

SMC-2000 & SMC+

ROS 설정 설명서

SYNEREX, Inc.

목 차

1	개요	3
2	제한사항	3
3	SMC-2000 & SMC+ 설정파일 적용	3
3.1	u-center 다운로드 및 설정파일 다운로드	3
3.2	설정 파일 적용	3
4	SMC-2000 단독구성	5
4.1	전원 및 통신 포트 연결	5
4.2	연결 확인	5
5	SMC-2000 & SMC+ 듀얼 GNSS 수신기 구성	6
5.1	SMC-2000 전원 및 통신 포트 연결	6
5.2	SMC+ 전원 및 통신포트 연결	6
5.3	연결 확인	7
5.4	GPS Heading 사용시 주의사항	7
6	ROS/ROS2에서 SMC-2000 & SMC+ 사용 방법	9
6.1	ROS-ublox 패키지 설치	9
6.2	패키지 설정 파일 및 실행 파일 수정(ROS1: noetic, melodic 등)	9
6.3	패키지 설정 파일 및 실행 파일 수정 (ROS2: foxy, humble)	13
6.4	.yaml 파일 옵션 변경	13
7	ROS2 Humble 에서 SMC-2000 & SMC+ 사용 방법	13
7.1	패키지 설치	13
7.2	패키지 수정, 빌드	13
7.3	패키지 실행	17
7.4	RQT를 통한 Topic 확인	17
8	SMC2000 & SMC+ 사용하여 Heading 확인	17

1 개요

SMC-2000 & SMC+를 ROS와 연동하는 방법에 대해서 기술한 문서입니다.

현재 ROS에서 SMC-2000 & SMC+의 GPS Heading 정보를 사용하기 위해서는 Moving base, Rover 로 각각 설정하여 사용해야 합니다.

2 제한사항

이 문서의 설정 방법은 아래 버전에서 테스트한 방법입니다.

- Ubuntu version : ROS(noetic, melodic), ROS2(foxy, humble)
- Ubuntu version : 22.04, ROS2 Humble

해당 버전이 아닌 경우에는 설정 방법이 상이할 수 있습니다.

RTCM 서버를 추가적으로 활용하실 경우 <https://github.com/KumarRobotics/ublox.git> 의 소스를 직접 colcon build를 활용(7장)하시는 것을 권장 드립니다. 현재 ros에서 제공하는 ublox 패키지의 경우 ros로 rtcn 노드를 만들어 보정 정보를 직접 넣어주었을 때 적용이 되지 않는 문제점이 있습니다. 추가적인 RTCM 서버를 활용하시는 경우가 아니라면 각 ros 버전별 ublox 패키지를 활용하셔도 smc-2000을 활용하는 것에 문제는 없습니다.

3 SMC-2000 & SMC+ 설정파일 적용

3.1 u-center 다운로드 및 설정파일 다운로드

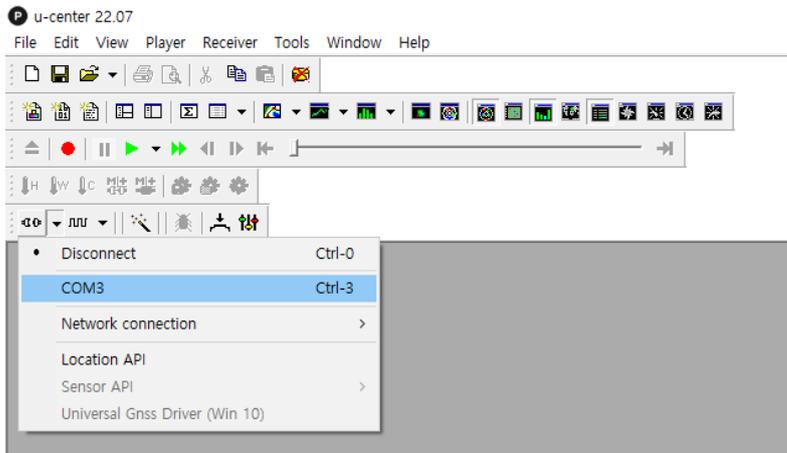
u-center 다운로드: <https://www.u-blox.com/en/product/u-center>

