

LIMO 환경설정

목 차

1. LIMO 기본 패키지 설치

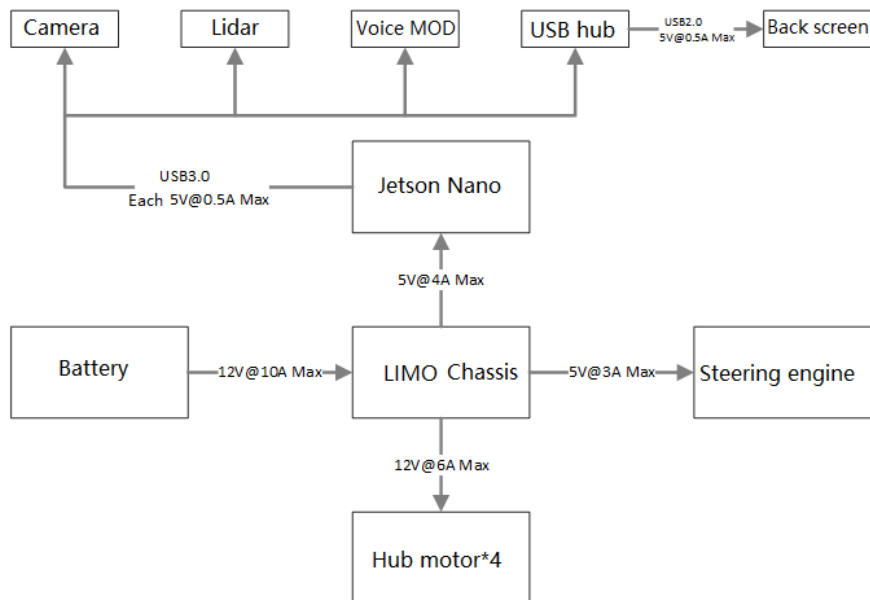
01

LIMO 기본
package
설치

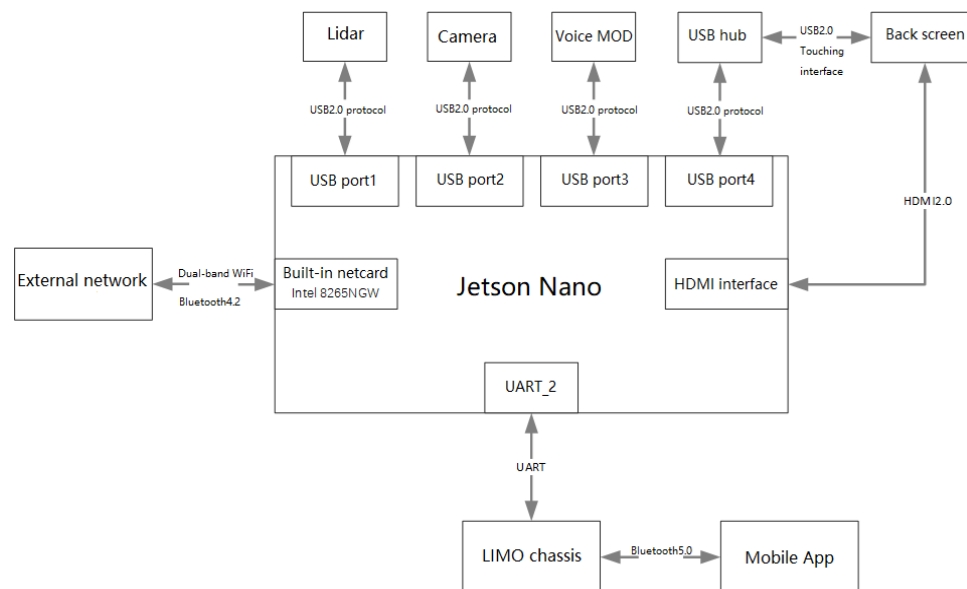
02 LIMO 기본 package 설치

- LIMO 구동을 위한 ros package 설치(모터 구동, imu, odom 등)

