

TETRA-DS-V ROS 매뉴얼



CONTENTS

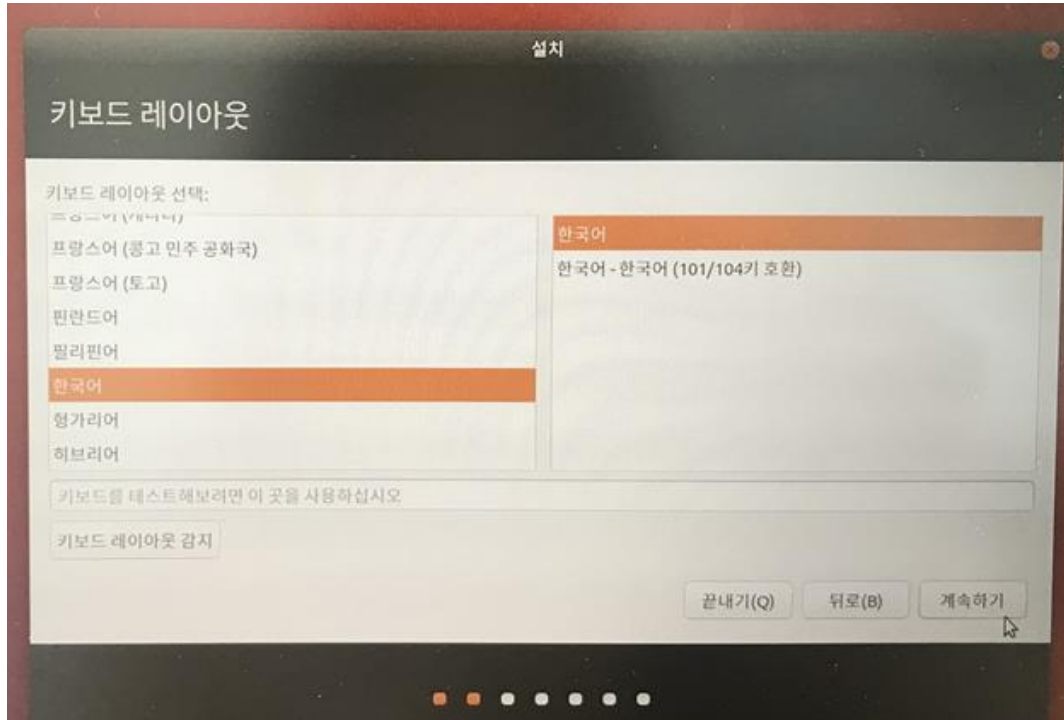
Chapter 1. Software Set up	3
1-0. PC Setup	3
1-1. ROS Melodic 설치	5
1-2. Realsense SDK 설치 (SDK 2.42.0)	7
1-3. Realsense ROS 설치 (v 2.2.22)	9
1-4. Cartographer 설치	12
1-5. ROS Package 설치	14
1-6. USBRule 설정	15
1-6-1 usb cam(AR marker) rule 설정	15
1-6-2 IMU 설정	17
Chapter 2. SLAM	19
2-1. Node 실행	19
2-2. JoyStick 조작	19
2-3. LandMark 등록	20
2-4. 지도 저장	21
Chapter 3. Navigation	23
3-1. Node 실행	23
3-2. 위치 이동	24
3-2-1 2D Nav Goal	24
3-2-2 Joystick 사용	24
3-2-3 서비스 호출	25
3-3. 위치 저장	26
3-4. 위치 보정	28
3-4-1 2D Pose Estimate	28
3-4-2 서비스 호출	28
Chapter 4. ROS	30
4-1. TETRA-DSV ROS Service	30
4-2. TETRA-DSV ROS TOPIC	35

Chapter 1. Software Set up

1-0. PC Setup

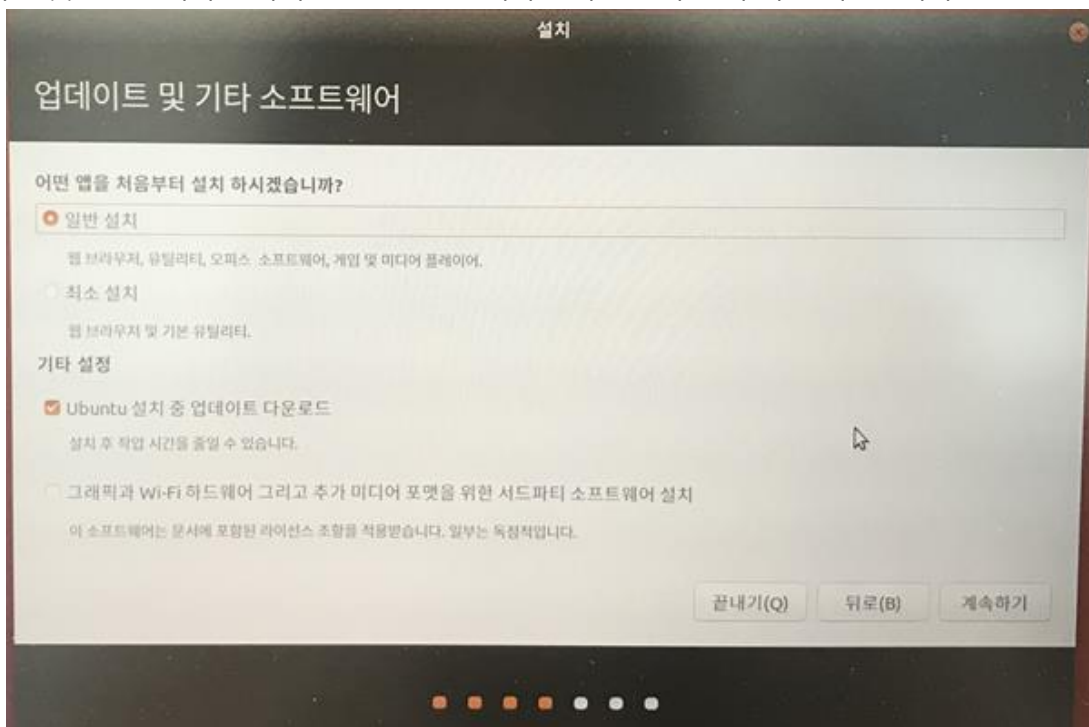
1) PC 설치

- USB 부팅 후 첫 화면이 뜨면 키보드 레이아웃 세팅을 요구합니다.아래 그림과 같이 세팅합니다.



2) 업데이트 및 기타 소프트웨어 설정

- 업데이트 및 소프트웨어 설치에 관련된 설정입니다.밑의 그림과 같이 기본 세팅합니다.

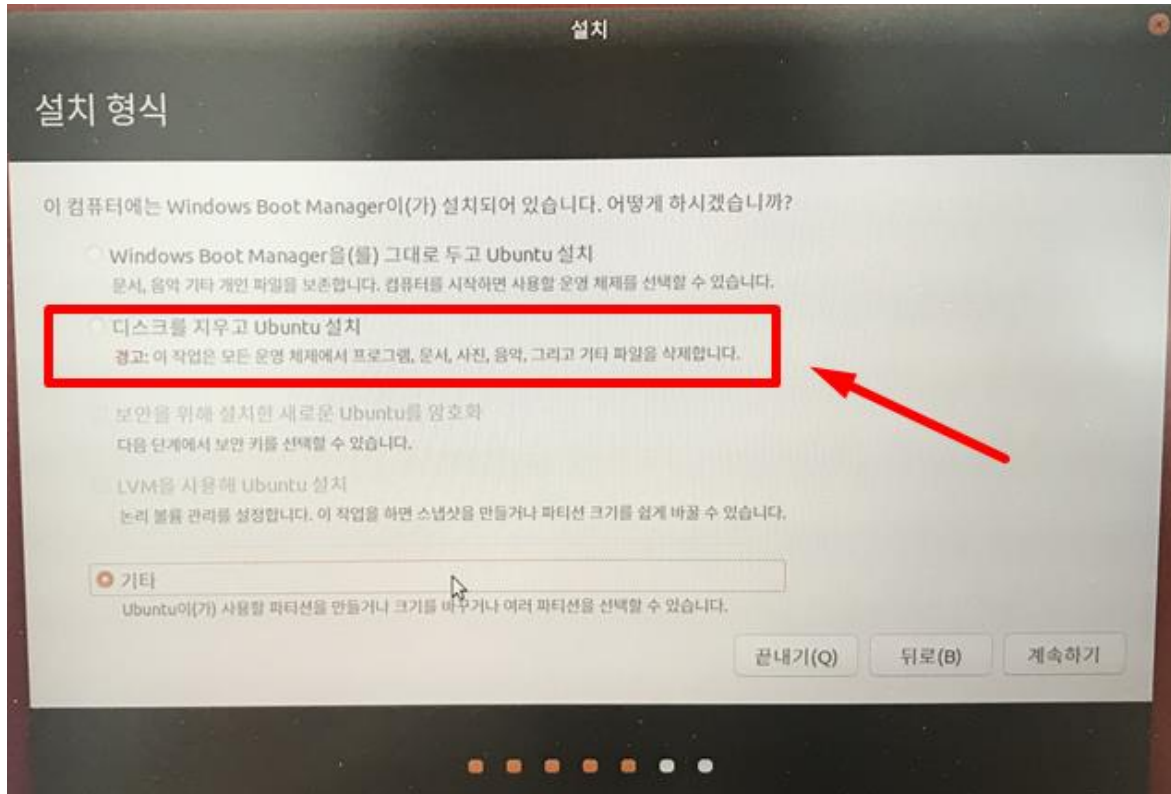


3)설치 형식 설정 및 Ubuntu 설치

- Ubuntu 18.04를 설치하기 전 마지막 단계입니다. 밑의 그림의 사각형을 보며 주의하여 디스크를 지우고 Ubuntu 를 설치합니다. (여기에서의 디스크는 H/D 를 말하므로 USB 의 디스크는 지워지지 않습니다.)

***PC 의 사양과 환경에 따라 설치 시간은 차이가 있을 수 있으며 시간이 꽤 소요되니 참고하세요.**

Ubuntu 가 설치되는 과정 중에는절대로 PC 를 끄거나 강제 종료하지 마십시오.



1-1. ROS Melodic 설치

1) source.list 설정

- package.ros.org 로부터 소프트웨어를 받아들이 수 있도록 PC 를 세팅 합니다.

```
task@task:~$ sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release -sc) main"
> /etc/apt/sources.list.d/ros-latest.list'
```

```
$ sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release -sc) main" > /etc/a
pt/sources.list.d/ros-latest.list'
```

2) keys 설정

```
task@task:~$ sudo apt-key adv --keyserver 'hkp://keyserver.ubuntu.com:80' --recv-key C1CF6E31E6
BADE8868B172B4F42ED6FBAB17C654
```

```
$ sudo apt-key adv --keyserver 'hkp://keyserver.ubuntu.com:80' --recv-key C1CF6E31E6BADE
8868B172B4F42ED6FBAB17C654
```

* keyserver 와의 연결 문제가 발생한다면 위의 명령어에서 " 안의 내용을 hkp:pgp.mit.edu:80 혹은 hkp://keyserver.ubuntu.com:80 으로 바꿔보시길 바랍니다.

3) melodic install

- Debian 패키지 목록을 최신화 합니다.

```
task@task:~$ sudo apt update
```

```
$ sudo apt update
```

- Desktop-full 설치를 합니다.

```
task@task:~$ sudo apt install ros-melodic-desktop
```

```
$sudo apt install ros-melodic-desktop-full
```

4) 환경변수 설정

```
task@task:~$ echo "source /opt/ros/melodic/setup.bash" >> ~/.bashrc
task@task:~$ source ~/.bashrc
```

```
$ echo "source /opt/ros/melodic/setup.bash" >> ~/.bashrc
$ source ~/.bashrc
```

5) 의존성 패키지 빌드 설치

```
task@task:~$ sudo apt install python-rosdep python-rosinstall python-rosinstall-
generator python-wstool build-essential
```

```
$ sudo apt install python-rosdep python-rosinstall python-rosinstall-generator python-wstool
build-essential
```

-의존성 패키지 빌드 설치 후 터미널 창에 roscore 를 실행하여 정상적으로 설치되었는지 확인

```
task@task:~$ roscore
... logging to /home/task/.ros/log/e07c8f10-df90-11eb-84fe-80ee73f1c046/roslaunch-task-27304.log
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://task:45415/
ros_comm version 1.14.11

SUMMARY
=====

PARAMETERS
* /rostdistro: melodic
* /rosversion: 1.14.11

NODES
auto-starting new master
process[master]: started with pid [27315]
ROS_MASTER_URI=http://task:11311/

setting /run_id to e07c8f10-df90-11eb-84fe-80ee73f1c046
process[roscout-1]: started with pid [27326]
started core service [/roscout]
```

6) rosdep 초기화

```
task@task:~$ sudo apt install python-rosdep
Reading package lists... Done
Building dependency tree
Reading state information... Done
python-rosdep is already the newest version (0.21.0-1).
0 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.
task@task:~$ sudo rosdep init
Wrote /etc/ros/rosdep/sources.list.d/20-default.list
Recommended: please run

    rosdep update

task@task:~$ rosdep update
reading in sources list data from /etc/ros/rosdep/sources.list.d
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/osx-homebrew.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/base.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/python.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/ruby.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/releases/fuerte.yaml
Query rosdistro index https://raw.githubusercontent.com/ros/rosdistro/master/index-v4.yaml
Skip end-of-life distro "ardent"
Skip end-of-life distro "bouncy"
Skip end-of-life distro "crystal"
Skip end-of-life distro "dashing"
Skip end-of-life distro "eloquent"
Add distro "foxy"
Add distro "galactic"
Skip end-of-life distro "groovy"
Skip end-of-life distro "hydro"
Skip end-of-life distro "indigo"
Skip end-of-life distro "jade"
Skip end-of-life distro "kinetic"
Skip end-of-life distro "lunar"
Add distro "melodic"
Add distro "noetic"
Add distro "rolling"
updated cache in /home/task/.ros/rosdep/sources.cache
```

```
$sudo apt install python-rosdep
$sudorosdep init
$ rosdep update
```

1-2.Realsense SDK 설치 (SDK 2.42.0)

1) source code 다운

<https://github.com/IntelRealSense/librealsense/releases/tag/v2.42.0> 사이트에 접속하여

'Source code(tar.gz)'를 다운받고 압축을 해제합니다.

