

Remote pc booting시 HDMI(DVI) 자동인식 안될 수 있음
→ 모니터 전원을 다시 한번 OFF-ON

ROS Programming

(file #2 / ?) ver1.0 Noetic

Yongseok Chi

Turtlebot3 burger : Standard Specification

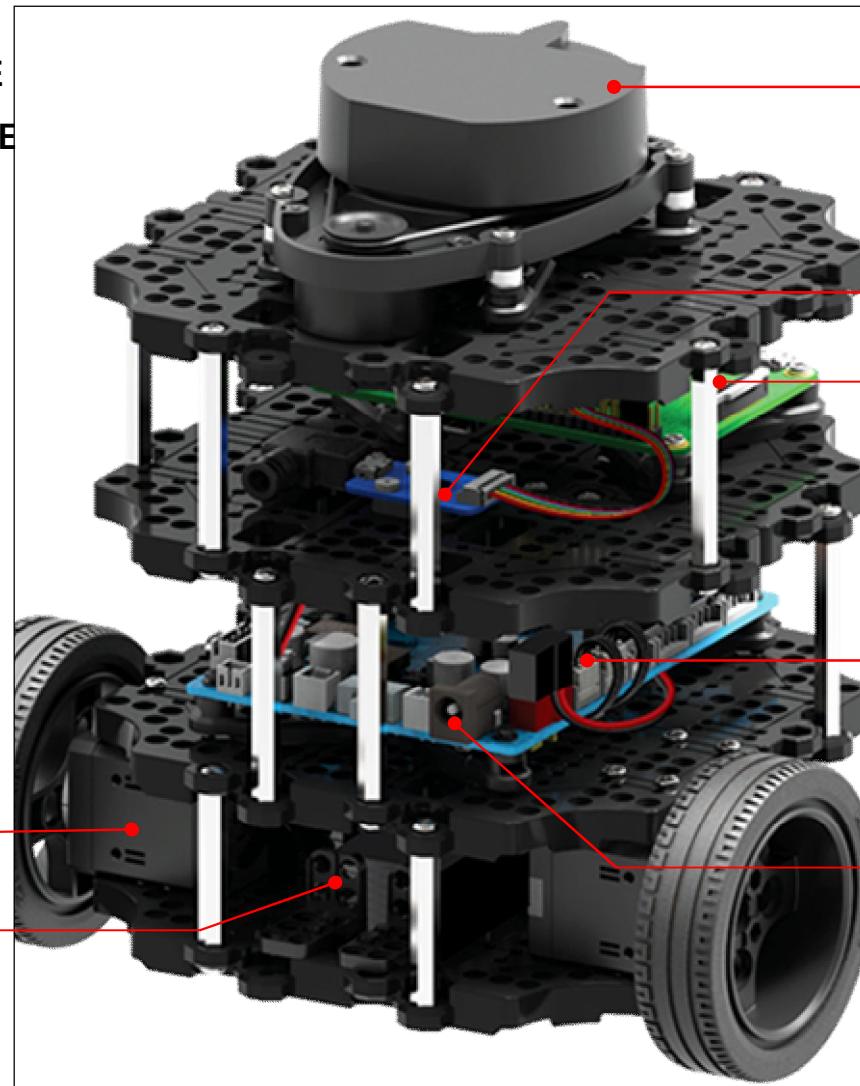
1. Specification

- (1) OPEN SOURCE SOFTWARE
- (2) OPEN SOURCE HARDWARE

DYNAMIXEL

- : XL430-W350-T
- : Stall Torque
 - 4.1 [N.m] (at 12.0 [V], 2.3 [A])
- : No Load Speed
 - 46 [rev/min] (at 12.0 [V])

Li-Po battery 11.1V 1800mAh



360° LiDAR

- : HLS-LFCD LDS(Laser Distance Sensor)
- : Detection distance 120mm~3500mm
- : Angular Resolution 1 degree

: USB2LDS (115200 baudrate)

SBC(Single Board Computer)

- : Raspberry PI 3 B+ / [PI 4](#)
- Broadcom BCM2837B0 / [2711](#)
- Cortex-A53 64bits Quad-core
- 1.4GHz frequency

OpenCR(Control module for ROS)

- : [STM32F746](#)
- ARM Cortex-M7 32bits RISC core
- 216MHz frequency

Input 12V 5A

Reference books

2. Reference books

(1) 주교재

: ROS 로봇 프로그래밍 : Ruby paper 3-1 ROS 설치

: Robotis e-manual 3장

<https://emanual.robotis.com/docs/en/platform/turtlebot3/quick-start/>

The screenshot shows the Robotis e-Manual website interface. At the top, there's a navigation bar with tabs: DYNAMIXEL, DYNAMIXEL SYSTEM, EDUCATIONAL KITS, SOFTWARE, and PARTS. Below the navigation bar, there's a search bar labeled "Enter Search Terms" with a magnifying glass icon. To the right of the search bar is a row of buttons for different ROS distributions: Kinetic, Melodic, Noetic, Dashing, Foxy, Humble, and Windows. The "Noetic" button is highlighted with a red box and has the word "click" written below it. Below these buttons, there's explanatory text: "O : Available", "Δ : Need to check", and "X : Unavailable". A table follows, showing the availability of various features across the different ROS distributions. The table has columns for Features (Teleop, SLAM, Navigation, Simulation, Manipulation, Home Service Challenge, Autonomous Driving, Machine Learning) and rows for ROS versions (Kinetic, Melodic, Noetic, Dashing, Foxy, Galactic, Humble). The availability is indicated by symbols: a circle for available, a triangle for need to check, and an X for unavailable.

Features	Kinetic	Melodic	Noetic	Dashing	Foxy	Galactic	Humble
Teleop	O	O	O	O	O	O	O
SLAM	O	O	O	O	O	O	O
Navigation	O	O	O	O	O	O	O
Simulation	O	O	O	O	O	O	O
Manipulation	O	O	O	O	O	Δ	O
Home Service Challenge	O	O	O	X	X	X	X
Autonomous Driving	O	X	O	X	X	X	X
Machine Learning	O	O	X	O	X	X	X

(2) 실험 재료 : Turtlebot3 burger

Reference books

TurtleBot3

1. Overview

2. Features

3. Quick Start Guide

 3. 1. PC Setup

 3. 2. SBC Setup

 3. 3. OpenCR Setup

 3. 4. Hardware Assembly

 3. 5. Bringup

 3. 6. Basic Operation

4. SLAM

5. Navigation

6. Simulation

7. Manipulation

8. Autonomous Driving

9. Machine Learning

10. Examples

11. Friends(Locomotion)

12. Learn

13. More Info

3. 1. PC Setup

WARNING: The contents in this chapter corresponds to the **Remote PC** (your desktop or laptop PC) which will control your TurtleBot3.

Compatibility WARNING

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

NOTE: This instruction was tested on Linux with **Ubuntu 20.04** and **ROS1 Noetic Ninjemys**.

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

1. Download the proper **Ubuntu 20.04 LTS Desktop** image for your PC
 - o [Ubuntu 20.04 LTS Desktop image \(64-bit\)](#)
2. Follow the instruction below to install Ubuntu on PC.
 - o [Install Ubuntu desktop](#)

3. 1. 2. Install ROS on Remote PC

Open the terminal with **Ctrl**+**Alt**+**T** and enter below commands one at a time.

In order to check the details of the easy installation script, please refer to [the script file](#).

```
$ sudo apt update  
$ sudo apt upgrade  
$ wget https://raw.githubusercontent.com/ROBOTIS-GIT/robotis_tools/master/install_ros_noetic.sh  
$ chmod 755 ./install_ros_noetic.sh  
$ bash ./install_ros_noetic.sh
```

If the above installation fails, please refer to [the official ROS1 Noetic installation guide](#).

Select an image

Ubuntu is distributed on three types of images described below.

Desktop image

The desktop image allows you to try Ubuntu without changing your computer at all, and at your option to install it permanently later. This type of image is what most people will want to use. You will need at least 1024MB of RAM to install from this image.

64-bit PC (AMD64) desktop image
Choose this if you have a computer based on the AMD64 or EM64T architecture (e.g., Athlon64, Opteron, EM64T Xeon, Core 2). Choose this if you are at all unsure.

4. ROS 개발환경 구축

강의실 PC에서



개인PC에서 강의자료 열기

MSI Cubi N100 ADL Win11Home (4GB, M.2 128GB)
i-5 2500



Monitor, keyboard, Mouse 연결



수업 마치기 전에, Monitor를 원래 강의실 PC에 연결하기

미수행 경우, 수업태도 감점

4. ROS 개발환경 구축

개인PC(노트북)에서



개인PC에서

- (1) Ubuntu 20.04 LTS Desktop image (64-bit)
- (2) Boot USB Tool download
- (3) USB에 Ubuntu 20.04 LTS 설치

→ 미니 PC에 Ubuntu와 ROS 설치



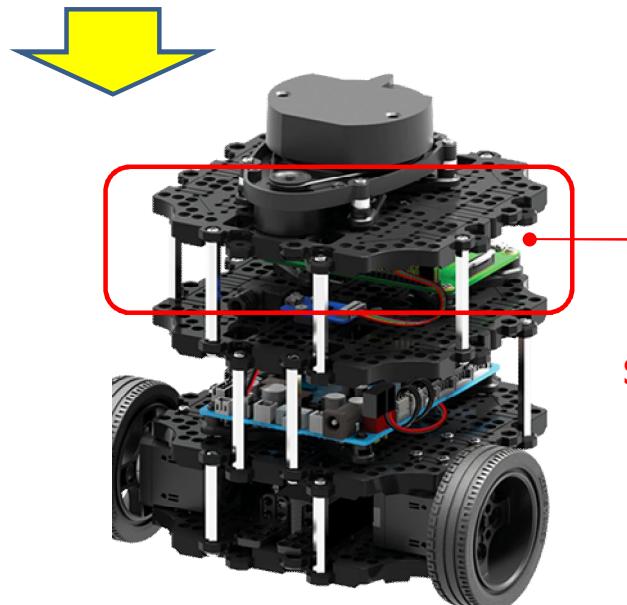
Remote PC

- : 제공되는 미니PC
- : Boot USB로 Ubuntu 20.04 LTS 설치
- : ROS 설치

개인PC에서

- (4) RASPBIAN download
- (5) Boot microSD Tool download
- (6) microSD에 RASPBIAN 설치

→ Raspberry Pi 4B 에 Raspbian과 ROS 설치



SBC(Single Board Computer)

- : Raspberry Pi 4B
- : microSD card로 RASPBIAN 설치
- : ROS 설치

4. ROS 개발환경 구축

<https://emanual.robotis.com/docs/en/platform/turtlebot3/quick-start/#pc-setup>

4-1. ROS 개발환경 구축

: Turtlebot 3 개발 환경 구축을 위해 **Remote PC**와 **SBC(Single Board Computer)**를 설정해야 함

- **Remote PC**(제공되는 미니PC)는 ROS를 실행하고 모니터링하기 위한 역할
- **SBC(Raspberry Pi 4B)**는 Turtlebot 3 burger(H/W-sensor, openCR)을 직접 구동하기 위한 역할

4-2. Remote PC setup

4-2-1. Download and Install Ubuntu on PC

(1) Ubuntu 20.04 LTS Desktop image (64-bit)

: Ubuntu 20.04 LTS 상위나 하위 버전은 **ROS Noetic version** 지원 안됨

: PC의 가상 머신(Virtual box)를 사용할 경우 ROS가 정상작동하지 않을 수 있어 **우분투를 단독 설치**

: <https://releases.ubuntu.com/20.04/>를 개인 PC(Remote PC가 아닌 별도 PC)에 download ↪ boot USB 만들기 위함

(2) install Ubuntu on Remote PC

: 다운로드 받은 파일은 ISO 디스크 파일이므로 **USB(개인 USB 최소 6GB 필요)**에 바로 옮기면 파일이 손상됨

➔ 이후, 다른 용도로 USB 사용불가

4. ROS 개발환경 구축

: 다운로드 받은 Ubuntu 파일은 ISO 디스크 파일이므로 **USB(개인 USB 최소 6GB 필요)**에 바로 옮기면 파일이 손상됨

(ubuntu-20.04.6-desktop-amd64.iso)

→ Boot USB 설치하기 위하여 <https://www.balena.io/etcher/> 에서 “**balenaEtcher-Setup-1.18.11.exe**” 파일 download

