

Week_1

환경

- 라즈베리파이4 8GB (터틀봇)
- 우분투 22.04 LTS
- ROS2 humble

3. 1. PC Setup

WARNING: The contents in this chapter corresponds to the `Remote PC` (your desktop or laptop PC) which will control TurtleBot3. Do not apply this instruction to your TurtleBot3.

Compatibility WARNING

- `Jetson Nano` does not support Ubuntu 20.04 and later. Please refer to [NVIDIA developer forum](#) for more details.

NOTE: This instruction was tested on Linux with `Ubuntu 22.04` and `ROS2 Humble Hawksbill`.

3. 1. 1. Download and Install Ubuntu on PC

1. Download the proper `Ubuntu 22.04 LTS Desktop` image for your PC from the links below.
 - [Ubuntu 22.04 LTS Desktop image \(64-bit\)](#)
2. Follow the instruction below to install Ubuntu on PC.
 - [Install Ubuntu desktop](#)

3. 1. 2. Install ROS 2 on Remote PC

Please follow [the official ROS2 documentation](#) to install the ROS2 Humble.

For most Linux users, [Debian package installation](#) method is strongly recommended.

3. 1. 3. Install Dependent ROS 2 Packages

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

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

gazebo 설치에서 문제 발생

```
turtle@turtle:~$ sudo apt install ros-humble-gazebo-*
[sudo] turtle 암호:
패키지 목록을 읽는 중입니다... 완료
의존성 트리를 만드는 중입니다... 완료
상태 정보를 읽는 중입니다... 완료
glob 'ros-humble-gazebo-msgs-dbgsym'에 대해 'ros-humble-gazebo-*'를 선택합니다.
glob 'ros-humble-gazebo-dev'에 대해 'ros-humble-gazebo-*'를 선택합니다.
glob 'ros-humble-gazebo-msgs'에 대해 'ros-humble-gazebo-*'를 선택합니다.
몇몇 패키지를 설치할 수 없습니다. 요청한 상황이 불가능할 수도 있고,
불안정 배포판을 사용해서 일부 필요한 패키지를 아직 만들지 않았거나,
아직 Incoming에서 나오지 않은 경우일 수도 있습니다.
이 상황을 해결하는데 다음 정보가 도움이 될 수도 있습니다:

다음 패키지의 의존성이 맞지 않습니다:
ros-humble-gazebo-dev : 의존: gazebo 하지만 설치할 수 없습니다
                        의존: libgazebo-dev 하지만 설치할 수 없습니다
E: 문제를 바로잡을 수 없습니다. 망가진 고정 패키지가 있습니다.
```

- gazebo의 버전문제로 판단하고, gazebo Fortress(LTS)버전으로 설치

Summary of Compatible ROS and Gazebo Combinations

This table includes all currently supported versions of ROS and Gazebo. All other ROS and Gazebo releases are end of life and we do not recommend their continued use.

	GZ Citadel (LTS)	GZ Fortress (LTS)	GZ Garden
ROS 2 Rolling	X	✓	P
ROS 2 Humble (LTS)	X	✓	P
ROS 2 Foxy (LTS)	✓	X	X
ROS 1 Noetic (LTS)	✓	P	X

- ✓ - Recommended combination
- X - Incompatible / not possible.
- P - Possible, *but not recommended*. These combinations of ROS and Gazebo can be made to work together, but some effort is required.

Docs / Gazebo Garden

Supported Sep, 2022 to Sep, 2024

Get Started

> Install

> Tutorials

[Migration From Ignition](#)[Feature Comparison](#)[Ros/gazebo Installation](#)[Roadmap](#)[Release Features](#)[Releases](#)

> Development

[Sim Architecture](#)

> Library Reference

Getting Started with Gazebo?

Welcome to Gazebo!

When you're ready, follow the next few steps to get up and running with simulation using Gazebo.

Step 1: Install

Note: If you are a [ROS](#) user, please first read our tutorial about the [ROS/Gazebo installation](#).

