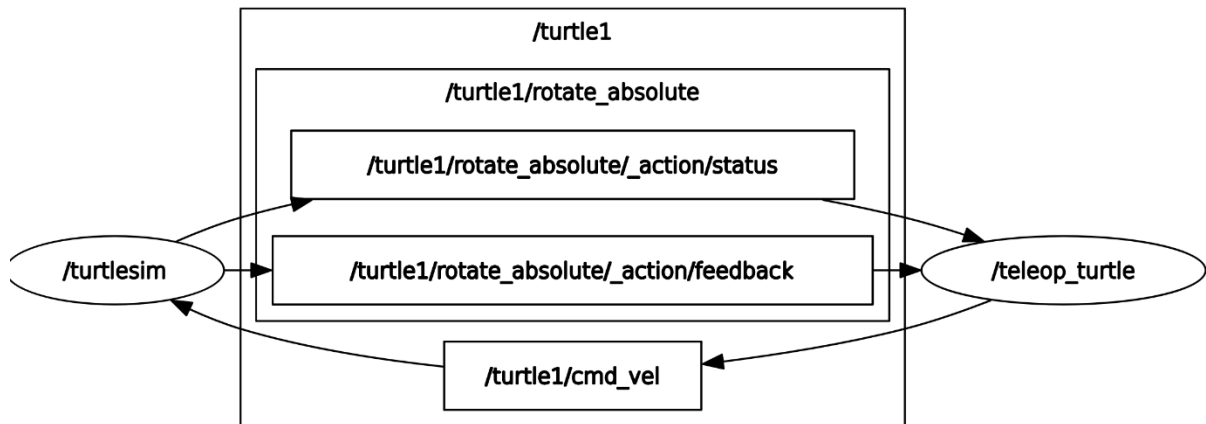
## 목차

- (1) turtlesim 실행과 토픽 발행
- (2) turtlesim 원, 정사각형, 정삼각형  
그리기

## (1) turtlesim 실행과 토픽 발행



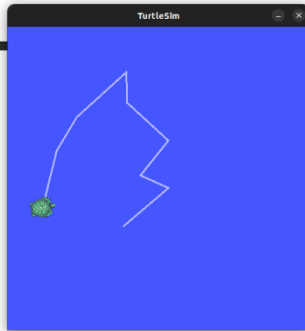
위의 사진은 turtlesim\_node, turtle\_teleop\_key 라는 2개의 노드 역할을 할 각각의 터미널을 열고 또 하나의 터미널을 열어 rqt\_graph라는 명령어를 실행시킨 모습입니다. 이렇게 터미널을 실행하여 rqt\_graph로 설명드리기 전의 모습을 보여드리기 위해 잠시 설명드렸습니다.



위에 보이시는 사진이 rqt\_graph를 확대한 부분입니다. 현재 2개의 노드끼리 토픽 통신을 하고 있다는 것을 볼 수 있습니다. 그것이 토픽 네임인 /turtle1/cmd\_vel을 통해 이루어짐을 알 수 있습니다.

위 사진에서는 \_action에 관한 터틀심의 상태, 터틀심의 feedback등이 teleop\_turtle에게 가는 모습도 보여줍니다. 여기서 주되게 이루어지는 중요한 통신은 teleop노드가 던지는 토픽입니다. teleop 노드에서 토픽을 토픽 네임을 통해 전달하여 turtlesim이 이를 받고 있는 것입니다. 그래서 해당 토픽 이름에 관련해서는 teleop쪽이 publisher, turtlesim\_node가 subscriber가 되는 것입니다.