압축 해제 후 압축을 푼 폴더로 이동 후 빌드를 하기 위한 폴더를 생성합니다.

```
task@task:~$ cd librealsense-2.42.0/
task@task:~/librealsense-2.42.0$ mkdir build
task@task:~/librealsense-2.42.0$ cd build
task@task:~/librealsense-2.42.0/build$
```

```
$cd librealsense-2.42.0
$mkdir build
$cd build
```

2) Download lib

```
task@task:~/librealsense-2.42.0/build$ sudo apt-get update
Hit:1 http://packages.ros.org/ros/ubuntu bionic InRelease
Hit:2 http://security.ubuntu.com/ubuntu bionic-security InRelease
Hit:3 http://kr.archive.ubuntu.com/ubuntu bionic InRelease
Hit:4 http://kr.archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:5 http://kr.archive.ubuntu.com/ubuntu bionic-backports InRelease
Reading package lists... Done
task@task:~/librealsense-2.42.0/build$ sudo apt-get install python3 python3-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
python3 is already the newest version (3.6.7-1~18.04).
python3 set to manually installed.
python3-dev is already the newest version (3.6.7-1~18.04).
python3-dev set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.
task@task:~/librealsense-2.42.0/build$ sudo apt install libgtk-3-dev libxcursor-dev libxinerama-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
libxcursor-dev is already the newest version (1:1.1.15-1).
libxcursor-dev set to manually installed.
libxinerama-dev is already the newest version (2:1.1.3-1).
libxinerama-dev set to manually installed.
The following additional packages will be installed:
  libatk-bridge2.0-dev libatspi2.0-dev libdbus-1-dev libepoxy-dev libxkbcommon-dev
  libxtst-dev wayland-protocols x11proto-record-dev
Suggested packages:
  libgtk-3-doc
The following NEW packages will be installed:
  libatk-bridge2.0-dev libatspi2.0-dev libdbus-1-dev libepoxy-dev libgtk-3-dev
  libxkbcommon-dev libxtst-dev wayland-protocols x11proto-record-dev
0 upgraded, 9 newly installed, 0 to remove and 2 not upgraded.
Need to get 1,460 kB of archives.
After this operation, 15.6 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
```



```
$ sudo apt-get update
$ sudo apt-get install python3 python3-dev
$ sudo apt install libgtk-3-dev libxcursor-dev libxinerama-dev
```

3) Make and Build

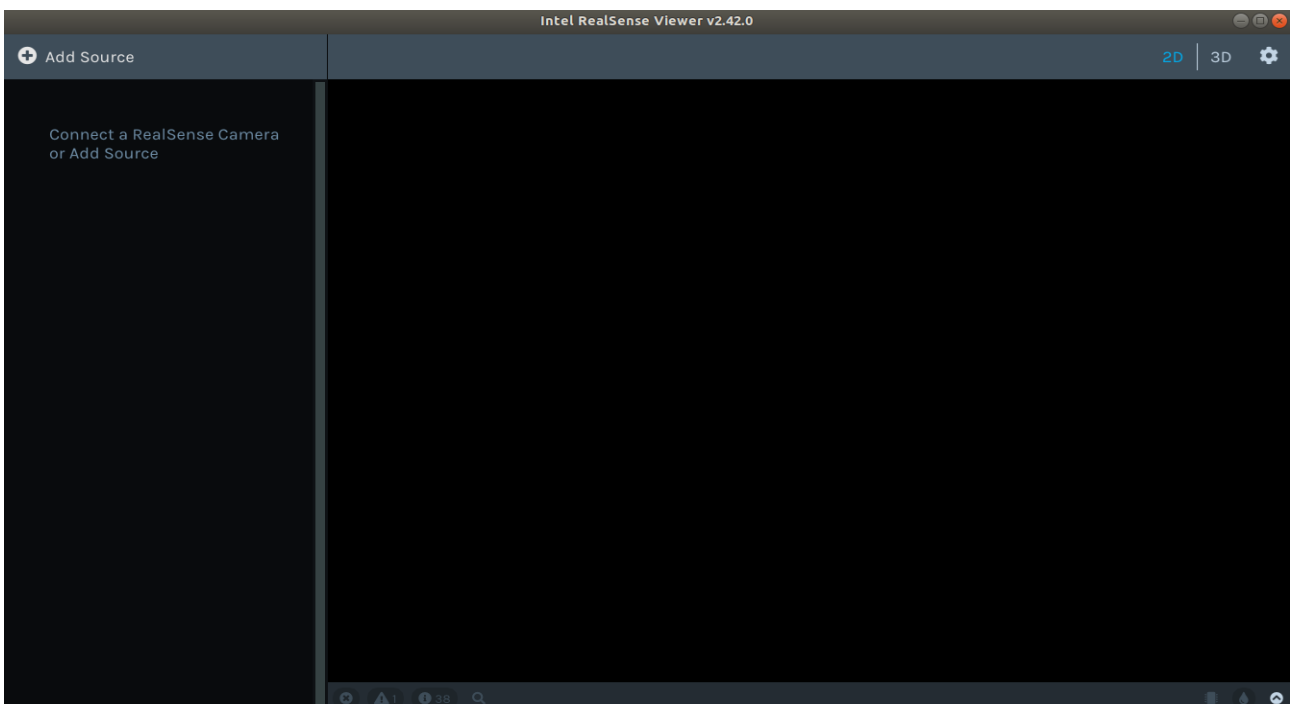
```
$ cmake ..
$ make -j4
$ sudo make install
$ export PYTHONPATH=$PYTHONPATH:/usr/local/lib
```

4) launch

-빌드 완료 후 realsense-viewer 를 실행하여 정상적으로 설치 되었는지 확인합니다.

```
task@task:~/librealsense-2.42.0/build$ realsense-viewer
```

```
$ realsense-viewer
```



5) rule 설정

```
task@task:~$ cd librealsense-2.42.0/
task@task:~/librealsense-2.42.0$ cd config/
task@task:~/librealsense-2.42.0/config$ sudo cp 99-realsense-libusb.rules /etc/udev/rules.d/
```



```
$ cd librealsense-2.42.0
$ cd config
$ sudo cp 99-realsense-libusb.rules /etc/udev/rules.d
$ sudo service udev restart
```

1-3.Realsense ROS 설치 (v 2.2.22)

1) realsense2 camera 설치

```
task@task:~$ export ROS_VER=melodic
task@task:~$ sudo apt-get install ros-$ROS_VER-realsense2-camera
[sudo] password for task:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  ros-melodic-ddynamic-reconfigure ros-melodic-librealsense2
The following NEW packages will be installed:
  ros-melodic-ddynamic-reconfigure ros-melodic-librealsense2
  ros-melodic-realsense2-camera
0 upgraded, 3 newly installed, 0 to remove and 2 not upgraded.
Need to get 13.1 MB of archives.
After this operation, 36.5 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
```

```
$ export ROS_VER=melodic
$ sudo apt-get install ros-$ROS_VER-realsense2-camera
```

2) realsense ros download

catkin_ws 폴더가 없으면 mkdir catkin_ws 명령어로 폴더를 생성합니다.

```
$ mkdir catkin_ws
$ cd catkin_ws
$ mkdir src
$ git clone https://github.com/IntelRealSense/realsense-ros.git
$ cd realsense-ros/
$ git checkout `git tag | sort -V | grep -P "^2.\d+\.\d+" | tail -1`
$ cd ..
```

3) Make

```
task@task:~$ cd catkin_ws/
task@task:~/catkin_ws$ catkin_make
```

```
$cd catkin_ws
$ catkin_make
```

4) package 적용

```
$echo "source ~/catkin_ws/devel/setup.bash" >> ~/.bashrc
$source ~/.bashrc
```

5) realsense_ros 확인

realsense ros 패키지가설치되었는지확인하기위해터미널창을연후다음과같이입력합니다.

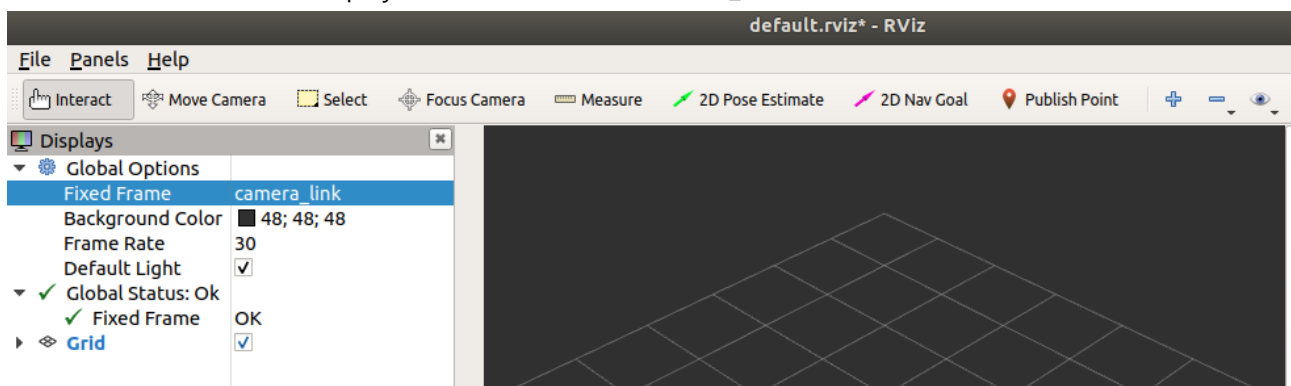
```
task@task:~$ roslaunch realsense2_camera rs_camera.launch

$ roslaunch realsense2_camera rs_camera.launch
```

터미널창을추가로실행한후 rviz 를입력합니다.

```
task@task:~$ rviz
[ INFO] [1625728372.216886193]: rviz version 1.13.17
[ INFO] [1625728372.216936272]: compiled against Qt version 5.9.5
[ INFO] [1625728372.216943572]: compiled against OGRE version 1.9.0 (Ghadamon)
[ INFO] [1625728372.220708926]: Forcing OpenGL version 0.
[ INFO] [1625728372.336498295]: Stereo is NOT SUPPORTED
[ INFO] [1625728372.336537627]: OpenGL device: Mesa DRI Intel(R) UHD Graphics 630 (CFL GT2)
[ INFO] [1625728372.336563607]: OpenGL version: 3.0 (GLSL 1.3).
```

rviz 프로그램이실행되면 Displays 의 Fixed Frame 을 camera_link 로선택합니다.



Add 를클릭한후 By topic 의 Image 를선택한후 OK 를클릭합니다.

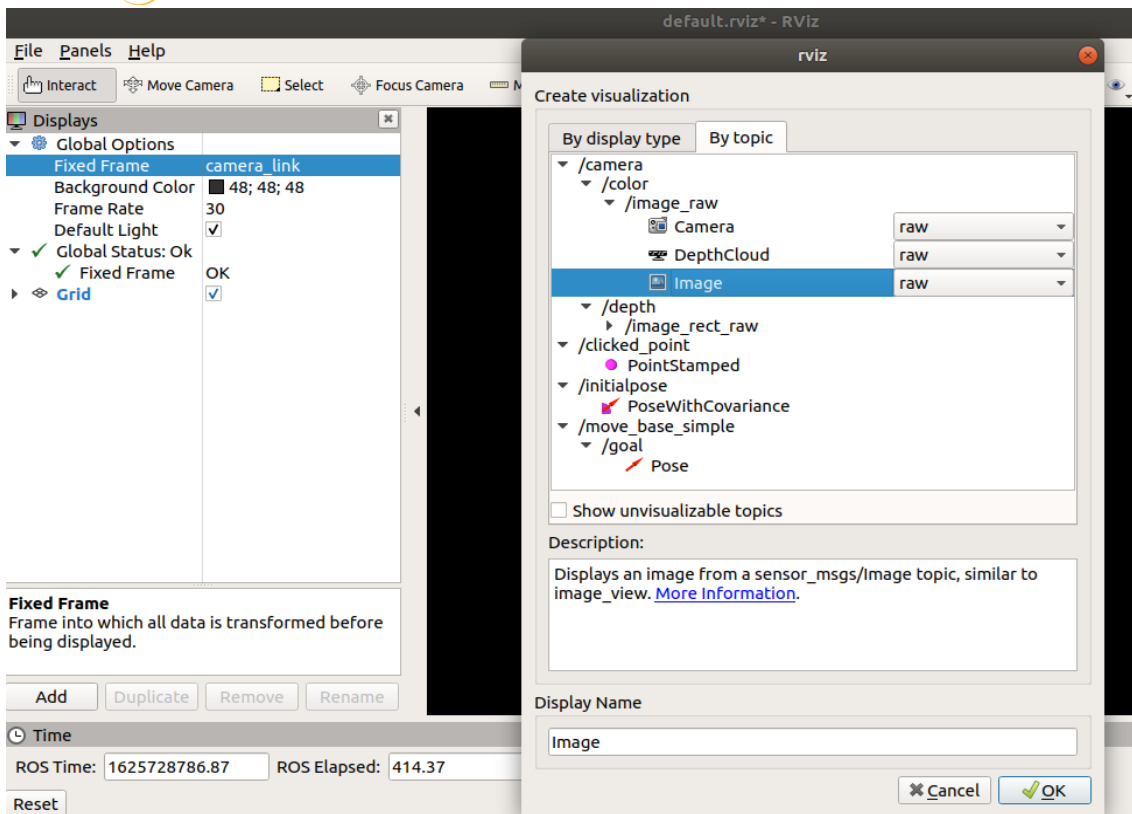
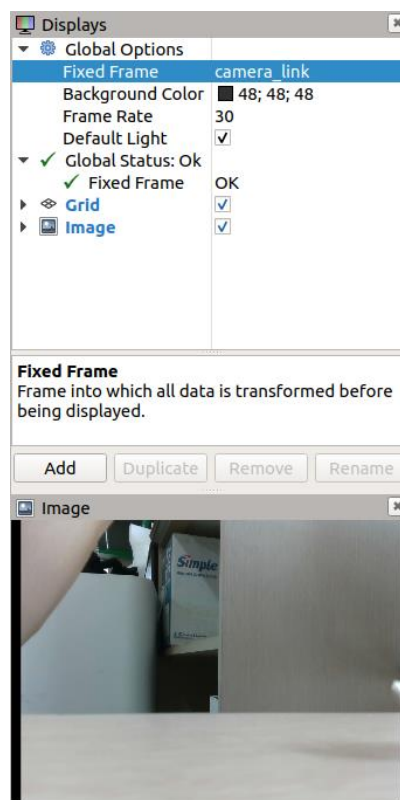


Image 화면이정상로나오는지확인합니다.