설정파일 다운로드: <https://synerex.tistory.com/4>

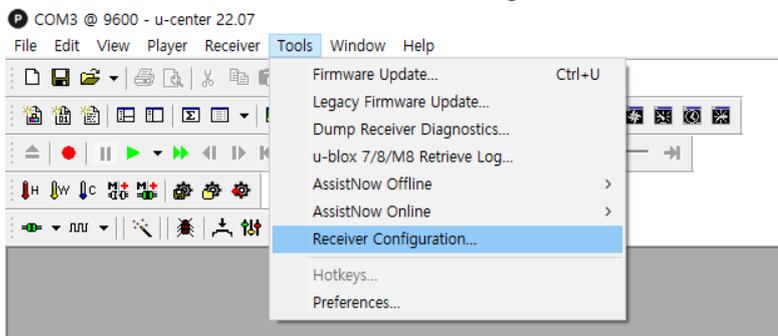
3.2 설정 파일 적용

- ① SMC-2000(or SMC+)이 동작 중일 때, SMC-2000 상부의 GPS USB 와 PC 를 연결하십시오.
SMC+는 동봉된 USB 연결 케이블을 사용해 연결해 주십시오.

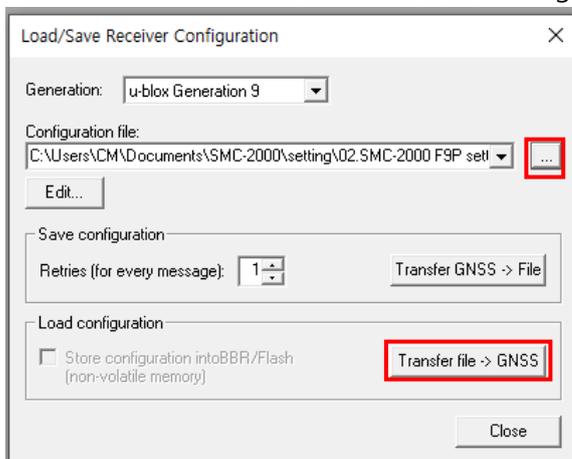
- ② 2. U-center 실행 후 인식된 포트로 연결하십시오.



- ③ 3. 상단 메뉴바 > Tools > Receiver Configuration 을 실행하여 주십시오.

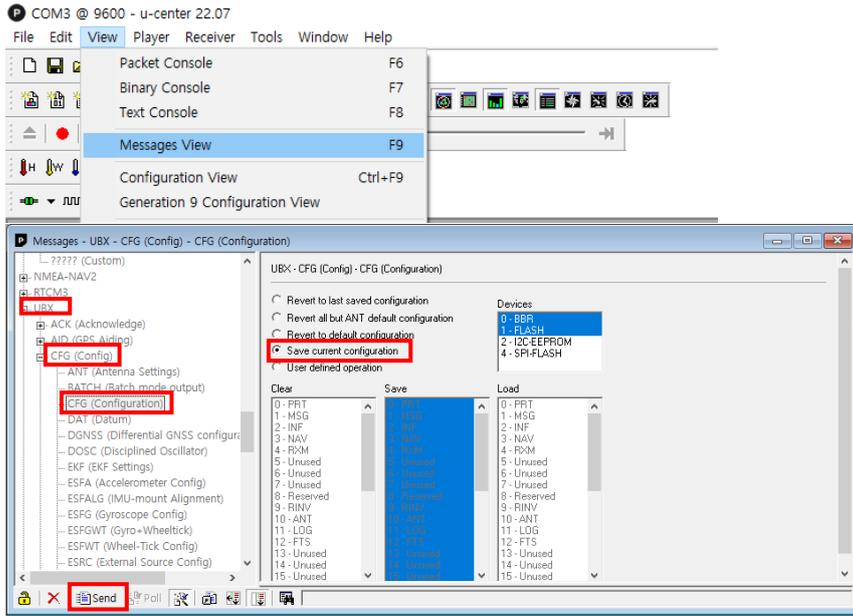


- ④ 4. Configuration file 의 ...버튼을 클릭하여 용도에 적합한 Configuration file 을 선택한 후 Transfer GNSS -> File 버튼을 클릭하여 Configuration file 을 적용하여 주십시오.