214



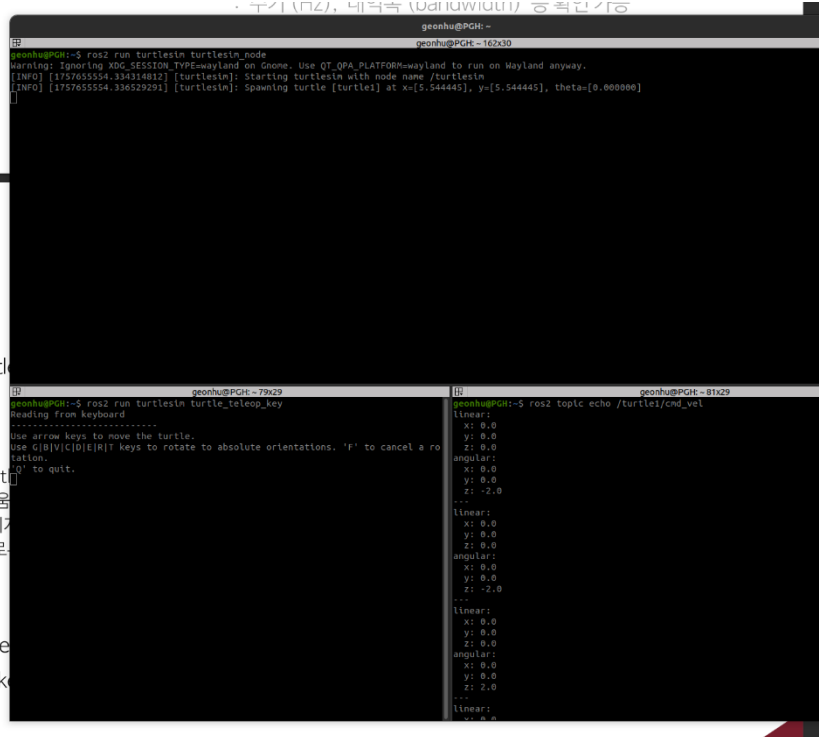
- draw\_square : 사각형 모양으로 turtle
- mimic : 유저가 지정한 토픽으로 동일 움직
- turtle\_teleop\_key : turtle 을 움직이기
- turtlesim\_node : teleop\_key 노드로

#### - 노드 실행

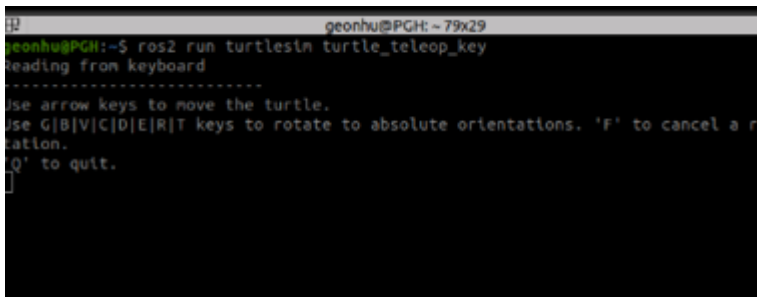
```
$ ros2 run turtlesim turtlesim_node
```

```
$ ros2 run turtlesim turtle_teleop_key
```

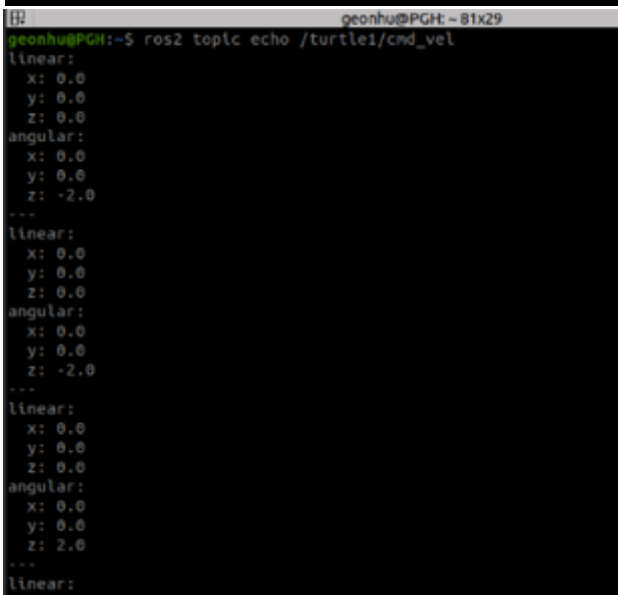
#### - 토픽 발행



위의 사진은 방향키를 통해 그림을 그리는 장면입니다. 터미널의 상부가 turtlesim\_node, 좌하단의 터미널이 teleop\_key, 우하단의 터미널이 topic echo를 한 상황입니다.



옆 사진은 좌하단의 터미널 부분으로 키보드로만 조작하고 토픽 명령어를 직접 작성하지 않은 상태입니다.



그리고 옆 사진인 이 우하단의 터미널 부분은 방향키를 통해서만 거북이를 제어하고 있고 방향키를 입력할 때마다 토픽이 전달되고 그 전달되는 토픽의 정보를 출력해주고 있는 것입니다. 그래서 옆의 사진에는 linear, angular에 관한 정보를 알려주고 있습니다. 이 점을 미루어보아, 직접 teleop\_key에 토픽 명령어를 입력해줘야 토픽이 발행되는 것이 아니라 방향키를 통해서도 토픽 발행이 된다는 것을 알았습니다.