1-4.Cartographer 설치

1) ninja-build 패키지 설치

-패키지 목록 최신화 후 패키지 설치를 진행합니다.

```
task@task:~$ sudo apt-get install -y python-wstool python-rosdep ninja-build stow
$ sudo apt-get update
$ sudo apt-get install -y python-wstool python-rosdep ninja-build stow
```

2) cartographer 작업공간 생성

```
task@task:~$ mkdir cartographer_ws
task@task:~$ cd cartographer_ws/
task@task:~/cartographer_ws$ wstool init src
Writing /home/task/cartographer_ws/src/.rosinstall

update complete.
task@task:~/cartographer_ws$ wstool merge -t src https://raw.githubusercontent.com/cartographer-project/cartographer_ros/master/cartographer_ros.rosinstall
Performing actions:

  Add new elements:
  cartographer, cartographer_ros

Config changed, maybe you need run wstool update to update SCM entries.
Overwriting /home/task/cartographer_ws/src/.rosinstall

update complete.
task@task:~/cartographer_ws$ wstool update -t src
[cartographer] Fetching https://github.com/cartographer-project/cartographer.git (version master) to /home/task/cartographer_ws/src/cartographer
Cloning into '/home/task/cartographer_ws/src/cartographer'...
remote: Enumerating objects: 14658, done.
remote: Counting objects: 100% (12/12), done.
remote: Compressing objects: 100% (12/12), done.
remote: Total 14658 (delta 1), reused 8 (delta 0), pack-reused 14646
Receiving objects: 100% (14658/14658), 5.78 MiB | 6.68 MiB/s, done.
Resolving deltas: 100% (11691/11691), done.
[cartographer] Done.
[cartographer_ros] Fetching https://github.com/cartographer-project/cartographer_ros.git (version master) to /home/task/cartographer_ws/src/cartographer_ros
Cloning into '/home/task/cartographer_ws/src/cartographer_ros'...
remote: Enumerating objects: 4490, done.
remote: Counting objects: 100% (10/10), done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 4490 (delta 2), reused 6 (delta 2), pack-reused 4480
Receiving objects: 100% (4490/4490), 4.28 MiB | 6.01 MiB/s, done.
Resolving deltas: 100% (3169/3169), done.
[cartographer_ros] Done.
```

```
$ mkdir cartographer_ws
$ cd cartographer_ws
$ wstool init src
$ wstool merge -t src https://raw.githubusercontent.com/cartographer-project/cartographer_ros/master/cartographer_ros.rosinstall
$ wstool update -t src
```

3) cartographer ros 종속성 설치

```
task@task:~/cartographer_ws$ sudo rosdep init
ERROR: default sources list file already exists:
/etc/ros/rosdep/sources.list.d/20-default.list
Please delete if you wish to re-initialize
task@task:~/cartographer_ws$ rosdep update
reading in sources list data from /etc/ros/rosdep/sources.list.d
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/osx-homebrew.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/base.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/python.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/ruby.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/releases/fuerte.yaml
Query rosdistro index https://raw.githubusercontent.com/ros/rosdistro/master/index-v4.yaml
Skip end-of-life distro "ardent"
Skip end-of-life distro "bouncy"
Skip end-of-life distro "crystal"
Skip end-of-life distro "dashing"
Skip end-of-life distro "eloquent"
Add distro "foxy"
Add distro "galactic"
Skip end-of-life distro "groovy"
Skip end-of-life distro "hydro"
Skip end-of-life distro "indigo"
Skip end-of-life distro "jade"
Skip end-of-life distro "kinetic"
Skip end-of-life distro "lunar"
Add distro "melodic"
Add distro "noetic"
Add distro "rolling"
updated cache in /home/task/.ros/rosdep/sources.cache
task@task:~/cartographer_ws$ rosdep install --from-paths src --ignore-src --rosdistro=${ROS_DISTRO} -y

$ sudo rosdep init
$ rosdep update
$ rosdep install --from-paths src --ignore-src --rosdistro=${ROS_DISTRO} -y
```

4) abseil-cpp 라이브러리 설치

```
task@task:~/cartographer_ws$ src/cartographer/scripts/install_abseil.sh

$ src/cartographer/scripts/install_abseil.sh
```

5) Build & install

```
task@task:~/cartographer_ws$ catkin_make_isolated --install --use-ninja

$ catkin_make_isolated --install --use-ninja
```

1-5. ROS Package 설치

```
$ sudo apt-get install ros-melodic-move-base
$ sudo apt-get install ros-melodic-teb-local-planner
$ sudo apt-get install ros-melodic-serial
$ sudo apt-get install ros-melodic-navigation
$ sudo apt-get install ros-melodic-spatio-temporal-voxel-layer
$ sudo apt-get install ros-melodic-ar-track-alvar
$ sudo apt-get install ros-melodic-rgbd-launch
```

1-6.USBRule 설정

1-6-1 usb cam(AR marker) rule 설정

ll /dev 를 이용한 장치 확인

```
tetra@tetra:~$ ll /dev
```

```
$ll /dev
```

list 에서 video 번호 확인(video 번호는 PC 마다 틀릴 수 있음)

```
crw-rw---- 1 root tty 7, 1 7월 28 08:42 vcs1
crw-rw---- 1 root tty 7, 2 7월 28 08:42 vcs2
crw-rw---- 1 root tty 7, 3 7월 28 08:42 vcs3
crw-rw---- 1 root tty 7, 4 7월 28 08:42 vcs4
crw-rw---- 1 root tty 7, 5 7월 28 08:42 vcs5
crw-rw---- 1 root tty 7, 6 7월 28 08:42 vcs6
crw-rw---- 1 root tty 7, 128 7월 28 08:42 vcsa
crw-rw---- 1 root tty 7, 129 7월 28 08:42 vcsa1
crw-rw---- 1 root tty 7, 130 7월 28 08:42 vcsa2
crw-rw---- 1 root tty 7, 131 7월 28 08:42 vcsa3
crw-rw---- 1 root tty 7, 132 7월 28 08:42 vcsa4
crw-rw---- 1 root tty 7, 133 7월 28 08:42 vcsa5
crw-rw---- 1 root tty 7, 134 7월 28 08:42 vcsa6
crw-rw---- 1 root tty 7, 64 7월 28 08:42 vcsu
crw-rw---- 1 root tty 7, 65 7월 28 08:42 vcsu1
crw-rw---- 1 root tty 7, 66 7월 28 08:42 vcsu2
crw-rw---- 1 root tty 7, 67 7월 28 08:42 vcsu3
crw-rw---- 1 root tty 7, 68 7월 28 08:42 vcsu4
crw-rw---- 1 root tty 7, 69 7월 28 08:42 vcsu5
crw-rw---- 1 root tty 7, 70 7월 28 08:42 vcsu6
drwxr-xr-x 2 root root 60 7월 28 08:42 vfio/
crw----- 1 root root 10, 63 7월 28 08:42 vga_arbiter
crw----- 1 root root 10, 137 7월 28 08:42 vhci
crw----- 1 root root 10, 238 7월 28 08:42 vhost-net
crw----- 1 root root 10, 241 7월 28 08:42 vhost-vsock
crw-rw----+ 1 root video 81, 12 7월 28 08:42 video12
crw-rw----+ 1 root video 81, 13 7월 28 08:42 video13
crw-rw-rw- 1 root root 1, 5 7월 28 08:42 zero
crw----- 1 root root 10, 249 7월 28 08:42 zfs
```

video 번호 확인 후 연결된 장치의 속성 확인을 위하여 아래와 같이 cmd 창에 입력

-list 에서 확인 해야 할 부분은 ATTR{index}, ATTR{idProduct}, ATTR{idVendor} 부분을 확인하여 rules 파일에 등록해주어야 함

```
tetra@tetra:~$ udevadm info --attribute-walk /dev/video12
```

```
$ udevadm info --attribute-walk /dev/video12
```



```

looking at device '/devices/pci0000:00/0000:00:14.0/usb1/1-6/1-6:1.0/video4linux/video12':
  KERNEL=="video12"
  SUBSYSTEM=="video4linux"
  DRIVER==" "
  ATTR{dev_debug}=="0"
  ATTR{index}=="0"
  ATTR{name}=="USB 2.0 HD Camera : USB 2.0 HD "

looking at parent device '/devices/pci0000:00/0000:00:14.0/usb1/1-6/1-6:1.0':
  KERNELS=="1-6:1.0"
  SUBSYSTEMS=="usb"
  DRIVERS=="uvcvideo"
  ATTRS{authorized}=="1"
  ATTRS{bAlternateSetting}==" 0"
  ATTRS{bInterfaceClass}=="0e"
  ATTRS{bInterfaceNumber}=="00"
  ATTRS{bInterfaceProtocol}=="00"
  ATTRS{bInterfaceSubClass}=="01"
  ATTRS{bNumEndpoints}=="01"
  ATTRS{iad_bFirstInterface}=="00"
  ATTRS{iad_bFunctionClass}=="0e"
  ATTRS{iad_bFunctionProtocol}=="00"
  ATTRS{iad_bFunctionSubClass}=="03"
  ATTRS{iad_bInterfaceCount}=="02"
  ATTRS{interface}=="USB 2.0 HD Camera "
  ATTRS{supports_autosuspend}=="1"

looking at parent device '/devices/pci0000:00/0000:00:14.0/usb1/1-6':
  KERNELS=="1-6"
  SUBSYSTEMS=="usb"
  DRIVERS=="usb"
  ATTRS{authorized}=="1"
  ATTRS{avoid_reset_quirk}=="0"
  ATTRS{bConfigurationValue}=="1"
  ATTRS{bDeviceClass}=="ef"
  ATTRS{bDeviceProtocol}=="01"
  ATTRS{bDeviceSubClass}=="02"
  ATTRS{bMaxPacketSize0}=="64"
  ATTRS{bMaxPower}=="200mA"
  ATTRS{bNumConfigurations}=="1"
  ATTRS{bNumInterfaces}==" 2"
  ATTRS{bcdDevice}=="1001"
  ATTRS{bmAttributes}=="80"
  ATTRS{busnum}=="1"
  ATTRS{configuration}==" "
  ATTRS{devnum}=="4"
  ATTRS{devpath}=="6"
  ATTRS{idProduct}=="3822"
  ATTRS{idVendor}=="058f"
  ATTRS{ltm_capable}=="no"
  ATTRS{maxchild}=="0"

```

udev rules 파일 경로로 이동 한 후 webcam.rules 파일을 생성

```

tetra@tetra:~$ cd /etc/udev/rules.d/
tetra@tetra:/etc/udev/rules.d$ sudo gedit webcam.rules

```

```
$cd /etc/udev/rules.d
```

```
$sudo gedit webcam.rules
```

아래와 같이 문구를 추가

```
webcam.rules
/etc/udev/rules.d

KERNEL=="video*", GROUP="video",ATTRS{idVendor}=="058f", ATTRS{idProduct}=="3822",ATTR{index}=="0", SYMLINK+="cam_rear"

KERNEL=="video*", GROUP="video",ATTRS{idVendor}=="058f",
ATTRS{idProduct}=="3822",ATTR{index}=="0", SYMLINK+="cam_rear"
```

-ATTR{index}, ATTR{idProduct}, ATTR{idVendor} 부분을 usb 정보에 맞게 입력 한 후 저장

1-6-2 IMU 설정

usb rule 설정과 마찬가지로 ll /dev 를 이용하여 장치 확인

```
crw-rw---- 1 root dialout 4, 72 4월 26 08:21 ttyS8
crw-rw---- 1 root dialout 4, 73 4월 26 08:21 ttyS9
crw-rw-rw- 1 root dialout 188, 0 4월 26 17:15 ttyUSB0
crw----- 1 root root 10, 60 4월 26 08:21 udmabuf
crw----- 1 root root 10, 239 4월 26 08:21 uhid
crw----- 1 root root 10, 223 4월 26 08:21 uinput
```

장치 속성 확인을 위하여 아래와 같이 cmd 창에 입력

```
tetra@tetra:~$ udevadm info --attribute-walk /dev/ttyUSB0
```

```
$ udevadm info --attribute-walk /dev/ttyUSB0
```

```
looking at parent device '/devices/pci0000:00/0000:00:14.0/usb1/1-8':
KERNELS=="1-8"
SUBSYSTEMS=="usb"
DRIVERS=="usb"
ATTRS{authorized}=="1"
ATTRS{avoid_reset_quirk}=="0"
ATTRS{bConfigurationValue}=="1"
ATTRS{bDeviceClass}=="00"
ATTRS{bDeviceProtocol}=="00"
ATTRS{bDeviceSubClass}=="00"
ATTRS{bMaxPacketSize0}=="64"
ATTRS{bMaxPower}=="100mA"
ATTRS{bNumConfigurations}=="1"
ATTRS{bNumInterfaces}==" 1"
ATTRS{bcdDevice}=="0100"
ATTRS{bmAttributes}=="80"
ATTRS{busnum}=="1"
ATTRS{configuration}==" "
ATTRS{devnum}=="4"
ATTRS{devpath}=="8"
ATTRS{idProduct}=="ea60"
ATTRS{idVendor}=="10c4"
ATTRS{ltm_capable}=="no"
ATTRS{manufacturer}=="Silicon Labs"
ATTRS{maxchild}=="0"
ATTRS{product}=="CP2102 USB to UART Bridge Controller"
ATTRS{quirks}=="0x0"
ATTRS{removable}=="removable"
ATTRS{rx_lanes}=="1"
ATTRS{serial}=="0001"
ATTRS{speed}=="12"
ATTRS{tx_lanes}=="1"
ATTRS{urbnum}=="13151339"
ATTRS{version}==" 1.10"
```

udev rules 파일 경로로 이동 한 후 IMU.rules 파일을 생성

```
tetra@tetra:~$ cd /etc/udev/rules.d/
tetra@tetra:/etc/udev/rules.d$ sudo gedit IMU.rules
```



```
IMU.rules
/etc/udev/rules.d
KERNEL=="ttyUSB*", ATTRS{idVendor}=="10c4", ATTRS{idProduct}=="ea60", ATTRS{serial}=="0001", MODE:="0666", GROUP:="dialout", SYMLINK+="IMU"
```

```
KERNEL=="ttyUSB*", ATTRS{idVendor}=="10c4", ATTRS{idProduct}=="ea60", ATTRS{serial}=="0001",
MODE:="0666", GROUP:="dialout", SYMLINK+="IMU"
```

-ATTR{serial}, ATTR{idProduct}, ATTR{idVendor} 부분을 usb 정보에 맞게 입력 한 후 저장

- ttyUSB 권한설정

```
tetra@tetra:~$ sudo usermod -a -G dialout $USER
```

```
$ sudo usermod -a -G dialout $USER
```

- 저장 후 터미널 창에 `sudo service udev restart` 를 한 후 PC 를 재부팅 하면 설정이 완료 됨

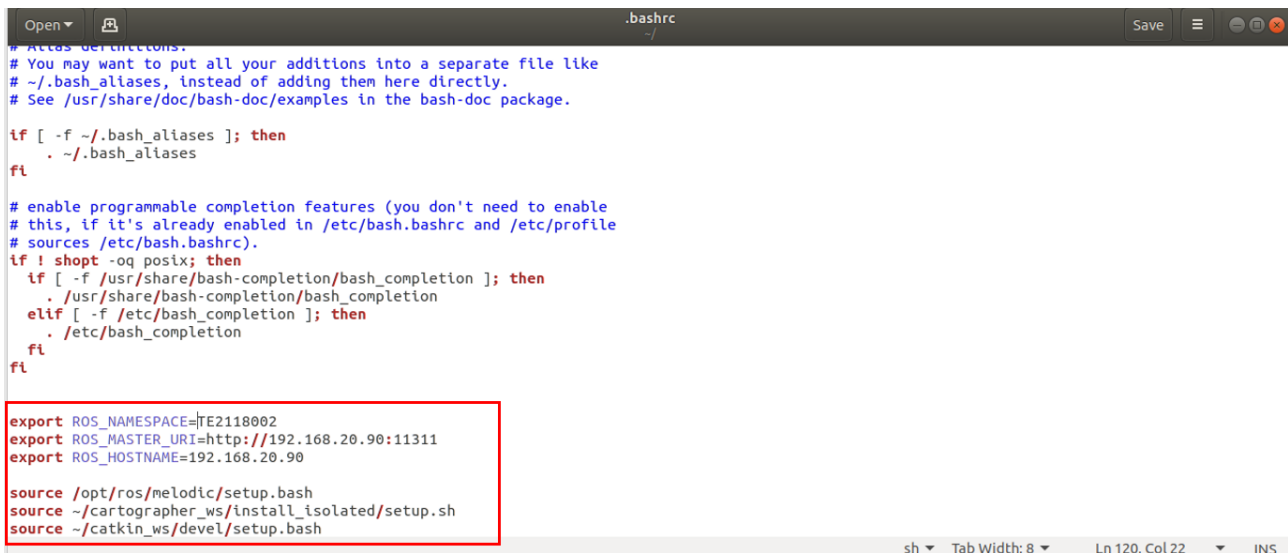
※bashrc

마지막으로 ROS 환경 설정 파일을 자동으로 읽어오도록해주기 위하여 bashrc 파일 수정이 필요함

```
tetra@tetra:~$ gedit ~/.bashrc
```

```
$gedit ~/.bashrc
```

bashrc 파일이 열리면 맨 마지막에 아래와 같이 IP 와 path 를 확인하여 MASTER_URI, HOSTNAME IP 를 PC 에 맞게 수정해야 함



```
.bashrc
~/
# Alias definitions.
# You may want to put all your additions into a separate file like
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/.bash_aliases ]; then
. ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
if [ -f /usr/share/bash-completion/bash_completion ]; then
. /usr/share/bash-completion/bash_completion
elif [ -f /etc/bash_completion ]; then
. /etc/bash_completion
fi
fi

export ROS_NAMESPACE=TE2118002
export ROS_MASTER_URI=http://192.168.20.90:11311
export ROS_HOSTNAME=192.168.20.90

source /opt/ros/melodic/setup.bash
source ~/cartographer_ws/install_isolated/setup.sh
source ~/catkin_ws/devel/setup.bash
```

