

Part III. Autoware Programming (2 hours)

숭실대학교 ICHTHUS 팀
김강희 교수(khkim@ssu.ac.kr)
이효은, 최규진, 백한나 연구원

Outline

- ❖ pcap play
- ❖ Lidar sensors
- ❖ nodelet lidar driver

pcap Play

Step 1: Setup, Build, Install nodelet_lidar_driver

- ❖ \$ cd # change to home directory
- ❖ \$ cp -r ~/shared_dir/pcap_demo/nodelet_lidar_driver
Autoware/src/autoware/core_perception
- ❖ \$ cd Autoware
- ❖ \$ colcon build --packages-select nodelet_lidar_driver
setup, build and install nodelet_lidar_driver
- ❖ \$ source install/setup.bash
to register nodelet_lidar_driver package into ROS pkg path
- ❖ \$ gnome-terminal # prepare another terminal to run rviz
- ❖ \$ vi install/nodelet_lidar_driver/share/nodelet_lidar_driver/cfg/
nodelet_frontend_default.yaml
lidar_num 값 설정: 3
pcap file 경로 설정: "/home/autoware/shared_dir/pcap_demo/20190821_kcity_ouster_degree0.pcap"
pcap_wait_factor 값 설정: 0.6
- ❖ \$ roslaunch nodelet_lidar_driver nodelet_lidar_driver.launch
- ❖ [terminal] \$ rviz

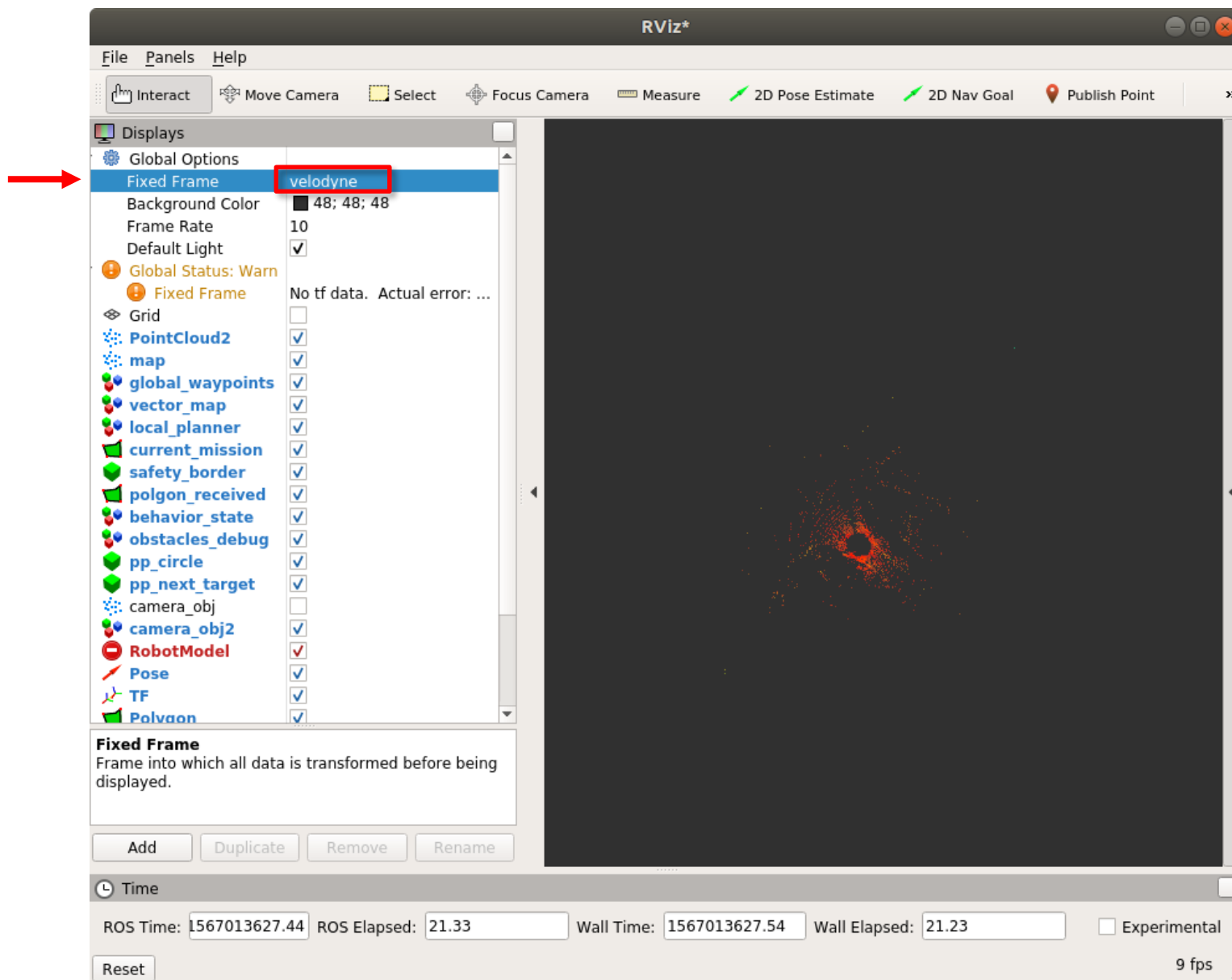
Note: Ouster OS-1-64 라이더를 구동하고 **socket**를 통해 **lidar msg**를 수신하는 경우에는 **nodelet_lidar_driver** 실행 전에 다음 명령어를 먼저 입력해야 함