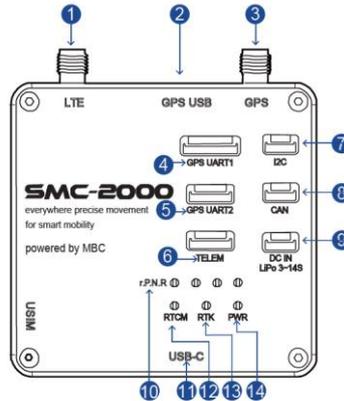
SMC-2000 에는 05.MB_SMC-2000_F9P setting_FW1.30_5Hz_5_BASE.txt 파일을 적용,
SMC+에는 06.MB_SMC+PLS_F9P setting_FW1.30_5Hz_ROVER.txt 파일을 적용해 주십시오.

- ⑤ 5. 상단 메뉴바 > View > Messages View 를 실행 한 후 UBX>CFG>CFG 항목에서 Save current configuration 을 선택 한 후 왼쪽 하단의 Send 버튼을 클릭하여 현재값을 저장해 주십시오.



4 SMC-2000 단독구성

4.1 전원 및 통신 포트 연결



SMC-2000을 단독으로 연결시에는 아래 포트를 기본적으로 연결을 해야 합니다.

LTE 안테나 포트(①)

- LTE Antenna 연결

GPS 안테나 포트(③)

- GPS Antenna 연결

GPS USB 포트(②)

- 사용할 장치와 연결

4.2 연결 확인

Ubuntu 에 SMC2000 의 GPS USB 로 연결을 했을 때 Ubuntu 에서 장치 이름은 일반적으로 `"/dev/ttyACM0"`로 인식이 되고 이 장치명은 시스템에 따라서 상이할 수 있습니다.

Ubuntu 에서 터미널을 열고 입력 메시지 확인 명령 실행합니다. Permission denied 라는 메시지가 출력되면 sudo 를 추가해서 명령을 실행하거나 장치 접근권한 변경 명령을 실행한 뒤에 입력 메시지 명령을 실행하면 SMC2000 에서 출력되는 NMEA 메시지를 확인할 수 있습니다..

- 입력 메시지 확인 명령 : cat /dev/ttyACM0
- 장치 접근권한 변경 : sudo chmod a+rw /dev/ttyACM0

```

synerex@synerex-ThinkPad-E14-Gen-3:~$ cat /dev/ttyACM0
$GNRMC,0.0,0.0,A,3733.63615,N,12659.60725,E,0.0,28.0,070323.0000,R,V*05
$GNVTG,T,M,0.028,N,0.052,K,D*35
$GNGGA,0.0,0.0,3733.63615,N,12659.60725,E,4,12,0.56,76.9,M,18.5,M,21.0,0020*61
$GNGSA,A,3,08,16,04,18,26,07,27,,,,,1.04,0.56,0.87,1*0C
$GNGSA,A,3,68,87,86,76,77,88,,,,,1.04,0.56,0.87,2*05
$GNGSA,A,3,02,07,08,11,30,34,36,25,,,,,1.04,0.56,0.87,3*01
$GNGSA,A,3,06,16,44,26,24,40,39,09,,,,,1.04,0.56,0.87,4*09
$GNGSA,A,3,02,03,07,,,,,,1.04,0.56,0.87,5*0A
$GPGSV,3,1,12,04,32,236,35,07,25,314,42,08,58,233,47,09,29,277,19,1*6D
$GPGSV,3,2,12,16,59,042,46,18,16,044,25,21,09,182,39,26,31,082,41,1*62
$GPGSV,3,3,12,27,84,027,47,31,07,138,39,41,27,238,26,50,47,180,42,1*69
$GPGSV,2,1,08,04,32,236,35,07,25,314,36,08,58,233,47,09,29,277,15,6*6F
$GPGSV,2,2,08,18,16,044,29,26,31,082,41,27,84,027,47,31,07,138,36,6*62
$GLGSV,2,1,08,67,13,264,22,68,13,313,43,76,49,079,52,77,55,347,40,1*7B
$GLGSV,2,2,08,85,06,025,23,86,45,047,52,87,57,160,52,88,13,190,45,1*70
$GLGSV,2,1,08,67,13,264,17,68,13,313,34,75,02,112,30,76,49,079,47,3*7E
$GLGSV,2,2,08,77,55,347,41,86,45,047,47,87,57,160,07,88,13,190,43,3*78
$GLGSV,1,1,01,78,08,307,,0*4B
$GAGSV,2,1,08,02,62,045,49,07,27,286,25,08,26,228,44,11,18,041,31,2*74
$GAGSV,2,2,08,25,20,086,44,30,45,303,50,34,31,158,47,36,45,091,49,2*79
$GAGSV,2,1,08,02,62,045,47,07,27,286,27,08,26,228,40,11,18,041,30,7*78
$GAGSV,2,2,08,25,20,086,37,30,45,303,45,34,31,158,45,36,45,091,45,7*72