← **balenaEtcher-Setup-1.18.11.exe** 를 관리자 권한으로 개인 PC(**Remote PC가 아님**)에 설치하기

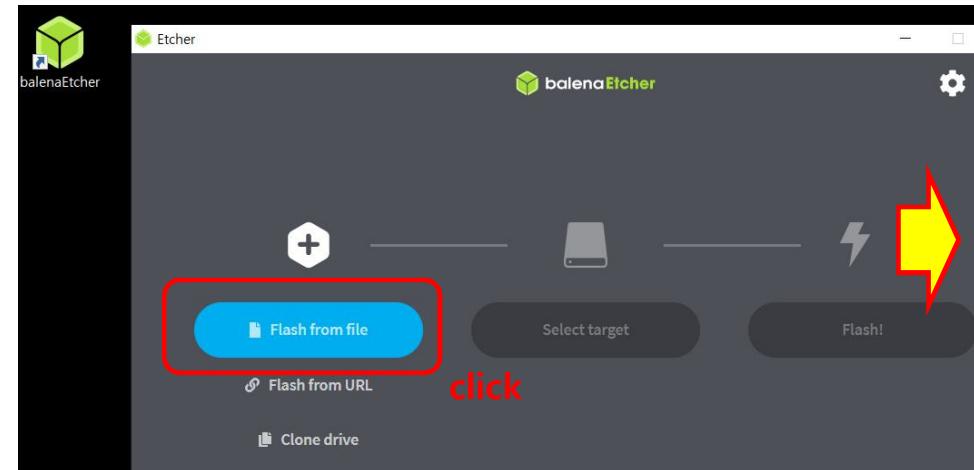
: 위 사이트(balena.io/etcher)에서 프로그램을 다운받고



(1) 개인 PC에 관리자권한으로 etcher 프로그램 실행

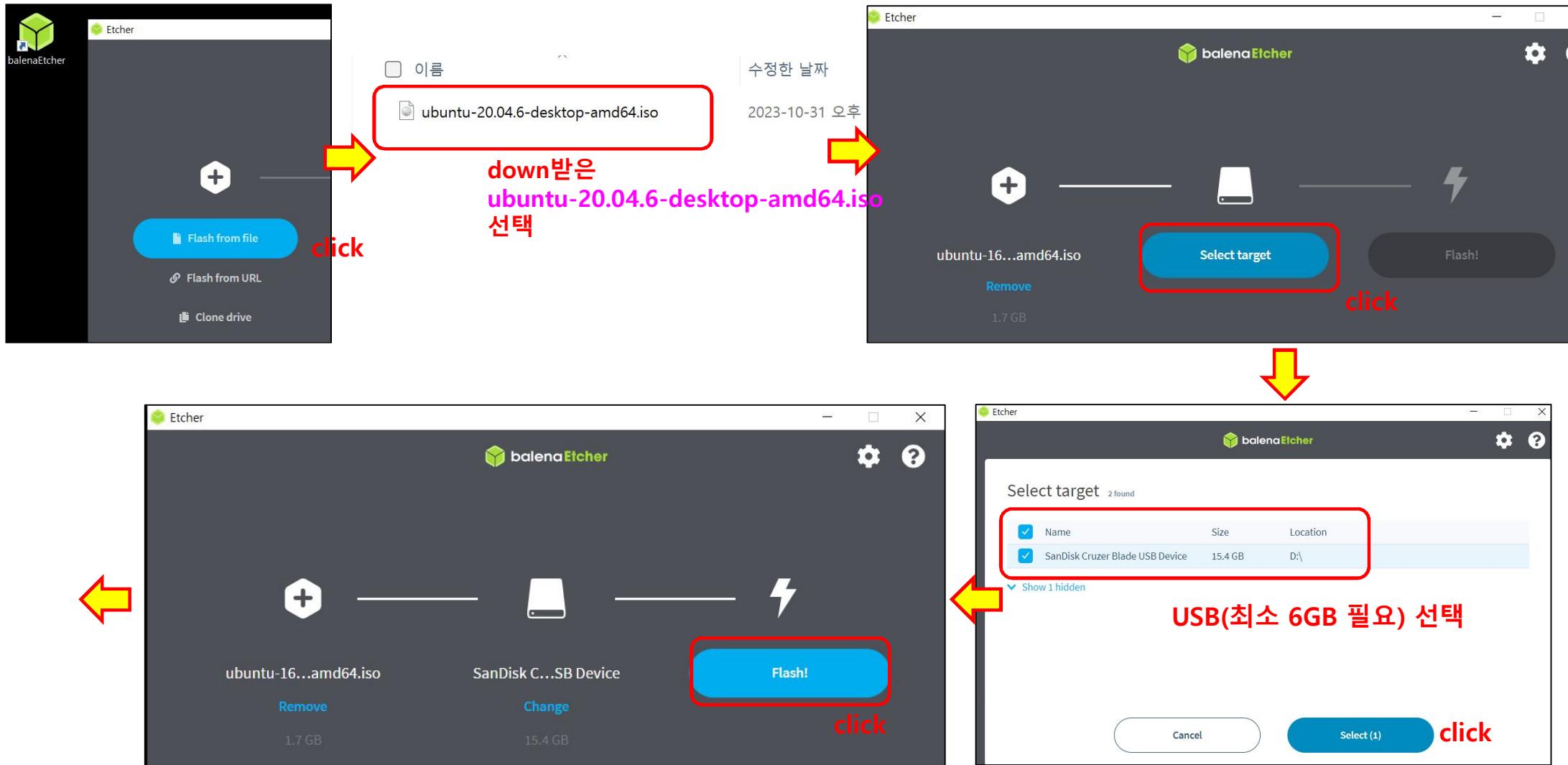
(2) USB(최소 6GB 필요)를 개인 PC에 연결한 후

balena_etcher에서

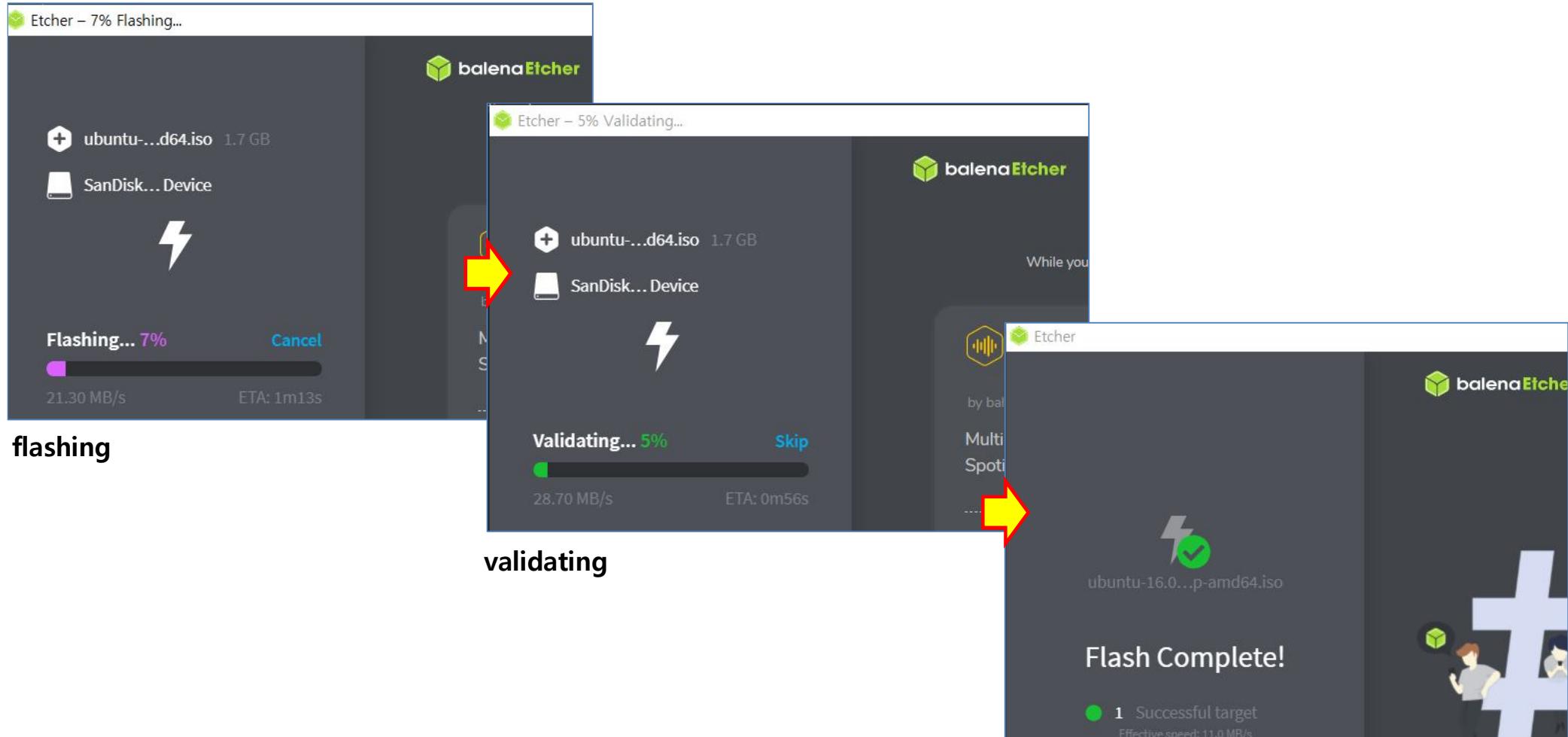


click

4. ROS 개발환경 구축



4. ROS 개발환경 구축



USB에 ubuntu-20.04.6-desktop-amd64.iso 설치 완료

4. ROS 개발환경 구축

- install Ubuntu on Remote PC(미니PC) 설치 여부 확인

① Remote PC(미니PC)에 Keyboard, Mouse, Monitor 연결 후 전원 ON

: 암호는 1234 (keyboard의 NumLock 해제)

: Ubuntu 설치여부 확인 (아래 Ubuntu 화면)



지난 과정에 Ubuntu 설치가 되어 있는 경우



◆ ubuntu version 확인하기

: CTRL + ALT + T 를 사용해 터미널 창 열기

: lsb_release -a

→ ubuntu version 확인

```
yongseok@yongseok:~$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 16.04.7 LTS
Release:        16.04
Codename:       xenial
yongseok@yongseok:~$
```

입력 후 enter

Version 확인

4. ROS 개발환경 구축

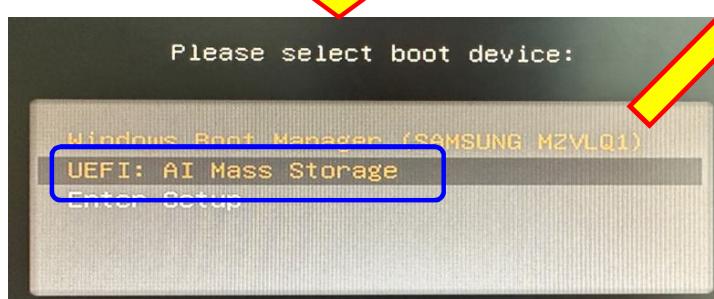
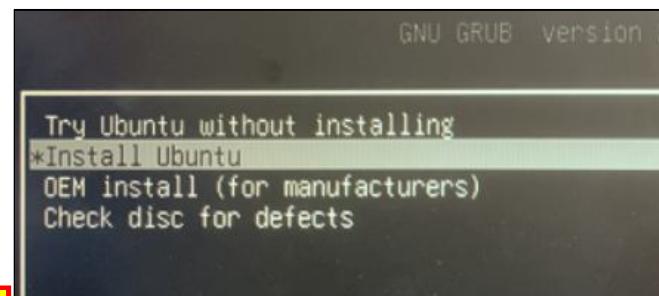
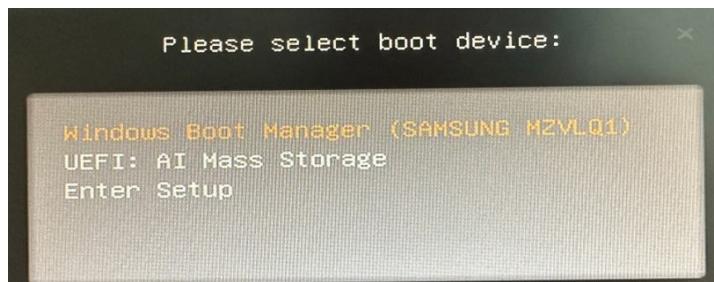
- install Ubuntu on Remote PC(미니PC- MSI Cubi N100 ADL Win11Home)

- windows, centos, Ubuntu 등이 설치된 Remote PC(미니PC)에 Ubuntu 설치하기

: 미니 PC에 마우스, 모니터, 키보드 연결하기

: ubuntu-20.04.6-desktop-amd64.iso가 설치된 USB를 미니PC에 연결

: powering on the PC 이후에 F11(억수로 많이) 누르기(MSI boot menu)

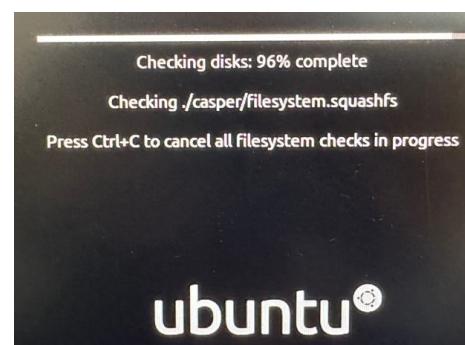


↑ ↓로 USB 선택 후 enter

(본인 USB 이름을 기억하기)

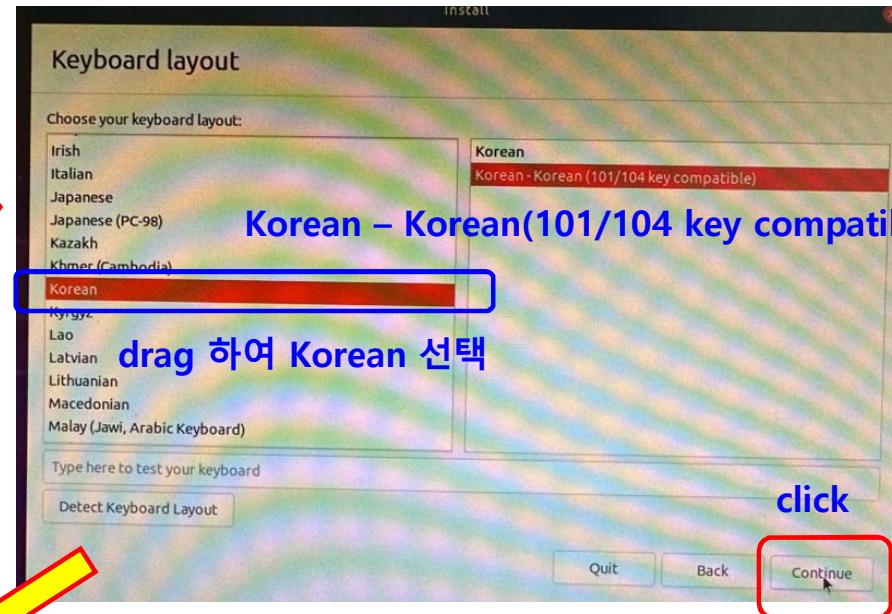
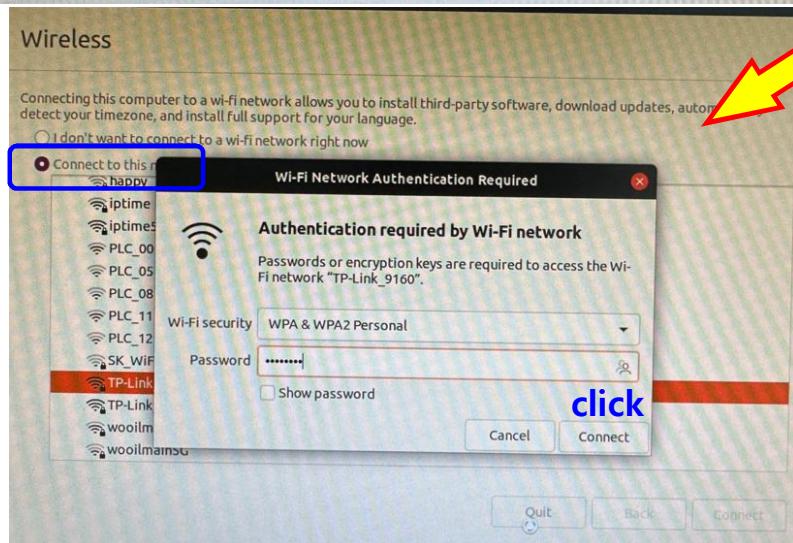
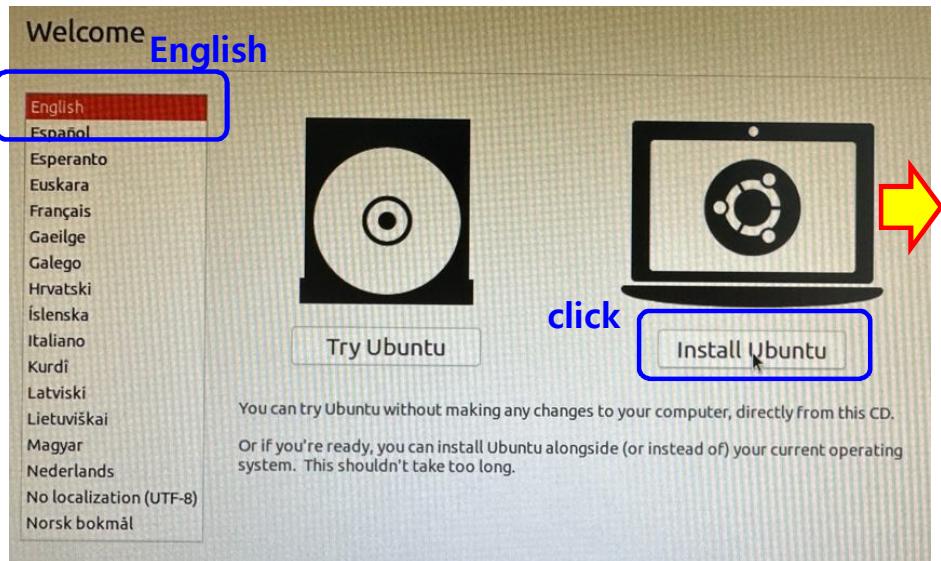