저장 후 수정한 bashrc 파일을 적용

```
tetra@tetra:~$ source ~/.bashrc
```

```
$source ~/.bashrc
```

Chapter 2. SLAM

2-1. Node 실행

TETRA-DS 와관련된기본 Node 들을실행하기위하여 configuration launch 파일실행

```
tetra@tetra:~$ roslaunch tetraDS_2dnv tetra_configuration.launch
```

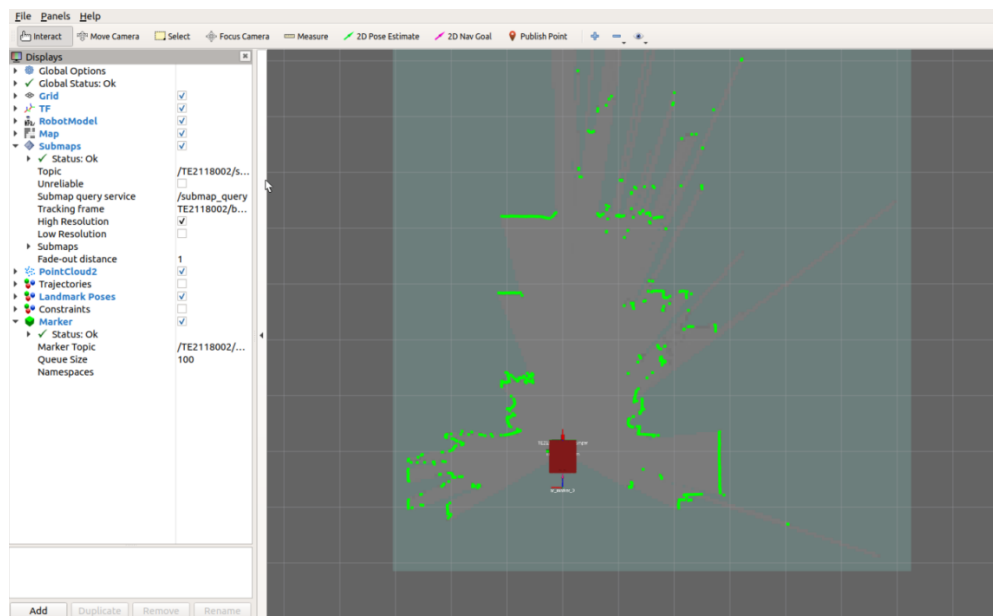
```
$ roslaunch tetraDS_2dnv tetra_configuration.launch
```

cartographer 를이용하여환경지도를작성하기위한 mapping launch 파일실행

```
tetra@tetra:~$ roslaunch tetraDS_2dnv cartographer_mapping.launch
```

```
$roslaunch tetraDS_2dnv cartographer_mapping.launch
```

정상적으로실행되면아래와같은화면이나타나게됨



*지도작성시 로봇의 속도를 빠르게 설정하게 되면 환경에 따라 미끄러짐이 발생되어 위치오차 또는 맵이 틀어질 수 있기 때문에 mapping 시에는속도를천천히하여 mapping 을해야함 (권장속도 : 0.3m/s)

2-2. JoyStick 조작

조이스틱을 조작하여 주변환경에 대하여 mapping 시작할 수 있음

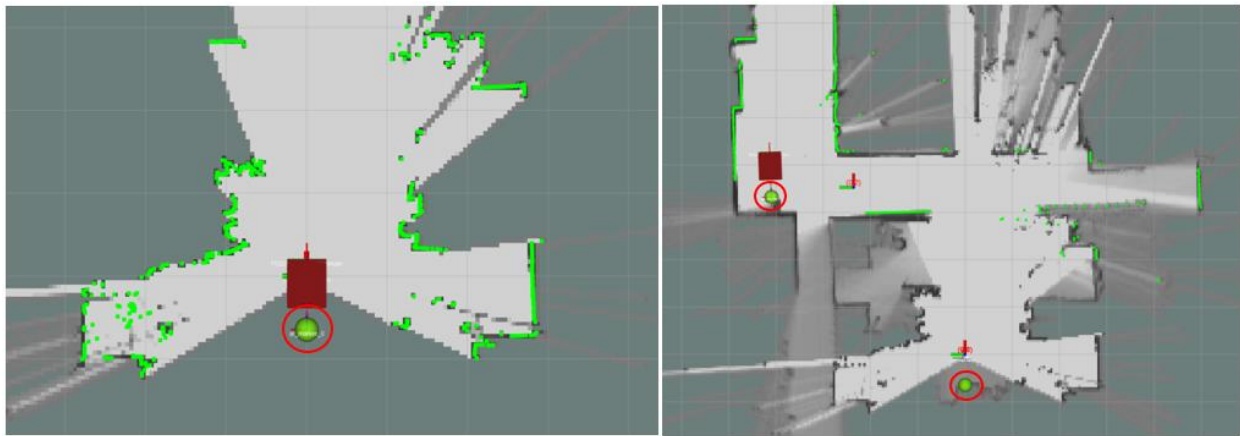


2-3.LandMark 등록

Navigation(자율주행)에서충전스테이션,컨베이어도킹시 AR Tag 를통해 TETRA 의위치보정을할수있으며,위치보정하기위해 Mapping 시 Landmark 를등록해 주어야 함
 Mapping 중 AR Tag 를인식하게되면아래와같이 TF(ar_marker_ID)가생성되고,조이스틱 RB 버튼을누르면 Landmark 를저장 할 수 있음



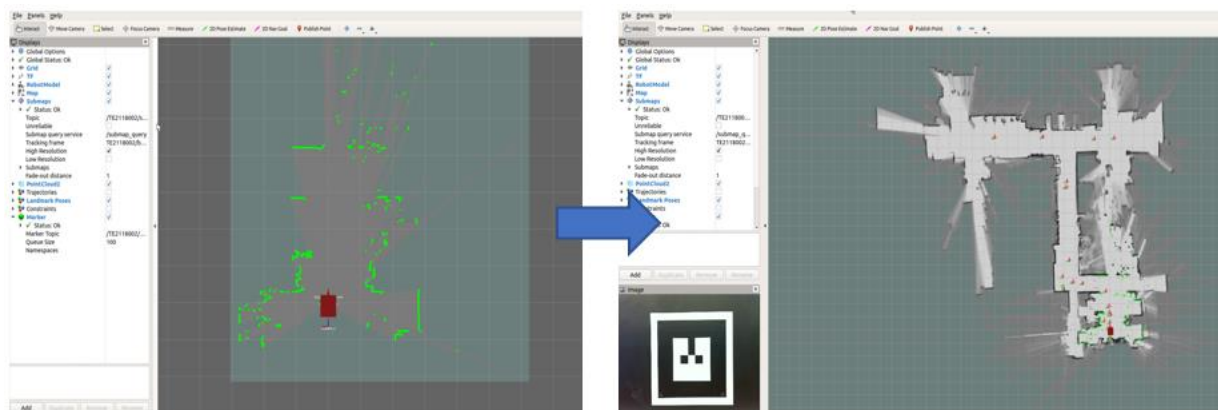
조이스틱 RB 버튼을눌러 Landmark 를저장하게되면 RVIZ 화면에서초록색 구모양으로나타나게됨



*등록한 Landmark list 는 /home/tetra/LANDMARK 에저장 됨

2-4.지도 저장

Mapping 완료후 환경지도를 저장하기 위해 service 호출

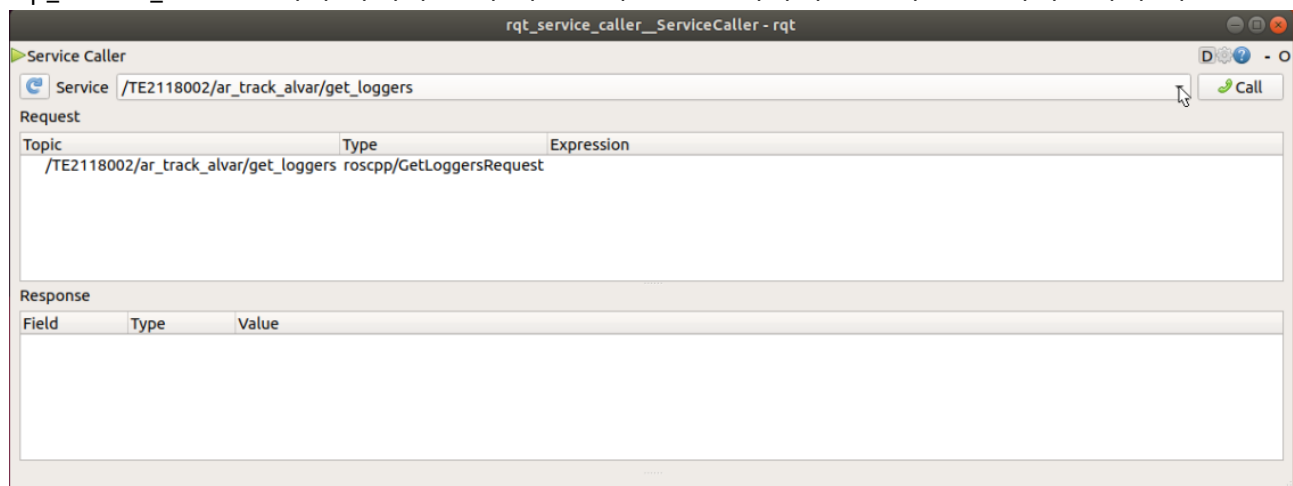


rqt service caller 실행

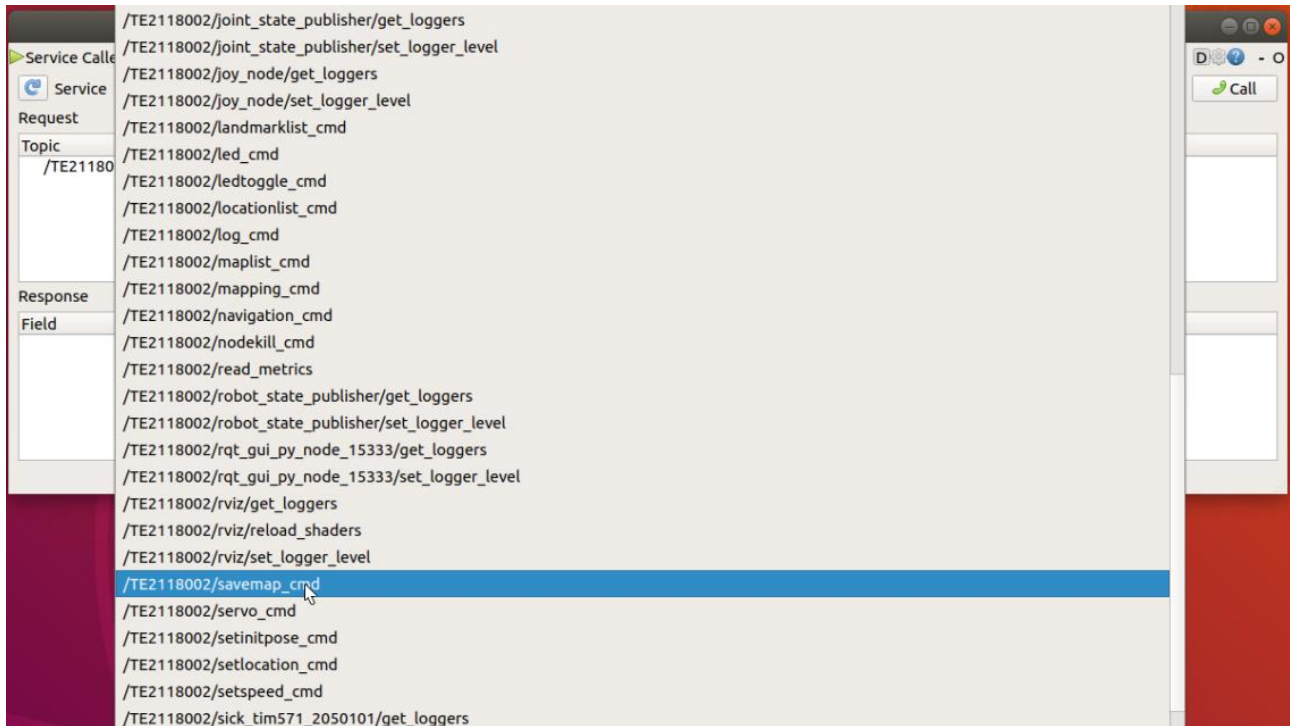
```
tetra@tetra:~$ rosrn rqt_service_caller rqt_service_caller
```

```
$ rosrn rqt_service_caller rqt_service_caller
```

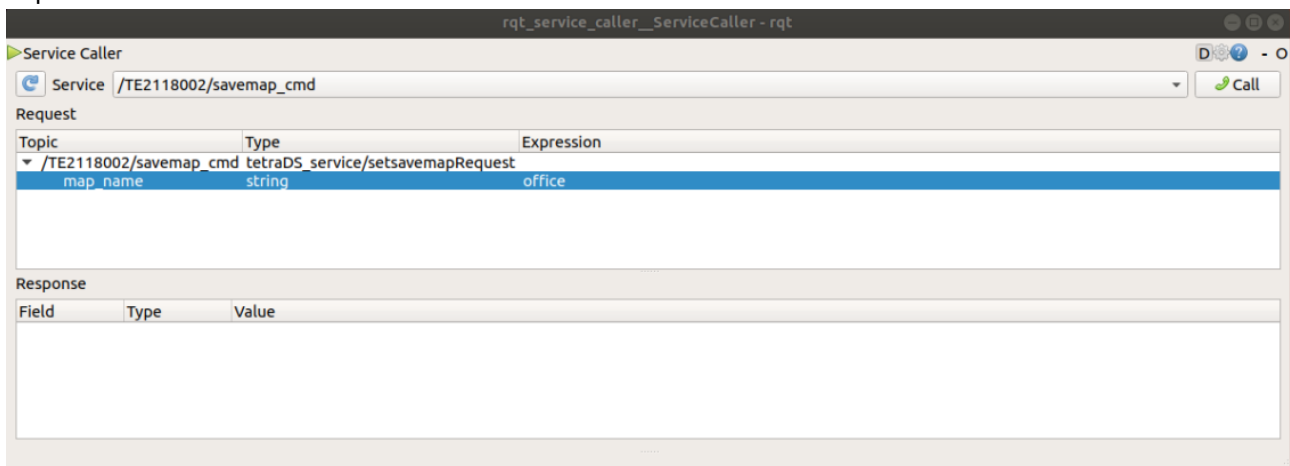
rqt_service_caller 창에서 아래와 같이 화살표 부분을 클릭하여 호출가능 한 서비스목록 확인