```

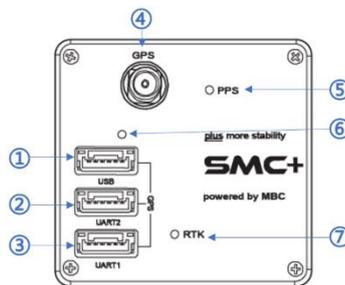
NMEA 메시지가 출력이 안되는 경우에는 시스템의 인식된 SMC2000 의 장치명을 다시 확인하고 시도해야 합니다.

5 SMC-2000 & SMC+ 듀얼 GNSS 수신기 구성

5.1 SMC-2000 전원 및 통신 포트 연결

“4.1 전원 및 통신 포트 연결” 절에 있는 SMC-2000 단독 구성시의 연결과 동일하게 진행합니다.

5.2 SMC+ 전원 및 통신포트 연결



SMC+ 의 전원 및 기본 통신 포트 연결은 아래와 같은 포트를 연결합니다.

GPS 안테나 포트(④)

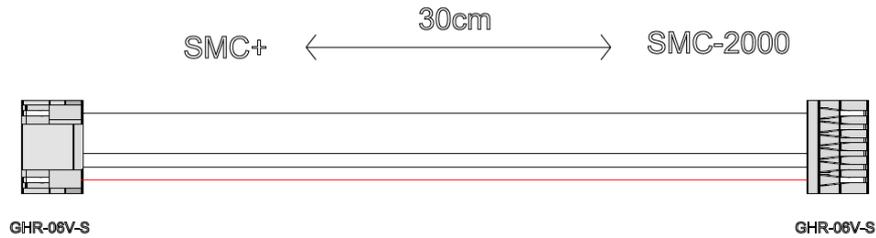
- GPS Antenna 연결

GPS USB 포트(①)

- 사용할 장치와 연결
- 동봉된 USB연결 케이블을 사용해 연결

GPS UART2 포트(②)

- SMC-2000과 SMC+ 연결
- SMC-2000의 GPS UART2 포트와 SMC+의 GPS UART2 포트는 아래 케이블을 사용



5.3 연결 확인

“4.2 전원 및 통신 포트 연결” 절에 있는 SMC-2000 단독 구성시의 확인과 동일하게 진행합니다. SMC+의 연결 확인 시 장치명은 시스템에 인식된 SMC+의 장치명을 이용해서 진행해야 합니다.