Ubuntu 설치 시작 화면



이미 Ubuntu가 설치 되어 있어도 F11을 누르면 설치 화면으로 진입

4. ROS 개발환경 구축

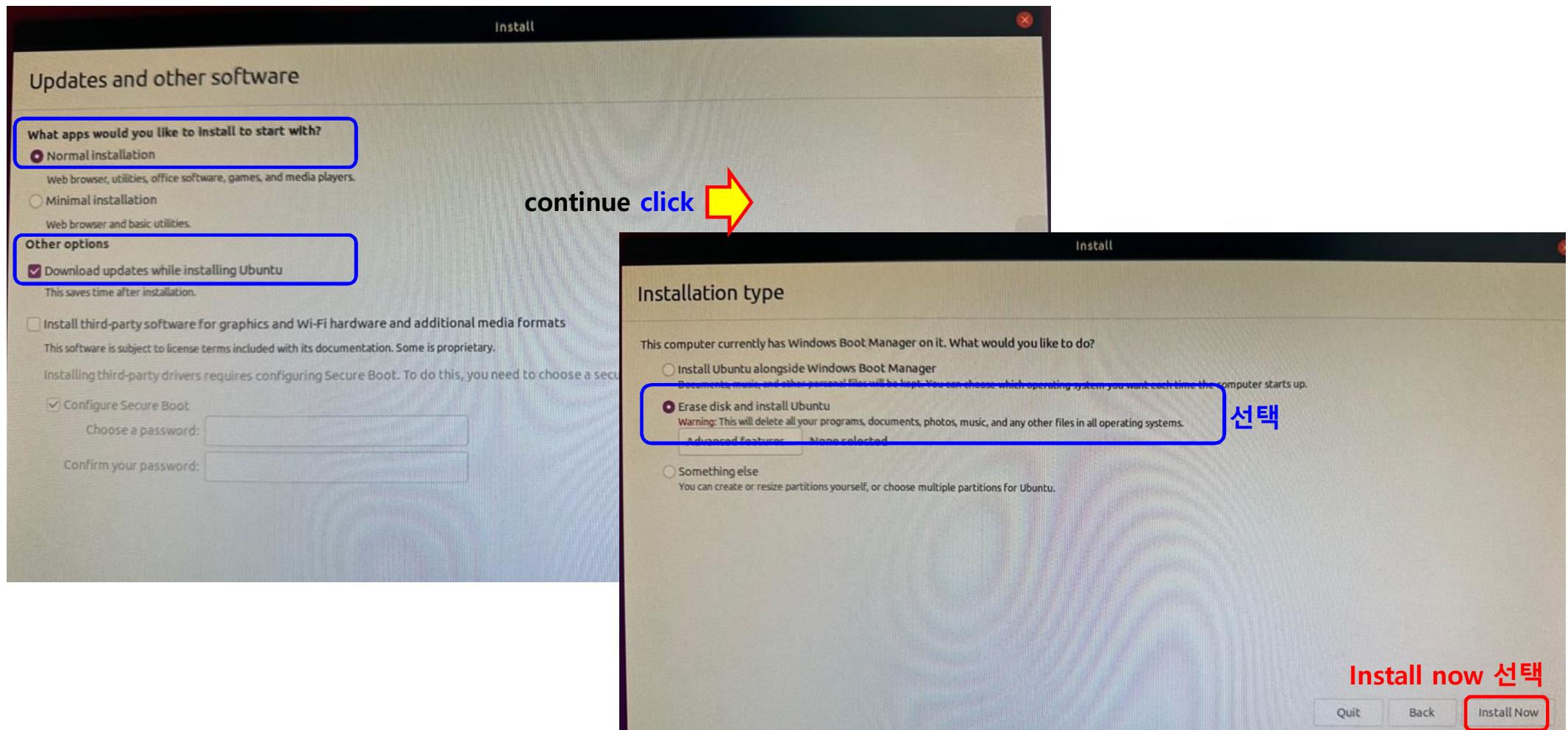


Network 연결 선택 후 TP-Link 9160 WiFi 선택

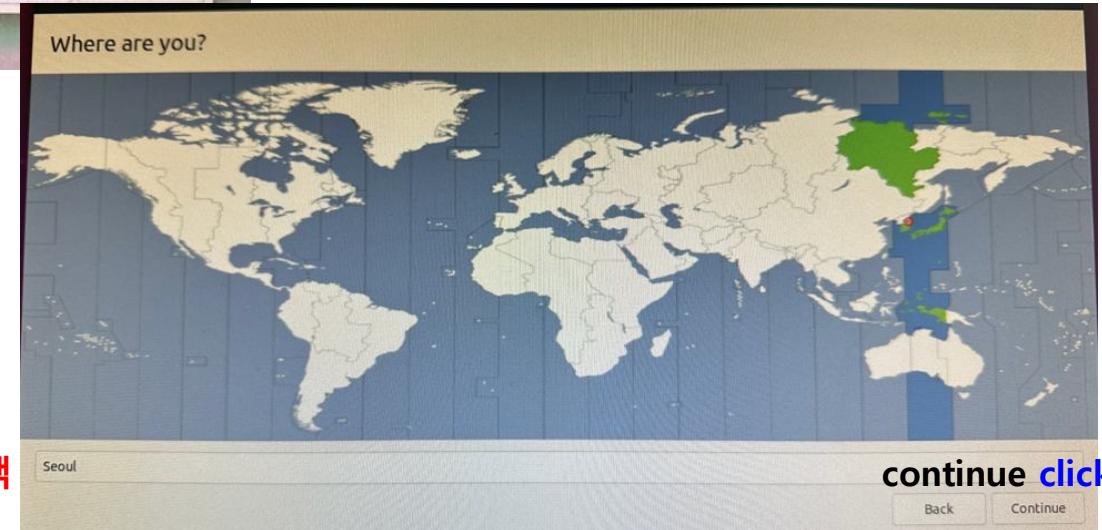
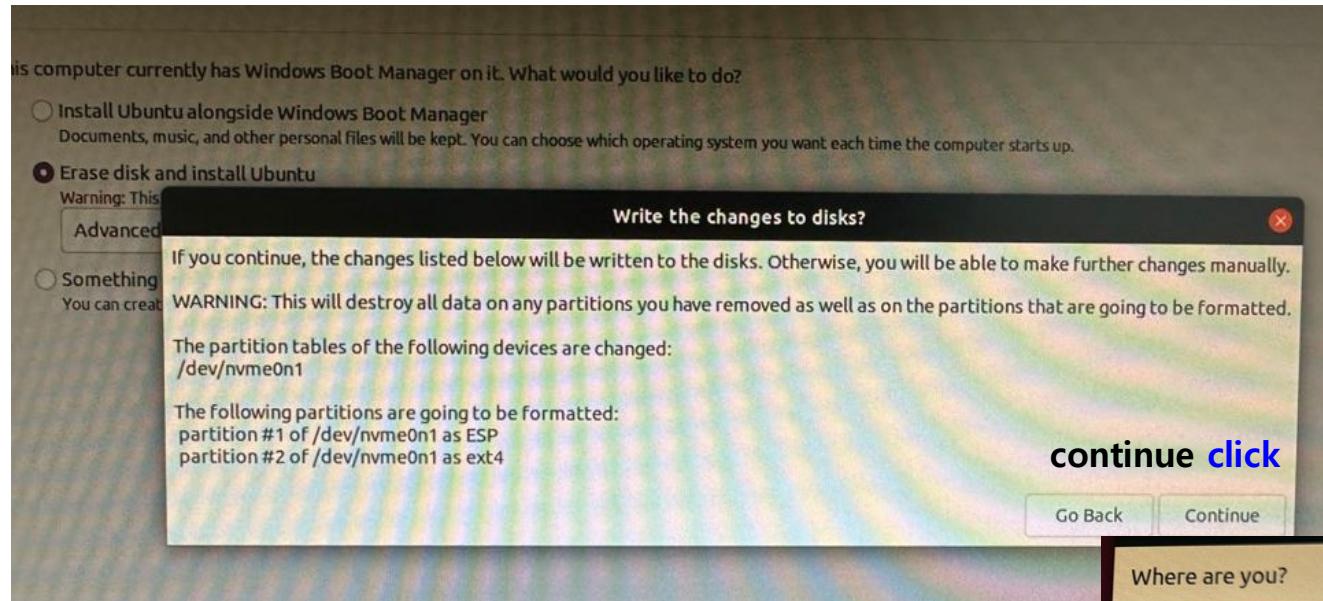
TP-Link 9160 5G WiFi 선택

(pw : 91919533)

4. ROS 개발환경 구축



4. ROS 개발환경 구축

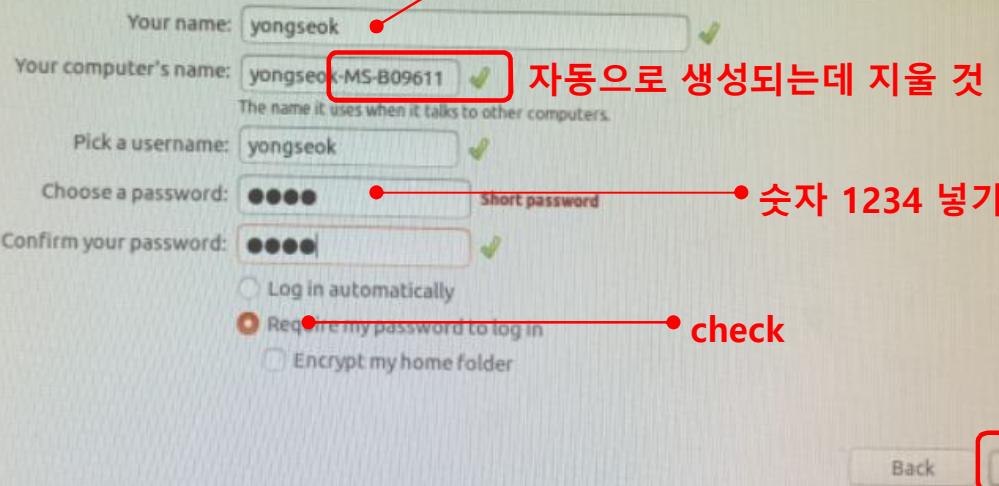


4. ROS 개발환경 구축

all (as superuser)

Who are you?

오직 영문 소문자 이름 넣기



Your name: yongseok

Your computer's name: yongseok-MS-B09611 ✓
The name it uses when it talks to other computers.

Pick a username: yongseok ✓

Choose a password: short password

Confirm your password: ✓

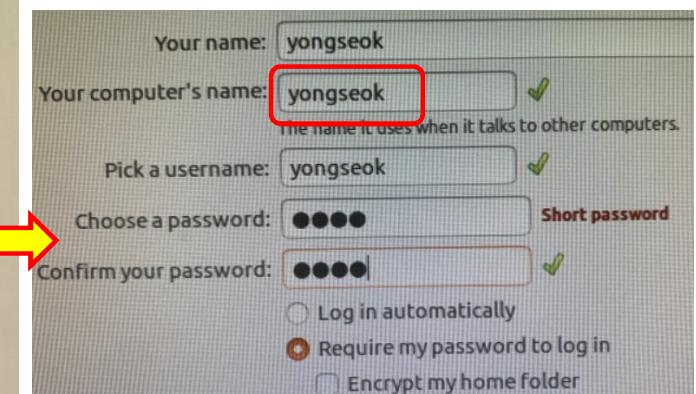
Log in automatically
 Require my password to log in
 Encrypt my home folder

Back Continue

자동으로 생성되는데 지울 것

숫자 1234 넣기

check



Your name: yongseok

Your computer's name: yongseok ✓
The name it uses when it talks to other computers.

Pick a username: yongseok ✓

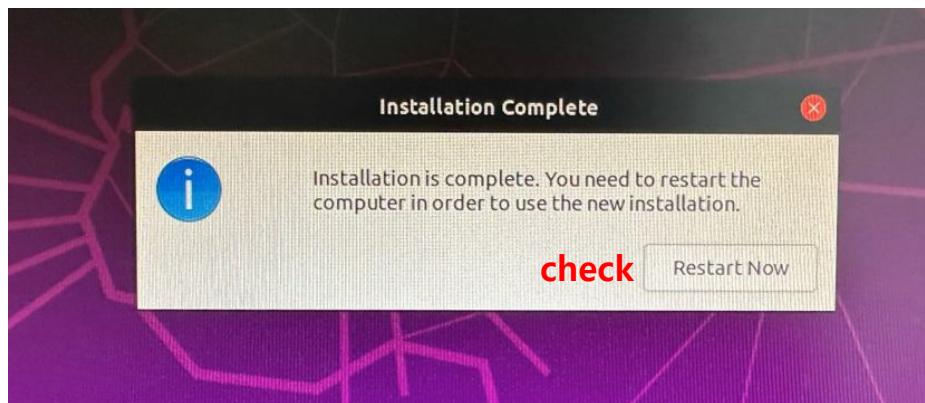
Choose a password: Short password

Confirm your password: ✓

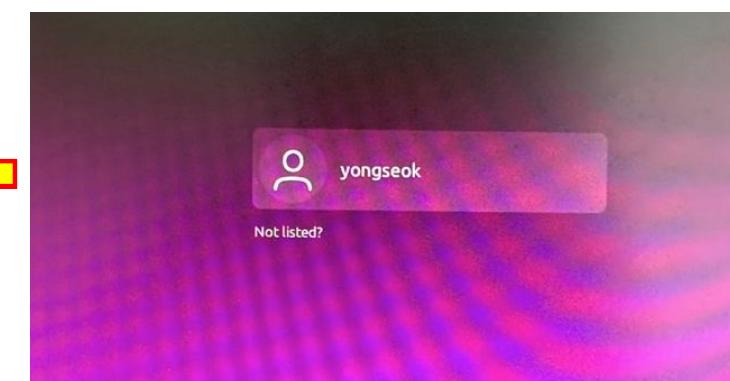
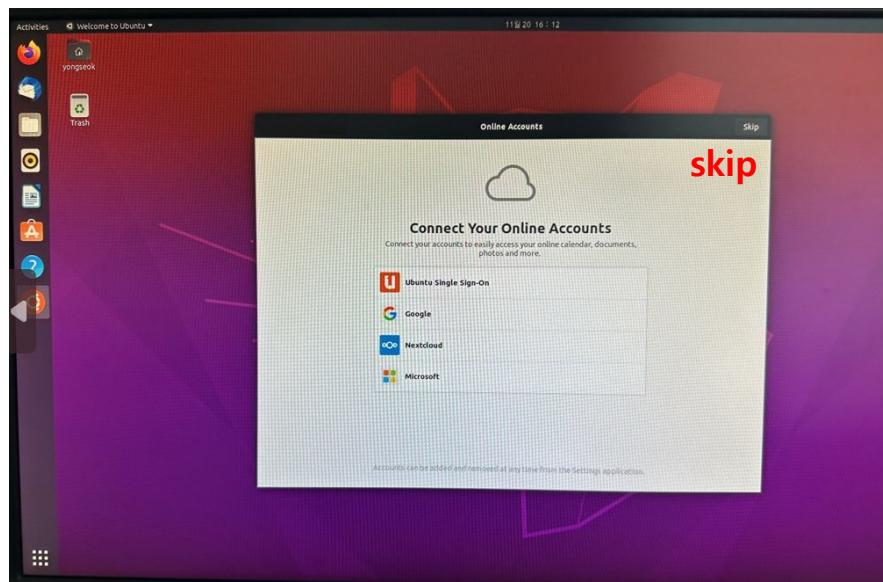
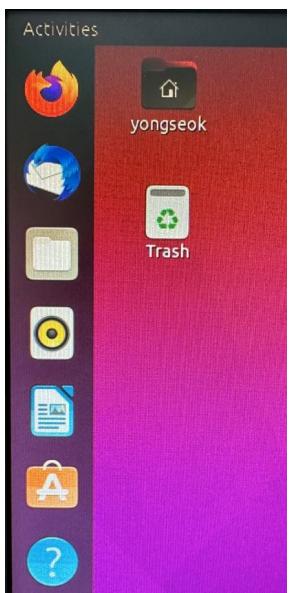
Log in automatically
 Require my password to log in
 Encrypt my home folder



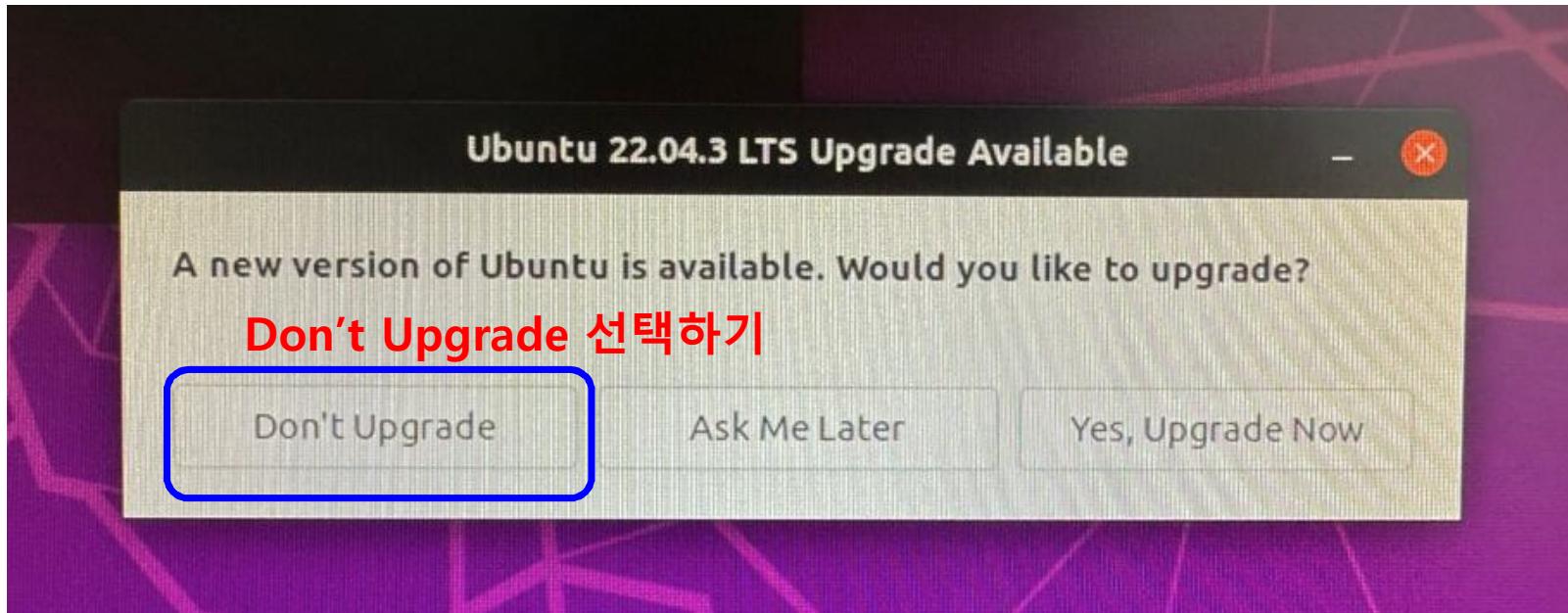
4. ROS 개발환경 구축



Booting USB를 제거하고 enter 누르기



4. ROS 개발환경 구축



4. ROS 개발환경 구축

Terminal 창 열기



: CTRL + ALT + T 를 사용해 터미널 창 열기

→ 반복 실행 시, 반복한 횟수 만큼 터미널 창이 여러 개 열림

yongseok@yongseok-MS-B09611: ~
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
yongseok@yongseok-MS-B09611: ~\$

: sudo 명령어(super_user do)

→ Linux에서 특정 명령을 실행하거나 파일에 접근하기 위해 root 권한이 필요.