서비스목록에서 savemap_cmd 를선택



Expression 부분의 표시된 곳을 더블 클릭하여 저장하고자 하는 이름 작성 후 Call 버튼클릭



*지도가 저장되는 경로는/home/tetra/catkin_ws/tetraDS_2dnv/maps 에저장됨

Chapter 3. Navigation

3-1. Node 실행

TETRA-DS 와 관련된 기본 Node 들을 실행하기 위하여 configuration launch 파일 실행

```
tetra@tetra:~$ roslaunch tetraDS_2dnnav tetra_configuration.launch
```

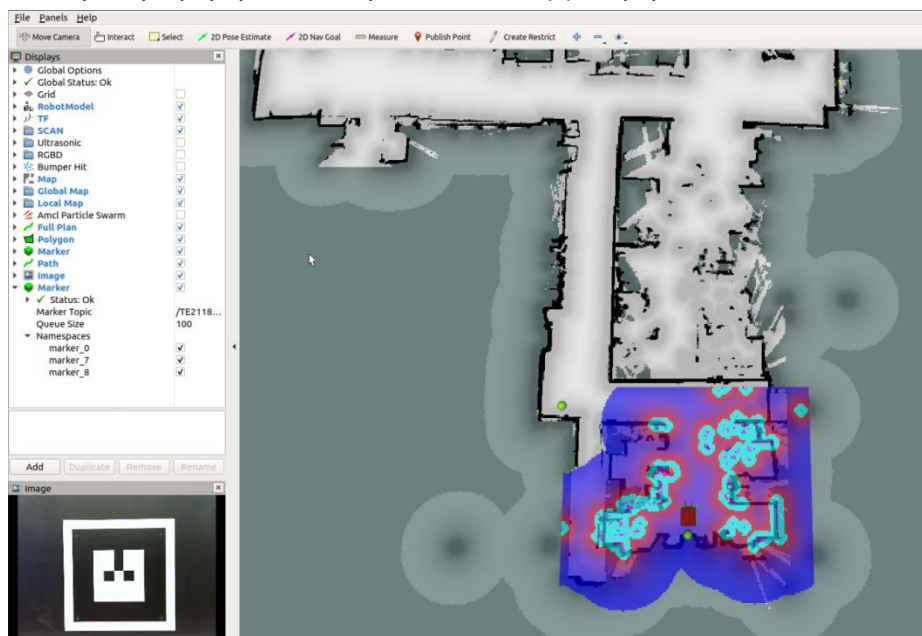
```
$ roslaunch tetraDS_2dnnav tetra_configuration.launch
```

TETRA-DS 를 자율주행하기 위한 launch 파일 실행

```
tetra@tetra:~$ roslaunch tetraDS_2dnnav move_base_tetra.launch
```

```
$ roslaunch tetraDS_2dnnav move_base_tetra.launch
```

아래와 같은 RVIZ 화면이 나타나면 정상적으로 실행된 것입니다



주의사항

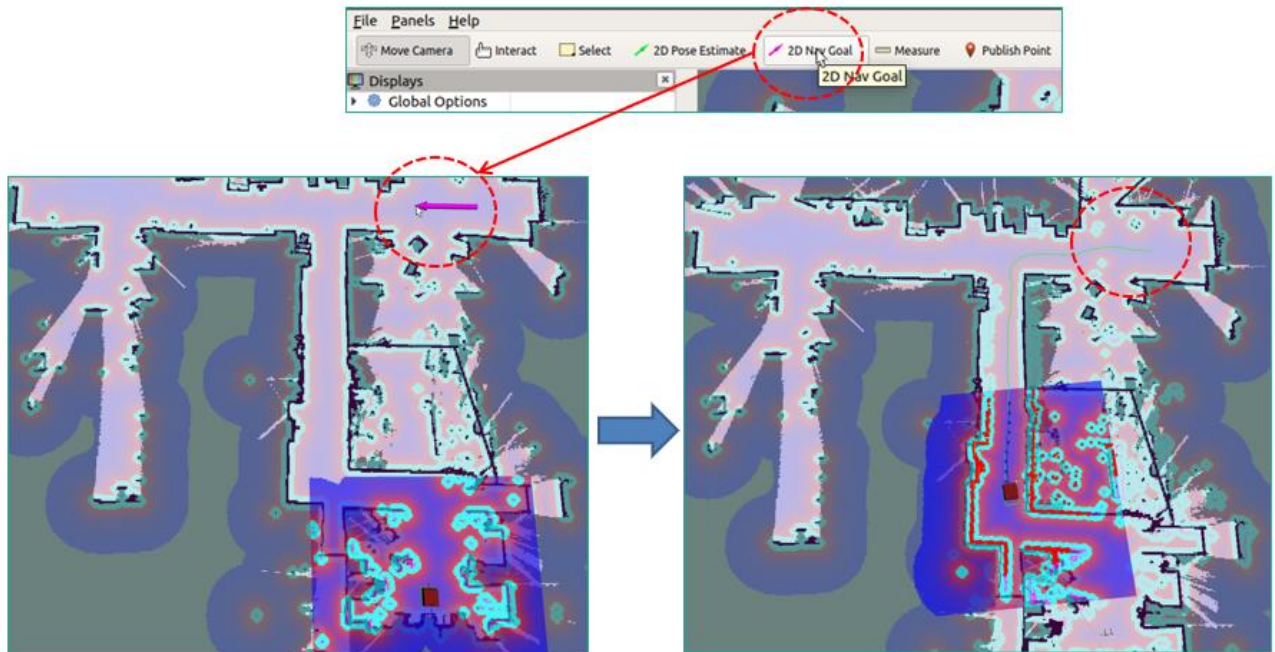
*TETRA-DS5는 충전스테이션에도킹되어있는상태여야함

*실행순서는 tetra_configuration launch 파일이실행되고후에 cartographer mapping launch 또는 move_base_tetra launch 가실행되어야함

3-2. 위치 이동

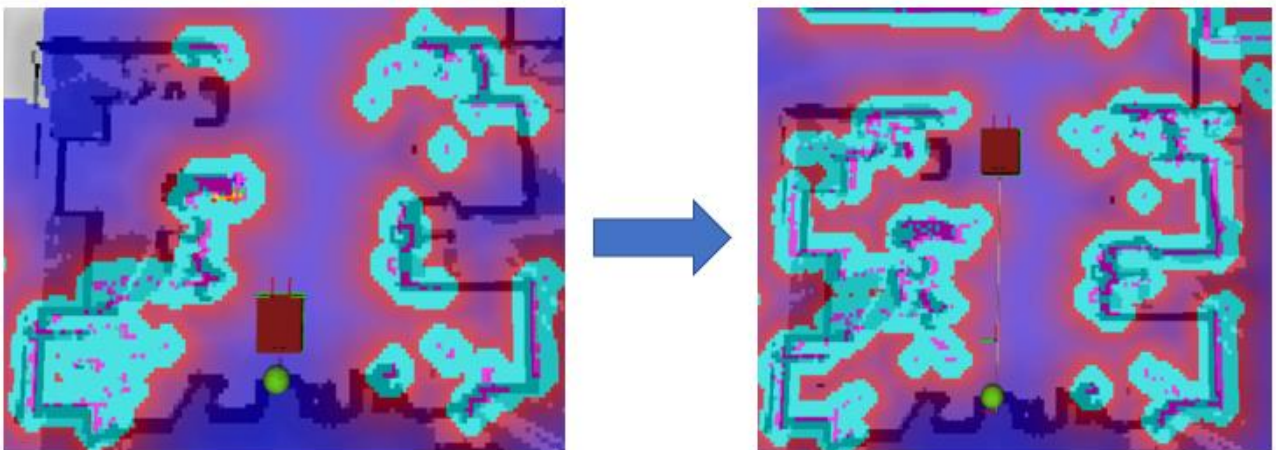
3-2-1 2D Nav Goal

RVIZ 화면에서 상단의 2D Nav Goal 을
클릭후 원하는 위치에 마우스로 드래그하면 해당 위치로 TETRA 가 이동하게 됩니다.



3-2-2 Joystick 사용

조이스틱을 움직여 TETRA 를 해당 위치로 이동시킬 수 있습니다.



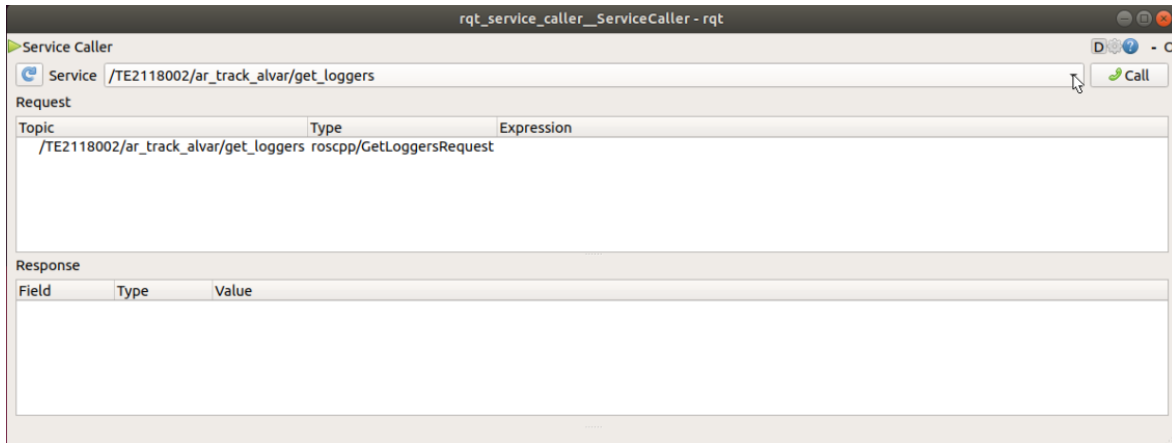
3-2-3서비스 호출

rqt service caller 실행

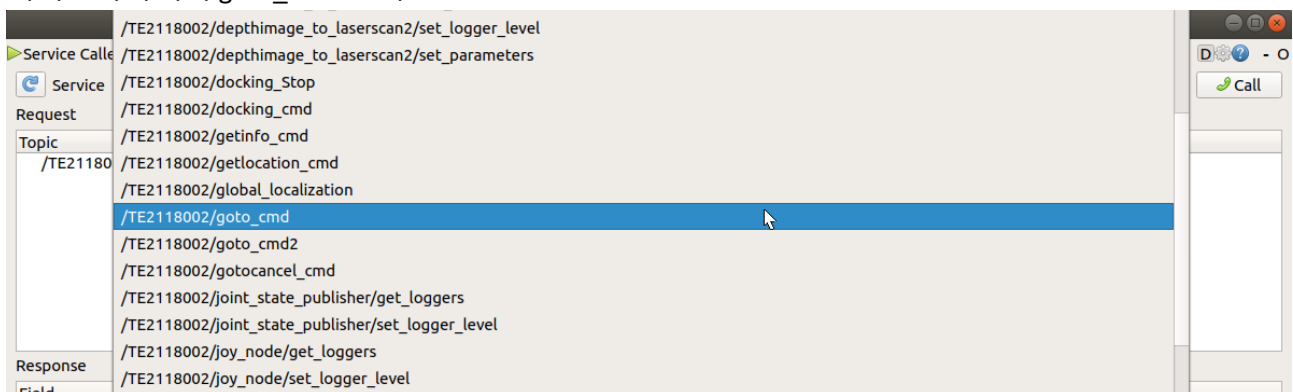
```
tetra@tetra:~$ rosrn rqt_service_caller rqt_service_caller
```

```
$ rosrn rqt_service_caller rqt_service_caller
```

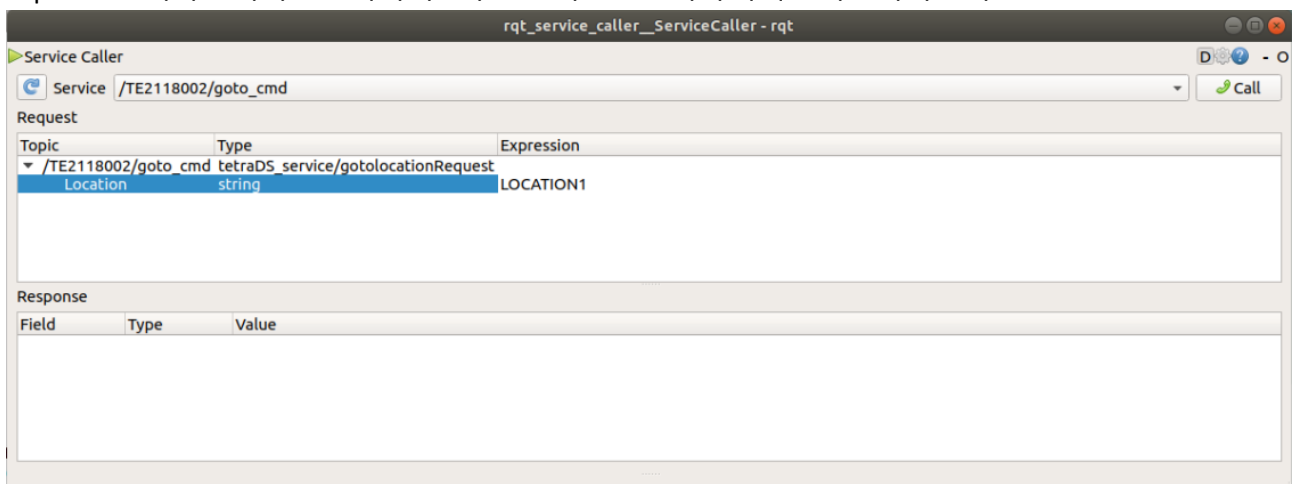
화살표를눌러서비스목록확인



서비스목록에서 goto_cmd 클릭



Expression 에사전에저장한위치기입후 Call 버튼을클릭하게되면해당위치로이동함



3-3. 위치 저장

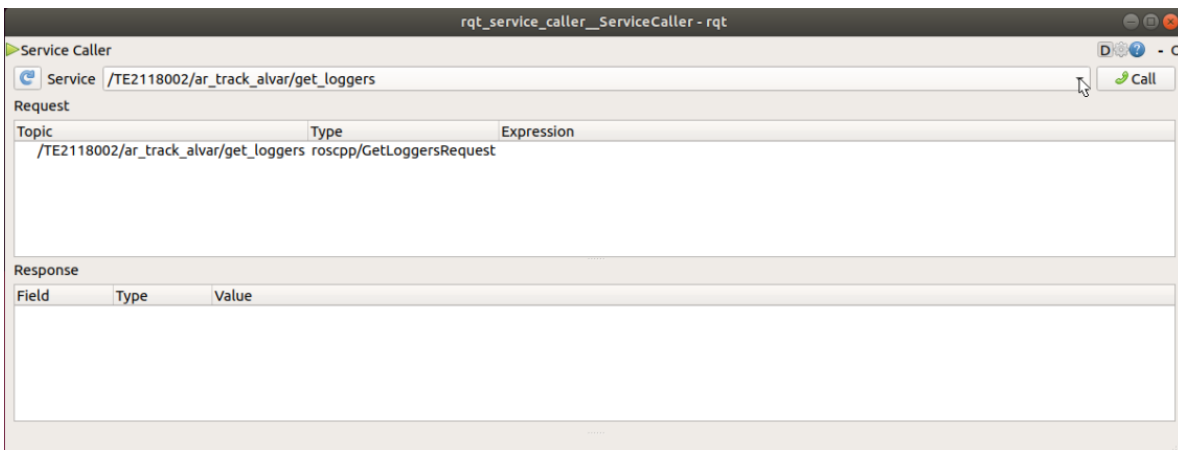
위치가동후해당위치를저장하고자할때 setlocation_cmd 서비스를호출하여저장할수있습니다.