5.4 GPS Heading 사용시 주의사항

두 안테나 사이의 거리가 너무 가까우면 Heading값이 출력되지 않을 수 있습니다. (아래 그래프 참고)

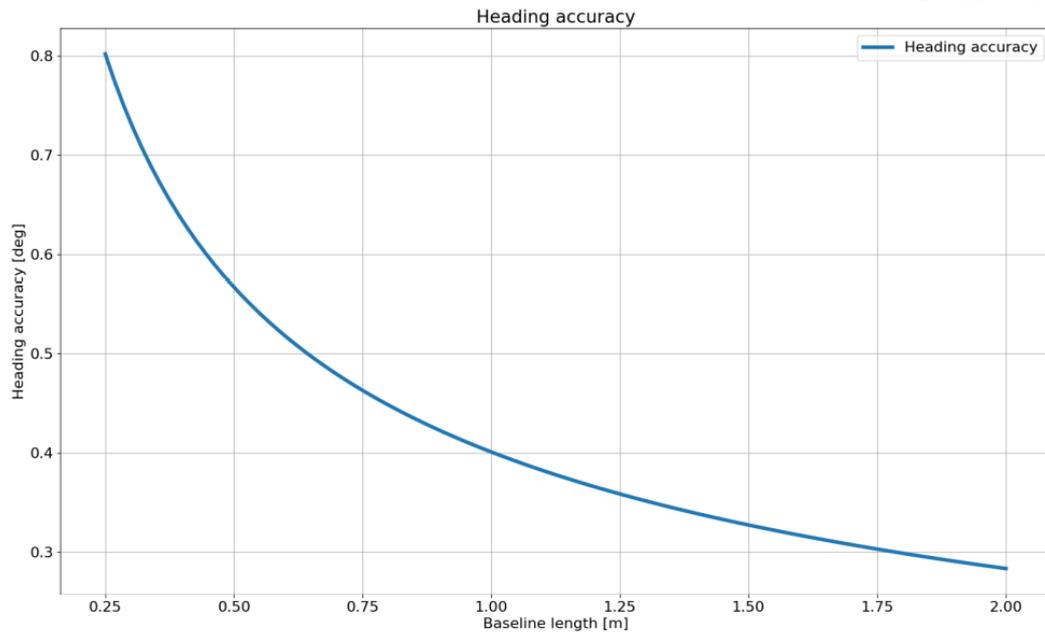


Figure 1: ZED-F9P-04B moving base RTK heading accuracy versus baseline length

6 ROS/ROS2에서 SMC-2000 & SMC+ 사용 방법

6.1 ROS-ublox 패키지 설치

ROS에서 제공하는 ublox 패키지를 설치합니다.

```
$ apt install ros-<version>-ublox
```

정상적으로 설치되었을 경우 다음과 같이 ublox 패키지가 설치됩니다.

```
$ ls /opt/ros/<version>/include | grep ublox
```

```
ublox
ublox_gps
ublox_msgs
```

6.2 패키지 설정 파일 및 실행 파일 수정(ROS1: noetic, melodic 등)

/opt/ros/<version>/share/ublox_gps의 하위 폴더 config와 launch에 다음과 같은 파일을 추가합니다.

launch/ublox_f9p_base.launch

```
<?xml version="1.0" encoding="UTF-8"?>

<launch>
  <arg name="node_name"          value="smc" />
  <arg name="param_file_name"    value="zed_f9p_base" />
  <arg name="param_file_dir"     default="$(find ublox_gps)/config" />
  <arg name="output"            default="screen" />
  <arg name="respawn"           default="true" />
  <arg name="respawn_delay"     default="30" />
  <arg name="clear_params"      default="true" />

  <node pkg="ublox_gps" type="ublox_gps" name="$(arg node_name)"
        output="$(arg output)"
        clear_params="$(arg clear_params)"
        respawn="$(arg respawn)"
        respawn_delay="$(arg respawn_delay)">
    <rosparam command="load"
          file="$(arg param_file_dir)/$(arg param_file_name).yaml" />
  </node>
```