## (2) turtlesim 원, 정사각형, 정삼각형 그리기

turtlesim으로 원, 정삼각형, 정사각형을 그려보았습니다. 이번에는 방향키가 아닌 토픽을 teleop\_key에서 직접 명령어를 작성하여 거북이를 움직이게 하였습니다. 이 때 angular의 단위는 라디안입니다. 그래서 실제 원하는 각도만큼 돌게 할 때는 원하는 각도에 원주율을 곱하고 360으로 나눈 값을 z에 넣어야 합니다.

```
geonhu@PGH: ~80x3
Previous goal finished. Aborting previous goal
[INFO] [1757656711.128822055] [turtlesim]: Rotation goal completed successfully

geonhu@PGH: ~80x36
geonhu@PGH:~$ 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.
geonhu@PGH:~$ ros2 topic pub --once /turtle1/cmd_vel geometry_msgs/msg/Twist "{l
linear: {x: 3.0, y: 0.0, z: 0.0}, angular: {x: 0.0, y: 0.0, z: 0.0}}"
publisher: beginning loop
publishing #1: geometry_msgs/msg/Twist(linear=geometry_msgs/msg/Vector3(x=3.0, y
=0.0, z=0.0), angular=geometry_msgs/msg/Vector3(x=0.0, y=0.0, z=0.0))

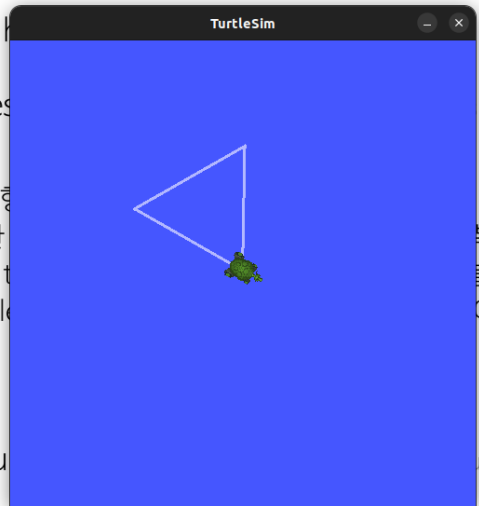
geonhu@PGH:~$
geonhu@PGH:~$
geonhu@PGH:~$ ros2 topic pub --once /turtle1/cmd_vel geometry_msgs/msg/Twist "{l
linear: {x: 0.0, y: 0.0, z: 0.0}, angular: {x: 0.0, y: 0.0, z: 2.09}}"
publisher: beginning loop
publishing #1: geometry_msgs/msg/Twist(linear=geometry_msgs/msg/Vector3(x=0.0, y
=0.0, z=0.0), angular=geometry_msgs/msg/Vector3(x=0.0, y=0.0, z=2.09))

geonhu@PGH:~$ ros2 topic pub --once /turtle1/cmd_vel geometry_msgs/msg/Twist "{l
linear: {x: 3.0, y: 0.0, z: 0.0}, angular: {x: 0.0, y: 0.0, z: 0.0}}"
publisher: beginning loop
publishing #1: geometry_msgs/msg/Twist(linear=geometry_msgs/msg/Vector3(x=3.0, y
=0.0, z=0.0), angular=geometry_msgs/msg/Vector3(x=0.0, y=0.0, z=0.0))

geonhu@PGH:~$ ros2 topic pub --once /turtle1/cmd_vel geometry_msgs/msg/Twist "{l
linear: {x: 0.0, y: 0.0, z: 0.0}, angular: {x: 0.0, y: 0.0, z: 2.09}}"
publisher: beginning loop
publishing #1: geometry_msgs/msg/Twist(linear=geometry_msgs/msg/Vector3(x=0.0, y
=0.0, z=0.0), angular=geometry_msgs/msg/Vector3(x=0.0, y=0.0, z=2.09))

geonhu@PGH:~$ ros2 topic pub --once /turtle1/cmd_vel geometry_msgs/msg/Twist "{l
linear: {x: 3.0, y: 0.0, z: 0.0}, angular: {x: 0.0, y: 0.0, z: 0.0}}"
publisher: beginning loop

geonhu@PGH:~$ ros2 run topic_echo /turtle1/cmd_vel
Package 'topic_echo' not found
geonhu@PGH:~$ ros2 run topic_echo /turtle1/cmd_vel
Package 'topic' not found
geonhu@PGH:~$ ros2 topic echo /turtle1/cmd_vel
linear:
  x: 3.0
  y: 0.0
  z: 0.0
angular:
  x: 0.0
  y: 0.0
  z: 0.0
---
linear:
  x: 0.0
  y: 0.0
  z: 0.0
angular:
  x: 0.0
  y: 0.0
  z: 2.09
---
```



topic\_name> <msg\_type> “<args>”  
/turtle1/cmd\_vel geometry\_msgs/msg/Twist "

