

4.24 스터디 송주훈

목표: ROS2 터틀봇 패키지 설치 및 분석

설치 과정

다음의 명령어를 터미널에 입력하여 설치를 진행한다

3. 1. 3. Install Dependent ROS 2 Packages

1. Open the terminal with `Ctrl`+`Alt`+`T` on the **Remote PC**.
2. Install Gazebo

[Remote PC]

```
$ sudo apt install ros-humble-gazebo-
```

3. Install Cartographer

[Remote PC]

```
$ sudo apt install ros-humble-cartographer  
$ sudo apt install ros-humble-cartographer-ros
```

4. Install Navigation2

[Remote PC]

```
$ sudo apt install ros-humble-navigation2  
$ sudo apt install ros-humble-nav2-bringup
```

시뮬레이션, slam, navigation을 위해 위의 세 가지를 설치한다.

3. 1. 4. Install TurtleBot3 Packages

Install the required TurtleBot3 Packages.

[Remote PC]

```
$ source /opt/ros/humble/setup.bash  
$ mkdir -p ~/turtlebot3_ws/src  
$ cd ~/turtlebot3_ws/src/  
$ git clone -b humble https://github.com/ROBOTIS-GIT/DynamixelSDK.git  
$ git clone -b humble https://github.com/ROBOTIS-GIT/turtlebot3_msgs.git  
$ git clone -b humble https://github.com/ROBOTIS-GIT/turtlebot3.git  
$ sudo apt install python3-colcon-common-extensions  
$ cd ~/turtlebot3_ws  
$ colcon build --symlink-install  
$ echo 'source ~/turtlebot3_ws/install/setup.bash' >> ~/.bashrc  
$ source ~/.bashrc
```

다음 명령어들을 입력해 터틀봇3과 관련된 패키지를 설치한다.

6. 1. 1. Install Simulation Package

The **TurtleBot3 Simulation Package** requires `turtlebot3` and `turtlebot3_msgs` packages. Without these prerequisite packages, the Simulation cannot be launched.

Please follow the [PC Setup](#) instructions if you did not install required packages and dependent packages.

```
$ cd ~/turtlebot3_ws/src/  
$ git clone -b humble https://github.com/ROBOTIS-GIT/turtlebot3_simulations.c  
$ cd ~/turtlebot3_ws && colcon build --symlink-install
```

터틀봇 시뮬레이션을 위한 패키지도 설치한다.

빌드 과정에서 다음과 같은 경고가 발생하나 실행에는 문제가 되지 않는 경고라고 함.

```
--- stderr: turtlebot3_gazebo  
CMake Warning (dev) at /usr/share/cmake-3.22/Modules/FindPackageHandleStandardArgs.  
cmake:438 (message):  
  The package name passed to 'find_package_handle_standard_args' (PkgConfig)  
  does not match the name of the calling package (gazebo). This can lead to  
  problems in calling code that expects 'find_package' result variables  
  (e.g., `_FOUND`) to follow a certain pattern.  
Call Stack (most recent call first):  
  /usr/share/cmake-3.22/Modules/FindPkgConfig.cmake:99 (find_package_handle_standar  
d_args)  
  /usr/lib/x86_64-linux-gnu/cmake/gazebo/gazebo-config.cmake:72 (include)  
  CMakeLists.txt:23 (find_package)  
This warning is for project developers. Use -Wno-dev to suppress it.  
---
```

```
$ export TURTLEBOT3_MODEL=waffle  
$ ros2 launch turtlebot3_gazebo turtlebot3_world.launch.py
```