\$ nc 192.168.1.2 7501 set_udp_ip 192.168.1.77

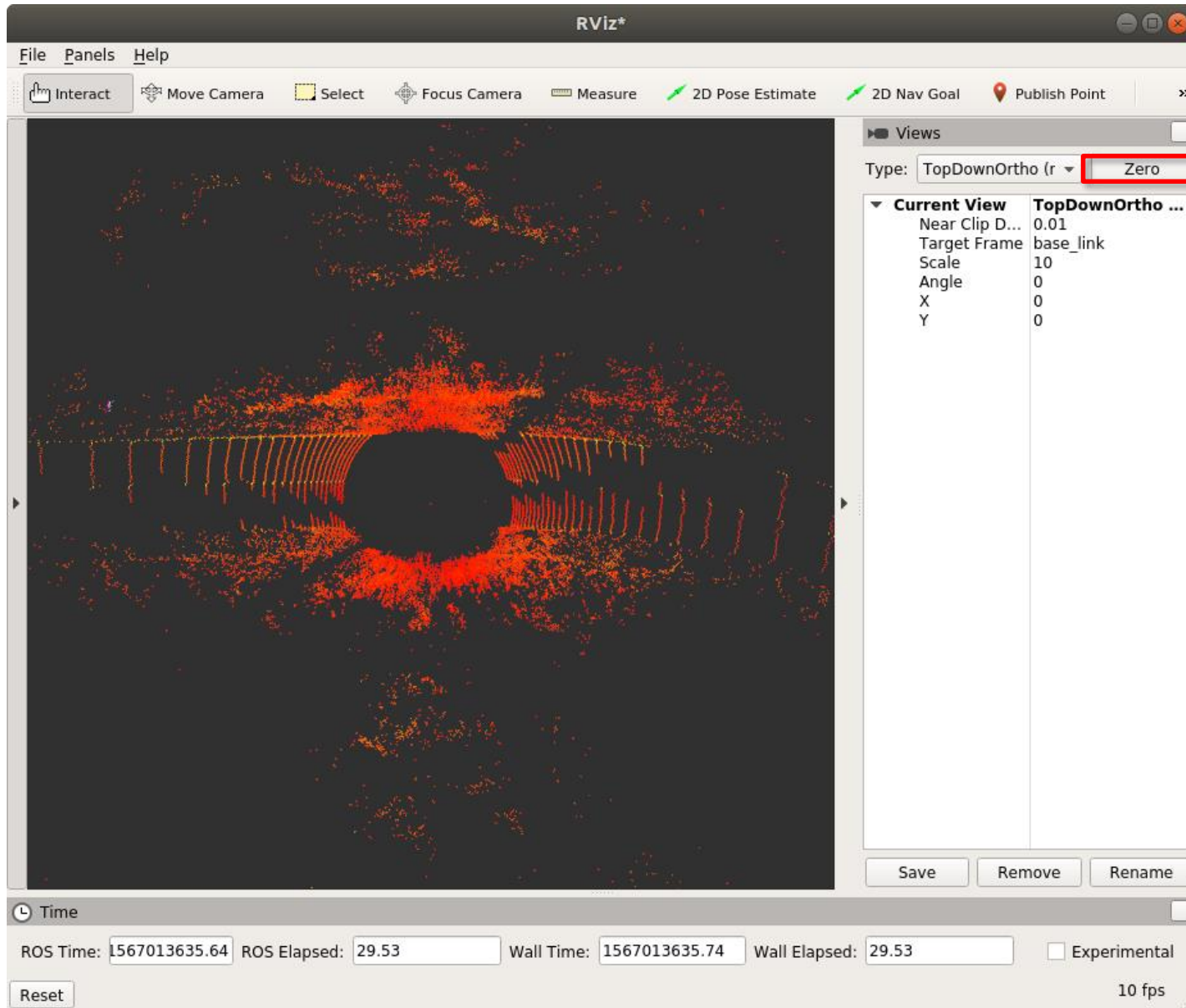
192.168.1.2 ouster ip,

192.168.1.77 host ip

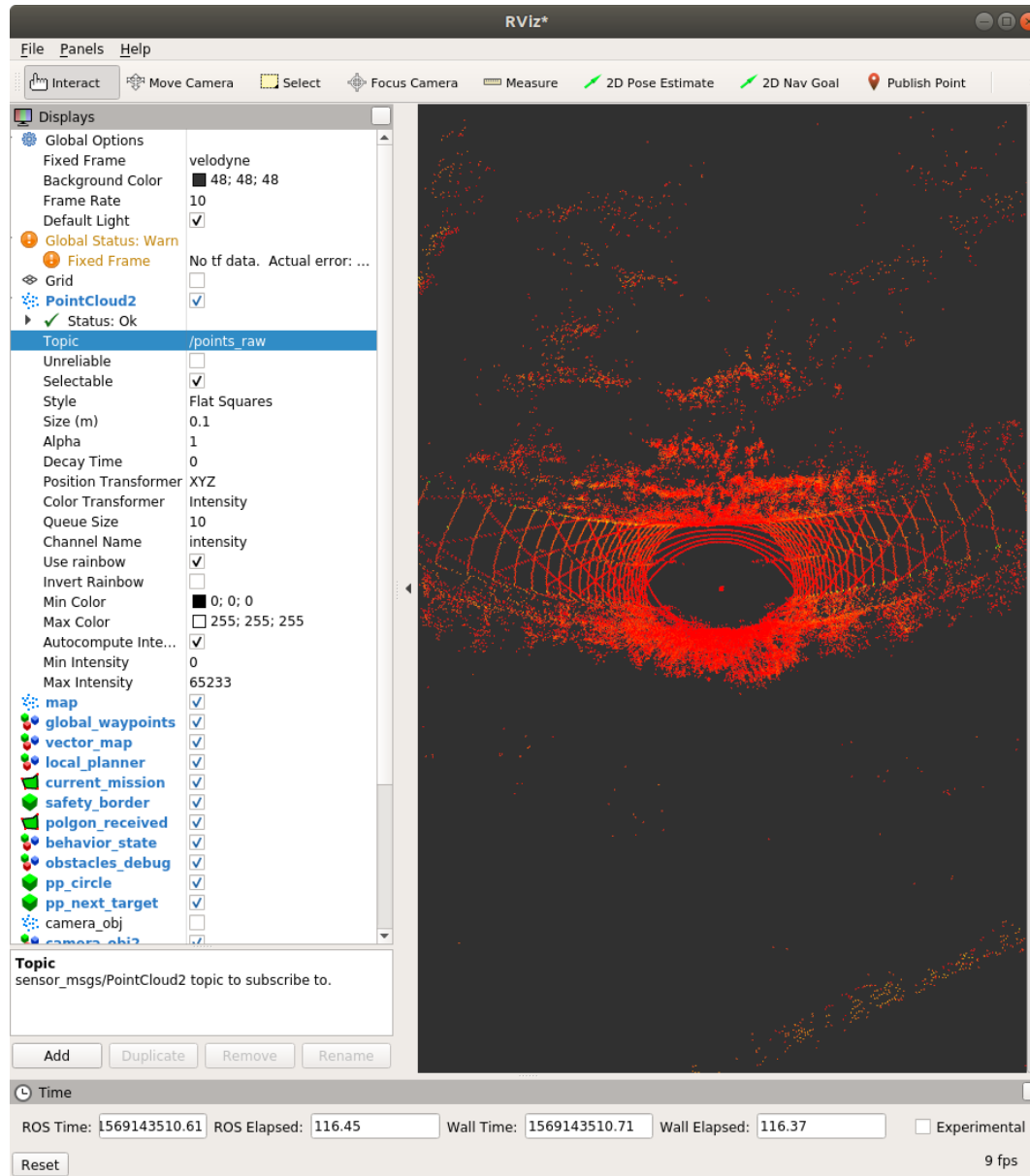
Step 2: Run 'rviz'