<LIMO 전원 diagram>



<LIMO 통신 diagram>



02 LIMO 기본 package 설치

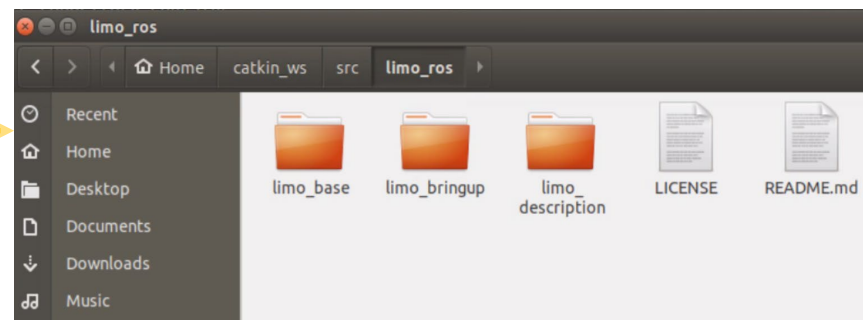
- LIMO 구동을 위한 ros package 설치(모터 구동, imu, odom 등)

- 터미널에 다음을 차례로 입력

\$ `cd ~/catkin_ws/src` or `cs`

\$ `git clone https://github.com/agilexrobotics/limo_ros.git`

- 폴더 생성 확인



- 터미널에 다음을 차례로 입력

\$ `cd ..` or `cd ~/catkin_ws/` or `cw`

\$ `catkin_make`

```
agilex@nano: ~/catkin_ws/src
agilex@nano: ~/catkin_ws/src 80x24
agilex@nano:~$ cd ~/catkin_ws/src
agilex@nano:~/catkin_ws/src$ git clone https://github.com/agilexrobotics/limo_ros.git
Cloning into 'limo_ros'...
remote: Enumerating objects: 998, done.
remote: Total 998 (delta 0), reused 0 (delta 0), pack-reused 998
Receiving objects: 100% (998/998), 50.27 MiB | 8.15 MiB/s, done.
Resolving deltas: 100% (485/485), done.
Checking out files: 100% (53/53), done.
agilex@nano:~/catkin_ws/src$ cd ..
agilex@nano:~/catkin_ws$ catkin_make
Base path: /home/agilex/catkin_ws
Source space: /home/agilex/catkin_ws/src
Build space: /home/agilex/catkin_ws/build
Devel space: /home/agilex/catkin_ws/devel
Install space: /home/agilex/catkin_ws/install
####
#### Running command: "cmake /home/agilex/catkin_ws/src -DCATKIN_DEVEL_PREFIX=/home/agilex/catkin_ws/devel -DCMAKE_INSTALL_PREFIX=/home/agilex/catkin_ws/install -G Unix Makefiles" in "/home/agilex/catkin_ws/build"
####
-- Using CATKIN_DEVEL_PREFIX: /home/agilex/catkin_ws/devel
-- Using CMAKE_PREFIX_PATH: /home/agilex/catkin_ws/devel;/opt/ros/melodic
-- This workspace overlays: /home/agilex/catkin_ws/devel;/opt/ros/melodic
-- Found PythonInterp: /usr/bin/python2 (found suitable version "2.7.17", minimum required is "2")
```