rqt service caller 실행

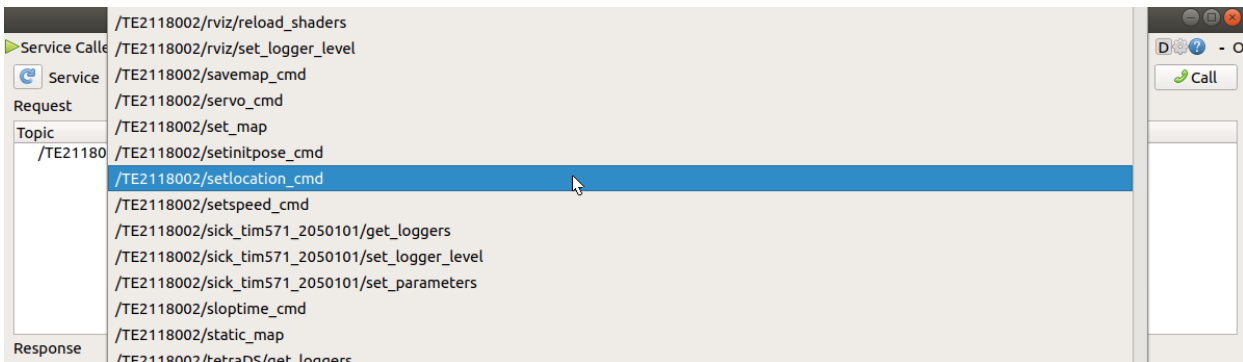
```
tetra@tetra:~$ rosrn rqt_service_caller rqt_service_caller
```

```
$ rosrn rqt_service_caller rqt_service_caller
```

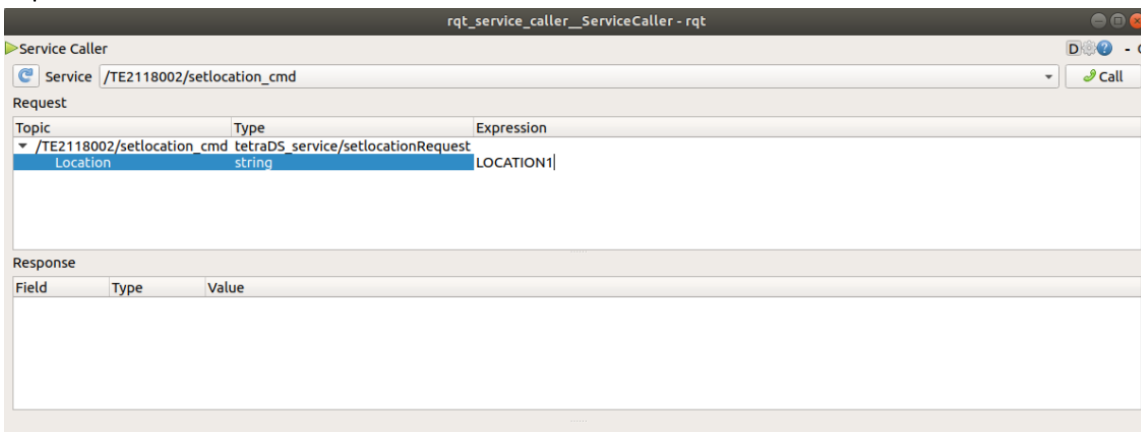
화살표를눌러서비스목록확인



서비스목록에서 setlocation_cmd 선택

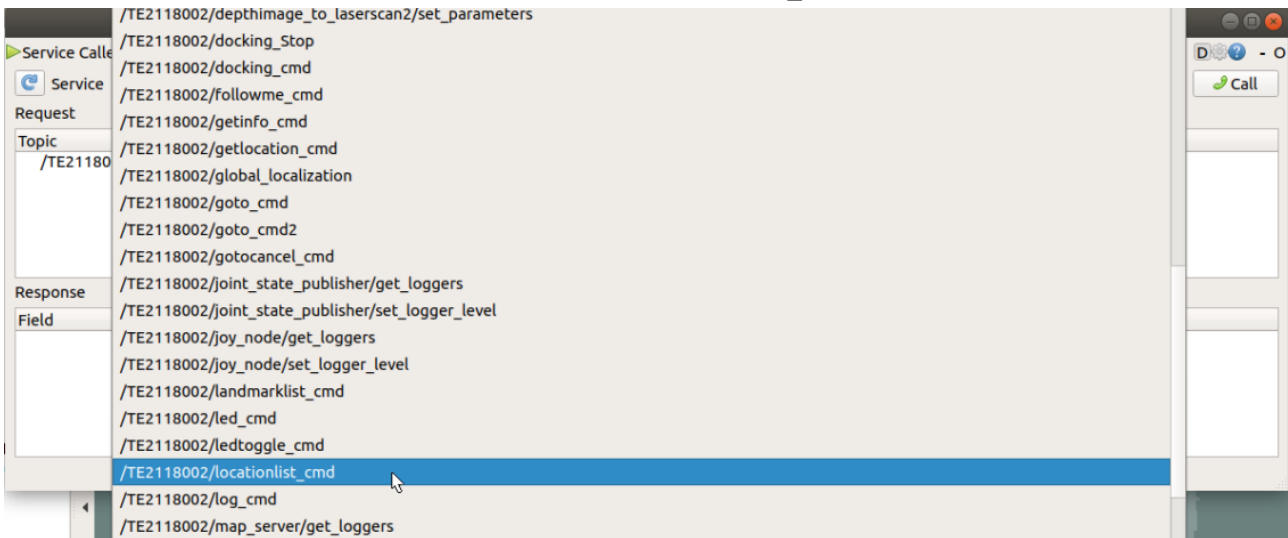


Expression 에저장하고자하는이름기입후 Call 버튼클릭

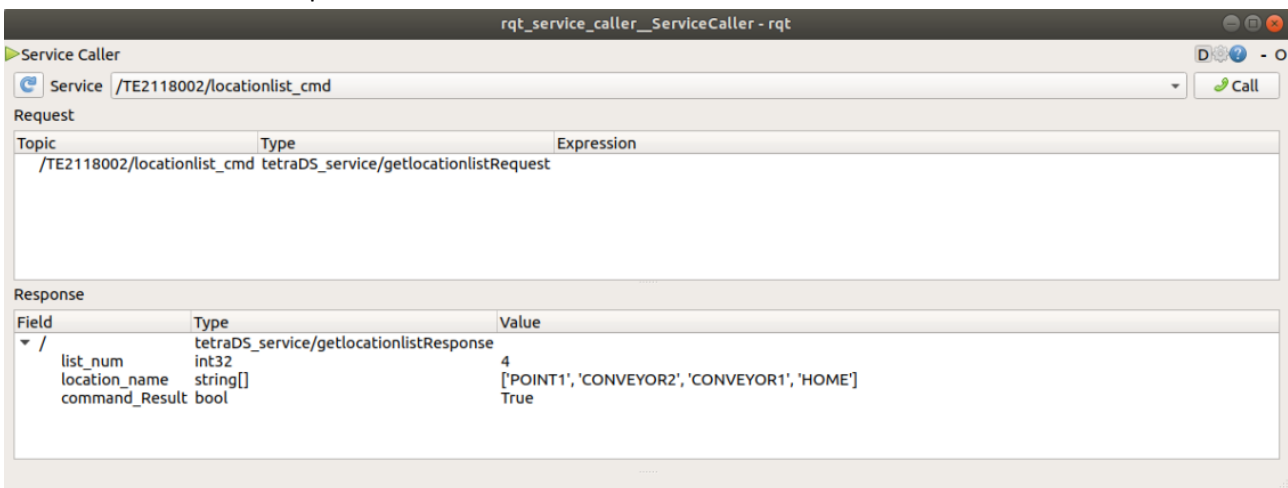


*위치파일들은/home/tetra/DATA 에저장됩니다

저장된 위치를 모두 확인하고 싶을 때 서비스 목록에서 locationlist_cmd 선택



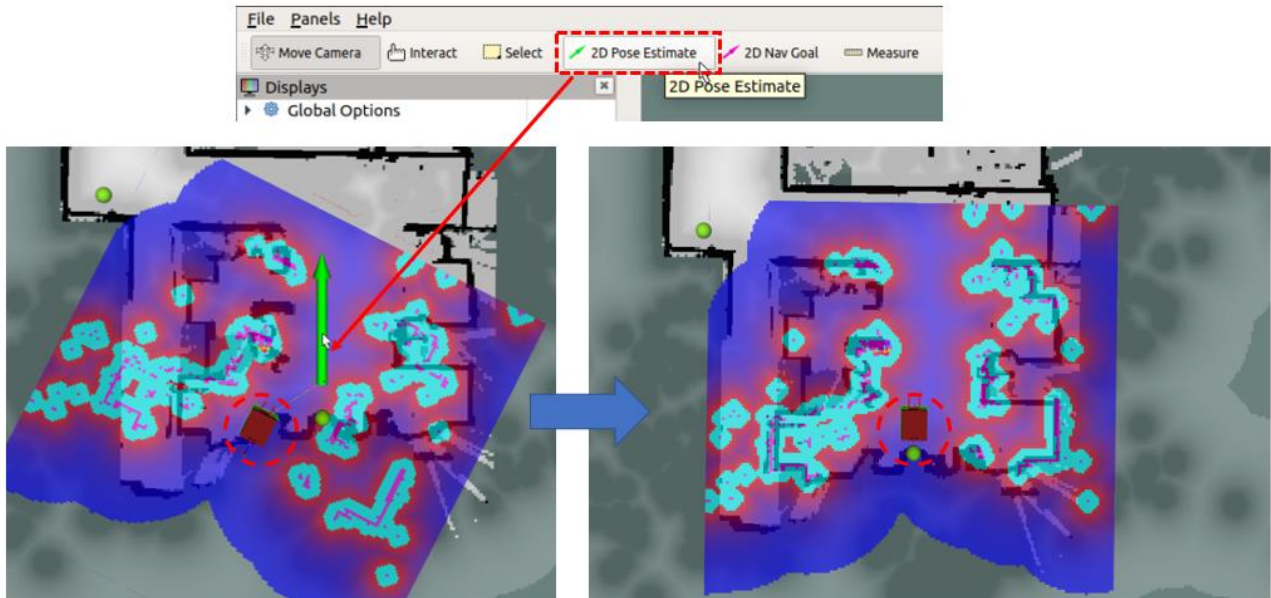
Call 버튼을 클릭하면 Response 창에 저장된 위치 개수와 목록이 나타남



3-4. 위치 보정

3-4-12D Pose Estimate

위치를수동으로보정하고싶은경우에는상단에위치한‘2D Pose Estimate’ 버튼을클릭한후 map 상에서현재로봇에위치에좌클릭후로봇방향으로녹색화살표를정하게되면해당위치로이동후보정하게됩니다.

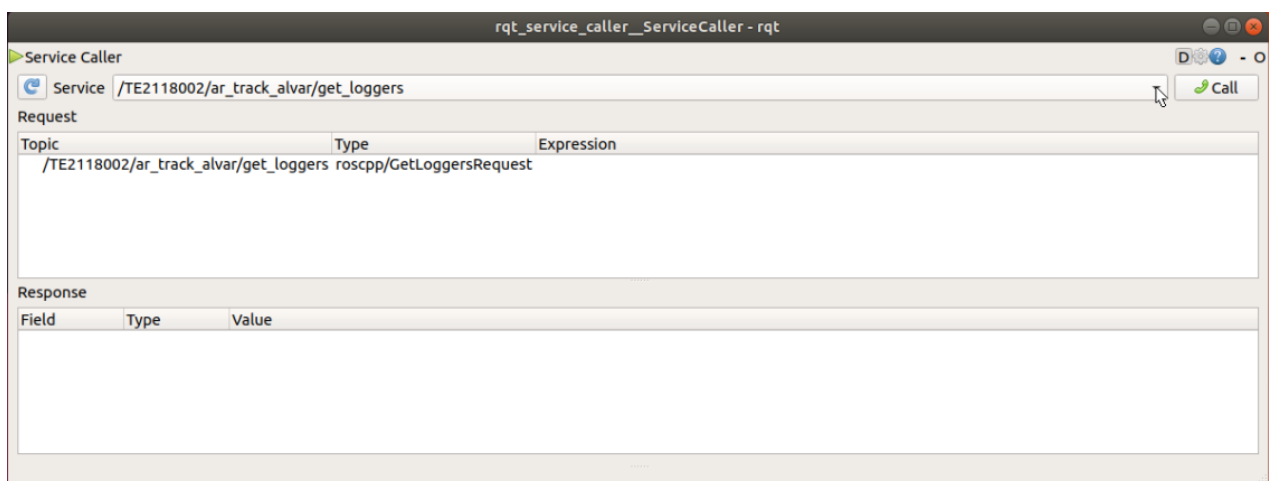


3-4-2서비스 호출

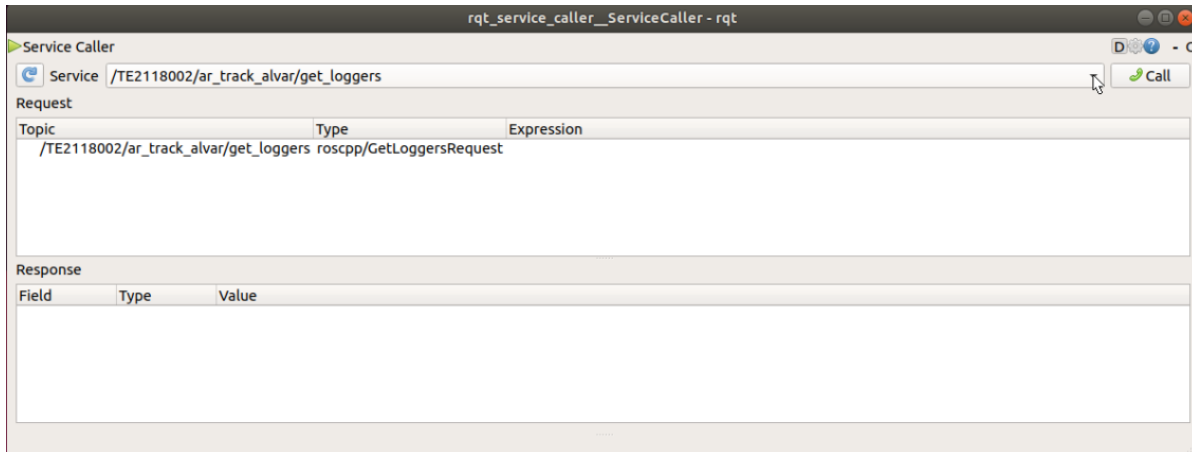
서비스호출로위치를보정하고자할때에는 rqt service caller 실행한후

```
tetra@tetra:~$ rosrn rqt_service_caller rqt_service_caller
```

```
$rosrn rqt_service_caller rqt_service_caller
```



