포팅메뉴얼

▼ Dual Boot for Ubuntu

1. Get Ubuntu 20.04 image
2. Make bootable USB using Ventoy tool
   - plug USB
   - install Ventoy Tool
   - run Ventoy2Dist.exe
   - select plugged USB
   - click Install button
   - put ISO files in USB
3. Disk partition division
   - click disk management on window search bar
   - select partition to be divided & click vulumn reduction
   - choose disk size
4. BIOS mode
   - Push F2 while re-booting
   - change Boot priority to USB
   - select ubuntu iso
   - reboot

▼ Install MORAI simulator in Ubuntu

```
$ cd ~/MoraiLauncher_Lin
$ chmod +x MORAISim.sh
$ chmod +x MoraiLauncher_Lin.x86_64
$ ./MORAISim # 실행
```

▼ ROS noetic setting

1. noetic setting

```
$ sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release -sc) main" > /etc/apt/sources.list.d/ros-latest.list'
$ sudo apt install curl
$ curl -s https://raw.githubusercontent.com/ros/rosdistro/master/ros.asc | sudo apt-key add -
$ sudo apt update
$ sudo apt install ros-noetic-desktop-full
$ source /opt/ros/noetic/setup.bash
$ sudo apt install python3-rosdep python3-rosinstall python3-rosinstall-generator python3-wstool build-essential
$ sudo apt install python3-rosdep
$ sudo rosdep init
$ rosdep update
$ echo "source /opt/ros/melodic/setup.bash" >> ~/.bashrc
$ source ./.bashrc
$ roscore # check if installed correctly
```

2. Create workspace & package and then build

```
$ cd ~/
$ mkdir -p catkin_ws/src
$ cd catkin_ws
$ catkin_make
$ cd ~/catkin_ws/src
$ catkin_create_pkg ssafy_ad rospy std_msgs
$ cd ~/catkin_ws
$ catkin_make
$ source ~/catkin_ws/devel/setup.bash
$ rospack profile
```

3. MORAI message download

```
$ sudo apt-get install git
$ cd ~/catkin_ws/src
$ git clone https://github.com/MORAI-Autonomous/MORAI-ROS_morai_msgs.git
$ cd ~/catkin_ws
$ catkin_make
```

4. Install dependent package

```
$ sudo apt-get install python-pip
$ sudo apt-get install net-tools
$ sudo apt-get install ros-melodic-rosbridge-server
$ sudo apt-get install ros-melodic-velodyne
$ sudo apt install terminator
$ mkdir -p catkin_ws/src
$ pip install pyproj
$ pip install scikit-learn
$ cd ~/catkin_ws
$ catkin_make
$ sudo apt install libvulkan1
```

▼ Docker setting & ROS melodic setting

1. Install Docker

```
$ sudo apt update
$ sudo apt install docker.io
```

2. Get ROS Melodic image in Docker-hub

```
$ sudo systemctl enable docker
$ sudo systemctl start docker
$ sudo docker pull ros:melodic
```

3. Enter ROS Melodic container

```
$ sudo docker run -it --name my_melodic_container ros:melodic /bin/bash
```

4. Set ROS bridge

```
# in DOCKER CONTAINOR
```

```
# Create workspace & package and then build
$ cd ~/
$ mkdir –p catkin_ws/src
$ cd catkin_ws
$ catkin_make
$ cd ~/catkin_ws/src
$ catkin_create_pkg ssafy_ad rospy std_msgs
$ cd ~/catkin_ws
$ catkin_make
$ source ~/catkin_ws/devel/setup.bash
$ rospack profile

# MORAI message download
$ apt-get install git
$ cd ~/catkin_ws/src
$ git clone https://github.com/MORAI-Autonomous/MORAI-ROS_morai_msgs.git
$ cd ~/catkin_ws
$ catkin_make

# Install ros bridge
$ apt-get install ros-melodic-rosbridge-server
$ apt-get install ros-melodic-velodyne
$ mkdir -p catkin_ws/src
$ pip install pyproj
$ cd ~/catkin_ws
$ catkin_make
```

5. Export docker image

```
# Exit Containor
$ exit

# make docker image
$ sudo docker commit my_melodic_container my_melodic_image
```

6. Make & run new container

```
$ sudo docker run -d --name melodic_bridge -p 11311:11311 my_melodic_image
```

▼ Project download

1. Clone our project in ROS build directory

```
$ cd ~/catkin_ws/src/ssafy_ad
$ git clone https://lab.ssafy.com/s09-mobility-autodriving-sub2/S09P22A701.git
$ cd ~/catkin_ws/src/ssafy_ad/S09P22A701/auto_drive/perception/Perception/YOLO/YOLOv5
$ pip install -r requirements.txt # yolo package
```

2. Make our custom message

```
$ cp ./S09P22A701/auto_drive/custom_msg_make/*.* ./
```

3. Re-build ROS project

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

▼ Run process

You have to open 3 or more new terminals.