The recommended installation for new users is the use of binary packages available for the platform to use:

Platform	Gazebo Versions
Ubuntu 22.04 Jammy	Gazebo Garden (recommended) and Gazebo Fortress
Ubuntu 20.04 Focal	Gazebo Garden (recommended), Gazebo Fortress and Gazebo Citadel
Ubuntu 18.04 Bionic	Gazebo Citadel
Mac Monterey	Gazebo Garden (recommended), Gazebo Fortress and Gazebo Citadel
Mac BigSur	Gazebo Garden (recommended), Gazebo Fortress and Gazebo Citadel
Mac Catalina	Gazebo Garden (recommended), Gazebo Fortress and Gazebo Citadel
Windows	Support via Conda-Forge is not fully functional, as there are known runtime issues see this issue for details .

- 이후 ign gazebo 실행에서 OpenGL 3.3을 지원하지 않는 문제 발생
- 오류가 발생하면서 ign gazebo창이 꺼진다.
 - Open Graphics Library 오픈 그래픽스 라이브러리의 줄임말이고, 3차원의 그래픽 표준 API 규격이다.

```
turtle@turtle:~$ ign gazebo
Warning: Ignoring XDG_SESSION_TYPE=wayland on Gnome. Use QT_QPA_PLATFORM=wayland to run on Wayland an
yway.
[GUI] [Err] [Ogre2RenderEngine.cc:986] Unable to create the rendering window: OGRE EXCEPTION(3:Rende
ringAPIException): OpenGL 3.3 is not supported. Please update your graphics card drivers. in GL3PlusR
enderSystem::initialiseContext at /build/ogre-next-phBY09/ogre-next-2.2.5+dfsg3/RenderSystems/GL3Plus
/src/OgreGL3PlusRenderSystem.cpp (line 3363)
[GUI] [Err] [Ogre2RenderEngine.cc:986] Unable to create the rendering window: OGRE EXCEPTION(3:Rende
ringAPIException): OpenGL 3.3 is not supported. Please update your graphics card drivers. in GL3PlusR
enderSystem::initialiseContext at /build/ogre-next-phBY09/ogre-next-2.2.5+dfsg3/RenderSystems/GL3Plus
/src/OgreGL3PlusRenderSystem.cpp (line 3363)
[GUI] [Err] [Ogre2RenderEngine.cc:986] Unable to create the rendering window: OGRE EXCEPTION(3:Rende
ringAPIException): OpenGL 3.3 is not supported. Please update your graphics card drivers. in GL3PlusR
enderSystem::initialiseContext at /build/ogre-next-phBY09/ogre-next-2.2.5+dfsg3/RenderSystems/GL3Plus
/src/OgreGL3PlusRenderSystem.cpp (line 3363)
[GUI] [Err] [Ogre2RenderEngine.cc:986] Unable to create the rendering window: OGRE EXCEPTION(3:Rende
ringAPIException): OpenGL 3.3 is not supported. Please update your graphics card drivers. in GL3PlusR
enderSystem::initialiseContext at /build/ogre-next-phBY09/ogre-next-2.2.5+dfsg3/RenderSystems/GL3Plus
/src/OgreGL3PlusRenderSystem.cpp (line 3363)
[GUI] [Err] [Ogre2RenderEngine.cc:986] Unable to create the rendering window: OGRE EXCEPTION(3:Rende
ringAPIException): OpenGL 3.3 is not supported. Please update your graphics card drivers. in GL3PlusR
enderSystem::initialiseContext at /build/ogre-next-phBY09/ogre-next-2.2.5+dfsg3/RenderSystems/GL3Plus
/src/OgreGL3PlusRenderSystem.cpp (line 3363)
[GUI] [Err] [Ogre2RenderEngine.cc:986] Unable to create the rendering window: OGRE EXCEPTION(3:Rende
ringAPIException): OpenGL 3.3 is not supported. Please update your graphics card drivers. in GL3PlusR
enderSystem::initialiseContext at /build/ogre-next-phBY09/ogre-next-2.2.5+dfsg3/RenderSystems/GL3Plus
/src/OgreGL3PlusRenderSystem.cpp (line 3363)
[GUI] [Err] [Ogre2RenderEngine.cc:986] Unable to create the rendering window: OGRE EXCEPTION(3:Rende
ringAPIException): OpenGL 3.3 is not supported. Please update your graphics card drivers. in GL3PlusR
enderSystem::initialiseContext at /build/ogre-next-phBY09/ogre-next-2.2.5+dfsg3/RenderSystems/GL3Plus
/src/OgreGL3PlusRenderSystem.cpp (line 3363)
[GUI] [Err] [Ogre2RenderEngine.cc:986] Unable to create the rendering window: OGRE EXCEPTION(3:Rende
ringAPIException): OpenGL 3.3 is not supported. Please update your graphics card drivers. in GL3PlusR
enderSystem::initialiseContext at /build/ogre-next-phBY09/ogre-next-2.2.5+dfsg3/RenderSystems/GL3Plus
/src/OgreGL3PlusRenderSystem.cpp (line 3363)
[GUI] [Err] [Ogre2RenderEngine.cc:986] Unable to create the rendering window: OGRE EXCEPTION(3:Rende
ringAPIException): OpenGL 3.3 is not supported. Please update your graphics card drivers. in GL3PlusR
enderSystem::initialiseContext at /build/ogre-next-phBY09/ogre-next-2.2.5+dfsg3/RenderSystems/GL3Plus
/src/OgreGL3PlusRenderSystem.cpp (line 3363)
[GUI] [Err] [Ogre2RenderEngine.cc:994] Unable to create the rendering window after [11] attempts.
[GUI] [Err] [Ogre2RenderEngine.cc:915] Failed to create dummy render window.
Stack trace (most recent call last) in thread 26944:
#31 Object "/lib/aarch64-linux-gnu/libc.so.6", at 0xffff89425d1b, in
#30 Object "/lib/aarch64-linux-gnu/libc.so.6", at 0xffff893bd5c7, in
```