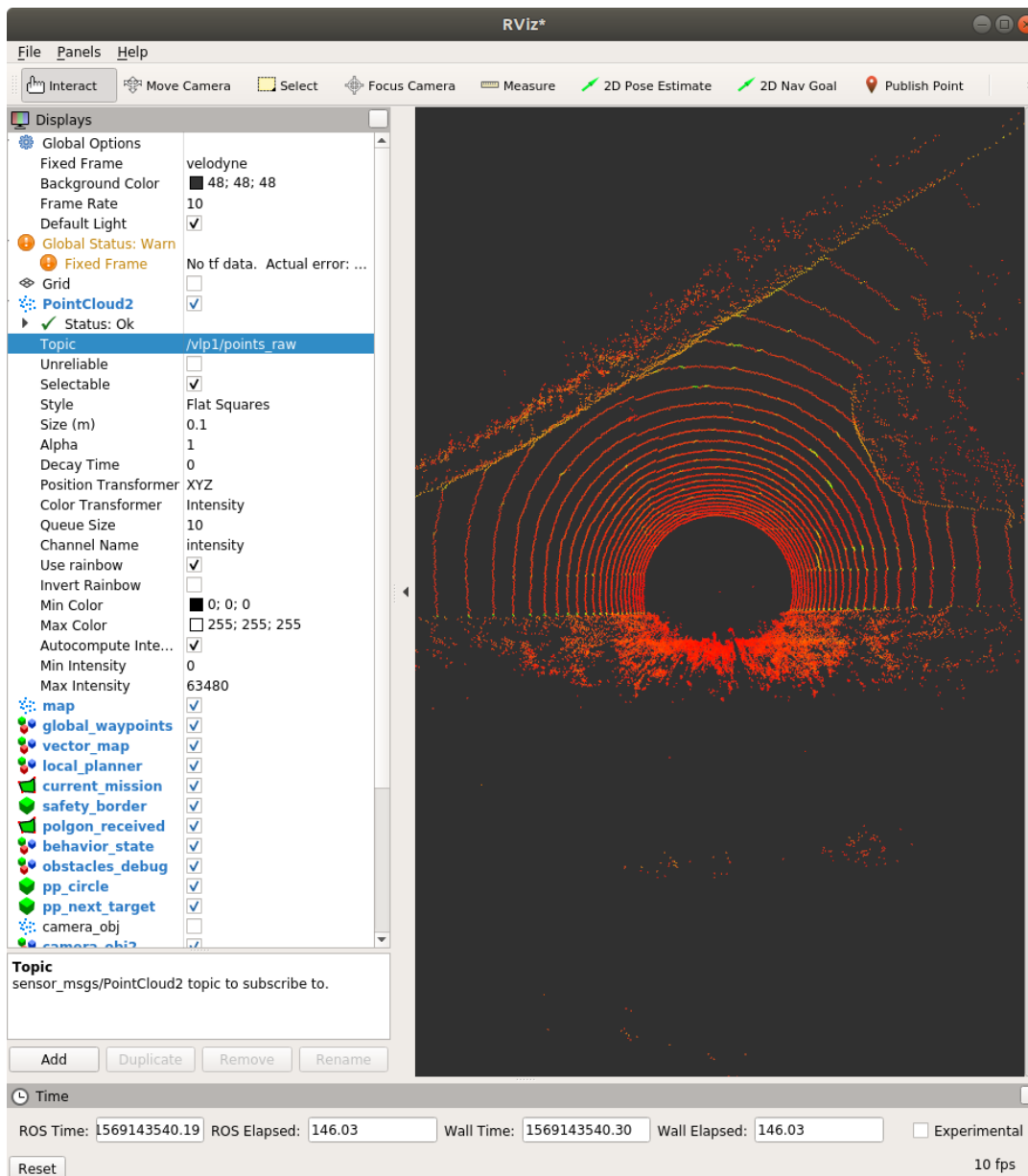
Step 2: Run 'rviz' : /points_raw (lidar_num == 1)



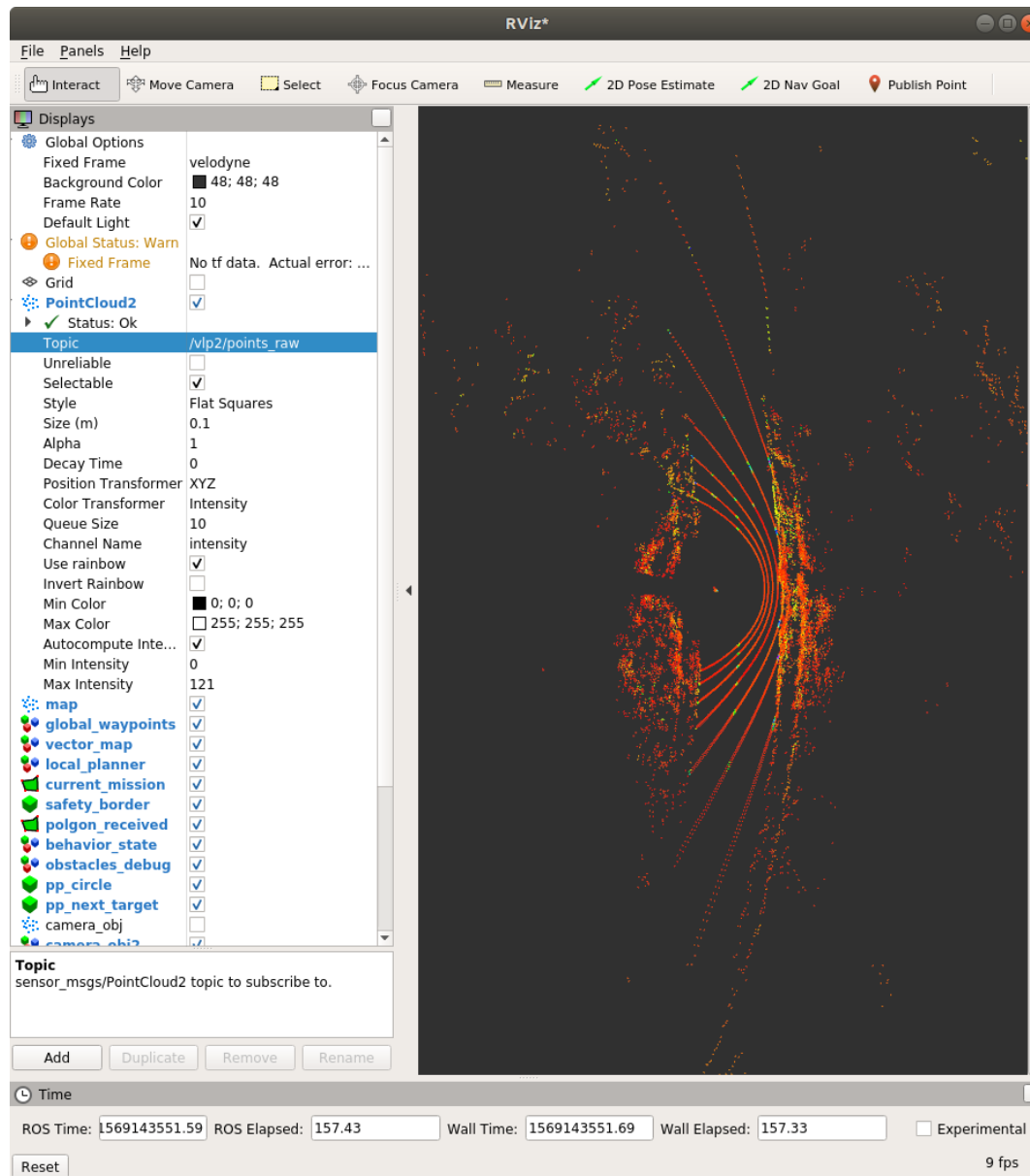
Step 2: Run 'rviz' : /points_raw (lidar_num == 3)