ROS Melodic bridge

1. Enter the Docker container

```
$ cd ~
$ sudo docker ps -a # config your container name
$ sudo docker restart (your_container_name)
$ sudo docker attach (your_container_name)
```

2. Run rosbridge

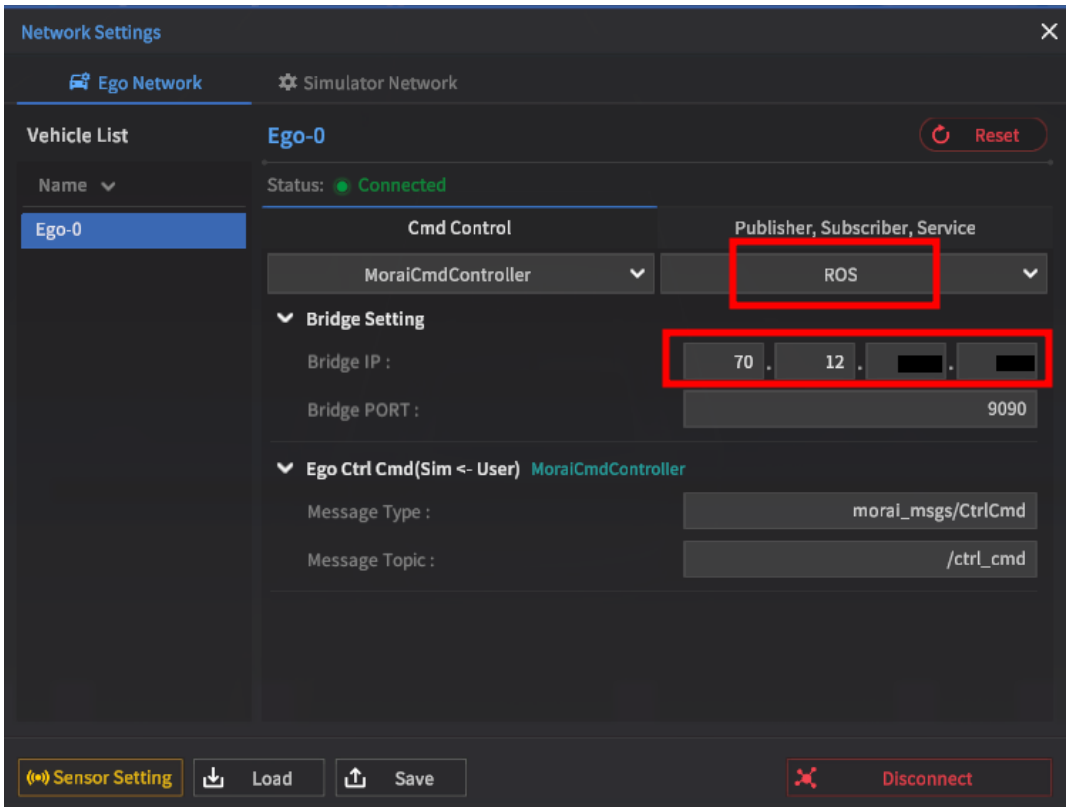
```
source ~/catkin_ws/devel/setup.bash
roslaunch rosbridge_server rosbridge_websocket.launch
```

Simulator

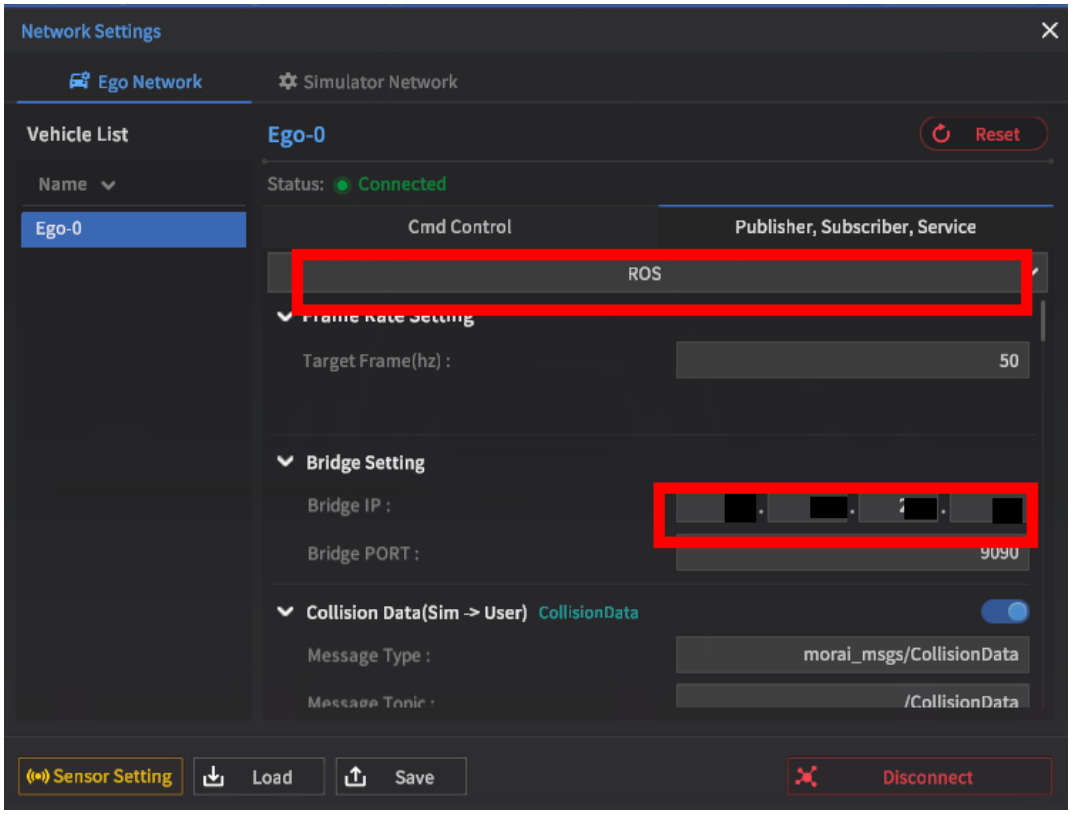
1. Start MORAI Simulator

```
$ cd ~/MoraiLauncher_Lin
$ ./MORAISim
```