launch/ublox_f9p_rover.launch

```
<?xml version="1.0" encoding="UTF-8"?>

<launch>
  <arg name="node_name"          value="smc_plus" />
  <arg name="param_file_name"    value="zed_f9p_rover" />
  <arg name="param_file_dir"     default="$(find ublox_gps)/config" />
  <arg name="output"             default="screen" />
  <arg name="respawn"            default="true" />
  <arg name="respawn_delay"      default="30" />
  <arg name="clear_params"       default="true" />

  <node pkg="ublox_gps" type="ublox_gps" name="$(arg node_name)"
        output="$(arg output)"
        clear_params="$(arg clear_params)"
        respawn="$(arg respawn)"
        respawn_delay="$(arg respawn_delay)">
    <rosparam command="load"
          file="$(arg param_file_dir)/$(arg param_file_name).yaml" />
  </node>
```

config/zed_f9p_base.yaml

```
debug: 0
device: /dev/<device> # smc device port
frame_id: gps
uart1:
  baudrate: 115200 # config buadrate
  in: 35
  out: 35

rate: 5.0 # config hz

tmode3: 0

inf:
  all: true

publish: #원하시는 topic 명
  all: false
  aid:
    hui: false
  nav:
    posecef: false

gnss:      # default = gps only
  sbas: True
  galileo: True
  beidou: True
  qzss: True
  glonass: True
```

config/zed_f9p_rover.yaml

```
debug: 0
device: /dev/<device> # smc_plus device name
frame_id: gps
uart1:
  baudrate: 115200 # config buadrate
  in: 35
  out: 35

rate: 5.0 # config hz

tmode3: 0

inf:
  all: true

publish: #원하시는 topic 명
  all: false
  aid:
    hui: false
  nav:
    posecef: false
    relposned: true

gnss:      # default = gps only
  sbas: True
  galileo: True
  beidou: True
  qzss: True
  glonass: True
```

다음의 명령어를 통해 패키지 및 SMC-2000이 정상적으로 동작하는 것을 확인하실 수 있습니다.

```
roslaunch ublox_gps ublox_f9p_base.launch
```

```
roslaunch ublox_gps ublox_f9p_rover.launch
```

6.3 패키지 설정 파일 및 실행 파일 수정 (ROS2: foxy, humble)

각 설정 파일의 저장 경로는 ROS1과 동일합니다.(첨부)

ublox_f9p_base_launch.py

ublox_f9p_rover_launch.py

zed_f9p_base.yaml(디바이스 명 확인 및 변경)

zed_f9p_rover.yaml(디바이스 명 확인 및 변경)

다음의 명령어를 통해 패키지 및 SMC-2000이 정상적으로 동작하는 것을 확인하실 수 있습니다.

```
ros2 launch ublox_gps ublox_f9p_base_launch.py
```

```
ros2 launch ublox_gps ublox_f9p_rover_launch.py
```

6.4 .yaml 파일 옵션 변경

변경 가능 옵션 확인: <https://github.com/KumarRobotics/ublox.git>

7 ROS2 Humble 에서 SMC-2000 & SMC+ 사용 방법

7.1 패키지 설치

[KumarRobotics/ublox](https://github.com/KumarRobotics/ublox) 패키지의 ROS2 branch와 RTCM 패키지를 설치합니다.

```
$ git clone -b ros2 --single-branch https://github.com/KumarRobotics/ublox.git
```

```
$ git clone https://github.com/tilk/rtcm\_msgs
```

7.2 패키지 수정, 빌드

이후 해당 패키지를 설치한 워크스페이스 하단의 ublox 폴더에 설정 파일과 launch파일을 생성합니다.

ublox/ublox_gps/config 폴더에 zed_f9p_base.yaml, zed_f9p_rover.yaml 파일을 각각 생성합니다.