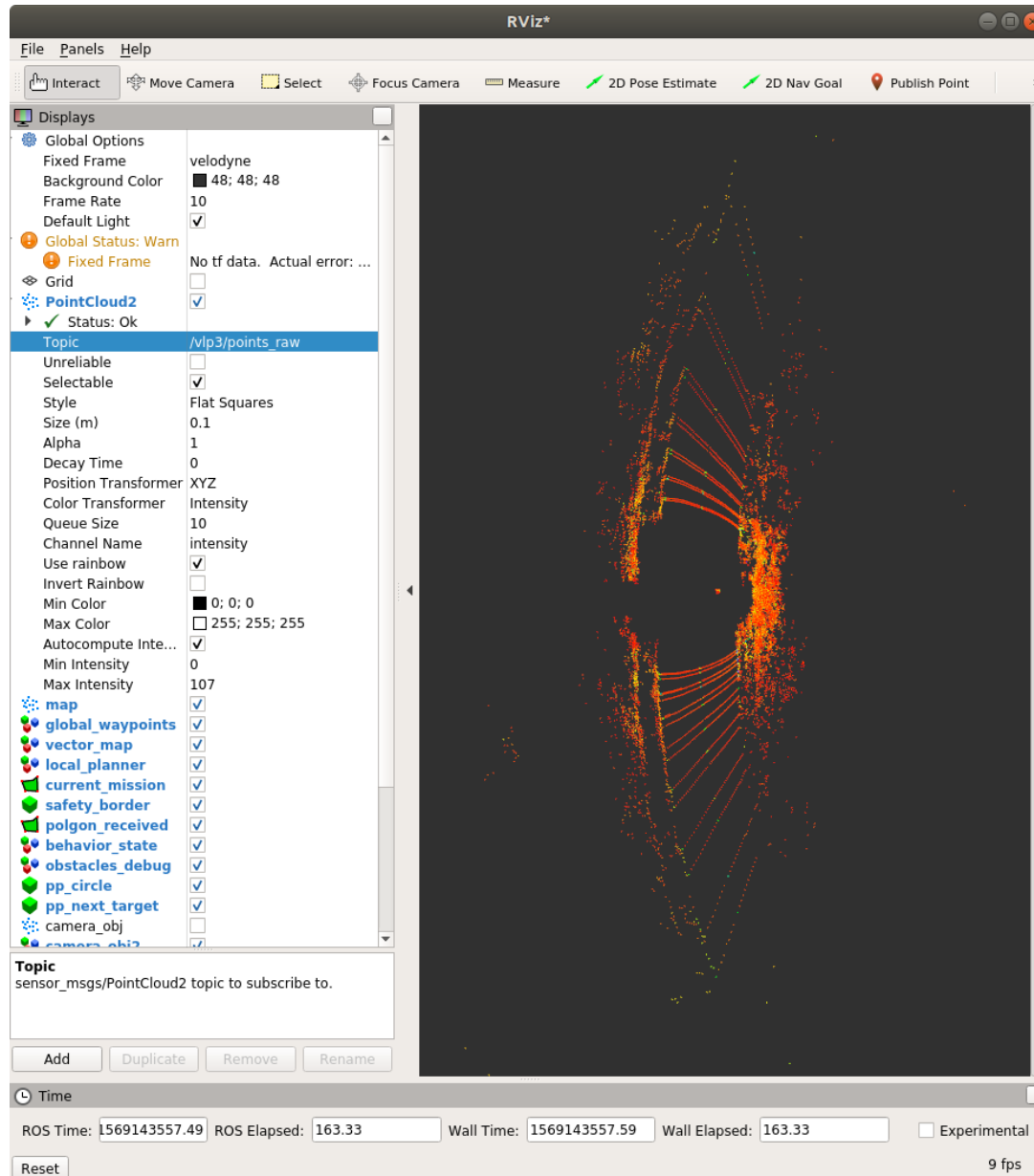
Step 2: Run 'rviz' : /vlp1/points_raw



Step 2: Run 'rviz' : /vlp2/points_raw



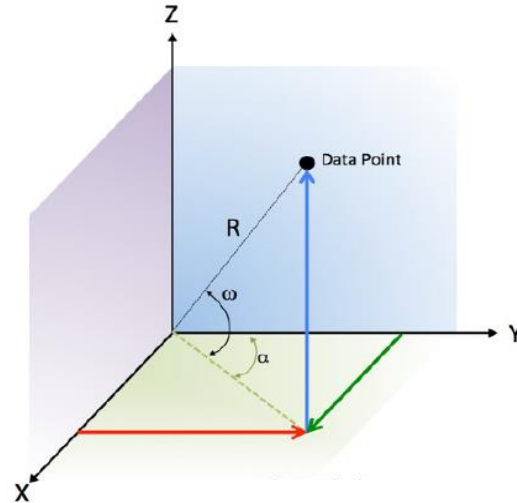
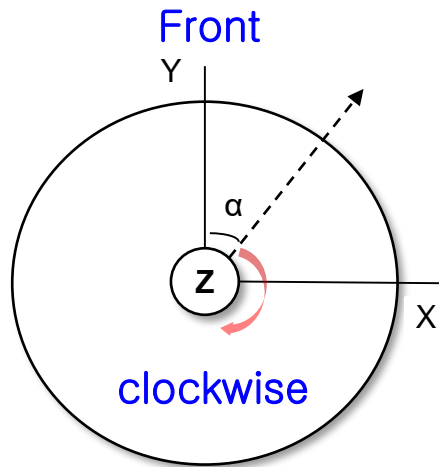
Step 2: Run 'rviz' : /vlp3/points_raw



LiDAR Sensors

Lidar : Geometry

Velodyne VLP16



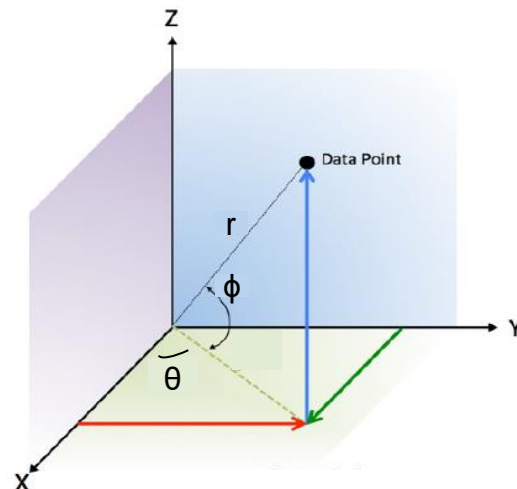
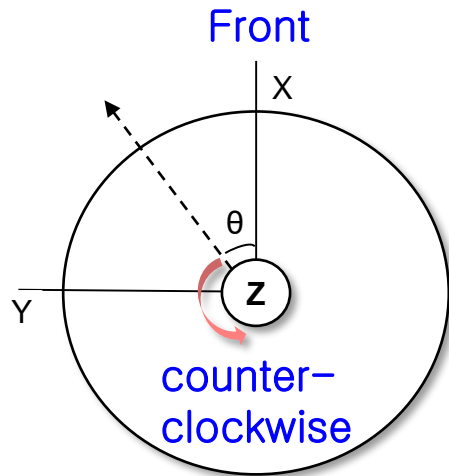
수식

$$x = R * \cos \omega * \sin \alpha$$

$$y = R * \cos \omega * \cos \alpha$$

$$z = R * \sin \omega$$

Ouster OS-1-64



수식

$$x = r \cos \theta \cos \varphi$$

$$y = -r \sin \theta \cos \varphi$$

$$z = r \sin \varphi$$

Lidar : Message Structure (VLP16)

Velodyne VLP16 (Single Return Mode)