일반 사용자(user)가 root 권한을 사용하기 위해서 sudo 명령어를 사용

→ sudo 다음에 실행할 명령을 입력하면 root 권한으로 명령어를 실행

★: CTRL + Shift + Q 열려있는 터미널 창 닫기

→ 반복 실행 시, 반복한 횟수 만큼 터미널 창이 닫힘

root : 최고 관리자.

: 운영체제의 모든 것을 제어할 권리를 지님

: 시스템 파괴할 권한도 포함(실수든 해킹이든)

컴퓨터 주인의 계정 :

리눅스 운영체제에서는 기본적으로 컴퓨터 주인 계정이 root보다 권한이 낮은 일반 사용자로 만들어짐.

시스템 관리 작업 등을 할 때, sudo라는 명령을 사용해서 임시 root 권한을 얻음

4. ROS 개발환경 구축

◆ ubuntu version 확인하기

: CTRL + ALT + T 를 사용해 터미널 창 열기

: lsb_release -a

→ ubuntu version 확인

```
yongseok@yongseok:~$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 20.04.6 LTS
Release:        20.04
Codename:       focal
yongseok@yongseok:~$
```

[명령어]

apt-get : 또는 **apt**(Advanced Packaging Tool)은 **Debian** 계열의 **Ubuntu**에서 쓰이는 **package** 관리 명령어

sudo : super-user 권한으로 실행하기 위함

wget : **web get**. 웹에서 파일을 다운로드할 때 사용되는 명령어

chmod 755: change mode. 대상 파일과 디렉토리의 사용권한을 변경할 때 사용. 755 소유자 **권한(읽고, 쓰기, 실행)**

bash : shell은 운영 체제 상에서 다양한 운영 체제 기능과 서비스를 구현하는 인터페이스를 제공하는 프로그램

: bash는 unix의 shell. / bash: 파일을 읽고 실행 **bashrc** 를 호출

4. ROS 개발환경 구축

: sudo apt-get update 가장 중요(ROS 환경을 위해)

: sudo apt-get upgrade

The terminal window shows the following output:

```
yongseok@yongseok: ~
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

yongseok@yongseok:~$ sudo apt-get update
[sudo] password for yongseok: ●
Hit:1 http://kr.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://security.ubuntu.com/ubuntu focal-security InRelease
Get:3 http://kr.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Hit:4 http://kr.archive.ubuntu.com/ubuntu focal-backports InRelease
Fetched 114 kB in 3s (42.9 kB/s)
Reading package lists... Done
yongseok@yongseok:~$ sudo apt-get upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following packages have been kept back:
  thunderbird-advantage-tools
  ●
```

Annotations in red:

- 입력 후 enter (Red dot at the end of the password line)
- 1234 enter(입력 안보임) (Red dot at the end of the update command line)
- update 화면(완료 확인할 것) (Red dot at the end of the update output)
- 입력 후 enter (Red dot at the end of the upgrade command line)
- upgrade 화면(완료 확인할 것) (Red dot at the end of the upgrade output)

생략

The terminal window shows the following output:

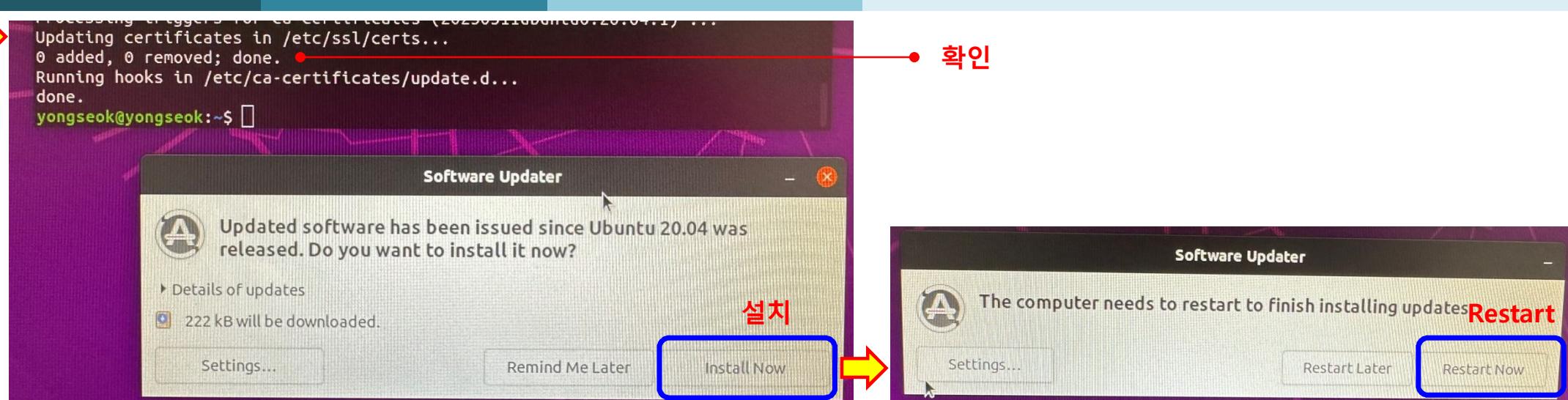
```
xxd
243 upgraded, 0 newly installed, 0 to remove and 1 not upgraded.
Need to get 71.1 MB/528 MB of archives.
After this operation, 27.6 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://kr.archive.ubuntu.com/ubuntu focal-updates/main amd64 thunderbird-
```

Annotation in red:

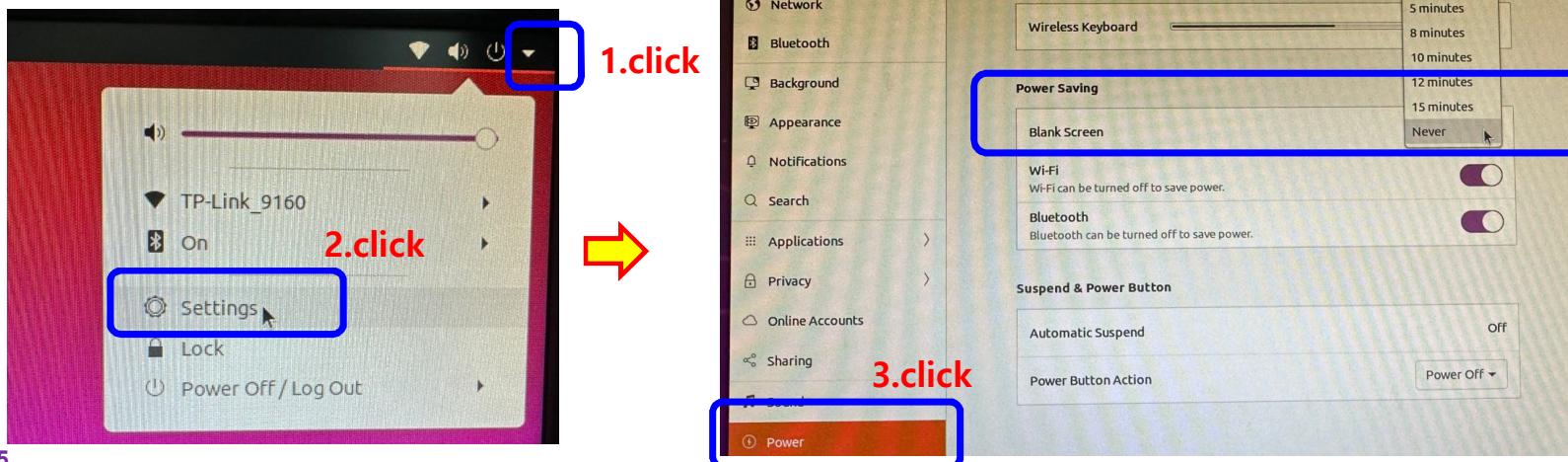
- Y 입력 (Red dot at the end of the Y/n prompt)

생략

4. ROS 개발환경 구축



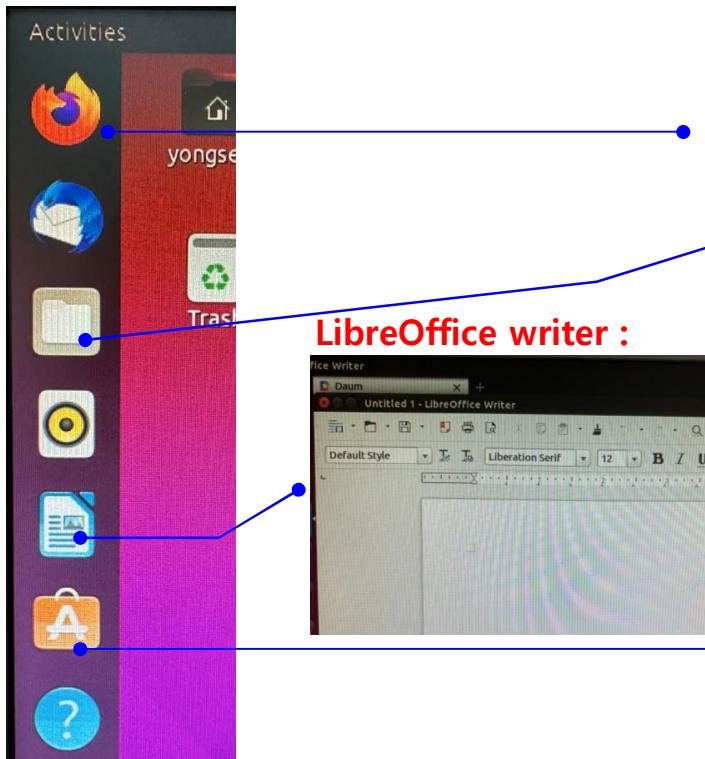
◆ Ubuntu setting



4. ROS 개발환경 구축

(2) install Ubuntu on Remote PC

② Ubuntu OS 이해하기

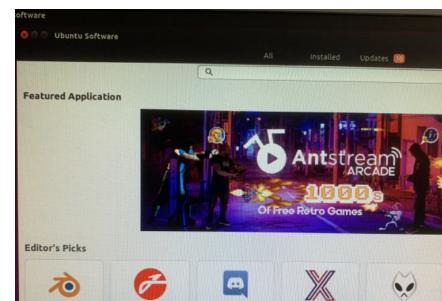


③ Terminal Window 열기 : Ctrl + Alt + T 누르기

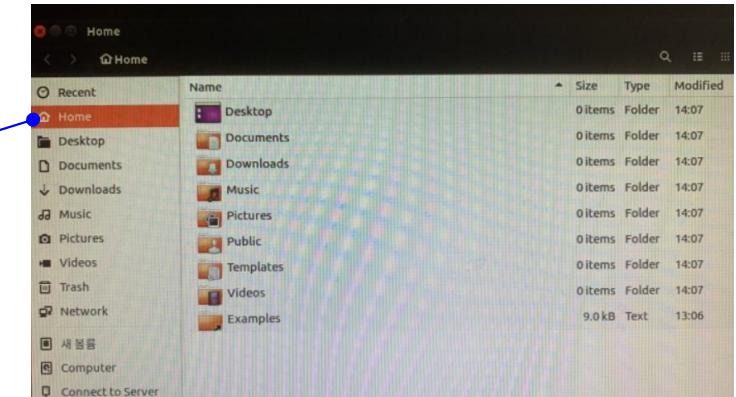
Firefox web browser :
windows OS에서의 internet browser



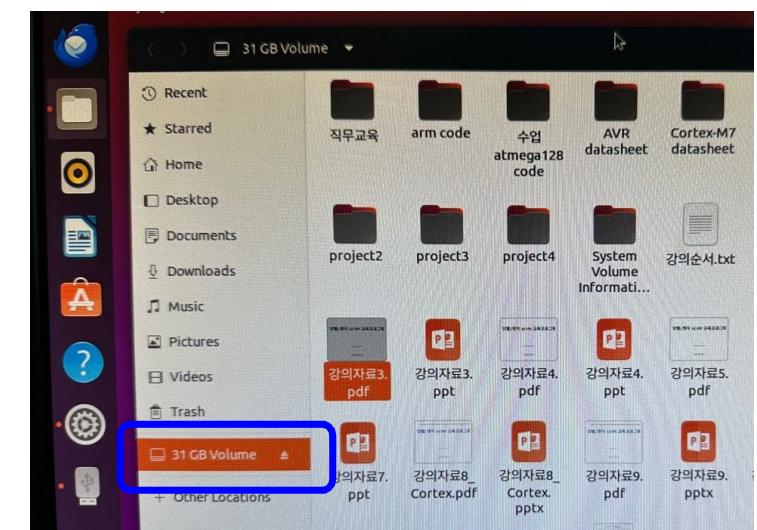
Ubuntu software : Applications



Files : windows OS에서의 탐색기 동일



USB : USB's name display



4. ROS 개발환경 구축

Ubuntu 16.04 SKIP



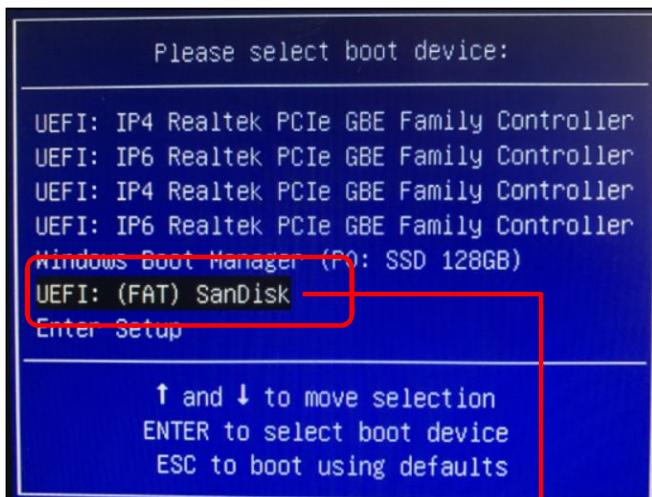
- install Ubuntu on Remote PC(미니PC)

- ① windows, centos, Ubuntu 등이 설치된 Remote PC(미니PC)에 Ubuntu 설치하기

: 미니 PC에 마우스, 모니터, 키보드 연결하기

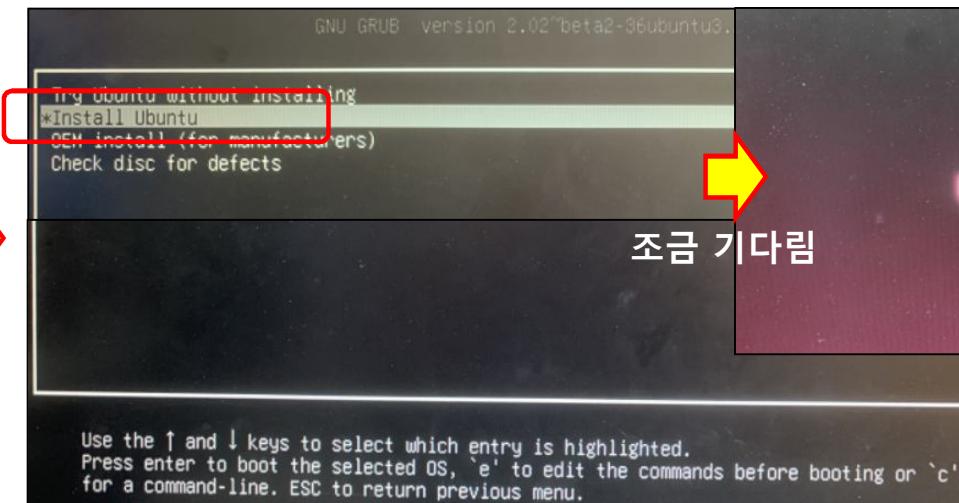
: ubuntu-20.04.6-desktop-amd64.iso가 설치된 USB를 미니PC에 연결

: powering on the PC 이후에 F11 누르기(MSI boot menu)



↑ ↓로 USB 선택 후 enter

(본인 USB 이름을 기억하기)