02 LIMO 기본 package 설치

- LIMO를 위한 ros package 구동 확인

- 터미널 3개에 각각을 입력

터미널 1 : `$ roslaunch limo_base limo_base.launch`

터미널 2 : `$ rostopic list`

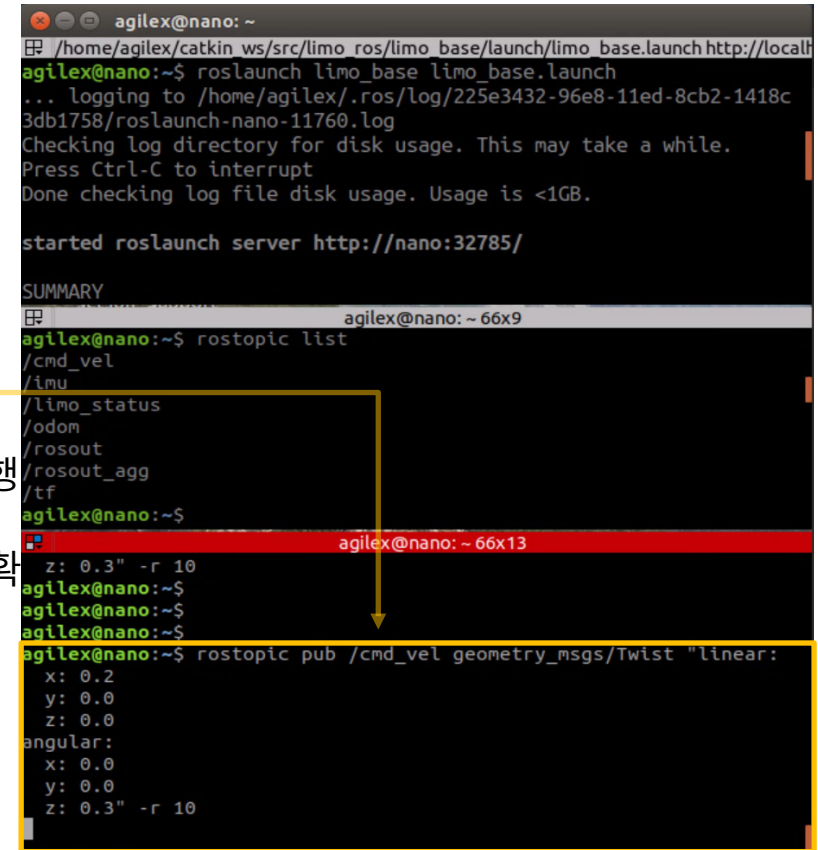
터미널 3 : `$ rostopic pub /topic_type [args ...] -r 10`

- 리모 구동 확인 : 0.2 m/s 속도, 0.3 rad 조향

- 터미널 3 을 활성화하여 ctrl + c로 종료하면 토픽 발행

- 조향 (x: 0.2, y: 0.0, z: 0.0, angular: x: 0.0, y: 0.0, z: 0.3")을 변경해서 실행하여 조향 영점 확인

```
x: 0.2
y: 0.0
z: 0.0
angular:
x: 0.0
y: 0.0
z: 0.3" -r 10
```



```
agilex@nano: ~
/home/agilex/catkin_ws/src/limo_ros/limo_base/launch/limo_base.launch http://localhost:32785/
agilex@nano:~$ roslaunch limo_base limo_base.launch
... logging to /home/agilex/.ros/log/225e3432-96e8-11ed-8cb2-1418c3db1758/roslaunch-nano-11760.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://nano:32785/

SUMMARY
agilex@nano: ~ 66x9
agilex@nano:~$ rostopic list
/cmd_vel
/limo
/limo_status
/odom
/rosout
/rosout_agg
/tf
agilex@nano:~$
agilex@nano:~ 66x13
z: 0.3" -r 10
agilex@nano:~$
agilex@nano:~$
agilex@nano:~$ rostopic pub /cmd_vel geometry_msgs/Twist "linear:
x: 0.2
y: 0.0
z: 0.0
angular:
x: 0.0
y: 0.0
z: 0.3" -r 10
```

*모두 typing 할 필요 없고 /cmd_vel까지 작성 후 tap키를 이용

02 LIMO 기본 package 설치

- LIMO를 위한 ros package 구동 확인(스크립트 작성)