Azimuth N+23
(Firing sequence 23) ←

size	Data Block 0	Data Block 1	...	Data Block 11
2 bytes	Flag x FFEE	Flag x FFEE	...	Flag x FFEE
2 bytes	Azimuth N	Azimuth N+2	...	Azimuth N+22
3 bytes	Channel 0 Data	Channel 0 Data	...	Channel 0 Data
...	...	Channel 1~14 Data	...	Channel 1~14 Data
3 bytes	Channel 15 Data	Channel 15 Data	...	Channel 15 Data
3 bytes	Channel 0 Data	Channel 0 Data	...	Channel 0 Data
...	...	Channel 1~14 Data	...	Channel 1~14 Data
3 bytes	Channel 15 Data	Channel 15 Data	...	Channel 15 Data
				TimeStamp (4 bytes)
				Factory (2 bytes)

Azimuth N 을 의미
(Firing sequence 0)

Azimuth N+1 필드가 없지만,
Azimuth N+1 을 의미함
(Firing sequence 1)

총 1206 bytes

Lidar : Message Structure (VLP16)

❖ Firing sequence

- 하나의 azimuth 에 대해서 모든 채널들의 레이저가 발사됨을 의미
- 레이저 재충전 시간이 포함되어 있음

❖ Laser channel

- 하나의 channel 은 단일 903 nm 레이저 emitter 및 detector 쌍임
- 각 channel 은 센서의 수평면을 기준으로 특정 높이 각으로 고정됨
- 각 channel 은 고유한 laser ID 번호가 부여됨

❖ Data point

- data point 는 하나의 channel 로부터 발사된 laser 파동의 반사 신호에 대한 측정값
- data point 는 3 byte (distance 2 bytes, calibrated reflectivity 1 byte) 로 구성됨
 - ❖ distance 는 부호 없는 정수
 - ❖ calibrated reflectivity 는 NIST 기준으로 하고, 0~255의 크기

Lidar : Message Structure (VLP16)

❖ Azimuth

- laser firing 들의 사이 각으로서 2 bytes 크기임
- azimuth 값은 부호 없는 정수로 1/100 의 각도를 표현함
 - ❖ 예: 27742 \rightarrow 277.42°

❖ Data block

- 16개 laser channel 로 구성된 2개의 firing sequence 를 표현함
- 각 패킷에는 12개의 data block , 24개의 firing sequence 가 포함됨

❖ Time stamp

- 4 bytes 크기의 부호 없는 정수로서 단위는 μs 임
- 첫번째 data block 의 첫번째 firing sequence 에서 첫번째 data point 의 측정 순간을 의미함

❖ Factory bytes

- 2 bytes 크기로서 모든 data 패킷에 포함됨
- 패킷에서 azimuths 와 data point 가 구성되는 방식을 표현함

Lidar : Message Structure (OS-1-64)

Ouster OS-1-64

word	Azimuth Block 0	Azimuth Block 1	...	Azimuth Block 15
word 0,1	Timestamp	Timestamp	...	Timestamp
word 2[0:15]	Measurement ID	Measurement ID	...	Measurement ID
word 2[16:31]	Frame ID	Frame ID	...	Frame ID
word 3	Encoder Count	Encoder Count	...	Encoder Count
word 4,5,6	Channel 0 Data Block	Channel 0 Data Block	...	Channel 0 Data Block
word 7,8,9	Channel 1 Data Block	Channel 1 Data Block	...	Channel 1 Data Block
...
word 193,194,195	Channel 63 Data Block	Channel 63 Data Block	...	Channel 63 Data Block
word 196	Packet Status	Packet Status	...	Packet Status

*word = 32 bits

Lidar : Message Structure (OS-1-64)

- ❖ Time stamp
 - ns 단위의 측정 time stamp
- ❖ Measurement ID
 - 0부터 시작하여 순차적으로 증가하는 azimuth ID
 - 512 mode 에서는 0 부터 511 까지, 1024 mode 에서는 0 부터 1023 까지, 2048 mode 에서는 0 부터 2047 까지 값을 가짐
- ❖ Frame ID
 - Lidar 의 rotation count 를 의미함
- ❖ Encoder count
 - azimuth angle 를 의미하는 encoder tick 개수
 - 0 부터 최대 90111 의 값을 가짐
 - azimuth angle 마다 2048 mode 에서는 44 ticks 가 증가하고, 1024 mode 에서는 88 ticks 가 증가하고, 512 mode 에서는 176 ticks 가 증가함

Lidar : Message Structure (OS-1-64)

- ❖ Data block (사실상 data point를 의미함)
 - 64 pixel 각각에 대한 3개의 data words 로 구성됨
 - ❖ 20 bits 의 range : 12 mm 단위의 거리 측정값
 - ❖ 16 bits 의 reflectivity : laser 반사 신호의 세기
 - ❖ 16 bits 의 signal photons : laser 반사 신호의 photons
 - ❖ 16 bits 의 ambient noise photons : 주변 noise 의 photons
- ❖ Packet status
 - Good = 0xFFFFFFFF or Bad = 0x0

Lidar : Message Structure

Velodyne VLP16

Ouster OS-1-64

Distance – 2 bytes,
Calibrated reflectivity (intensity) – 1 byte

Terms

Firing sequence
Laser channel
Data block
Factory bytes
Data point
Timestamp μs
Azimuth