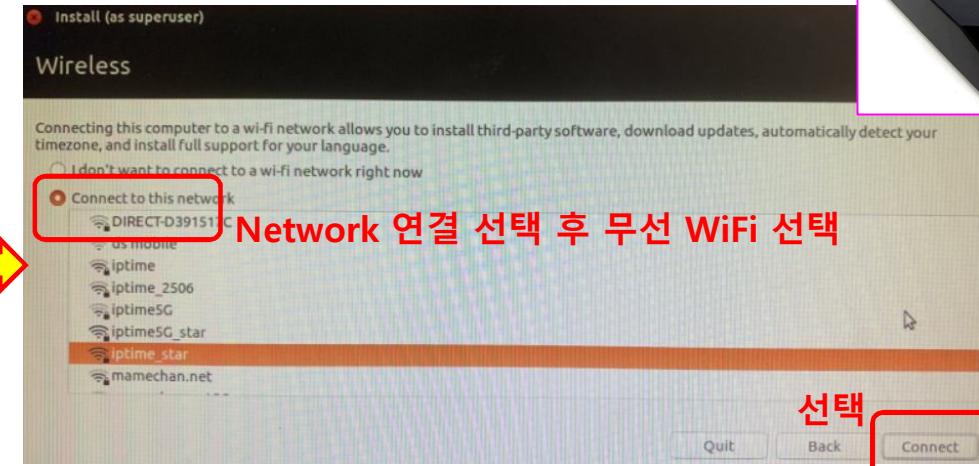
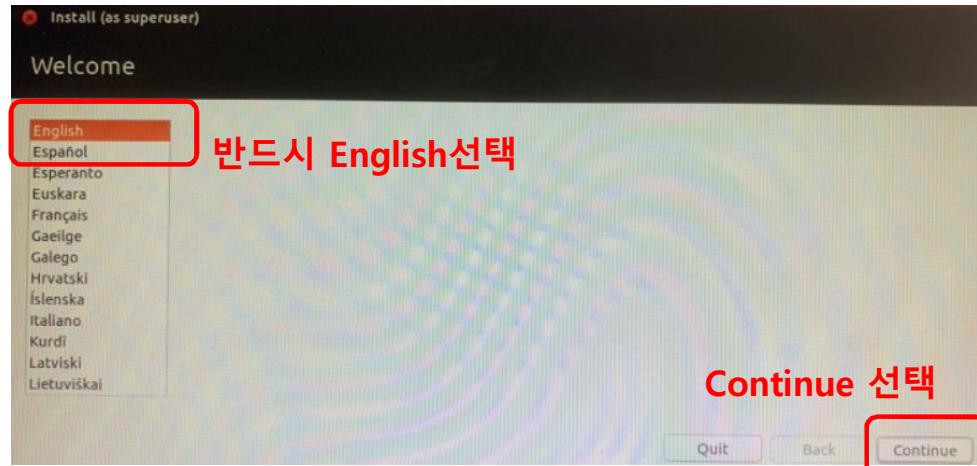


↑ ↓로 Install Ubuntu 선택 후 enter

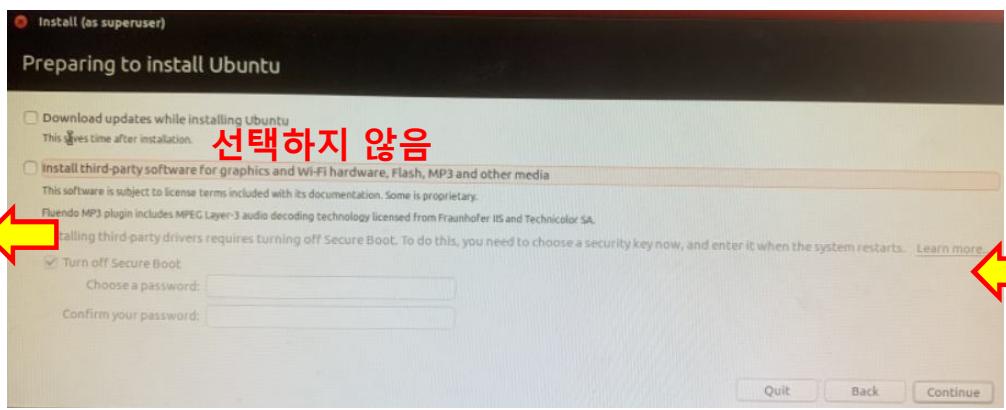
Ubuntu 설치 시작 화면

이미 Ubuntu가 설치 되어 있어도 F11을 누르면 설치 화면으로 진입

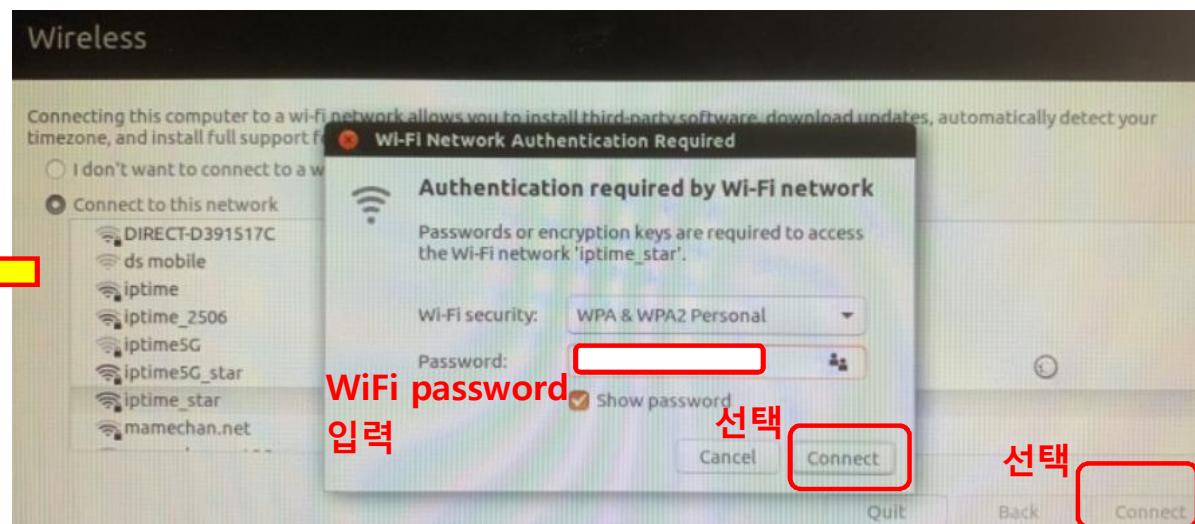
4. ROS 개발환경 구축



Ubuntu 16.04 SKIP



설치 중 업데이트 여부와 기타 소프트웨어 설치 항목
→ 체크하지 않고 '계속'클릭



4. ROS 개발환경 구축

Ubuntu 16.04 SKIP



Install (as superuser)

Installation type

This computer currently has Windows Boot Manager on it. What would you like to do?

- Erase Ubuntu 16.04.7 LTS and reinstall
Warning: This will delete all your Ubuntu 16.04.7 LTS programs, documents, photos, music, and any other files.
- Install Ubuntu 16.04.7 LTS alongside Ubuntu 16.04.7 LTS
Documents, music, and other personal files will be kept. You can choose which operating system you want each time the computer starts up.
- Erase disk and install Ubuntu
Warning: This will delete all your programs, documents, photos, music, and any other files in all operating systems.
- Encrypt the new Ubuntu installation for security
You will choose a security key in the next step.
- Use LVM with the new Ubuntu installation
This will set up Logical Volume Management. It allows taking snapshots and easier partition resizing.
- Something else
You can create or resize partitions yourself, or choose multiple partitions for Ubuntu.

Disk 지우고 Ubuntu 설치하기 선택

Continue 선택

Partition format 과정

Install (as superuser)

Erase disk and install Ubuntu

Select drive: SCSI1 (0,0,0) (sda) - 128.0 GB ATA SSD 128GB

The entire disk will be used:

Ubuntu /dev/sda (ext4)
128.0 GB

4 partitions will be deleted, use the [advanced partitioning tool](#) for more control

Install now 선택

Erase disk and install Ubuntu

Select drive: SCSI1 (0,0,0) (sda) - 128.0 GB ATA SSD 128GB

The entire disk will be used:

Write the changes to disks?
If you continue, the changes listed below will be written to the disks. Otherwise, you will be able to make further changes manually.

WARNING: This will destroy all data on any partitions you have removed as well as on the partitions that are going to be formatted.

The partition tables of the following devices are changed:
SCSI1 (0,0,0) (sda)

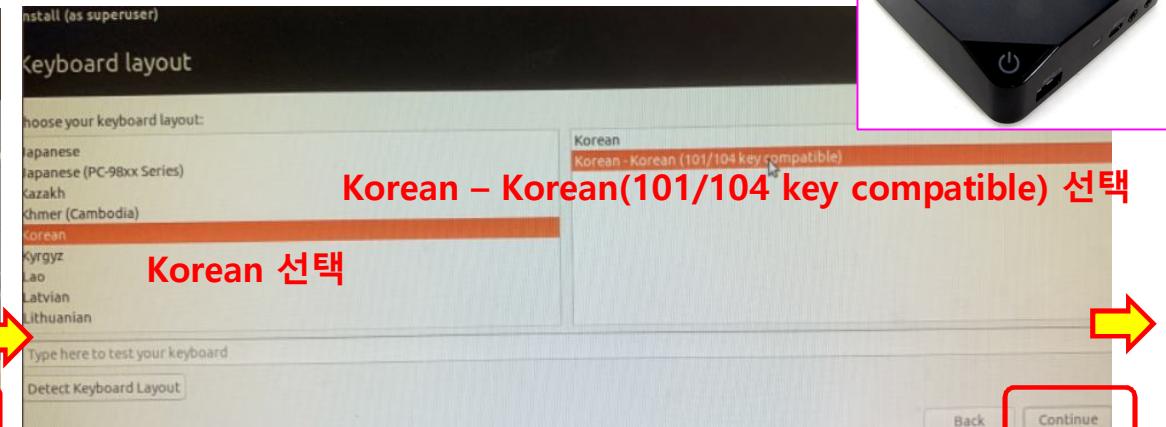
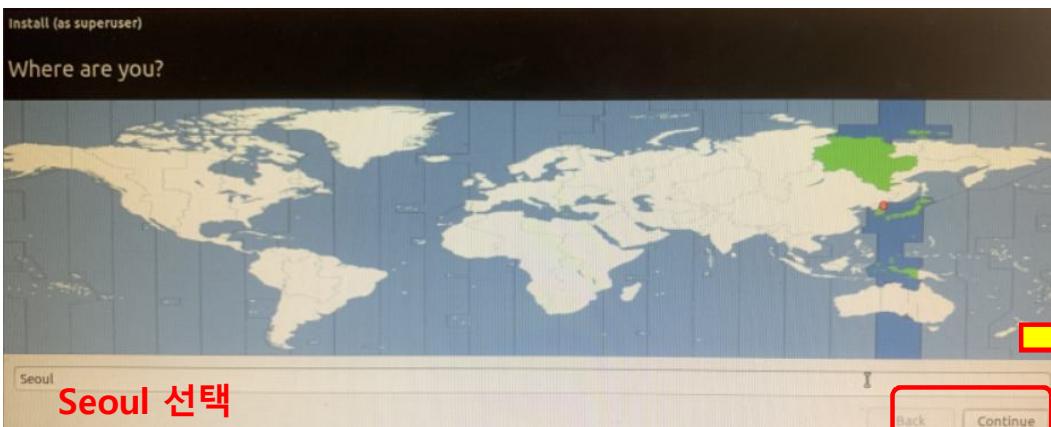
4 partitions will be deleted:

The following partitions are going to be formatted:
partition #1 of SCSI1 (0,0,0) (sda) as ESP
partition #2 of SCSI1 (0,0,0) (sda) as ext4
partition #3 of SCSI1 (0,0,0) (sda) as swap

Continue 선택

4. ROS 개발환경 구축

Ubuntu 16.04 SKIP



오직 영문 소문자 이름 넣기

자동으로 생성되는데 지울 것

숫자 1234 넣기

check

Your name: yongseok

Your computer's name: yongseok-MS-B09611

Pick a username: yongseok

Choose a password: Short password

Confirm your password:

Log in automatically

Require my password to log in

Encrypt my home folder

Back Continue

Your name: yongseok

Your computer's name: yongseok

Pick a username: yongseok

Choose a password: Short password

Confirm your password:

Log in automatically

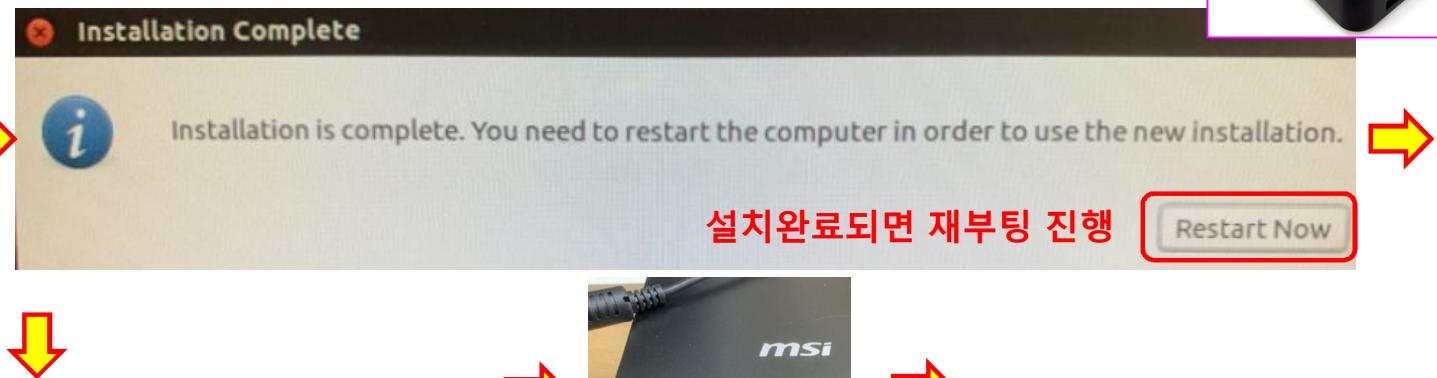
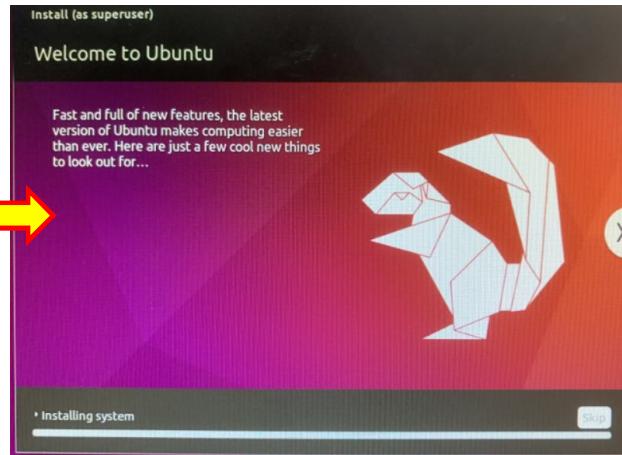
Require my password to log in

Encrypt my home folder

Back Continue

4. ROS 개발환경 구축

Ubuntu 16.04 SKIP



부팅 후, 아래와 같은 오류 message가 나온다면,

```
[ 9.320716] Couldn't get size: 0x800000000000000e  
→ NVIDIA 그래픽 카드가 문제임(bios setting에서 해결해야 함)
```



→ Power를 눌러 강제 종료 후 재 시작



암호는 1234 (keyboard의 NumLock 해제)



Ubuntu 시작 화면

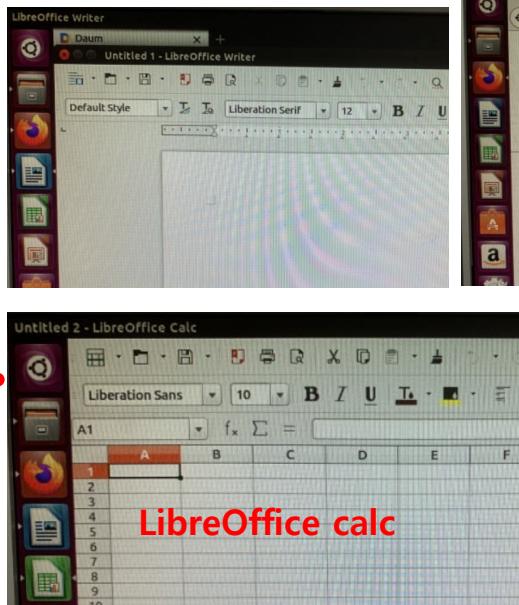
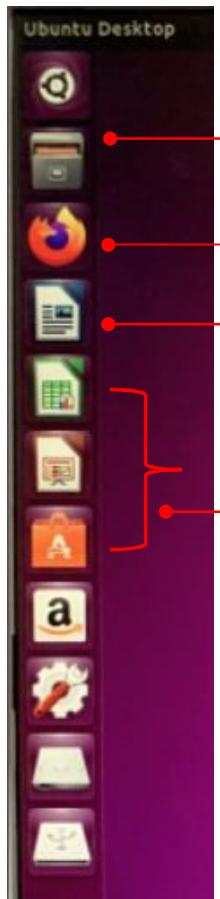
4. ROS 개발환경 구축