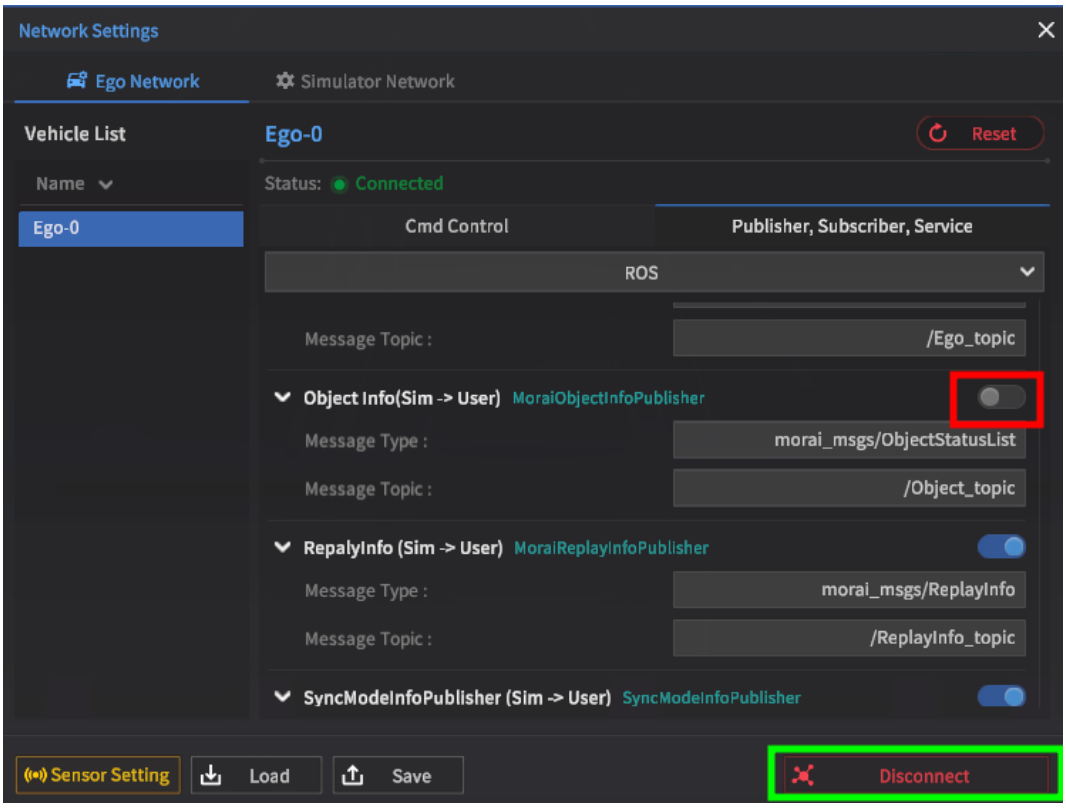
2. Connect with ROS (Press “F4” button)
   - Cmd Control connect setting
     - Connet Type : ROS / Bridge IP : your Bridge IP



- Pub, Sub connect setting
  - Connet Type : ROS / Bridge IP : your Bridge IP



  - Object info : No publish / Click “connect” button



3. Set sensors (Press “F3” button)
   - Add sensor setting
     - Set sensor’s coordinate

     - Connect with bridge

- You have to connect 6 sensors. Coordinate values are here.
  - GPS : (0.00, 0.00, 0.00)
  - IMU : (0.00, 0.00, 0.00)
  - Camera : (3.67, -0.01, 0.51)
  - 3D Lidar : (1.00, 0.00, 1.30)
  - Radar1 : (3.68, -0.14, 0.51)
  - Radar2 : (-0.6, 0.88, 0.41)

ROS Noetic

1. Setup

```
$ source ~/catkin_ws/devel/setup.bash
```

2. Run `.launch` file

```
$ roslaunch ssafy_ad sangam_final.launch
```