Range (distance) – 20 bits
Reflectivity(intensity) – 16 bits

Data block
Timestamp ns
Measurement ID
Frame ID
Encoder count
Packet status

*  같은 의미

Lidar : Timing

Velodyne VLP16

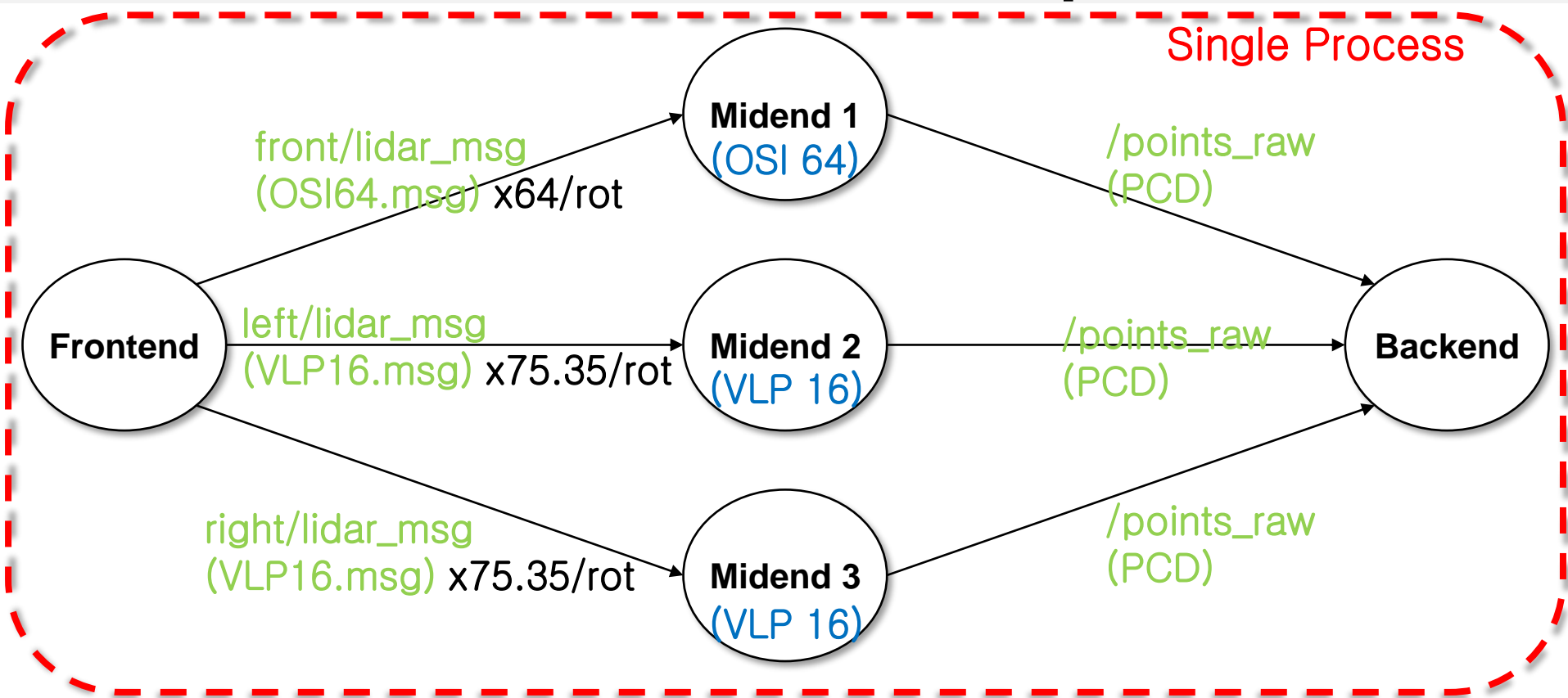
- ❖ Firing sequence timing = $55.296 \mu\text{s}$ (cannot be changed)
- ❖ Assuming lidar rotation rate = 10 hz
 - Azimuth = 0.2°
 - 18084 firings / sec
 - $753.5 \text{ messages / sec} = 18084 \text{ firings / sec} \times 1 \text{ message / 24 firings}$
 - 75.35 messages / rotation
 - 1206 bytes / message \rightarrow 1 packet

Ouster OS-1-64

- ❖ Assuming 1024 mode \times lidar rotation rate = 10 hz
- ❖ Firing sequence timing = $97.656 \mu\text{s}$ (programmable by selecting mode)
 - Azimuth = 0.35°
 - 10240 firings / sec
 - $640 \text{ messages / sec} = 10240 \text{ firings / sec} \times 1 \text{ message / 16 firings}$
 - 64 messages / rotation
 - 12608 bytes / message \rightarrow 9 packets { 1472, 1480, ..., 1480, 776 }

nodelet_lidar_driver

nodelet_lidar_driver : Node Graph



*VLP16.msg
time ts
unit8[1206] buf
unit32 size

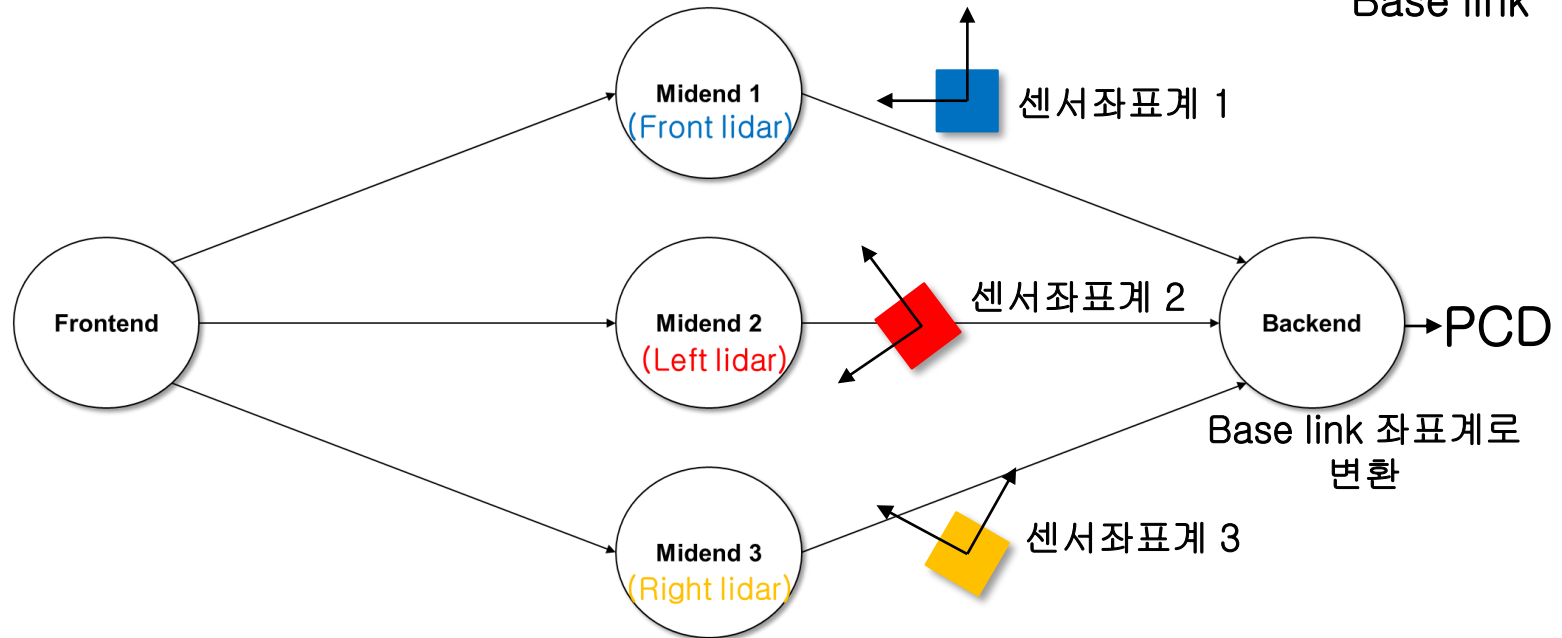
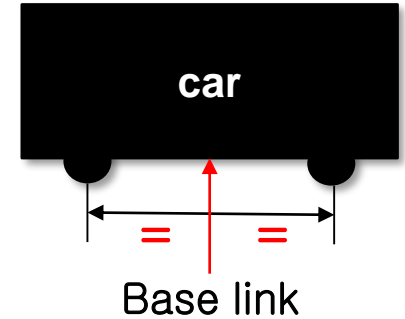
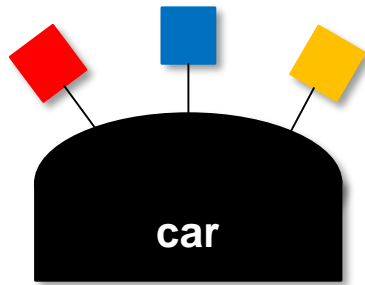
*OSI64.msg
time ts
unit8[12608] buf
unit32 size