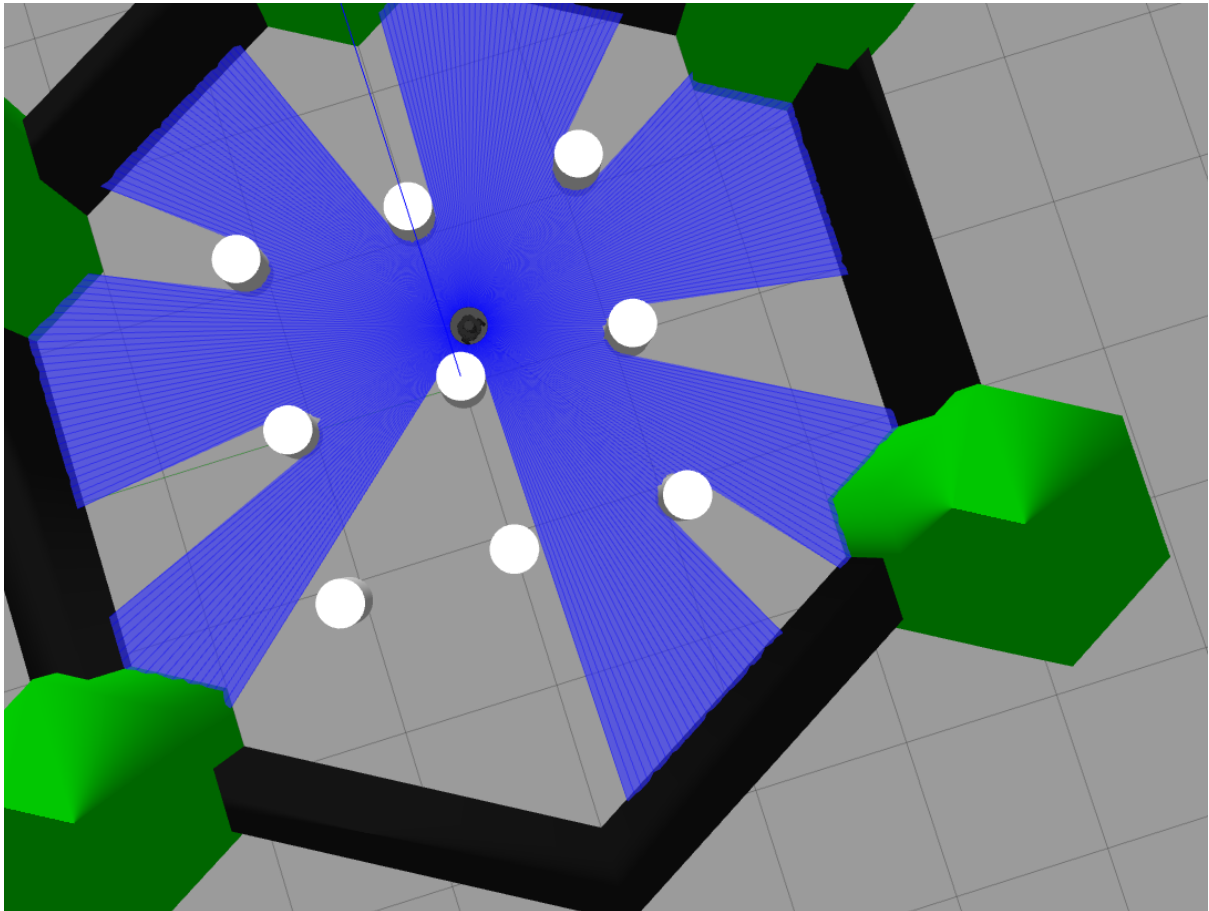
3. TurtleBot3 House

터틀봇 가제보 시뮬레이션을 진행하기 위해 다음과 같이 입력한다. TURTLEBOT3_MODEL을 waffle 혹은 burger로 입력하면 models 폴더의 해당 모델의 sdf 파일을 읽어 turtlebot3_world.launch.py에서의 spawn_turtlebot3.launch.py에서 소환을 하게 된다.