위의 사진은 삼각형을 그린 모습입니다. 터미널 중 중간 부분 터미널에서 topic 명령어인 ros2 topic pub --once /turtle1/cmd\_vel geometry\_msgs/msg/Twist "{linear에 대한 정보, angular에 대한 정보}"라는 명령어를 통해 teleop\_key에서 토픽을 발행한 것입니다.

```

[INFO] [17576546312.616883773] [turtlesim]: Spawning turtle [turtle1] at x=[5.544445], y=[5.544445], theta=[0.000000]

geonhu@PGH:~$ ros2 topic pub --once /turtle1/cmd_vel geometry_msgs/msg/Twist "{l
linear: {x: 4.0, y: 0.0, z: 0.0}, angular: {x: 0.0, y: 0.0, z: 0.0}}"
publisher: beginning loop
publishing #1: geometry_msgs/msg/Twist(llinear=geometry_msgs.msg.Vector3(x=4.0, y
=0.0, z=0.0), angular=geometry_msgs.msg.Vector3(x=0.0, y=0.0, z=0.0))

geonhu@PGH:~$ ros2 topic pub --once /turtle1/cmd_vel geometry_msgs/msg/Twist "{l
linear: {x: 0.0, y: 0.0, z: 0.0}, angular: {x: 0.0, y: 0.0, z: 1.57}}"
publisher: beginning loop
publishing #1: geometry_msgs/msg/Twist(llinear=geometry_msgs.msg.Vector3(x=0.0, y
=0.0, z=0.0), angular=geometry_msgs.msg.Vector3(x=0.0, y=0.0, z=1.57))

geonhu@PGH:~$ ros2 topic pub --once /turtle1/cmd_vel geometry_msgs/msg/Twist "{l
linear: {x: 4.0, y: 0.0, z: 0.0}, angular: {x: 0.0, y: 0.0, z: 0.0}}"
publisher: beginning loop
publishing #1: geometry_msgs/msg/Twist(llinear=geometry_msgs.msg.Vector3(x=4.0, y
=0.0, z=0.0), angular=geometry_msgs.msg.Vector3(x=0.0, y=0.0, z=0.0))

geonhu@PGH:~$ ros2 topic pub --once /turtle1/cmd_vel geometry_msgs/msg/Twist "{l
linear: {x: 0.0, y: 0.0, z: 0.0}, angular: {x: 0.0, y: 0.0, z: 1.57}}"
publisher: beginning loop
publishing #1: geometry_msgs/msg/Twist(llinear=geometry_msgs.msg.Vector3(x=0.0, y
=0.0, z=0.0), angular=geometry_msgs.msg.Vector3(x=0.0, y=0.0, z=1.57))

geonhu@PGH:~$ ros2 topic pub --once /turtle1/cmd_vel geometry_msgs/msg/Twist "{l
linear: {x: 4.0, y: 0.0, z: 0.0}, angular: {x: 0.0, y: 0.0, z: 0.0}}"
publisher: beginning loop
publishing #1: geometry_msgs/msg/Twist(llinear=geometry_msgs.msg.Vector3(x=4.0, y
=0.0, z=0.0), angular=geometry_msgs.msg.Vector3(x=0.0, y=0.0, z=0.0))

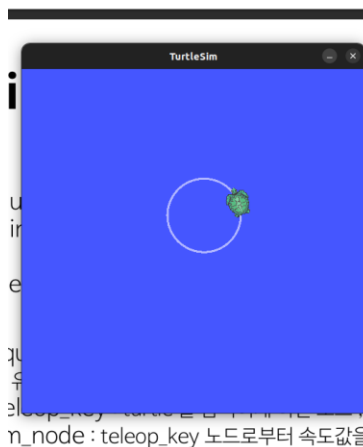
geonhu@PGH:~$ ros2 topic pub --once /turtle1/cmd_vel geometry_msgs/msg/Twist "{l
linear: {x: 0.0, y: 0.0, z: 0.0}, angular: {x: 0.0, y: 0.0, z: 1.57}}"
publisher: beginning loop
publishing #1: geometry_msgs/msg/Twist(llinear=geometry_msgs.msg.Vector3(x=0.0, y
=0.0, z=0.0), angular=geometry_msgs.msg.Vector3(x=0.0, y=0.0, z=1.57))

geonhu@PGH:~$ ros2 topic echo /turtle1/cmd_vel
linear:
  x: 4.0
  y: 0.0
  z: 0.0
angular:
  x: 0.0
  y: 0.0
  z: 0.0
---
linear:
  x: 0.0
  y: 0.0
  z: 0.0
angular:
  x: 0.0
  y: 0.0
  z: 1.57
---
linear:
  x: 4.0
  y: 0.0
  z: 0.0
angular:
  x: 0.0
  y: 0.0
  z: 0.0
---
linear:
  x: 0.0
  y: 0.0
  z: 0.0
angular:
  x: 0.0
  y: 0.0
  z: 1.57
---
linear:
  x: 4.0
  y: 0.0
  z: 0.0
angular:
  x: 0.0
  y: 0.0
  z: 0.0