*토픽명
(형식)
*PCD = point cloud

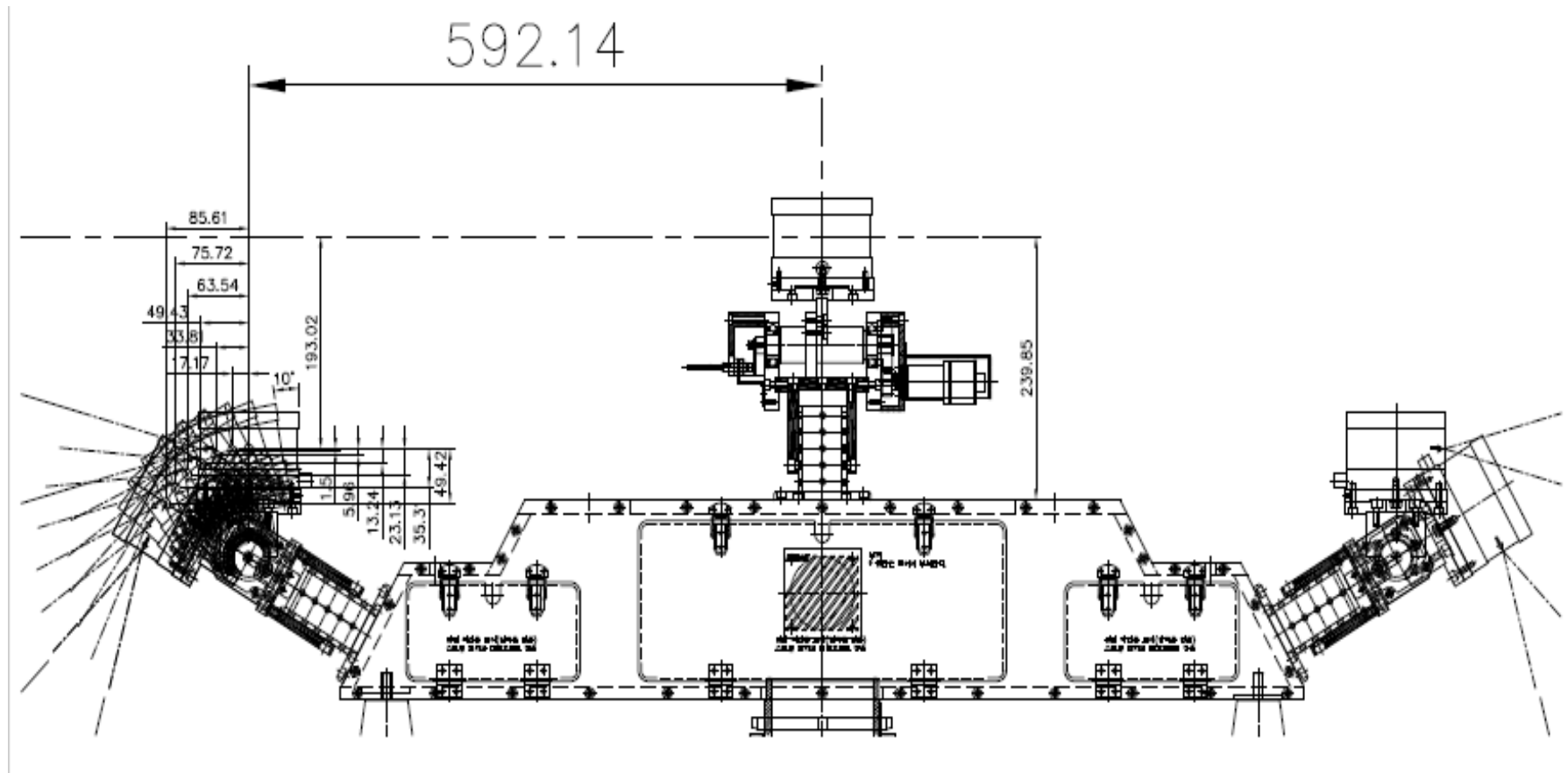
nodelet_frontend_default.yaml

```
1  ###
2  ### number of lidars used
3  ###
4  lidar_num : 1
5
6  ###
7  ### pcap file used to replay
8  ###
9  pcap_file : "/home/autoware/shared_dir/pcap_demo/20190821_kcity_ouster_degree0.pcap"
10 pcap_wait_factor : 0.6
11 ###
12 ### lidar model selection:
13 ### 0 means VLP16 (Velodyne Puck 16 model)
14 ### 1 means OSI64 (Ouster-1-64 model)
15 ###
16 mode_front : 1
17 mode_left : 0
18 mode_right : 0
19
20 ###
21 ### ip address of each lidar
22 ###
23 ip_add_front : "192.168.1.2"
24 ip_add_left : "192.168.1.202"
25 ip_add_right : "192.168.1.203"
```

nodelet_lidar_driver : Transform



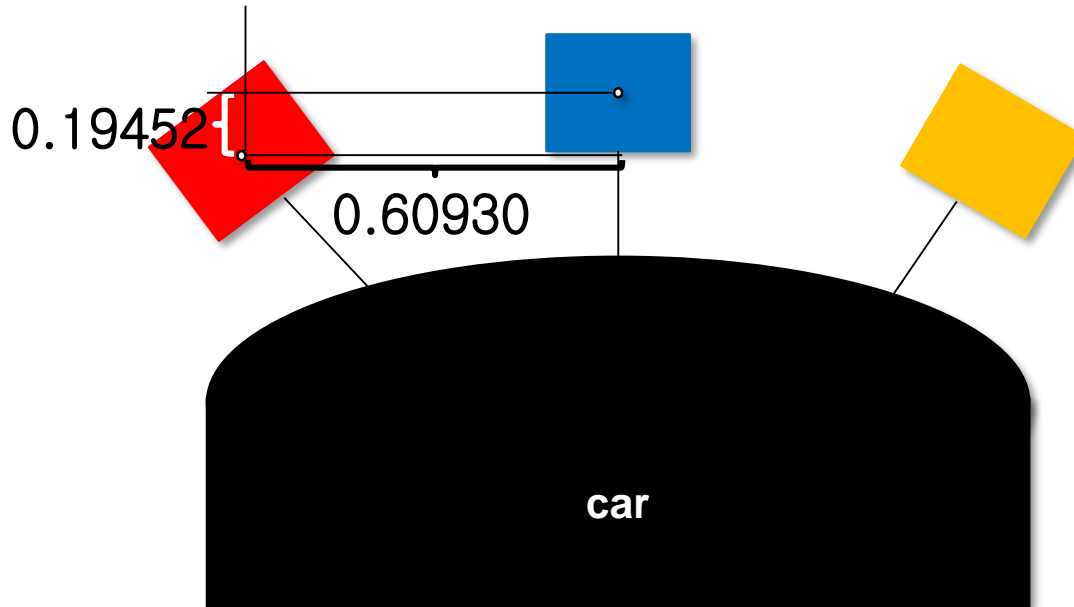
nodelet_lidar_driver : Transform



nodelet_lidar_driver : Transform



Front lidar 0°
Left, Right lidar 10°



*m 단위

```
CloudIn1: # Front
  topic_name: /vlp1/points_raw
  pose_yaw: 0
  pose_pitch: 0
  pose_roll: 0
  pose_x: 0
  pose_y: 0
  pose_z: 0.0045
CloudIn2: # Left
  topic_name: /vlp2/points_raw
  pose_yaw: 90
  pose_pitch: 10
  pose_roll: 0
  pose_x: 0
  pose_y: 0.60930
  pose_z: -0.19902
CloudIn3: # Right
  topic_name: /vlp3/points_raw
  pose_yaw: -90
  pose_pitch: 10
  pose_roll: 0
  pose_x: 0
  pose_y: -0.60930
  pose_z: -0.19902
```

nodelet_lidar_driver : Transform

❖ lidar 기울어진 각도에 따른 yaml 파일

- pose_yaw, pitch, roll 정보와 x, y, z 좌표 정보를 가지고 있음)
- Autoware/src/core_perception/nodelet_lidar_driver/cfg_backend