- 터미널 3개에 각각을 입력

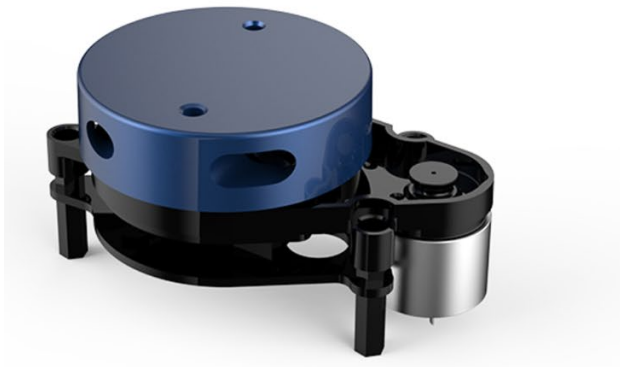
터미널 1 : `$ roslaunch limo_base limo_base.launch`

터미널 2 : `$ rosrn basic_ex limo_motor_test.py`

```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  # rospy 라이브러리와 필요한 메시지 유형(geometry_msgs.msg)을 가져옵니다.
5  import rospy
6  from geometry_msgs.msg import Twist
7
8
9  # Class_Name 클래스 정의: ROS 노드를 클래스로 정의하여 코드를 구조화합니다.
10 class Class_Name: # 1단계: 클래스 이름 정의
11     def __init__(self): # 2단계: 클래스 초기화 및 초기 설정
12         # ROS 노드를 초기화합니다. 노드 이름은 "wego_pub_node"로 지정됩니다.
13         rospy.init_node("wego_pub_node") # ROS 1단계(필수): 노드 이름 정의
14
15         # ROS 퍼블리셔(Publisher)를 설정합니다.
16         # "/cmd_vel" 토픽에 Twist 메시지를 발행하는 퍼블리셔를 만듭니다.
17         # "queue_size"는 메시지 대기열 크기를 나타냅니다.
18         self.pub = rospy.Publisher("/cmd_vel", Twist, queue_size=1) # ROS 2단계(필수): 퍼블리셔 설정
19
20         # 메시지 발행 주기를 설정합니다. 이 예제에서는 10Hz로 설정합니다.
21         self.rate = rospy.Rate(10) # ROS 2-1단계(옵션): 발행 주기 설정
22
23         # Twist 메시지 타입의 메시지 객체를 생성하고 초기화합니다.
24         self.msg = Twist() # 메시지 타입 설정 및 초기화
25
26     def func(self): # 3단계: 클래스 내의 함수 설정
27         # 메시지 항목(linear.x)을 1로 설정합니다.
28         self.msg.linear.x = 0.1 # 메시지 항목(linear.x) - 데이터 값 변경
29
30         # 퍼블리셔를 사용하여 Twist 메시지를 발행합니다.
31         self.pub.publish(self.msg) # ROS 3단계(필수): 퍼블리셔 - 퍼블리셔 실행
32
33         # 현재 메시지와 linear.x 값을 출력합니다.
34         print(f"self.msg: {self.msg}") # 출력: 메시지
35         print(f"self.msg.linear.x: {self.msg.linear.x}") # 출력: 메시지(linear.x)
36
37         # 지정한 발행 주기에 따라 슬립합니다. 이것은 메시지를 일정한 주기로 발행하기 위해 사용됩니다.
38         self.rate.sleep() # ROS 3-1단계(옵션): 퍼블리셔 - 주기 실행
39
40
41 # 메인 함수 정의: ROS 노드를 실행하기 위한 메인 함수입니다.
42 def main(): # 4단계: 메인 함수 정의
43     class_name = Class_Name() # Class_Name 클래스의 인스턴스를 생성합니다.
44
45     # 노드가 종료되지 않는 한 퍼블리셔를 실행하고 메시지를 발행합니다.
46     while not rospy.is_shutdown():
47         class_name.func()
48
49
50 # 직접 실행 코드: 스크립트가 직접 실행될 때 main() 함수를 호출합니다.
51 if __name__ == "__main__": # 5단계: 직접 실행 구문 정의
52     main()
```