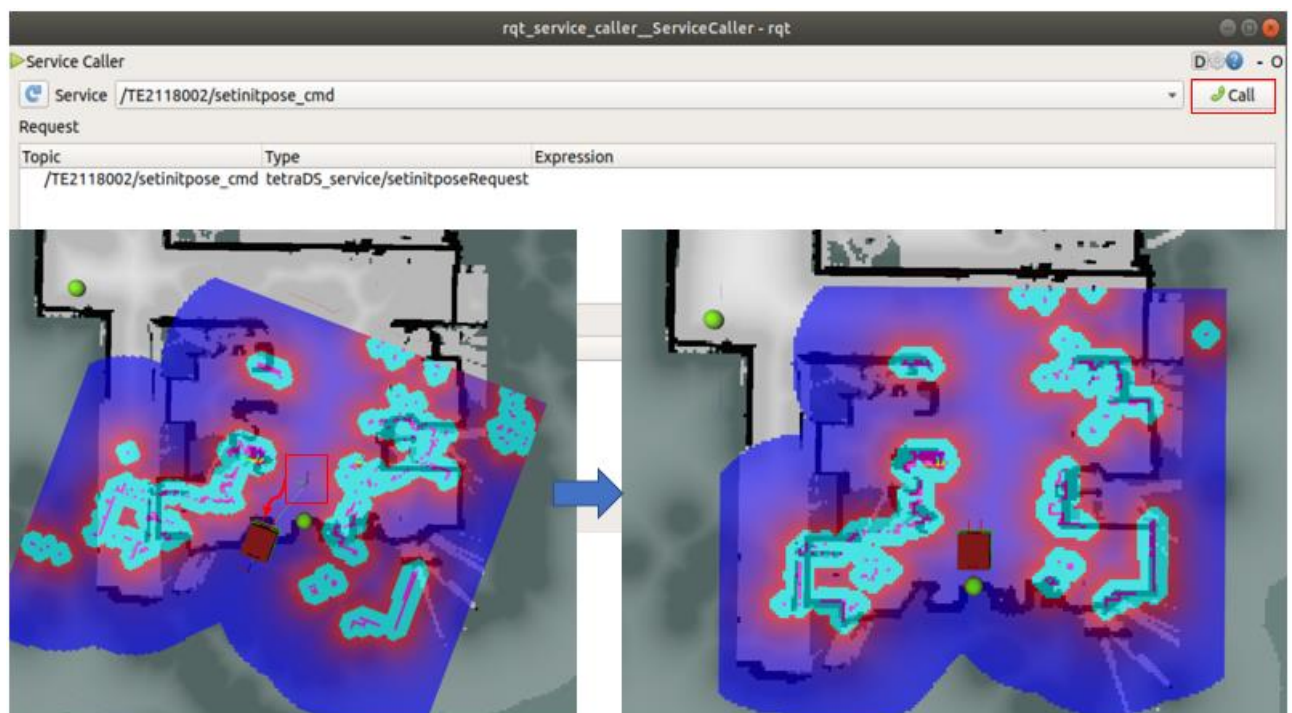
화살표를 눌러서 서비스 목록 확인



setinitpose_cmd 선택



Call 버튼을 클릭하면 위치 보정을 실시함



Chapter 4. ROS

4-1. TETRA-DSV ROS Service

service name	Input	Output	Description
getlocation_cmd		bool command_Result float32 poseAMCLx float32 poseAMCLy float32 poseAMCLqx float32 poseAMCLqy float32 poseAMCLqz float32 poseAMCLqw	로봇의 현재 위치정보데이터 확인명령. * 리턴으로는 X,Y좌표(m단위)와 쿼터니언(4개)가 수신.
goto_cmd	string Location	bool command_Result float32 goal_positionX float32 goal_positionY float32 goal_quarterX float32 goal_quarterY float32 goal_quarterZ float32 goal_quarterW	로봇의 저장위치 이동 명령 * 입력으로 저장된 위치(문자열)이름을 전송 * 리턴으로는 호출결과와 저장위치에 X,Y좌표, 쿼터니언(4개)가 수신
goto_cmd2	float32 goal_positionX float32 goal_positionY float32 goal_quarterX float32 goal_quarterY float32 goal_quarterZ float32 goal_quarterW	bool command_Result	로봇의 특정위치 이동 명령 * 입력으로 X,Y좌표와 쿼터니언(4개)데이터를 전송 * 리턴으로는 호출결과가 수신
setlocation_cmd	string Location	bool command_Result float32 goal_positionX float32 goal_positionY float32 goal_quarterX float32 goal_quarterY float32 goal_quarterZ float32 goal_quarterW	로봇에 현재위치 저장 명령. * 입력으로는 현재위치를 저장할 위치이름(문자열) 전송. * 리턴으로는 호출결과와, 현재위치에 X,Y좌표, 쿼터니언(4개)가 수신.
savemap_cmd	string map_name	bool command_Result	작성된 현재 환경지도의 지도파일 저장 명령. * 입력으로는 저장할 지도의 이름(문자열)전송. * 리턴으로는 호출결과 수신.
getinfo_cmd		bool command_Result int32 battery int32 Error_code int32 EMG int32 bumper int32 charging int32 running_mode	로봇의 상태정보 확인 명령. * 리턴으로는 배터리 레벨, 에러코드, 비상정지스위치 체크, 범퍼체크, 충전상태 체크, 동작모드가 수신. . 단, 에러코드는 에러 미발생 시 0을 리턴. . 동작모드(running_mode) 설명: 0: 기본상태 1 : 환경지도 작성 모드, 2: 자율주행 모드
locationlist_cmd		int32 list_num string[] location_name bool command_Result	로봇에 저장된 Waypoint정보 확인명령 * 리턴으로는 Waypoint의 총 개수와 각각의 Waypoint 이름(문자열) 배열 수신
delete_location_cmd	string location	bool command_Result	로봇의 저장된 Waypoint삭제 * 입력으로 Waypoint이름(문자열)을 전송 * 리턴으로는 삭제여부 결과가 수신
maplist_cmd	-	int32 list_num string[] location_name bool command_Result	'/home/tetra/catkin_ws/src/tetraDS_2dnv/maps 경로에 저장된 map정보확인 명령 * 리턴으로는 map의 총 개수와 각각의 map이름(문자

			<p>열) 배열 수신</p> <p>* .pgm 확장자만 검색</p>
mapping_cmd	boolflag_mapping	bool command_Result	<p>cartographer를 이용한 지도작성 기능을 호출하는 명령</p> <p>* 입력으로는 true를 전송</p> <p>* 리턴으로는 호출결과 수신</p> <p>* 주의사항: 자율주행 기능이 호출중이면, nodekill을 먼저 수행 해야함</p>
navigation_cmd	string map_name	bool command_Result	<p>로봇의 자율주행 기능을 호출하는 명령</p> <p>* 입력으로는 자율주행을 실시할 map의 이름을 전송.</p> <p>* 리턴으로는 호출결과 수신.</p> <p>* 주의사항: 지도작성 기능이 호출중이면, nodekill을 먼저 수행 해야 함</p>
nodekill_cmd	-	bool command_Result	<p>tetra_configuration.launch를 제외하고 로봇에 실행중인 ROS node들을 모두 종료하는 명령.</p> <p>* 입력으로는 true를 전송.</p> <p>* 리턴으로는 호출결과 수신.</p>
delete_map_cmd	string map_name	bool command_Result	<p>저장된 환경지도를 지우는 명령.</p> <p>* 입력으로는 환경지도 이름을 전송.</p> <p>* 리턴으로는 호출결과 수신.</p>
turnon_cmd	int32 ID	bool command_Result	<p>로봇에 장착된 LED On/Off 제어명령.</p> <p>* 입력으로는 LED의 ID, On이되는 가속시간, On시의 밝기 전송.</p> <p>* 출력으로는 호출결과 수신.</p>
led_cmd	int32 ID int32 led brightness	bool command_Result	<p>로봇에 장착된 LED On/Off 제어명령.</p> <p>* 입력으로는 LED의 ID 전송. (1: 좌, 2: 우, 3: 좌우On, 4: 좌우 Off)</p> <p>* 출력으로는 호출결과 수신.</p>
ledtoggle_cmd	int32 light_accel int32 led_High_brightness int32 light_decel int32 led Low_brightness	bool command_Result	<p>로봇에 장착된 LED Toggle 제어명령. (깜박이 동작)</p> <p>* 입력으로는 LED의 On이되는 가속시간, On시의 밝기, Off되는 가속시간, Off시의 밝기 값을 전송.</p> <p>* 출력으로는 호출결과 수신.</p>
gotocancel_cmd	string location_id	bool command_Result	<p>로봇의 특정위치 이동취소 명령.</p> <p>* 입력으로는 공백문자열을 전송.</p> <p>* 출력으로는 호출결과 수신.</p>
setspeed_cmd	float32 speed	float32 set_vel bool command_Result	<p>로봇의 최대속도를 설정하는 명령.</p> <p>* 입력으로는 로봇의 최대 선속도 값(double)을 전송. (기본값: 0.5) 단위는 m/s</p> <p>* 출력으로는 호출결과와 적용된 선속도 값을 전송.</p>
servo_cmd	int32 data	bool command_Result	<p>로봇의 구동부 서보모터를 강제로 On/Off하는 명령.</p> <p>* 입력으로는 data(1이면 On, 2이면 Off)를 전송.</p> <p>* 출력으로는 호출결과를 전송.</p>
docking_cmd	int32 id int32 data	bool command_Result	<p>충전스테이션과의 Docking명령.</p> <p>* 입력으로 id는 0, data는 1을 전송</p> <p>* 출력으로는 호출결과를 전송.</p> <p># 충전스테이션의 id가 변경되면, 변경된 id번호를 넣어야 함.</p>
docking_stop			<p>로봇과 충전스테이션과의 docking동작을 강제로 정지시키는 명령.</p> <p>* 입력과 출력이 없습니다.</p> <p>* 명령이 호출되면 로봇의 선속도와 각속도를 0으로 만들고 정지합니다.</p>
virtual_obstacle_	int32[] list_count	bool command_Result	<p>가상벽 /장애물을 그리는 명령.</p>

cmd	float32[] form_x float32[] form_y float32[] form_z		<p>*입력:</p> <ul style="list-style-type: none"> - list_count: 가상벽/장애물의 개수 - mode: 장애물의 형태를 정하는 변수.(1: 점 또는 원, 2: 선, 그외: 다각형) - form_x: 장애물의 X좌표. - form_y: 장애물의 Y좌표. - form_x: mode가 1인 상태에서 해당값이 0보다 크면 원의 지름을 의미하며, 그 이외에는 모두 0.0을 사용 <p>* 출력으로는 호출결과를 전송.</p> <p>※ 좌표는 amcl_pose의 좌표기준이며, 단위는 m단위임.</p> <p>※ 서비스 사용방법 예시</p> <ul style="list-style-type: none"> - 우측 그림과 같이 로봇 전방에 지름0.5m사이즈의 원을 1, 2, 3m지점에 하나씩 생성할 경우 <p>* list_count[]: [1, 1, 1]</p> <p>* form_x: [1.0, 2.0, 3.0]</p> <p>* form_y: [0.0, 0.0, 0.0]</p> <p>* form_z: [0.5, 0.5, 0.5]</p>
landmarklist_cmd		int32 list_num string[] landmark_name bool command_Result	<p>로봇에 저장된 Landmark의 정보 리스트 확인</p> <ul style="list-style-type: none"> - 입력은 없고 출력만 있음 <p>* list_num: Landmark정보 개수</p> <p>* landmark_name: Landmark이름(문자열) 배열 수신.</p> <p>* command_Result: 함수 호출결과</p>
delete_landmark_cmd	stringLandmark	bool command_Result	<p>로봇의 저장된 Landmark정보 삭제</p> <ul style="list-style-type: none"> * 입력으로 Landmark정보가 저장된 text file명을 입력 (예: marker_0 -> 0번 landmark) <p>* 리턴으로는 삭제여부 결과가 수신.</p>
setinitpose_cmd		int8 m_iAR_tag_id float64init_position_x float64init_position_y float64init_position_z float64init_orientation_z float64init_orientation_w bool command_Result	<p>해당 AR Tag ID를 확인 한 후그 위치로 위치보정 실시</p> <ul style="list-style-type: none"> -입력은 없고 AR Tag 가 인식되면 출력으로 해당 AR Tag ID 와 위치 정보가 출력됨
output_cmd	int8 Output0 int8 Output1 int8 Output2 int8 Output3 int8 Output4 int8 Output5 int8 Output6 int8 Output7	bool command_Result	<p>TETRA-DS5에 내장된 GPIO의 기능으로서 Digital Output 8개 채널을 개별 On/Off제어할 수 있는 기능.</p> <ul style="list-style-type: none"> * 입력으로 각각의 Digital Output에 1을 입력하면 On, 0을입력하면Off * 단, 1 또는 0 만 입력해야 함.
patrol_cmd	bool on int32 list_count string[] location_name	bool command_Result	<p>TETRA-DS5에 저장된 Locattion들을 Patrol하는 기능.</p> <p>*입력</p> <ul style="list-style-type: none"> .on: Patrol의 활성화 여부 (True는 enable, False는 disable) .list_count: location의 개수 .location_name: location문자열을 넣을 배열. <p>예시) P1과 P2를 Patrol하고 싶은 경우 on은 True, list_count는2, location_name은['P1','P2']를 입력</p> <p>* 출력으로는 호출결과를 리턴.</p>

gotoconveyor_cmd	string Location int32 id int32 movement	bool command_Result	TETRA-DS5를 고정형 Conveyor위치로 이동하는 명령 (Conveyor위치에는 지정된 AR_tag ID가 있어야 함) * 입력 . Location: Conveyor위치 이름을 저장한 문자열 . id: Conveyor에 부착된 AR_tag ID 번호 . movement: loading or unloading의 선택 (0: nomal, 1: Loading, 2: Unloading) * 출력: 함수호출 결과를 리턴
loadingcheck_service_cmd		bool command_Result	TETRA-DS5가 Conveyor에 Docking하여 Loading을 완료했는지를 확인하는 함수. * 출력으로 Loading완료여부를 리턴.
unloadingcheck_service_cmd		bool command_Result	TETRA-DS5가 Conveyor에 Docking하여 Unloading을 완료했는지를 확인하는 함수. * 출력으로 Unloading완료여부를 리턴.
savemark_cmd		bool command_Result	TETRA-DS5의 Mapping Mode에서만 사용이 가능합니다. * 해당 서비스를 호출하면, 확인된 AR_tag ID를 저장합니다.
deletedataall_cmd		bool command_Result	TETRA-DS5의 Map, Location, Landmark Data들을 모두 삭제합니다. * 해당 서비스를 호출하면, /home/tetra/catkin_ws/src/tetraDS_2dnav/maps/ 경로에 있는 Map data와 /home/tetra/DATA/ 경로에 있는 Location data, /home/tetra/LANDMARK/ 경로에 있는 Landmark data를 모두 삭제합니다.
CAL_cmd		bool command_Result	Conveyor에 장착된 Loadcell의 Callibration을 실시합니다. * 해당 함수를 호출한 상태에 무게를 0.0Kg으로 설정합니다. * 해당 함수는 초기에 한번만 호출하면 ROM에 저장됩니다.
Auto_Move_cmd	int32 start	bool command_Result	TETRA의 Conveyor 자동조작 기능 함수입니다. * start는 Loading or Unloading을 Start/Stop하는 설정입니다. . start: 0 --> Loading or Unloading Stop . start: 1 --> Loading or Unloading Strat
Manual_Move_cmd	int32 mode	bool command_Result	TETRA의 Conveyor 수동조작 기능 함수입니다. * mode에는 동작 방향에 대한 명령값을 입력합니다. . mode: 0 --> Conveyor정지 . mode: 1 --> Unloading 동작 실행. . mode: 2 --> Loading 동작 실행.
parameter_read_cmd	int32 num	int32 data bool command_Result	TETRA-DS5 Motor Drive관련 Parameter의 읽기 함수입니다. * Input: num에는 Parameter번호를 입력합니다. (0~30번까지) * Output: . data에는 해당 Parameter번호의 데이터 값이 리턴됩니다. . command_Result는 호출결과가 리턴됩니다. 단, HW의 변경이 없는 한 해당 기능은 사용하지 마세요.