Ubuntu 16.04 SKIP



(2) install Ubuntu on Remote PC

② Ubuntu OS 이해하기

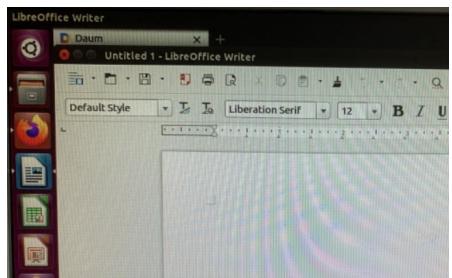


LibreOffice calc

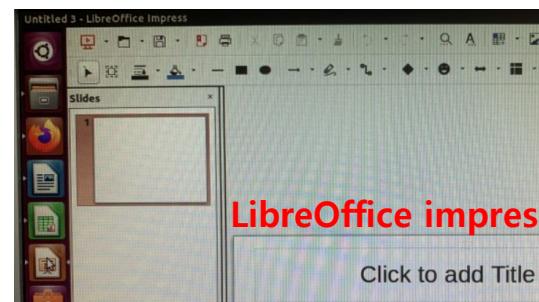
windows OS에서의 internet browser



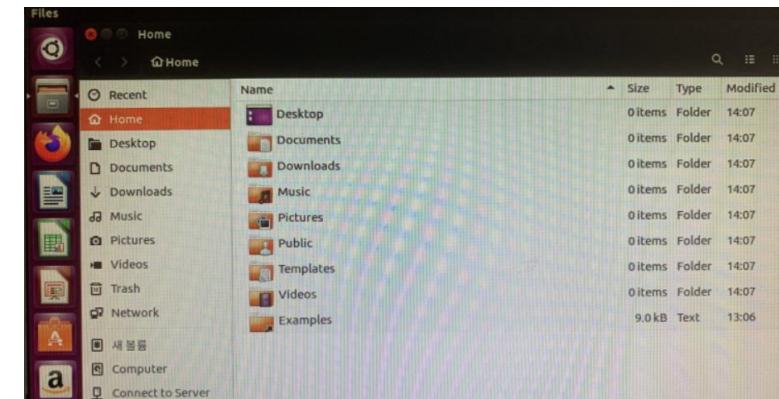
Firefox web browser :



LibreOffice writer :

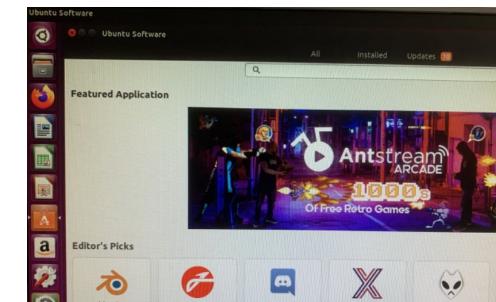


LibreOffice impress



Files : windows OS에서의 탐색기 동일

Ubuntu software : Applications



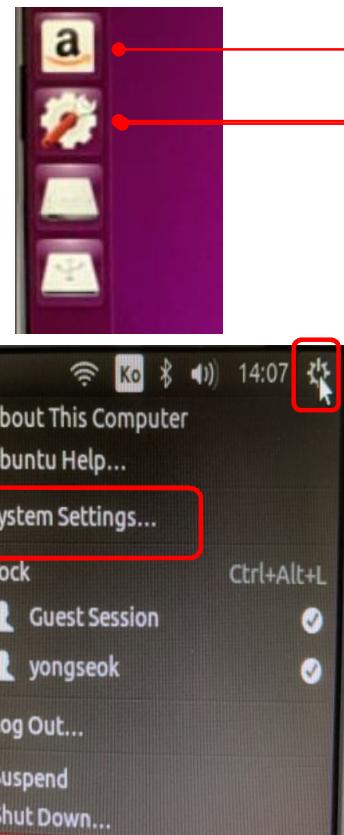
4. ROS 개발환경 구축

Ubuntu 16.04 SKIP

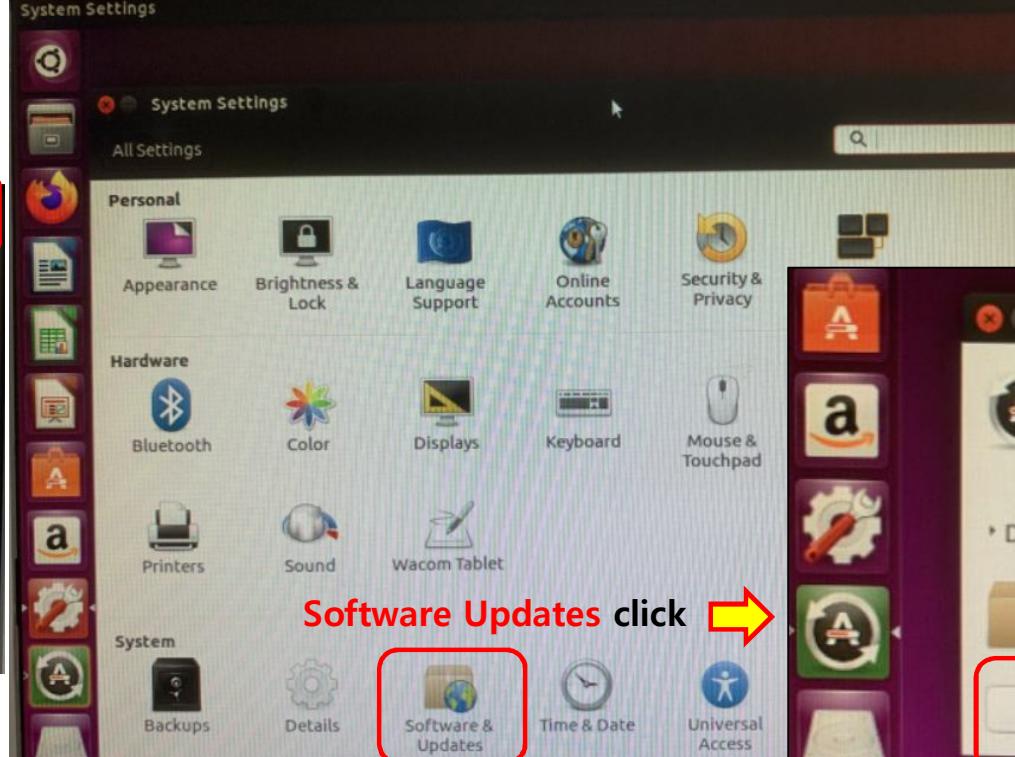


- install Ubuntu on Remote PC

② Ubuntu OS 이해하기

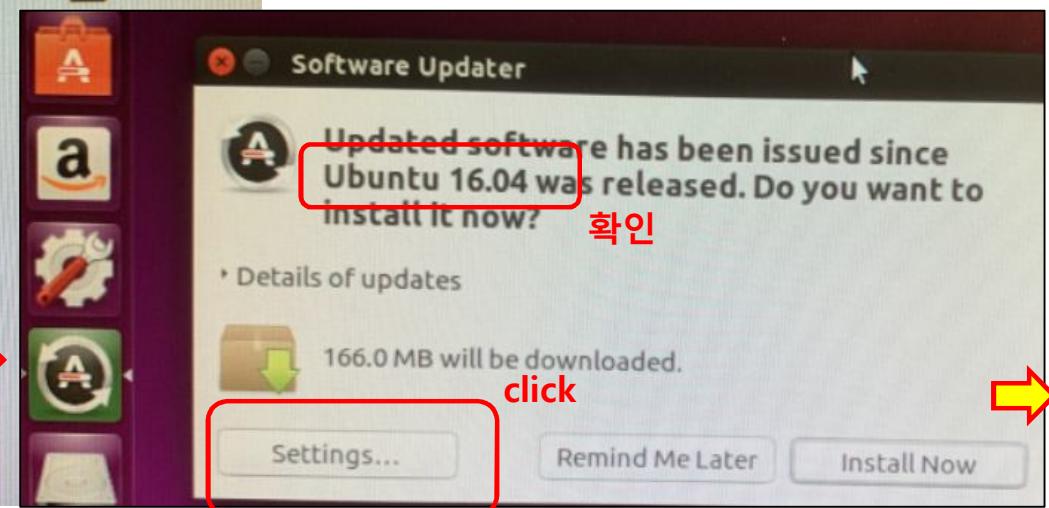
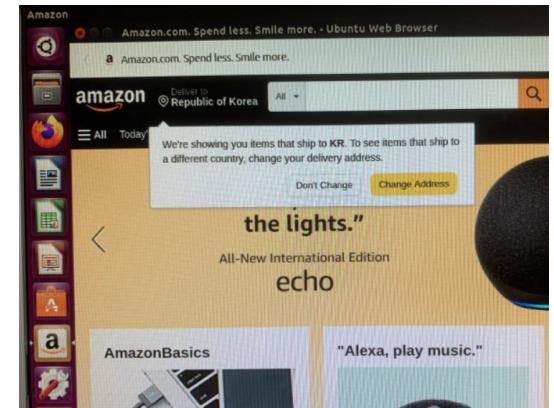


System setting : 설정



Software Updates click

Amazon



확인

click

click

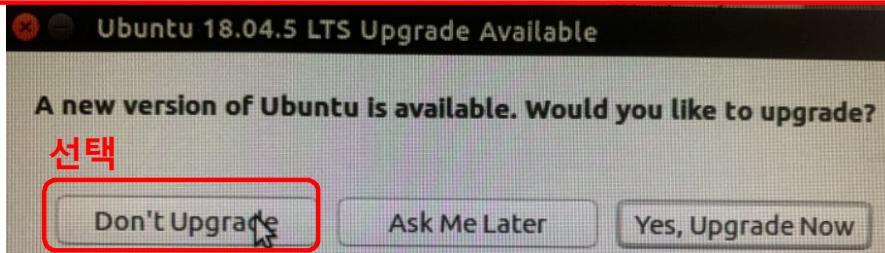
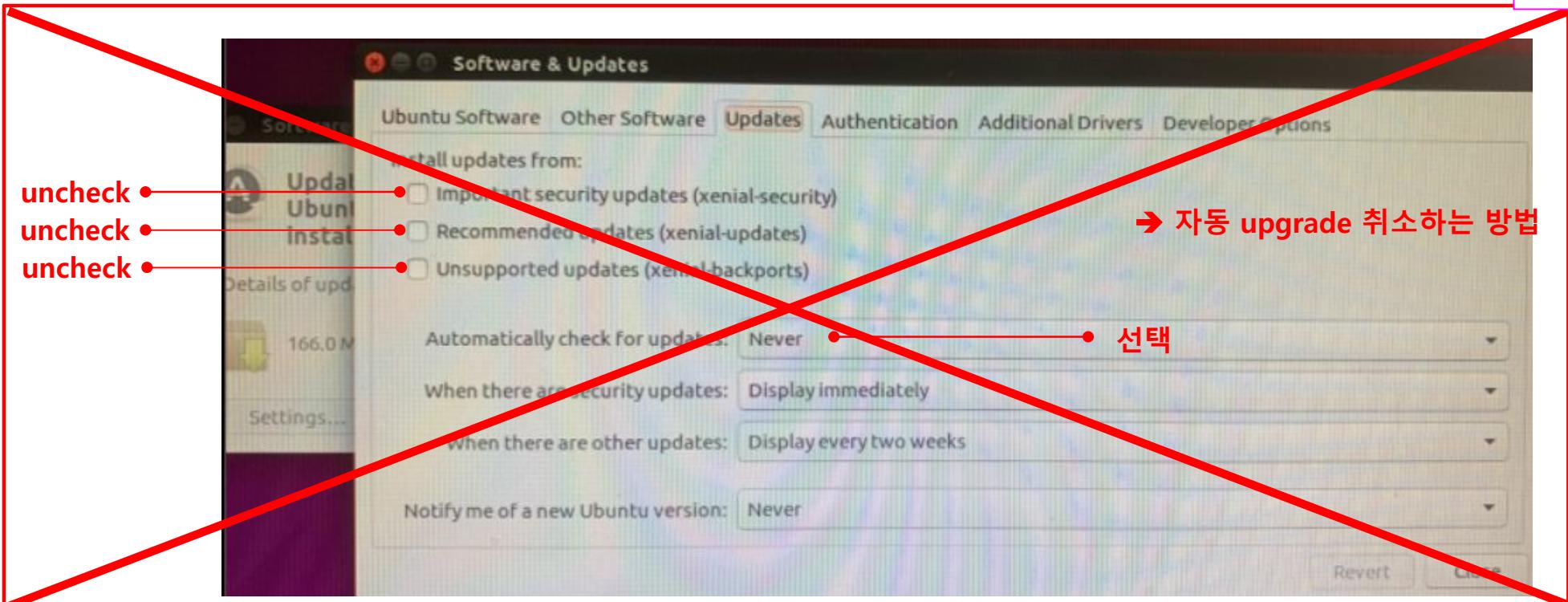
4. ROS 개발환경 구축

Ubuntu 16.04 SKIP



Ubuntu upgrade 금지 : ROS와 TurtleBot에서 지원 안됨 ★

Ubuntu 16.04.7 LTS 유지하기 (ubuntu-18.04.5 LTS upgrade 금지)



4. ROS 개발환경 구축

Ubuntu 16.04 SKIP



③ Terminal 창 열기



: CTRL + ALT + T 를 사용해 터미널 창 열기

→ 반복 실행 시, 반복한 횟수 만큼 터미널 창이 여러 개 열림

```
yongseok@yongseok-MS-B09611: ~  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
yongseok@yongseok-MS-B09611: ~$
```

: sudo 명령어(super_user do)

→ Linux에서 특정 명령을 실행하거나 파일에 접근하기 위해 root 권한이 필요.

일반 사용자(user)가 root 권한을 사용하기 위해서 sudo 명령어를 사용

→ sudo 다음에 실행할 명령을 입력하면 root 권한으로 명령어를 실행

★: CTRL + Shift + Q 열려있는 터미널 창 닫기

→ 반복 실행 시, 반복한 횟수 만큼 터미널 창이 닫힘

root : 최고 관리자.

: 운영체제의 모든 것을 제어할 권리를 지님

: 시스템 파괴할 권한도 포함(실수든 해킹이든)

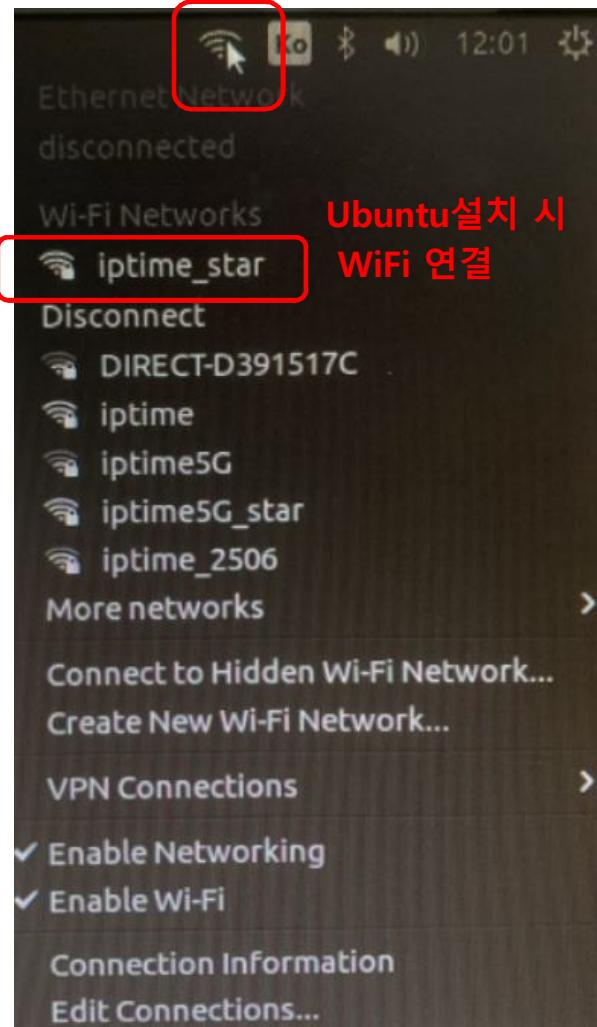
컴퓨터 주인의 계정 :

리눅스 운영체제에서는 기본적으로 컴퓨터 주인의 계정이 root보다 권한이 낮은 일반 사용자로 만들어짐.