02 LIMO 기본 package 설치

- LIMO : lidar model(YDLIDAR X2/X2L)
 - https://github.com/YDLIDAR/ydlidar_ros_driver : ydlidar driver package under ROS
 - Github를 참고하여 ydlidar driver package 설치



YDLIDAR X2



Range Frequency
3000Hz



Scan Frequency
5-8Hz



Range Distance
0.12-8m



Scan Angle
360°



Angle Resolution
0.6°-0.96°



Size
Φ60.5*50.3*96mm

02 LIMO 기본 package 설치

• ydlidar driver package 설치

- . Compile & Install YDLidar SDK

\$ cd

\$ git clone https://github.com/YDLIDAR/YDLidar-SDK.git

\$ cd ~/YDLidar-SDK

\$ mkdir build

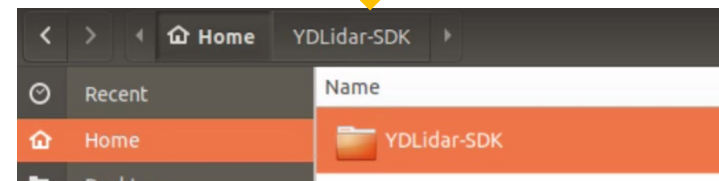
\$ cd ~/YDLidar-SDK/build

\$ cmake ..

\$ make

\$ sudo make install

```
agilex@nano: ~
agilex@nano: ~ 66x36
agilex@nano:~$ git clone https://github.com/YDLIDAR/YDLidar-SDK.git
Cloning into 'YDLidar-SDK'...
remote: Enumerating objects: 803, done.
remote: Counting objects: 100% (51/51), done.
remote: Compressing objects: 100% (27/27), done.
remote: Total 803 (delta 29), reused 26 (delta 24), pack-reused 752
Receiving objects: 100% (803/803), 11.28 MiB | 6.25 MiB/s, done.
Resolving deltas: 100% (484/484), done.
agilex@nano:~$
```



```
agilex@nano:~$ cd ~/YDLidar-SDK
agilex@nano:~/YDLidar-SDK$ mkdir build
agilex@nano:~/YDLidar-SDK$ cd ~/YDLidar-SDK/build
bash: cd: /home/agilex/YDLidar-SDK/bulid: No such file or directory
agilex@nano:~/YDLidar-SDK$ cd ~/YDLidar-SDK/build
agilex@nano:~/YDLidar-SDK/build$ cmake ..
-- The C compiler identification is GNU 7.5.0
-- The CXX compiler identification is GNU 7.5.0
-- Check for working C compiler: /usr/bin/cc
-- Check for working C compiler: /usr/bin/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
-- Detecting C compile features - done
-- Check for working CXX compiler: /usr/bin/c++
-- Check for working CXX compiler: /usr/bin/c++ -- works
```

```
agilex@nano: ~/YDLidar-SDK/build 66x36
agilex@nano:~/YDLidar-SDK/build$ make
Scanning dependencies of target ydlidar_sdk
[ 2%] Building CXX object CMakeFiles/ydlidar_sdk.dir/core/base/timer.cpp.o
[ 5%] Building CXX object CMakeFiles/ydlidar_sdk.dir/core/common/ydlidar_def.cpp.o
[ 8%] Building CXX object CMakeFiles/ydlidar_sdk.dir/core/network/ActiveSocket.cpp.o
[ 11%] Building CXX object CMakeFiles/ydlidar_sdk.dir/core/network/PassiveSocket.cpp.o
[ 14%] Building CXX object CMakeFiles/ydlidar_sdk.dir/core/network/SimpleSocket.cpp.o
[ 17%] Building CXX object CMakeFiles/ydlidar_sdk.dir/core/serial/serial.cpp.o
In file included from /home/agilex/YDLidar-SDK/core/serial/common.h:45:0,
                 from /home/agilex/YDLidar-SDK/core/serial/serial.cpp:1:
```

```
agilex@nano: ~/YDLidar-SDK/build 66x36
agilex@nano:~/YDLidar-SDK/build$ sudo make install
[sudo] password for agilex:
[ 50%] Built target ydlidar_sdk
[ 55%] Built target lidar_c_api_test
[ 61%] Built target gs_test
[ 67%] Built target tmini_test
[ 73%] Built target tof_test
[ 79%] Built target et_test
[ 85%] Built target tri_and_gs_test
[ 91%] Built target tri_test
[100%] Built target _ydlidar
Install the project...
-- Install configuration: ""
-- Installing: /usr/local/include/core/base/datas...
```