Front lidar 각도	Left, Right 각도	.yaml 파일
0°	0°	Clouds_0_0_height_45.yaml
	5°	Clouds_0_5_height_45.yaml
	10°	Clouds_0_10_height_45.yaml
	15°	Clouds_0_15_height_45.yaml
	20°	Clouds_0_20_height_45.yaml
5°	0°	Clouds_5_0_height_45.yaml
	5°	Clouds_5_5_height_45.yaml
	10°	Clouds_5_10_height_45.yaml
	15°	Clouds_5_15_height_45.yaml
	20°	Clouds_5_20_height_45.yaml
10°	0°	Clouds_10_0_height_45.yaml
	5°	Clouds_10_5_height_45.yaml
	10°	Clouds_10_10_height_45.yaml
	15°	Clouds_10_15_height_45.yaml
	20°	Clouds_10_20_height_45.yaml

nodelet_lidar_driver : Source Tree

- ❖ nodelet_lidar_driver/
 - cfg/ : .yaml files
 - cfg_backend/ : .yaml files for tf transform
 - include/ : header files
 - launch/ : launch file
 - msg/ : .msg files
 - src/ : source files

nodelet_lidar_driver : package.xml

```

1  <?xml version="1.0"?>
2  <package format="2">
3    <name>nodelet_lidar_driver</name>
4    <version>0.9.0</version>
5    <description>The nodelet_lidar_driver package</description>
6    <maintainer email="kim.kanghee@gmail.com">Kanghee Kim</maintainer>
7    <license>Apache 2</license>
8
9    <buildtool_depend>autoware_build_flags</buildtool_depend>
10   <buildtool_depend>catkin</buildtool_depend>
11
12   <build_depend>message_generation</build_depend>
13
14   <depend>roscpp</depend>
15   <depend>sensor_msgs</depend>
16   <depend>std_msgs</depend>
17   <depend>pcl_ros</depend>
18   <depend>pcl_conversions</depend>
19   <depend>nodelet</depend>
20
21   <exec_depend>message_runtime</exec_depend>
22
23   <export>
24     <nodelet plugin="${prefix}/nodelet_lidar_driver.xml"/>
25   </export>
26 </package>

```

build tool dependency

build dependency

build & execution
dependency

execution dependency

nodelet plugin xml

nodelet_lidar_driver : nodelet_lidar_driver.xml

```
1 <library path="lib/libnodelet_lidar_driver">
2   <class name="nodelet_lidar_driver/LidarNodeletFront" type="LidarNodeletFront::LidarNodeletFront"
   base_class_type="nodelet::Nodelet">
3     <description>
4       LidarNodeletFront.
5     </description>
6   </class>
7   <class name="nodelet_lidar_driver/LidarNodeletMid" type="LidarNodeletMid::LidarNodeletMid"
   base_class_type="nodelet::Nodelet">
8     <description>
9       LidarNodeletMid.
10    </description>
11  </class>
12  <class name="nodelet_lidar_driver/LidarNodeletBack" type="LidarNodeletBack::LidarNodeletBack"
   base_class_type="nodelet::Nodelet">
13    <description>
14      LidarNodeletBack.
15    </description>
16  </class>
17 </library>
```

nodelet_lidar_driver : CMakeLists.txt

```

1  cmake_minimum_required(VERSION 2.8.3)
2  project(nodelet_lidar_driver)
3
4  #find_package(PCL REQUIRED)
5  find_package(autoware_build_flags REQUIRED)
6  find_package(catkin REQUIRED COMPONENTS
7      roscpp
8      sensor_msgs
9      std_msgs
10     pcl_ros
11     pcl_conversions
12     nodelet
13     message_generation
14 )
15
16 add_message_files(FILES
17     OSI64.msg
18     VLP16.msg
19 )
20
21 generate_messages(DEPENDENCIES
22     std_msgs
23 )
24
25 catkin_package(
26     CATKIN_DEPENDS
27     roscpp
28     sensor_msgs
29     std_msgs
30     pcl_ros
31     pcl_conversions
32     nodelet
33     message_runtime
34     #DEPENDS PCL
35 )
36
37 # Resolve system dependency on yaml-cpp, which apparently does not
38 # provide a CMake find_package() module.
39 find_package(PkgConfig REQUIRED)
40 pkg_check_modules(YAML_CPP REQUIRED yaml-cpp)
41 find_path(YAML_CPP_INCLUDE_DIR NAMES yaml_cpp.h PATHS ${YAML_CPP_I
42 find_library(YAML_CPP_LIBRARY NAMES YAML_CPP PATHS ${YAML_CPP_LIBR
43 link_directories(${YAML_CPP_LIBRARY_DIRS})
44 if(NOT ${YAML_CPP_VERSION} VERSION_LESS "0.5")
45     add_definitions(-DHAVE_NEW_YAMLCPP)
46 endif(NOT ${YAML_CPP_VERSION} VERSION_LESS "0.5")
47

```

nodelet_lidar_driver : CMakeLists.txt (cont'd)

```

48 #####
49 ### build configuration ###
50 #####
51
52 include_directories(
53     ${catkin_INCLUDE_DIRS}
54     ${Boost_INCLUDE_DIR}
55     ${PCL_INCLUDE_DIRS}
56     include
57 )
58
59 add_library(nodelet_lidar_driver
60     src/nodelet_midend.cpp
61     src/calibration.cpp
62     src/lidar.cpp
63     src/packet.cpp
64     src/nodelet_frontend.cpp
65     src/nodelet_backend.cpp
66     src/util.cpp
67 )
68
69 set(libpcap_LIBRARIES "-lpcap")
70
71 target_link_libraries(nodelet_lidar_driver
72     ${catkin_LIBRARIES}
73     ${libpcap_LIBRARIES}
74     ${YAML_CPP_LIBRARIES}
75 )
76
77 install(TARGETS nodelet_lidar_driver
78     ARCHIVE DESTINATION ${CATKIN_PACKAGE_LIB_DESTINATION}
79     LIBRARY DESTINATION ${CATKIN_PACKAGE_LIB_DESTINATION}
80     RUNTIME DESTINATION ${CATKIN_PACKAGE_BIN_DESTINATION}
81 )
82
83 install(FILES nodelet_lidar_driver.xml
84     DESTINATION ${CATKIN_PACKAGE_SHARE_DESTINATION}
85 )
86
87 install(DIRECTORY launch/
88     DESTINATION ${CATKIN_PACKAGE_SHARE_DESTINATION}/launch
89 )
90
91 install(DIRECTORY cfg/
92     DESTINATION ${CATKIN_PACKAGE_SHARE_DESTINATION}/cfg
93 )

```