zed_f9p_base.yaml과 zed_f9p_rover.yaml 파일을 다음과 같이 수정합니다.

```
ublox_gps_node:
  ros__parameters:
    debug: 0
    device: /dev/<device> # smc device name
    frame_id: gps
    uart1:
      baudrate: 115200
      in: 35
      out: 35
    rate: 5.0
    # TMODE3 Config
    tmode3: 0 # Survey-In Mode

    inf:
      all: true # Whether to display all INF messages in console

    publish:
      all: false
      aid:
        hui: false
      nav:
        posecef: false

    gnss:
      sbas: True
      galileo: True
      beidou: True
      qzss: True
      glonass: True
```

```
ublox_gps_node:  
  ros__parameters:  
    debug: 0  
    device: /dev/<device> # smc_plus device name  
    frame_id: gps  
    uart1:  
      baudrate: 115200  
      in: 35  
      out: 35  
    rate: 5.0  
    # TMODE3 Config  
    tmode3: 0  
    inf:  
      all: true  
    publish:  
      all: false  
      aid:  
        hui: false  
      nav:  
        posecef: false  
        relposned: True  
    gnss:  
      sbas: True  
      galileo: True  
      beidou: True  
      qzss: True  
      glonass: True
```

ublox/ublox_gps/launch 폴더에 있는 ublox_gps_node-launch.py파일을 복사해 ublox_gps_node_base-launch.py, ublox_gps_node_rover-launch.py 파일을 각각 생성합니다.

ublox_gps_node_base-launch.py 파일을 다음과 같이 수정합니다.

```

35 import os
36
37 import ament_index_python.packages
38 import launch
39 import launch_ros.actions
40
41
42 def generate_launch_description():
43     config_directory = os.path.join(
44         ament_index_python.packages.get_package_share_directory('ublox_gps'),
45         'config')
46     params = os.path.join(config_directory, 'zed_f9p_base.yaml')
47     ublox_gps_node = launch_ros.actions.Node(package='ublox_gps',
48                                             executable='ublox_gps_node',
49                                             output='both',
50                                             parameters=[params],
51                                             remappings=[
52                                                 ('/ublox_gps_node/fix', '/smc_2000/fix'),
53                                                 ('/rtcm', 'smc_2000/rtcm'),
54                                                 ('/ublox_gps_node/fix_velocity', '/smc_2000/fix_velocity')
55                                             ]
56                                             )
57
58     return launch.LaunchDescription([ublox_gps_node,
59
60                                     launch.actions.RegisterEventHandler(
61                                         event_handler=launch.event_handlers.OnProcessExit(
62                                             target_action=ublox_gps_node,
63                                             on_exit=[launch.actions.EmitEvent(
64                                                 event=launch.events.Shutdown())],
65                                             )),
66                                     ])

```

ublox_gps_node_rover-launch.py 파일을 다음과 같이 수정합니다.

```

33 """Launch the ublox_gps node with c94-m8p configuration."""
34
35 import os
36
37 import ament_index_python.packages
38 import launch
39 import launch_ros.actions
40
41
42 def generate_launch_description():
43     config_directory = os.path.join(
44         ament_index_python.packages.get_package_share_directory('ublox_gps'),
45         'config')
46     params = os.path.join(config_directory, 'zed_f9p_rover.yaml')
47     ublox_gps_node = launch_ros.actions.Node(package='ublox_gps',
48                                             executable='ublox_gps_node',
49                                             output='both',
50                                             parameters=[params],
51                                             remappings=[
52                                                 ('/ublox_gps_node/fix', '/smc_plus/fix'),
53                                                 ('/rtcm', 'smc_plus/rtcm'),
54                                                 ('/ublox_gps_node/fix_velocity', '/smc_plus/fix_velocity'),
55                                                 ('/navrelposned', '/smc_plus/relposned')
56                                             ]
57                                             )
58
59     return launch.LaunchDescription([ublox_gps_node,
60
61                                     launch.actions.RegisterEventHandler(
62                                         event_handler=launch.event_handlers.OnProcessExit(
63                                             target_action=ublox_gps_node,
64                                             on_exit=[launch.actions.EmitEvent(
65                                                 event=launch.events.Shutdown())],
66                                             )),
67                                     ])