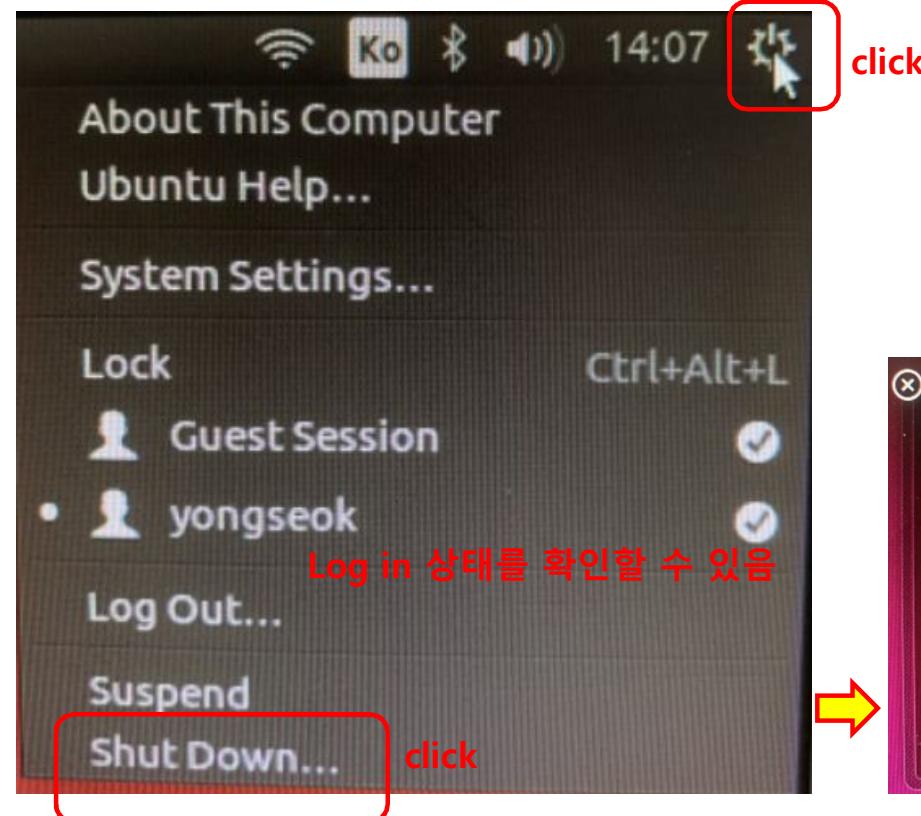
시스템 관리 작업 등을 할 때, sudo 라는 명령을 사용해서 임시 root 권한을 얻음

4. ROS 개발환경 구축

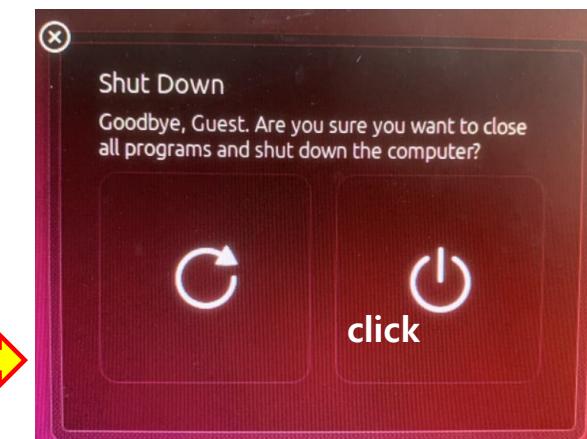
④ WiFi 확인하기



⑤ Shut down



Ubuntu 16.04 SKIP



교수용 : WiFi 설치 (TP-LINK 고객센터 1899 1086)

[2407호 TP-Link 0484]

: AX10 AX1500 WIFI MODEL

1. 고정 IP 확인
2. TP-Link 전원만 연결하기
3. 뒷면에 RESET Button 볼펜으로 10초 누르기
4. LED 등이 왼쪽부터 초록색 3개 들어옴
5. 핸드폰 설정에서 WIFI TP-Link 0484 선택(암호 뒷면 [31542418](#))
(못 찾으면 핸드폰 설정 나갔다. 들어갔다 반복)
6. 핸드폰에 주소 입력
192.168.0.1 (Password 임의 설정 a123456)
7. 선택
8. 선택
9. Advanced – Network – Internet – Static IP 선택
10. 고정 IP입력 후 저장
11. TP-Link 0484에 인터넷 케이블 연결. 왼쪽부터 LED 초록 4개 켜짐
12. 핸드폰에서 인터넷 되는지 확인. 완료

[2408호 TP-Link 1030]

: AX10 AX1500 WIFI MODEL

1. 고정 IP 확인
2. TP-Link 전원만 연결하기
3. 뒷면에 RESET Button 볼펜으로 10초 누르기
4. LED 등이 왼쪽부터 초록색 3개 들어옴
5. 핸드폰 설정에서 WIFI TP-Link 1030 선택(암호 뒷면 [56882230](#))
(못 찾으면 핸드폰 설정 나갔다. 들어갔다 반복)
6. 핸드폰에 주소 입력
192.168.0.1 (Password 임의 설정 a123456)
7. 선택
8. 선택
9. Advanced – Network – Internet – Static IP 선택
10. 고정 IP입력 후 저장
11. TP-Link 1030 **초록색(WAN)**에 인터넷 케이블 연결.
왼쪽부터 LED 초록 4개 켜짐
12. 핸드폰에서 인터넷 되는지 확인. 완료



교수용 : WiFi 설치 (TP-LINK 고객센터 1899 1086)

[스마트 팩토리실 TP-Link 9160]

: AX10 AX1500 WIFI MODEL

1. 고정 IP 확인
2. TP-Link 전원만 연결하기
3. 뒷면에 RESET Button 볼펜으로 10초 누르기
4. LED 등이 왼쪽부터 초록색 3개 들어옴
5. 핸드폰 설정에서 WIFI TP-Link 9160 선택
(못 찾으면 핸드폰 설정 나갔다. 들어갔다 반복)
6. 핸드폰에 주소 입력

192.168.0.1 (Password 임의 설정 a123456)

7.  선택
8.  선택
9. Advanced – Network – Internet – Static IP 선택
10. 고정 IP입력 후 저장
11. TP-Link 9160에 인터넷 케이블 연결. 왼쪽부터 LED 초록 4개 켜짐
12. 핸드폰에서 인터넷 되는지 확인. 완료

TP-Link 9160

: PW 91919533

: 거리가 길지만... 속도가 느리고

TP-Link 9160 5G

: PW 91919533

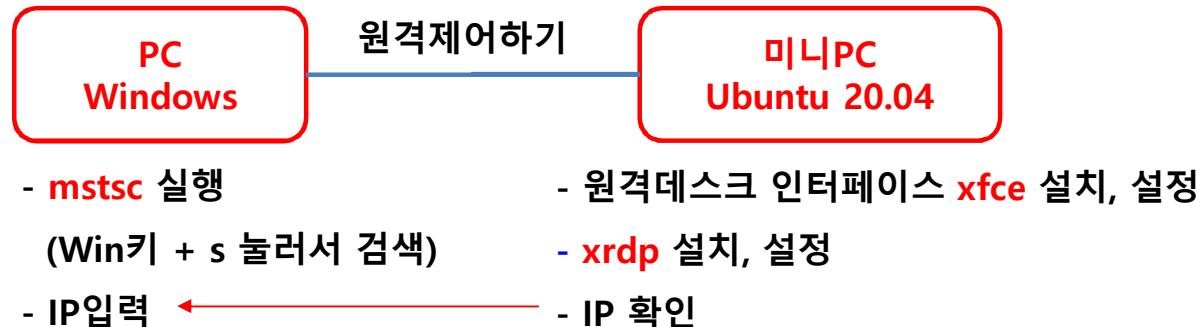
: 속도는 빠르지만... 거리가 짧고

4. ROS 개발환경 구축

Terminal Window 열기 : **Ctrl + Alt + T** 누르기

4-2-2. Linux on Remote PC

(1) Windows PC에서 ubuntu를 원격으로 제어하기



mstsc

: windows 환경의
원격데스크톱 연결 제어 프로그램

xrdp

... 원격 접속을 허용하는 프로그램으로, 윈도우 환경에서 GUI 환경으로 Linux를 사용하기 위함

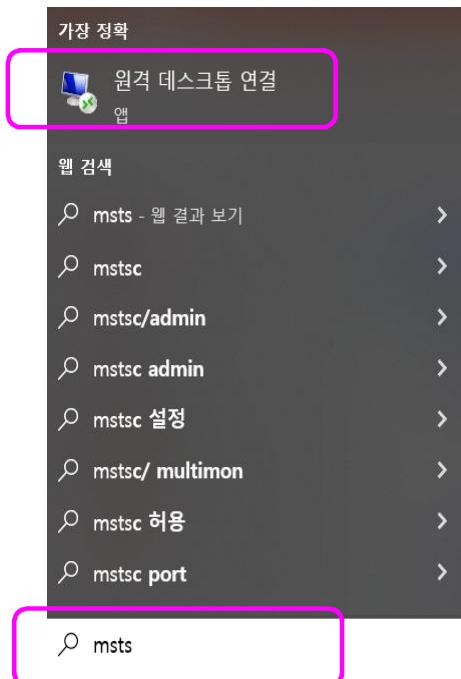
xfce

: 리눅스 플랫폼을 위한 자유 소프트웨어 데스크톱 환경의 GUI 환경

```
: sudo apt-get install xfce4
```

→ 원격데스크 인터페이스 설치

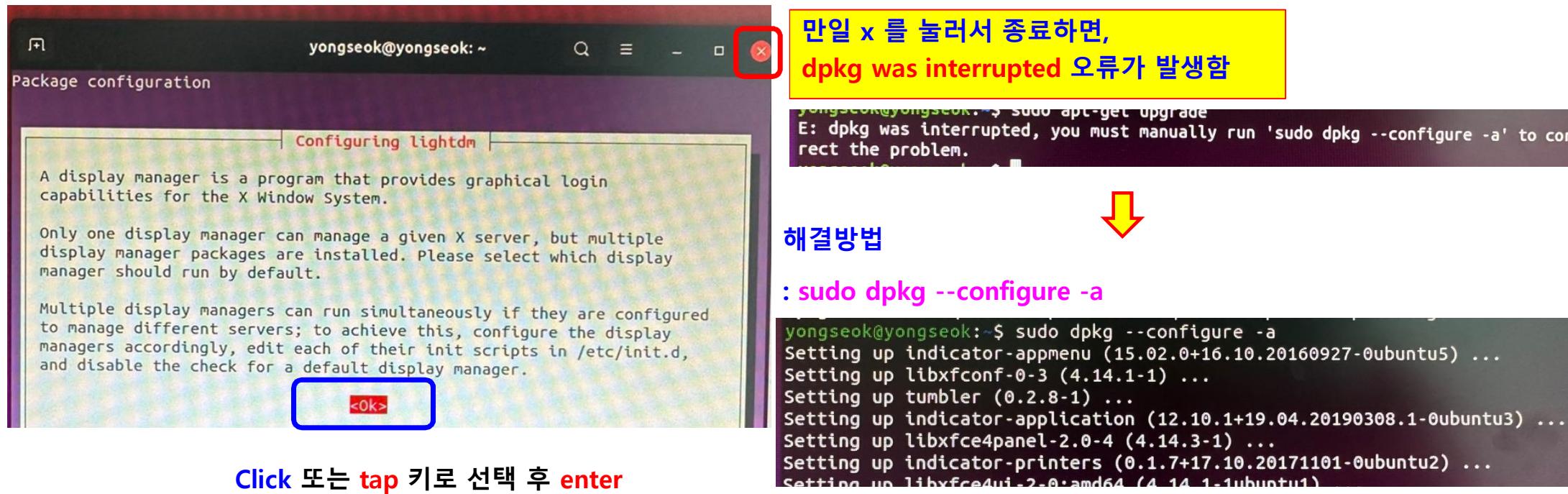
```
yongseok@yongseok:~$ sudo apt-get install xfce4
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
desktop-base exo-utils greybird-gtk-theme gtk2-engines-pixbuf
gtk2-engines-xfce libbonobo2-0 libbonobo2-common libbonoboui2-
libbonoboui2-common libexo-1.0.lite
```



4. ROS 개발환경 구축

: sudo apt-get install xfce4

→ 원격데스크 인터페이스 설치 후 화면



4. ROS 개발환경 구축

4-2-2. Linux on Remote PC

(1) Windows PC에서 ubuntu를 원격으로 제어

: `systemctl status xrdp`

→ ubuntu에서 xrdp 패키지 설치 여부 확인

: `sudo apt-get install xrdp` 또는

`sudo apt install xrdp`

→ ubuntu에서 xrdp 패키지 설치

※

RDP (Microsoft Remote Desktop Protocol)

.sh (shell script)

: 예, `startwm.sh` / `install_ros_kinetic.sh` 등 일반적으로 shell script는 확장자에 지니지 않음

: shell script는 명령어를 실행하거나 프로그램을 실행하는 구문으로 이해

(여러 명령어나 프로그램 실행을 파일 형식으로 갖고 있어, 한번에 실행해 줌)

The screenshot shows two terminal sessions. The top session shows the command `systemctl status xrdp` with output indicating it is inactive (dead) because it was not found. The bottom session shows the command `sudo apt-get install xrdp` being run, followed by a list of packages to be installed, including xorgxrdp and xrdp, along with suggested packages like guacamole and xrdp-pulseaudio-installer. It also shows the download progress from the kr.archive.ubuntu.com repository.

입력 후 enter

미설치 확인

입력 후 enter

입력

4. ROS 개발환경 구축

: sudo gedit /etc/xrdp/startwm.sh

→ xrdp 설정파일중에 startwm.sh 수정(edit) → 저장

: window pc에서 원격으로 Ubuntu pc 접속시
화면창 변경하기 위함(설치된 xfce4)

```
yongseok@yongseok:~$ sudo gedit /etc/xrdp/startwm.sh
```



Linux 프로그램 편집 tool
: vi edit , gedit

추가하기

#xrdp multiple users configuration

xfce4-session

#. /etc/X11/Xsession

. /usr/bin/startxfce4

Yellow arrow pointing right from the '수정화면' text to the code block.

: 수정화면

```
1#!/bin/sh
2# xrdp X session start script (c) 2015, 2017 mirabilis
3# published under The MirOS Licence
4
5if test -r /etc/profile; then
6    . /etc/profile
7fi
8
9if test -r /etc/default/locale; then
10    . /etc/default/locale
11    test -z "${LANG+x}" || export LANG
12    test -z "${LANGUAGE+x}" || export LANGUAGE
13    test -z "${LC_ADDRESS+x}" || export LC_ADDRESS
14    test -z "${LC_ALL+x}" || export LC_ALL
15    test -z "${LC_COLLATE+x}" || export LC_COLLATE
16    test -z "${LC_CTYPE+x}" || export LC_CTYPE
17    test -z "${LC_IDENTIFICATION+x}" || export LC_IDENTIFICATION
18    test -z "${LC_MEASUREMENT+x}" || export LC_MEASUREMENT
19    test -z "${LC_MESSAGES+x}" || export LC_MESSAGES
20    test -z "${LC_MONETARY+x}" || export LC_MONETARY
21    test -z "${LC_NAME+x}" || export LC_NAME
22    test -z "${LC_NUMERIC+x}" || export LC_NUMERIC
23    test -z "${LC_PAPER+x}" || export LC_PAPER
24    test -z "${LC_TELEPHONE+x}" || export LC_TELEPHONE
25    test -z "${LC_TIME+x}" || export LC_TIME
26    test -z "${LOCPATH+x}" || export LOCPATH
27fi
28
29if test -r /etc/profile; then
30    . /etc/profile
31fi
32
33xfce4-session
34
35./usr/bin/startfce4
```



4. ROS 개발환경 구축

: sudo systemctl enable --now xrdp

→ 부팅 후 자동 실행되도록 설정

방화벽(UFW, uncomplicated firewall) 제어

: sudo ufw allow from any to any port 3389 proto tcp

→ 방화벽 설정 추가. 3389 포트 열기

: sudo /etc/init.d/xrdp restart

→ xrdp를 재 시작

: ifconfig

(네트워크 설정 확인. 유무선)