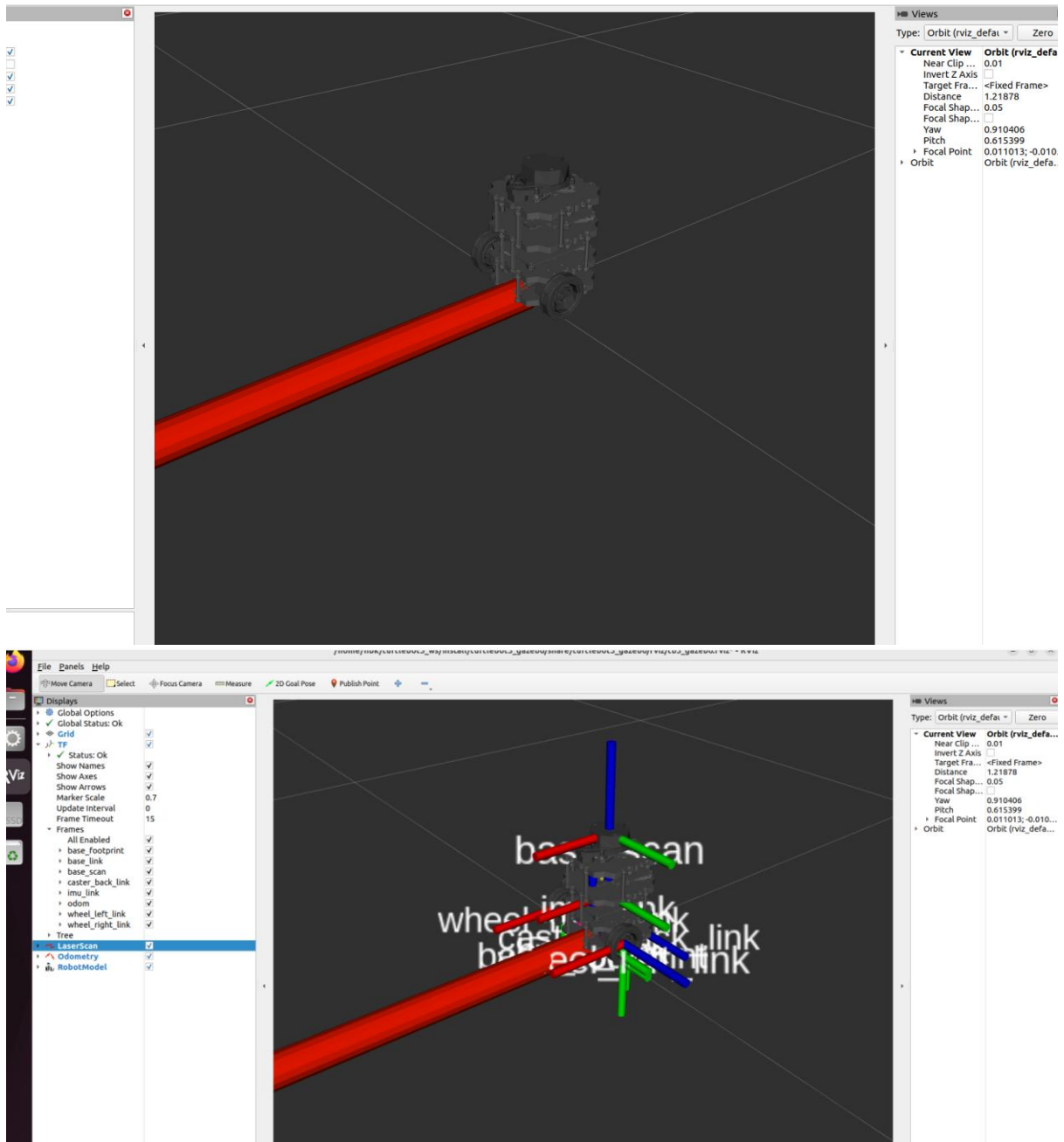
이번에는 burger 모델로 turtlebot3_world 에서 소환 후 teleop을 진행하였다.

```
hbk@hbk-IdeaPad-Slim-3-16ABR8:~/turtlebot3_ws$ export TURTLEBOT3_MODEL=burger
hbk@hbk-IdeaPad-Slim-3-16ABR8:~/turtlebot3_ws$ ros2 run turtlebot3_teleop teleop_keyboard
```

```
Control Your TurtleBot3!
-----
Moving around:
    w    a    s    d
    x
```



실행 결과로 다음과 같이 터틀봇 소환 후 움직이는 것을 확인할 수 있었다.



fake node 패키지를 통해 rviz2로 tf를 확인할 수 있었다.