02 LIMO 기본 package 설치

- ydlidar driver package 설치

- . ydlidar_ros_driver package 설치

```
$ cd ~/catkin_ws/src
```

```
$ git clone https://github.com/YDLIDAR/ydlidar_ros_driver.git
```

```
$ cd ~/catkin_ws/
```

```
$ catkin_make
```

* catkin_make 후 build 에러 시 아래 설치 후 다시 catkin_make

```
$ sudo apt-get install -y python-tf2-sensor-msgs
```

```
$ source ./devel/setup.sh
```

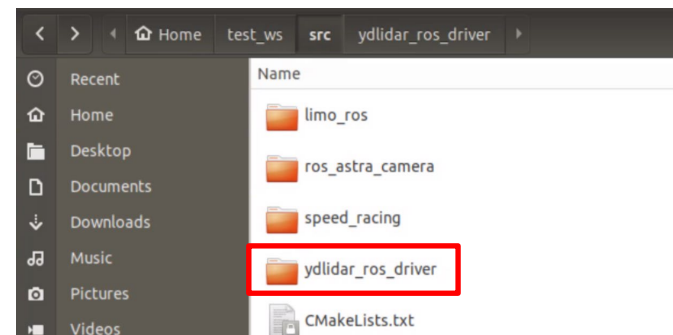
- . ydlidar_ros_driver 실행

```
$ roslaunch ydlidar_ros_driver X2.launch
```

- . Lidar 구동 확인

1. rviz

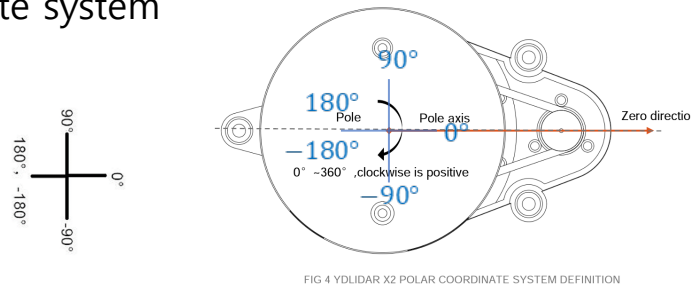
2. 예제 프로그램



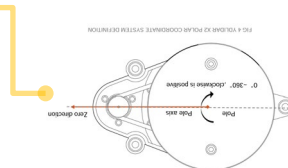
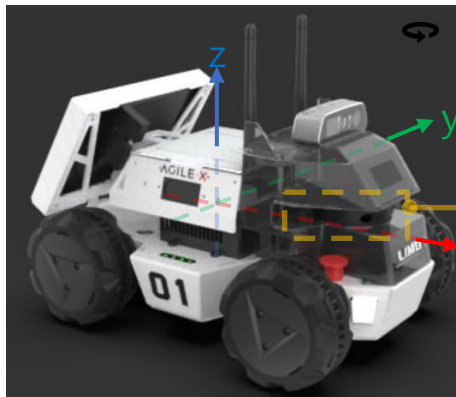
02 LIMO 기본 package 설치

- ydlidar driver package 설치

- 구동 전 ydlidar 센서 설정
- https://github.com/YDLIDAR/ydlidar_ros_driver 참고
- Lidar coordinate system