```

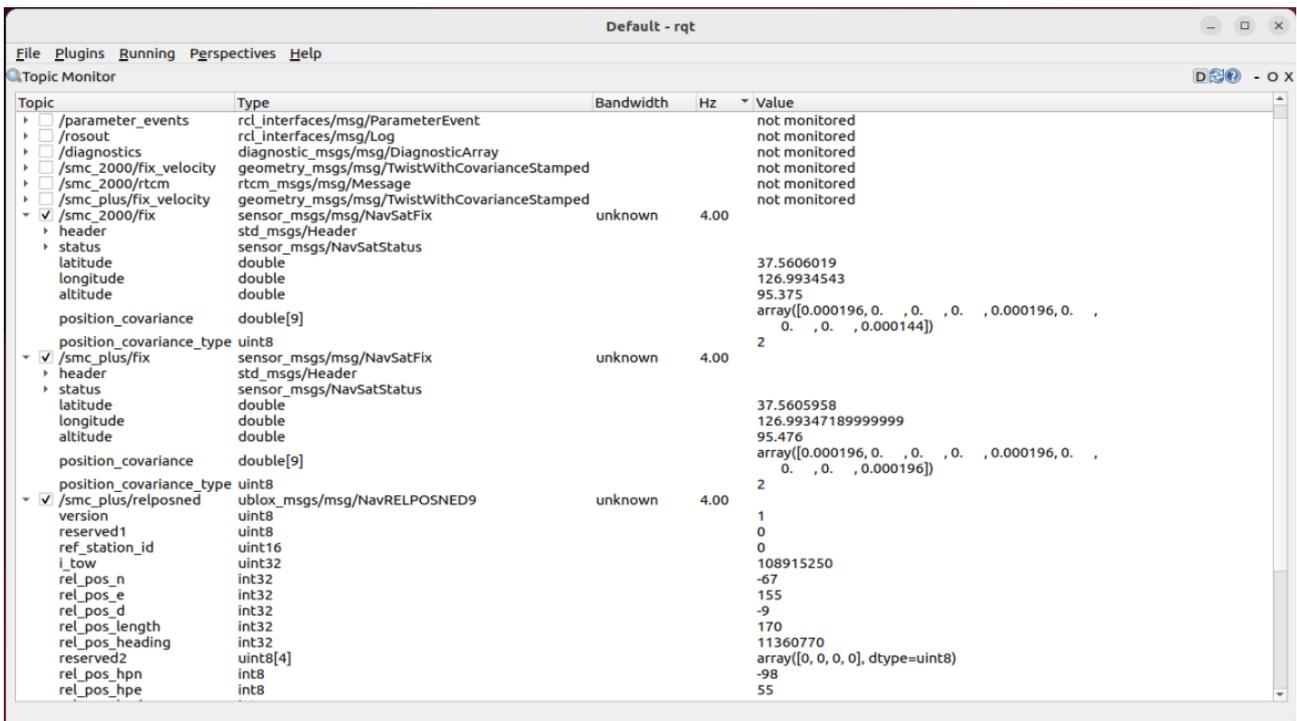
수정 후 \$ colcon build로 패키지를 빌드합니다.

7.3 패키지 실행

패키지 빌드 후 다음 명령어로 launch파일을 각각 실행시킵니다.

```
$ Ros2 launch ublox_gps ublox_gps_node_base-launch.py
$ Ros2 launch ublox_gps ublox_gps_node_rover-launch.py
```

7.4 RQT를 통한 Topic 확인



8 SMC2000 & SMC+ 사용하여 Heading 확인

Heading 값은 SMC+에서 출력이 되고 출력되는 데이터 UBX메시지 “UBX-NAV-RELPOSNED” 항목에서 확인하면 됩니다. 관련한 메시지에 정보는 아래 문서를 확인하면 됩니다.

https://www.u-blox.com/sites/default/files/documents/u-blox-F9-HPG-1.32_InterfaceDescription_UBX-22008968.pdf

그리고 “6절/7절”에서 Heading 값 확인은 “/smc_plus/relposned/rel_pos_heading” topic을 확인하면 됩니다.