parameter_write_cmd	int32 num int32 data	bool command_Result	TETRA-DS5 Motor Drive관련 Parameter의 쓰기 함수입니다. * Input: . num에는 Parameter번호를 입력합니다. (0~30번까지) . data에는 해당 Parameter번호에 적용할 값을 입력합니다. * Output: command_Result는 호출결과가 리턴됩니다. 단, HW의 변경이 없는 한 해당 기능은 사용하지 마세요.
mode_change_cmd	int32 mode	bool command_Result	TETRA-DS5 Motor Drive관련 동작모드 변환 함수입니다. * Input: mode가 1인 경우 Position Mode, 0인 경우 Velocity Mode입니다. (기본 0) * Output: command_Result는 호출결과가 리턴됩니다. 단, 초기설정 기능임으로 평상시에 사용하지 않습니다.
linear_move_cmd	int32 linear_position	bool command_Result	TETRA-DS5 Motor Drive관련 전/후진 동작 함수입니다. * Input: linear_position에 위치 값을 입력합니다. (1m를 이동할 경우 10000을 입력) * Output: command_Result는 호출결과가 리턴됩니다. 단, 초기설정 기능임으로 평상시에 사용하지 않습니다.
angular_move_cmd	int32 Angular_degree	bool command_Result	TETRA-DS5 Motor Drive관련 좌/우회전 동작 함수입니다. * Input: Angular_degree에 회전각도 값을 입력합니다. (90도를 회전할 경우 9000을 입력) * Output: command_Result는 호출결과가 리턴됩니다. 단, 초기설정 기능임으로 평상시에 사용하지 않습니다.
pose_estimate_cmd	float64 estimate_position_x float64 estimate_position_y float64 estimate_position_z float64 estimate_orientation_x float64 estimate_orientation_y float64 estimate_orientation_z float64 estimate_orientation_w	bool command_Result	RVIZ에 있는 2D Pose Estimate를 서비스로 호출할 수 있도록 제작한 함수입니다. * Input: 로봇을 이동하고자 하는 위치와 자세 데이터 * Output: 호출결과
gpio_status_cmd		uint8 Input0 uint8 Input1 uint8 Input2 uint8 Input3 uint8 Input4 uint8 Input5 uint8 Input6 uint8 Input7 uint8 Output0 uint8 Output1 uint8 Output2 uint8 Output3 uint8 Output4	TETRA-DS에 있는 GPIO의 상태를 확인하는 서비스입니다. *Input: 없음. *Output: Input 0~7 번, Output 0~7 번까지의 상태 리턴

		uint8 Output5 uint8 Output6 uint8 Output7 bool command_Result	
set_ekf_cmd	float64 init_position_x float64 init_position_y float64 init_position_z float64 init_orientation_x float64 init_orientation_y float64 init_orientation_z float64 init_orientation_w	bool command_Result	TETRA-DS에 현 위치로 EKF Pose와 Imu를 모두 초기화하는 함수입니다. *Input: map좌표 기준으로의 현 로봇 pose값

4-2. TETRA-DSV ROS TOPIC

Topic name	Message Type	Description
odom	nav_msgs/Odometry [http://docs.ros.org/en/noetic/api/nav_msgs/html/msg/Odometry.html]	로봇 기준의 position 정보와 orientation 정보, 선속도와 각속도 정보.
amcl_pose	geometry_msgs/PoseWithCovarianceStamped [http://docs.ros.org/en/api/geometry_msgs/html/msg/PoseWithCovarianceStamped.html]	공분산과 함께 맵에서 로봇의 추정된 포즈 지도상 기준의 로봇 position 정보와 orientation 정보.
joy	sensor_msgs/Joy axes[0.0, 0.0, 0.0, 0.0, 0.0, 0.0] buttons[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]	조이스틱 입력에 대한 스틱과 버튼정보. (단, 조이스틱 입력이 있을 때만 Update)
PoseRest	data	data 가 1이되면, 로봇의 Odometry 가 모두 0으로 리셋. (리셋 후에는 0으로 변경됨) 초기 프로그램 실행 시와 충전스테이션 도킹완료 시 자동호출 됨.
cmd_vel	data	로봇의 선속도와 각속도가 입력되는 정보. (Navigation 동작 시 Update 됨)
accel_vel	data	주의! _가/감속 슬로프 시간을 주는 기능으로서 주행성능에 영향을 줄 수 있습니다.
Servo_ON	data	로봇의 모터 전원 On/Off 정보. (Data 가 1이면 On, 2이면 Off, 0이면 마지막 상태 유지)
tetra_battery	Data	로봇 배터리 정보 0~100까지의 배터리 용량 %가 Update 됨
docking_status	data	로봇과 충전스테이션 간의 도킹 상태를 표시. 1: Nomal 상태. 2: 로봇의 단자와 충전스테이션의 단자가 접촉된 상태. 3: 도킹동작 완료 후 충전을 시작하는 상태. 4: Error 가 발생한 상태. 5: 충전이 완료된 상태. 11: Conveyor Loading station Contact & Loading possible

		12: Conveyor Loading station Contact & Loading impossible 13: Conveyor Unloading station Contact & Unloading possible 14: Conveyor Unloading station Contact & Unloading impossible
emg_state	data	로봇에 장착된 비상정지 스위치의 On/Off 상태 정보. 1: 비상정지 스위치 On 2: 비상정지 스위치 Off
bumper_data	data	로봇 전방에 장착된 범퍼스위치의 On/Off 상태 정보. 1: 범퍼입력 On 2: 범퍼입력 Off
bumper_pointcloud	data[(PointCloud2)]	로봇 전방에 장착된 범퍼스위치의 On/Off 상태를 Pointcloud2 데이터로 변환한 정보
move_base/result	move_base_msgs/MoveBaseActionResult - status.status - status.text	로봇의 Navigation 동작에 대한 상태 표시 status: 상태별 번호가 리턴 text: 상태 설명(영문 문자열)
move_base/TebLocalPlanner ROS/global_plan	http://docs.ros.org/en/api/nav_msgs/html/msg/Path.html	로봇 Navigation 글로벌 계획 (시각화 목적으로만 사용)
move_base/TebLocalPlanner ROS/local_plan	http://docs.ros.org/en/api/nav_msgs/html/msg/Path.html	로봇 Navigation 로컬 계획 (시각화 목적으로만 사용)
move_base/TebLocalPlanner ROS/obstacles	http://docs.ros.org/en/api/costmap_converter/html/msg/ObstacleArrayMsg.html	장애물의 정보제공 토픽
Ultrasonic_D_L	sensor_msgs/Range http://docs.ros.org/en/noetic/api/sensor_msgs/html/msg/Range.html	로봇의 좌측 하단에 장착된 초음파 센서 데이터 (단위는 m)
Ultrasonic_D_R	sensor_msgs/Range http://docs.ros.org/en/noetic/api/sensor_msgs/html/msg/Range.html	로봇의 우측 하단에 장착된 초음파 센서 데이터 (단위는 m)
Ultrasonic_R_L	sensor_msgs/Range http://docs.ros.org/en/noetic/api/sensor_msgs/html/msg/Range.html	로봇의 후방 좌측에 장착된 초음파 센서 데이터 (단위는 m)
Ultrasonic_R_R	sensor_msgs/Range http://docs.ros.org/en/noetic/api/sensor_msgs/html/msg/Range.html	로봇의 후방 우측에 장착된 초음파 센서 데이터 (단위는 m)
range_points_DL	sensor_msgs/PointCloud2 http://docs.ros.org/en/noetic/api/sensor_msgs/html/msg/PointCloud2.html	로봇의 좌측 하단에 장착된 초음파에 대한 range data 를 PointCloud2 data 로 변환
range_points_RL	sensor_msgs/PointCloud2 http://docs.ros.org/en/noetic/api/sensor_msgs/html/msg/PointCloud2.html	로봇의 우측 하단에 장착된 초음파에 대한 range data 를 PointCloud2 data 로 변환
range_points_RR	sensor_msgs/PointCloud2 http://docs.ros.org/en/noetic/api/sensor_msgs/html/msg/PointCloud2.html	로봇의 후방 좌측에 장착된 초음파에 대한 range data 를 PointCloud2 data 로 변환
range_points_DR	sensor_msgs/PointCloud2 http://docs.ros.org/en/noetic/api/sensor_msgs/html/msg/PointCloud2.html	로봇의 후방 우측에 장착된 초음파에 대한 range data 를 PointCloud2 data 로 변환
map	http://docs.ros.org/en/api/nav_msgs/html/msg/OccupancyGrid.html	환경지도 정보
docking_progress	data	도킹상태를 알 수 있는 토픽 -1: 도킹실패.

		1: 도킹명령 시작. 2: 마커의 위치까지 접근. 3: 자세에 대한 보정_1 4: 자세에 대한보정_2 5: 스테이션 위치까지 직진. 6: 도킹완료.
ar_pose_marker	ar_pose_marker (ar_track_alvar/AlvarMarkers)	검출된 ar_tag 의 포즈 목록이 출력됩니다. - ar_tag 의 ID 정보, 위치/자세 정보
camera1/.. camera2/..	https://github.com/IntelRealSense/realsense-ros	TETRA-DS5전방 하단 좌/우에 장착된 각각의 RGBD-Camera 관련 정보.
depthimage_to_laserscan1/.. depthimage_to_laserscan2/..	http://wiki.ros.org/depthimage_to_laserscan	TETRA-DS5전방 하단 좌/우에 장착된 각각의 RGBD-Camera 의 Pointcloud 정보를 Laser scan 정보로 변환한 정보.
particlecloud	geometry_msgs/PoseArray http://docs.ros.org/en/api/geometry_msgs/html/msg/PoseArray.html	필터에서 유지 관리하는 포즈 추정값 세트입니다.
scan	sensor_msgs/LaserScan http://docs.ros.org/en/api/geometry_msgs/html/msg/PoseArray.html	SICK TIM571에서 획득한 Laser Scan Data 정보입니다.
pcl_1	sensor_msgs/LaserScan http://docs.ros.org/en/api/geometry_msgs/html/msg/PoseArray.html	TETRA 의 전방 하단 좌측에 위치한 RGBD-Camera(camera1)의 Point Cloud 를 Laser Scan Data 로 변환한 정보입니다.
pcl_2	sensor_msgs/LaserScan http://docs.ros.org/en/api/geometry_msgs/html/msg/PoseArray.html	TETRA 의 전방 하단 우측에 위치한 RGBD-Camera(camera2)의 Point Cloud 를 Laser Scan Data 로 변환한 정보입니다.
usb_cam/..	http://wiki.ros.org/usb_cam	후방 USB 카메라 관련 정보입니다.
marker/node	http://docs.ros.org/en/api/visualization_msgs/html/msg/Marker.html	작성된 환경지도(map)상에 기록된 Landmark(ar_tag)의 정보가 담겨있습니다. 저장한 Landmark 의 개수 대로 출력됩니다.
gpio_status	GPIO_msg Input0, Input1,Input2,Input3,Input4,Input5,Input6, Input7, Output0,Output1,Output2,Output3,Output4,Output5,Output6,Output7	TETRA-DS5에 내장된 Digital Input8개와 Digital Output8개의 상태 정보. * 1은 On, 0은 Off 를 의미합니다. * GPIO 라는 custom msg 를 사용합니다.
Camera1/Imu	sensor_msgs/Imu http://docs.ros.org/en/noetic/api/sensor_msgs/html/msg/Imu.html	Realsense D435i 1번으로부터 전달받은 IMU 정보입니다. * geometry_msgs/Quaternion orientation: N/A * geometry_msgs/Vector3 angular_velocity: x, y, z axis * geometry_msgs/Vector3 linear_acceleration: x, y, z axis
battery_voltage	data (Float64)	TETRA-DS5 배터리의 전압(Voltage)정보입니다. * 0.1V 단위로 값이 업데이트 됩니다.
battery_current	data (Float64)	TETRA-DS5 배터리의 전류(Current)정보입니다. * 0.1A 단위로 값이 업데이트 됩니다.
conveyor_loadcell	data (Float64)	Conveyor 에 내장된 Loadcell 에서 측정한 weight 정보입니다. * 0.1Kg 단위로 업데이트 되며, 일정 시간평균을 낸 값을 업데이트 합니다.

		* 단, 물건이 없는 상태는 0.0~ 0.3Kg 값을 출력합니다. (노이즈 포함)
conveyor_sensor	data	TETRA 에 장착된 Conveyor 의 입구/출구 센서 상태값을 업데이트 합니다. * data: 0 --> 입구/출구 센서 모두 Off 상태. * data: 1 --> 입구 센서 On 상태. * data: 2 --> 출구 센서 On 상태. * data: 3 --> 입구/출구 센서 모두 On 상태.
conveyor_movement	data(Int32)	TETRA 에 장착된 Conveyor 의 동작 상태값을 업데이트 합니다. * data: 0 or 1 --> 동작정지 상태 * data: 3 --> 로딩 동작 중. * data: 5 --> 언로딩 동작 중.
imu/data	http://docs.ros.org/en/melodic/api/sensor_msgs/html/msg/Imu.html	TETRA 에 장착된 IMU 센서 데이터