```



“aras”  
msgs/msg/Twist "{linear: {x: 2.0, y: 0.0, z: 0.0}, angular: {x: 0.0, y: 0.0, z: 1.8}}"



turtlesim turtlesim\_node  
turtlesim turtle\_teleop\_key

c pub (topic name) (msg type) “aras”

c pub --once /turtle1/cmd\_vel geometry\_msgs/msg/Twist "{linear: {x: 2.0, y: 0.0, z: 0.0}, angular: {x: 0.0, y: 0.0, z: 1.8}}"

이렇게 네모, 원들 또한 그려보았습니다. 감사합니다.

```

geonhu@PGH:~$ ros2 run turtlesim turtlesim_node
Warning: Ignoring XDG_SESSION_PATH=/home/wayland on GNOME. Use QT_PLATFORM=wayland to run on Wayland anyway.
[INFO] [1757655788.91023426] [turtlesim]: Starting turtlesim with node name /turtlesim
[INFO] [1757655788.912225051] [turtlesim]: Spawning turtle [turtle1] at x=[5.544445], y=[5.544445], theta=[0.000000]

geonhu@PGH:~$ ros2 run turtlesim turtle_teleop_key
Heading from keyboard
Use arrow keys to move the turtle.
Use G|B|V|C|D|E|R|I keys to rotate to absolute orientations. 'F' to cancel a rotation.
'Q' to quit.

geonhu@PGH:~$ ros2 topic pub --rate /turtle1/cmd_vel geometry_msgs/msg/Twist "{linear: {x: 2.0, y: 0.0, z: 0.0}, angular: {x:
0.0, y: 0.0, z: 1.8}}}"
usage: ros2 topic pub [-h] [-r N]
  [-p N]
  [-t 1 | -t TIMES]
  [-w WAIT_MATCHING_SUBSCRIPTIONS]
  [--keep-alive N]
  [-n NODE_NAME]
  [--qos-profile {unknown,system_default,sensor_data,services_default,parameters,parameter_events,action_s
tatus_default}]
  [--qos-depth N]
  [--qos-history {system_default,keep_last,keep_all,unknown}]
  [--qos-reliability {system_default,reliable,best_effort,unknown}]
  [--qos-durability {system_default,transient_local,volatile,unknown}]
  [--spin-time SPIN_TIME]
  [-s]
  topic_name
  message_type
  [values]

ros2 topic pub: error: argument -r/--rate: invalid positive_float value: '/turtle1/cmd_vel'

geonhu@PGH:~$ ros2 topic pub --rate 1 /turtle1/cmd_vel geometry_msgs/msg/Twist "{linear: {x: 2.0, y: 0.0, z: 0.0}, angular: {x:
0.0, y: 0.0, z: 1.8}}}"
publisher: beginning loop
publishing #1: geometry_msgs/msg/Twist(llinear=geometry_msgs.msg.Vector3(x=2.0, y=0.0, z=0.0), angular=geometry_msgs.msg.Vector
3(x=0.0, y=0.0, z=1.8))

publishing #2: geometry_msgs/msg/Twist(llinear=geometry_msgs.msg.Vector3(x=2.0, y=0.0, z=0.0), angular=geometry_msgs.msg.Vector
3(x=0.0, y=0.0, z=1.8))

publishing #3: geometry_msgs/msg/Twist(llinear=geometry_msgs.msg.Vector3(x=2.0, y=0.0, z=0.0), angular=geometry_msgs.msg.Vector
3(x=0.0, y=0.0, z=1.8))

publishing #4: geometry_msgs/msg/Twist(llinear=geometry_msgs.msg.Vector3(x=2.0, y=0.0, z=0.0), angular=geometry_msgs.msg.Vector
3(x=0.0, y=0.0, z=1.8))

geonhu@PGH:~$ ros2 topic echo /turtle1/cmd_vel
linear:
  x: 2.0
  y: 0.0
  z: 0.0
angular:
  x: 0.0
  y: 0.0
  z: 1.8
---
linear:
  x: 2.0
  y: 0.0
  z: 0.0
angular:
  x: 0.0
  y: 0.0
  z: 1.8

```