- 라즈베리 파이에서는 일반적으로 2.1 버전을 지원하고, 일부에서 3.0을 지원하지만 3.3 버전을 지원하지는 않는다
- gazebo Fortress troubleshooting을 보고 문제 해결 시도

Unable to create the rendering window

If you're getting errors like "Unable to create the rendering window", it could mean you're using an old OpenGL version. Gazebo Sim uses the Ogre 2 rendering engine by default, which requires an OpenGL version higher than 3.3, preferably 4.3+.

This can be confirmed by checking the Ogre 2 logs at `~/ignition/rendering/ogre2.log`, which should have an error like:

```
"OGRE EXCEPTION(3:RenderingAPIException): OpenGL 3.3 is not supported. Please update your graphics card drivers."
```

You can also check your OpenGL version running:

```
glxinfo | grep "OpenGL version"
```

To enable Ogre 2 support, you'll need to update your computer's OpenGL version. As suggested on the Ogre logs, this may require updating your graphics card drivers.

If you still run into OpenGL issues when running Gazebo with Ogre 2, it could be that certain extensions are not supported by your driver or you are running inside a virtual machine. In this case, you can try disabling DRI:

```
export LIBGL_DRI_DISABLE=1
```

or force software rendering

```
export LIBGL_ALWAYS_SOFTWARE=1
```

If you are using MESA drivers, you can also try overriding the OpenGL version

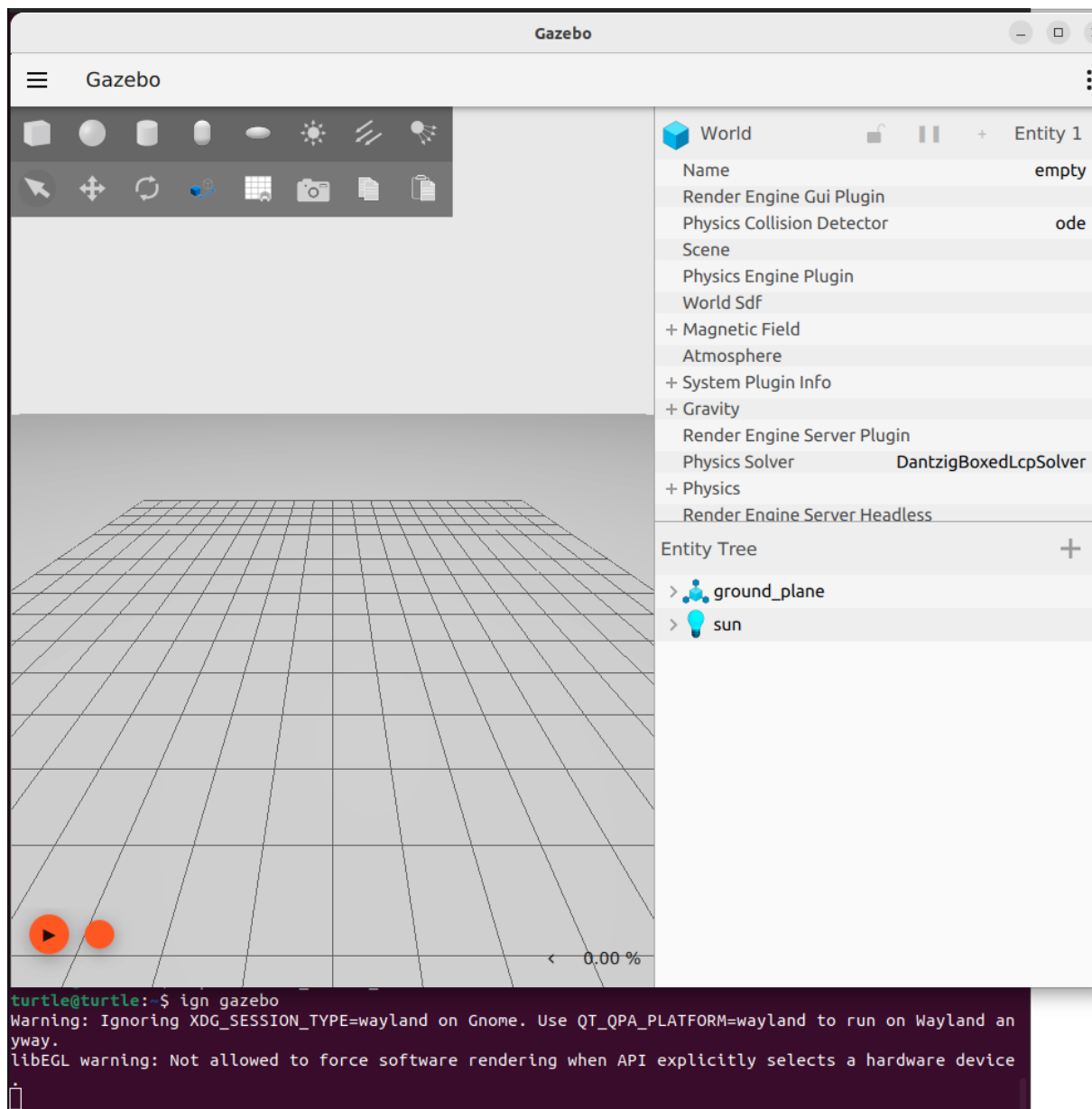
```
export MESA_GL_VERSION_OVERRIDE=3.3
```

```
turtle@turtle:~$ glxinfo | grep "OpenGL version"
OpenGL version string: 2.1 Mesa 22.2.5-0ubuntu0.1~22.04.3
```

- OpenGL의 버전을 조회한다.
- Ogre2 (Object-Oriented Graphics Rendering Engine, 3D 그래픽 렌더링 라이브러리)로 gazebo를 실행하는데 문제가 있을 경우 DRI (Direct Rendering Infrastructure, 사용자 공간 애플리케이션들이 비디오 하드웨어의 리소스를 직접 접근하게 해주는 API)를 비활성화한다.

```
turtle@turtle:~$ export LIBGL_DRI_DISABLE=1
turtle@turtle:~$
```

```
turtle@turtle:~$ export LIBGL_ALWAYS_SOFTWARE=1
turtle@turtle:~$
```



- If you still run into OpenGL issues when running Gazebo with Ogre 2, it could be that certain extensions are not supported by your driver or you are running inside a virtual machine. In this case, you can try disabling DRI:

3. Install Cartographer

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

4. Install Navigation2

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

- Cartographer, Navigation2 설치
- Turtlebot3 Packages 설치

3. 1. 4. Install TurtleBot3 Packages

Install TurtleBot3 via Debian Packages.

```
$ source ~/.bashrc
$ sudo apt install ros-humble-dynamixel-sdk
$ sudo apt install ros-humble-turtlebot3-msgs
$ sudo apt install ros-humble-turtlebot3
```

✧ [Click here to expand more details about building TurtleBot3 package from source.](#)

- Environment 설정

3. 1. 5. Environment Configuration

1. Set the ROS environment for PC.

```
$ echo 'export ROS_DOMAIN_ID=30 #TURTLEBOT3' >> ~/.bashrc
$ source ~/.bashrc
```

If you have installed TurtleBot3 using Debian packages with `apt install` command, you can ignore the warning below.

```
bash: /home/${YOUR_ACCOUNT}/turtlebot3_ws/install/setup.bash: No such file or directory
```

3. 2. 4. Configure the Raspberry Pi

HDMI cable must be connected before powering the Raspberry Pi, or else the HDMI port of the Raspberry Pi will be disabled.

1. Boot Up the Raspberry Pi
 - a. Connect the HDMI cable of the monitor to the HDMI port of Raspberry Pi.
 - b. Connect input devices to the USB port of Raspberry Pi.
 - c. Insert the microSD card.
 - d. Connect the power (either with USB or OpenCR) to turn on the Raspberry Pi.
 - e. Login with ID `ubuntu` and PASSWORD `ubuntu`. Once logged in, you'll be asked to change the password.
2. Open the network configuration file with the command below.

```
$ sudo nano /writable/etc/netplan/50-cloud-init.yaml
```

3. When the editor is opened, edit the content as below while replacing the `WIFI_SSID` and `WIFI_PASSWORD` with your actual wifi SSID and password.

```
network:
  version: 2
  renderer: networkd
  ethernets:
    eth0:
      dhcp4: yes
      dhcp6: yes
      optional: true
  wifis:
    wlan0:
      dhcp4: yes
      dhcp6: yes
      access-points:
        WIFI_SSID:
          password: WIFI_PASSWORD
```

- 문제2
- network 설정을 바꾸기 위해서 해당 디렉토리의 `50-cloud-init.yaml` 파일을 수정해야 하는데,

해당 디렉토리에는 `50-cloud-init.yaml` 파일이 없고, 대신에 `01-network-manager-all.yaml` 파일이 있다.

→ `/etc/netplan` 개념

- 이 디렉토리는 YAML 파일로 작성된 Netplan(네트워크 설정을 쉽게 구성할 수 있는 유틸리티이다) 파일을 저장한다.
- Netplan파일은 여러 개의 파일을 사용할 수 있고, 사전순으로 파일을 처리한다.
(`'01-xyz.yaml'` 부터 `'99-xyz.yaml'` 까지 파일을 처리한다.)

5. Enter the command below to edit automatic update setting file.

```
$ sudo nano /etc/apt/apt.conf.d/20auto-upgrades
```

6. Change the update settings as below.

```
APT::Periodic::Update-Package-Lists "0";  
APT::Periodic::Unattended-Upgrade "0";
```

7. Save the file with `(Ctrl)+(S)` and exit with `(Ctrl)+(X)`.

8. Set the `systemd` to prevent boot-up delay even if there is no network at startup. Run the command below to set mask the `systemd` process using the following command.

```
$ systemctl mask systemd-networkd-wait-online.service
```

- automatic update setting 설정 변경

- Suspend와 Hibernation 비활성화와 reboot

9. Disable Suspend and Hibernation

```
$ sudo systemctl mask sleep.target suspend.target hibernate.target hybrid-sleep.target
```

10. Reboot the Raspberry Pi.

```
$ reboot
```

11. After rebooting the Raspberry Pi, if you wish to work from the Remote PC using SSH, use below command from the remote PC terminal. Make sure to use the password you set in `Step 1`.

```
$ ssh ubuntu@{IP Address of Raspberry PI}
```

- SSH(Secure Shell)은
 - 네트워크 상에서 다른 컴퓨터에 로그인하거나 원격에서 명령을 실행하기 위한 프로토콜이고, 호스트로 암호화를 통해 정보를 보호하는 역할을 한다.

```
ssh ubuntu@192.168.1.100
```

12. Install ROS2 Humble Hawksbill

Follow the instruction in [the official ROS2 Humble installation guide](#).