- LIMO 차량의 좌표계에 대하여 라이다가 반대로 설치됨



라이다 설치 방향

Publish Topic

Topic	Type	Description
scan	sensor_msgs/LaserScan	2D laser scan of the 0-angle ring
point_cloud	sensor_msgs/PointCloud	2D point cloud of the 0-angle ring
laser_fan (removed in version 1.0.1)	ydlidar_ros_driver::LaserFan	2D Raw laser fan of the 0-angle ring

Subscribe Service

Service	Type	Description
stop_scan	std_srvs::Empty	turn off lidar
start_scan	std_srvs::Empty	turn on lidar

Configure ydlidar_ros_driver internal parameter

The ydlidar_ros_driver internal parameters are in the launch file, they are listed as below :

Parameter name	Data Type	detail
port	string	Set Lidar the serial port or IP address it can be set to /dev/ttyUSB0, 192.168.1.11, etc. default: /dev/ydlidar
frame_id	string	Lidar TF coordinate system name. default: laser_frame
ignore_array	string	LiDAR filtering angle area eg: -90, -80, 30, 40
baudrate	int	Lidar baudrate or network port. default: 230400
		Set lidar type

02 LIMO 기본 package 설치

• ydlidar driver package 설치

-. Lidar 구동 launch 파일 수정

```
$ cd ~/catkin_ws/src/ydlidar_ros_driver/launch
```

```
$ gedit X2.launch
```

-. 다음과 같이 수정

: 라이다 좌표 reversion 'true'

: 라이다 측정 범위 변환 -90°~90°

```
1 <launch>
2 <node name="ydlidar_lidar_publisher" pkg="ydlidar_ros_driver" type="ydlidar_
  respawn="false" >
3   <!-- string property -->
4   <param name="port" type="string" value="/dev/ydlidar"/>
5   <param name="frame_id" type="string" value="laser_frame"/>
6   <param name="ignore_array" type="string" value=""/>
7
8   <!-- int property -->
9   <param name="baudrate" type="int" value="115200"/>
10  <!-- 0:TYPE_TOF, 1:TYPE_TRIANGLE, 2:TYPE_TOF_NET -->
11  <param name="lidar_type" type="int" value="1"/>
12  <!-- 0:YDLIDAR_TYPE_SERIAL, 1:YDLIDAR_TYPE_TCP -->
13  <param name="device_type" type="int" value="0"/>
14  <param name="sample_rate" type="int" value="3"/>
15  <param name="abnormal_check_count" type="int" value="4"/>
16
17  <!-- bool property -->
18  <param name="resolution_fixed" type="bool" value="true"/>
19  <param name="auto_reconnect" type="bool" value="true"/>
20  <param name="reversion" type="bool" value="true"/>
21  <param name="inverted" type="bool" value="true"/>
22  <param name="isSingleChannel" type="bool" value="true"/>
23  <param name="intensity" type="bool" value="false"/>
24  <param name="support_motor_dtr" type="bool" value="true"/>
25  <param name="invalid_range_is_inf" type="bool" value="false"/>
26  <param name="point_cloud_preservative" type="bool" value="false"/>
27
28  <!-- float property -->
29  <param name="angle_min" type="double" value="-90" />
30  <param name="angle_max" type="double" value="90" />
31  <param name="range_min" type="double" value="0.1" />
32  <param name="range_max" type="double" value="12.0" />
33  <!-- frequency is invalid, External PWM control speed -->
34  <param name="frequency" type="double" value="10.0"/>
```

02 LIMO 기본 package 설치

- ydlidar driver package 설치

- Lidar 구동 확인 : rviz를 통한 확인

- 터미널 1 : `$ roslaunch ydlidar_ros_driver X2.launch`

- 터미널 2 : `$ rviz`

- laser_frame x축(빨간색) 방향이 차량 정면과 동일한지 확인