13. Install and Build ROS Packages.

Building the `turtlebot3` package may take longer than an hour. Please use the SMPS to ensure the system is always powered.

```
$ sudo apt install python3-argcomplete python3-colcon-common-extensions libboost-system-dev
$ sudo apt install ros-humble-hls-lfcd-lds-driver
$ sudo apt install ros-humble-turtlebot3-msgs
$ sudo apt install ros-humble-dynamixel-sdk
$ sudo apt install libudev-dev
$ mkdir -p ~/turtlebot3_ws/src && cd ~/turtlebot3_ws/src
$ git clone -b humble-devel https://github.com/ROBOTIS-GIT/turtlebot3.git
$ git clone -b ros2-devel https://github.com/ROBOTIS-GIT/ld08_driver.git
$ cd ~/turtlebot3_ws/src/turtlebot3
$ rm -r turtlebot3_cartographer turtlebot3_navigation2
$ cd ~/turtlebot3_ws/
$ echo 'source /opt/ros/humble/setup.bash' >> ~/.bashrc
$ source ~/.bashrc
$ colcon build --symlink-install --parallel-workers 1
$ echo 'source ~/turtlebot3_ws/install/setup.bash' >> ~/.bashrc
$ source ~/.bashrc
```

```

turtle@turtle:~/turtlebot3_ws$ colcon build --symlink-install --parallel-workers 1
Starting >>> turtlebot3_description
Finished <<< turtlebot3_description [0.72s]
Starting >>> turtlebot3_node
Finished <<< turtlebot3_node [1.29s]
Starting >>> turtlebot3_example
--- stderr: turtlebot3_example
/usr/lib/python3/dist-packages/setuptools/command/easy_install.py:158: EasyInstallDeprecationWarning:
easy_install command is deprecated. Use build and pip and other standards-based tools.
  warnings.warn(
/usr/lib/python3/dist-packages/setuptools/command/install.py:34: SetuptoolsDeprecationWarning: setup.
py install is deprecated. Use build and pip and other standards-based tools.
  warnings.warn(
/usr/lib/python3/dist-packages/pkg_resources/__init__.py:116: PkgResourcesDeprecationWarning: 1.1buil
d1 is an invalid version and will not be supported in a future release
  warnings.warn(
/usr/lib/python3/dist-packages/pkg_resources/__init__.py:116: PkgResourcesDeprecationWarning: 0.1.43u
buntu1 is an invalid version and will not be supported in a future release
  warnings.warn(
---
Finished <<< turtlebot3_example [4.46s]
Starting >>> turtlebot3_teleop
--- stderr: turtlebot3_teleop
/usr/lib/python3/dist-packages/setuptools/command/easy_install.py:158: EasyInstallDeprecationWarning:
easy_install command is deprecated. Use build and pip and other standards-based tools.
  warnings.warn(
/usr/lib/python3/dist-packages/setuptools/command/install.py:34: SetuptoolsDeprecationWarning: setup.
py install is deprecated. Use build and pip and other standards-based tools.
  warnings.warn(
/usr/lib/python3/dist-packages/pkg_resources/__init__.py:116: PkgResourcesDeprecationWarning: 1.1buil
d1 is an invalid version and will not be supported in a future release
  warnings.warn(
/usr/lib/python3/dist-packages/pkg_resources/__init__.py:116: PkgResourcesDeprecationWarning: 0.1.43u
buntu1 is an invalid version and will not be supported in a future release
  warnings.warn(
---
Finished <<< turtlebot3_teleop [4.45s]
Starting >>> turtlebot3_bringup
Finished <<< turtlebot3_bringup [0.65s]
Starting >>> ld08_driver
Finished <<< ld08_driver [0.92s]
Starting >>> turtlebot3
Finished <<< turtlebot3 [0.63s]

Summary: 7 packages finished [13.9s]
2 packages had stderr output: turtlebot3_example turtlebot3_teleop

```