→ 만일 ifconfig 명령어를 찾을 수 없다면,

sudo apt install net-tools

```
yongseok@yongseok: $ sudo apt install net-tools
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  net-tools
0 upgraded, 1 newly installed, 0 to remove and 12
```

```
yongseok@yongseok:~$ sudo systemctl enable --now xrdp
[sudo] password for yongseok:
xrdp.service is not a native service, redirecting to systemd-sysv-install
Executing /lib/systemd/systemd-sysv-install enable xrdp
yongseok@yongseok:~$ sudo ufw allow from any to any port 3389 proto tcp
Rules updated
Rules updated (v6)
yongseok@yongseok:~$
```

```
yongseok@yongseok:~$ ifconfig
enp2s0      Link encap:Ethernet HWaddr 4c:cc:6a:9d:47:16
            UP BROADCAST MULTICAST MTU:1500 Metric:1
            RX packets:0 errors:0 dropped:0 overruns:0 frame:0
            TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:0 (0.0 B)   TX bytes:0 (0.0 B)

lo         Link encap:Local Loopback
           inet addr:127.0.0.1 Mask:255.0.0.0
           inet6 addr: ::1/128 Scope:Host
             UP LOOPBACK RUNNING MTU:65536 Metric:1
             RX packets:3638 errors:0 dropped:0 overruns:0 frame:0
             TX packets:3638 errors:0 dropped:0 overruns:0 carrier:0
             collisions:0 txqueuelen:1000
             RX bytes:713589 (713.5 KB)   TX bytes:713589 (713.5 KB)

ip 기억
wlp3s0      Link encap:Ethernet HWaddr f4:06:69:f0:af:37
            inet addr:192.168.0.15 Bcast:192.168.0.255 Mask:255.255.255.0
            inet6 addr: fe80::5dc8:f53:79d6:78cf/64 Scope:Link
              UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
              RX packets:553870 errors:0 dropped:0 overruns:0 frame:0
              TX packets:224046 errors:0 dropped:0 overruns:0 carrier:0
              collisions:0 txqueuelen:1000
              RX bytes:810194447 (810.1 MB)   TX bytes:23074618 (23.0 MB)
```

→ Ubuntu PC 재시작

4. ROS 개발환경 구축

: netstat -antp

→ 원격데스크톱의 기본 포트 3389 확인하기

```
yongseok@yongseok:~$ netstat -antp
(Not all processes could be identified, non-owned process info
will not be shown, you would have to be root to see it all.)
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address          Foreign Address        State
PID/Program name
tcp            0      0 127.0.0.53:53           0.0.0.0:*
                                         LISTEN
-
tcp            0      0 127.0.0.1:631            0.0.0.0:*
                                         LISTEN
-
tcp6           0      0 ::1:631                ::*:*
                                         LISTEN
-
tcp6           0      0 ::::3389               ::*:*
                                         LISTEN
                                         확인하기
tcp6           0      0 ::1:3350               ::*:*
                                         LISTEN
-
yongseok@yongseok:~$
```

netstat : 현재 내 컴퓨터와 연결되었거나 연결될 목록을 프로토콜과 함께 보여주는 명령어

- a : 현재 다른 PC와 연결(ESTABLISHED)되어 있거나 대기(LISTENING)중인 모든 포트 번호를 확인
- n : 현재 다른 PC와 연결(ESTABLISHED)되어 있는 것들만 보여줌, 주소를 IP로 표시
- t : tcp 프로토콜
- u : udp 프로토콜
- p : pid와 program 명 표시
- l : 연결 가능한 상태

4. ROS 개발환경 구축

Ref. www.digitalocean.com/community/tutorials/how-to-set-up-a-firewall-with-ufw-on-ubuntu-20-04
Ref. <https://webdir.tistory.com/206>

Ubuntu 20.04 방화벽(ufw) 사용법

(1) ufw는 Ubuntu 20 설치시 기본으로 설치됨, 만일 없다면

```
sudo apt install ufw
```

(2) ufw는 기본적으로 모든 incoming connections과 outgoing connections 허용 안함, 그래서 허용해야 함

```
sudo ufw allow from any to any port 3389
```

(3) UFW 활성화/비활성화

```
sudo ufw enable
```

```
sudo ufw disable
```

(4) ufw 상태확인

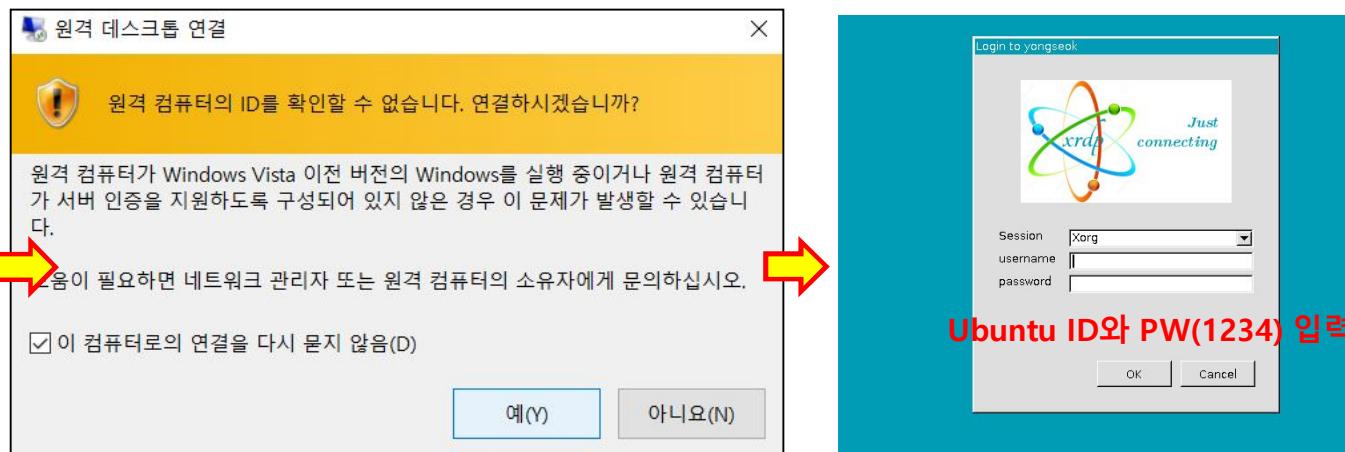
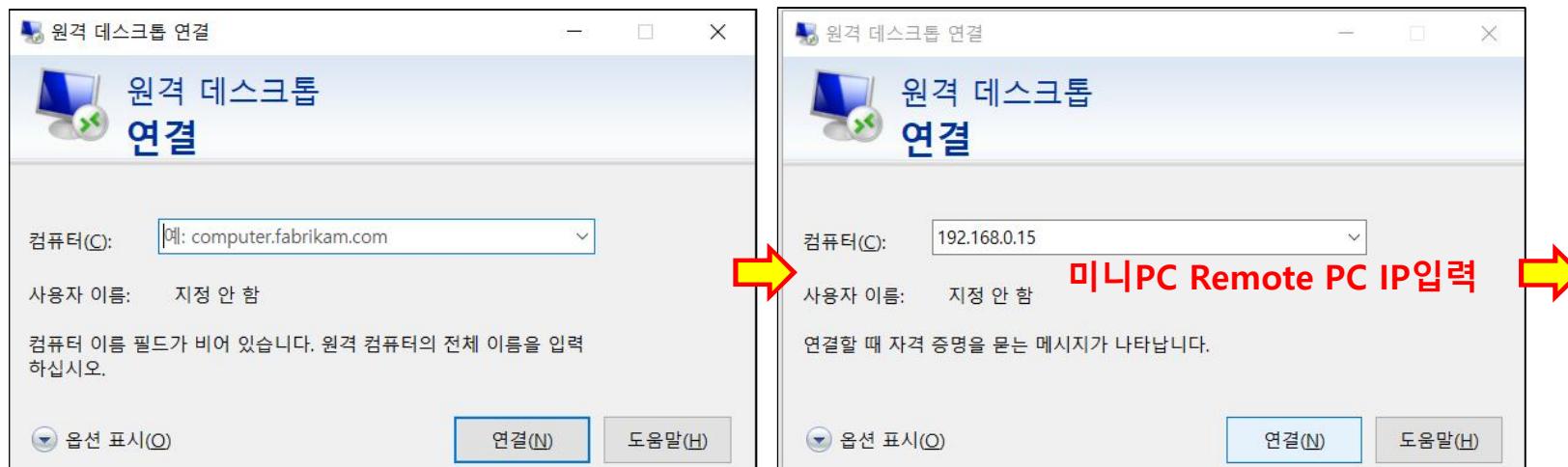
```
sudo ufw status verbose
```

```
yongseok@yongseok:~$ sudo ufw status verbose
Status: active
Logging: on (low)
Default: allow (incoming), allow (outgoing), disabled (routed)
New profiles: skip

To                         Action      From
--                         --          --
3389/tcp                   ALLOW IN   Anywhere
3389                       ALLOW IN   Anywhere
3389                       ALLOW IN   192.168.1.141
3389                       ALLOW IN   192.168.2.0/24
22/tcp                      ALLOW IN   Anywhere
22                          ALLOW IN   Anywhere
3389/tcp (v6)               ALLOW IN   Anywhere (v6)
3389 (v6)                   ALLOW IN   Anywhere (v6)
22/tcp (v6)                 ALLOW IN   Anywhere (v6)
22 (v6)                     ALLOW IN   Anywhere (v6)
```

4. ROS 개발환경 구축

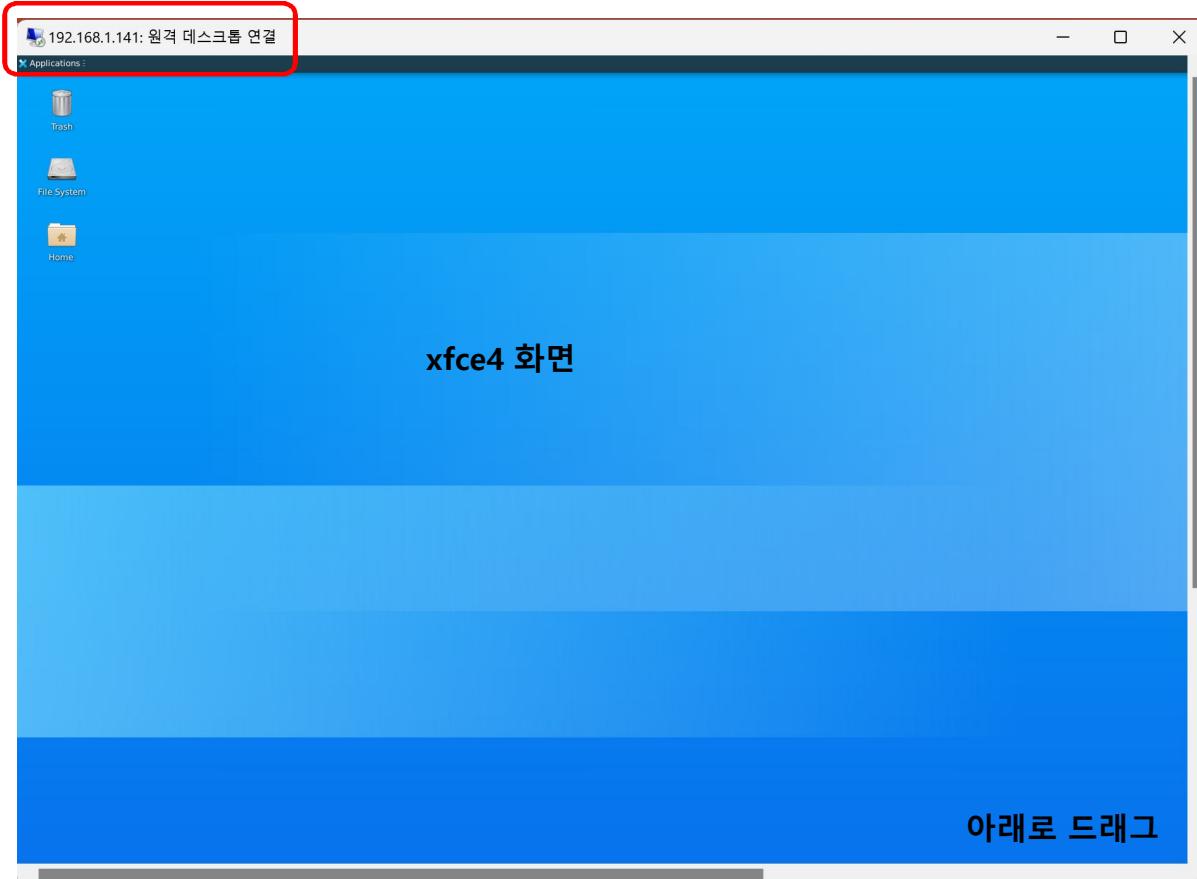
: Windows PC에서 **mstsc** 실행(Win키 + s 놀러서 검색)



4. ROS 개발환경 구축

: Windows PC에서 **mstsc** 실행(Win키 + s 놀러서 검색)

Connection 화면

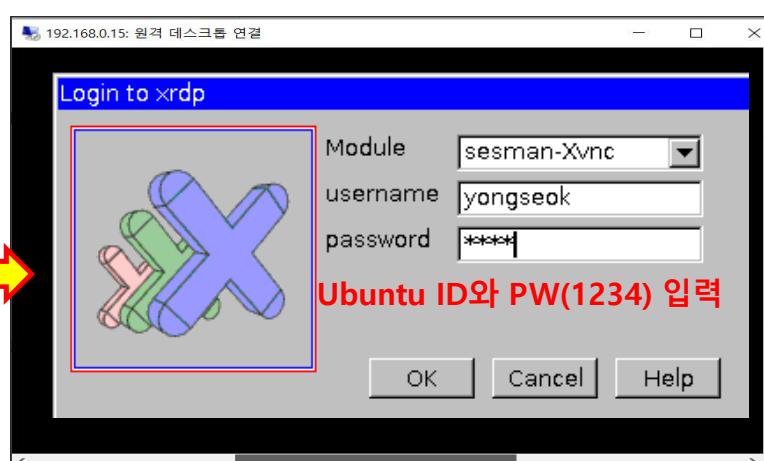
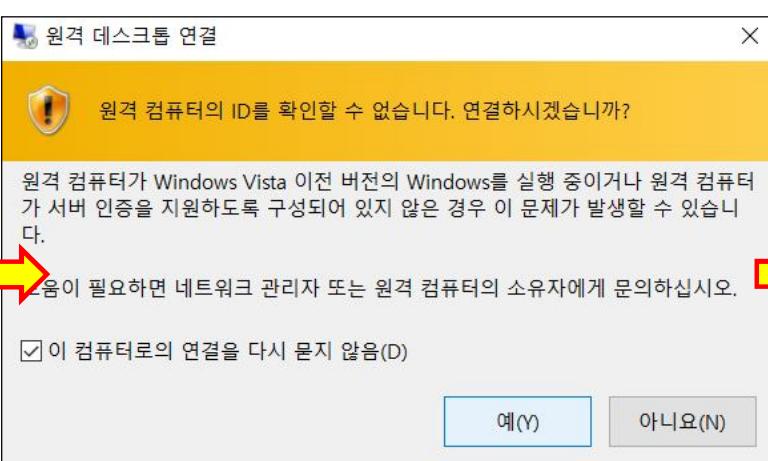
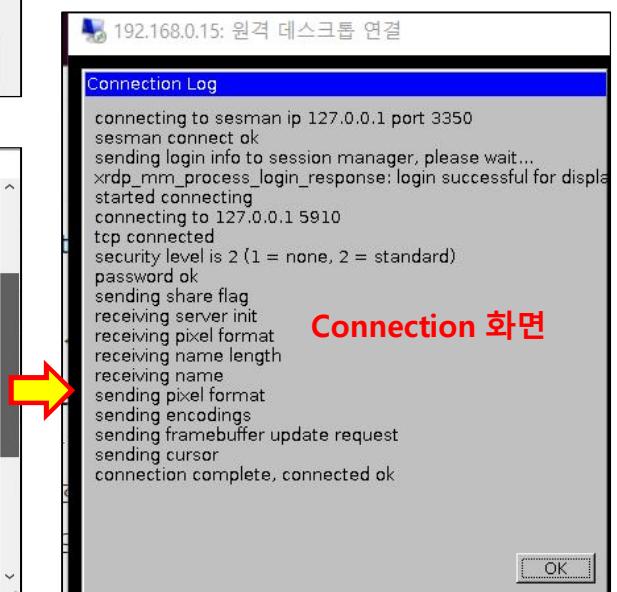
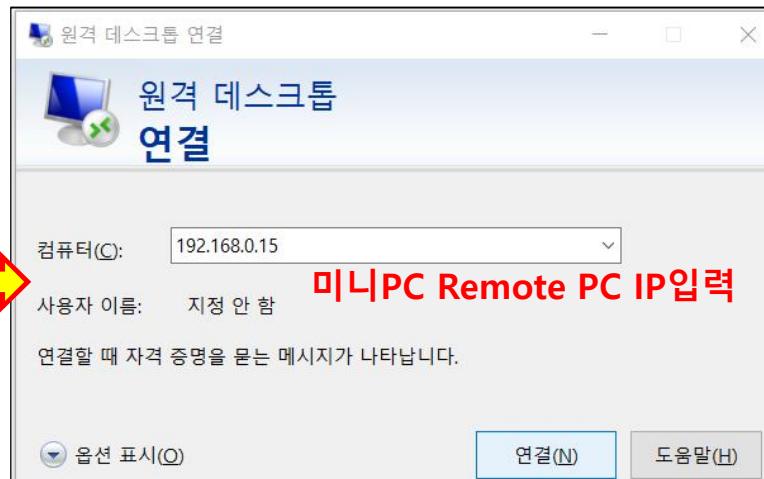
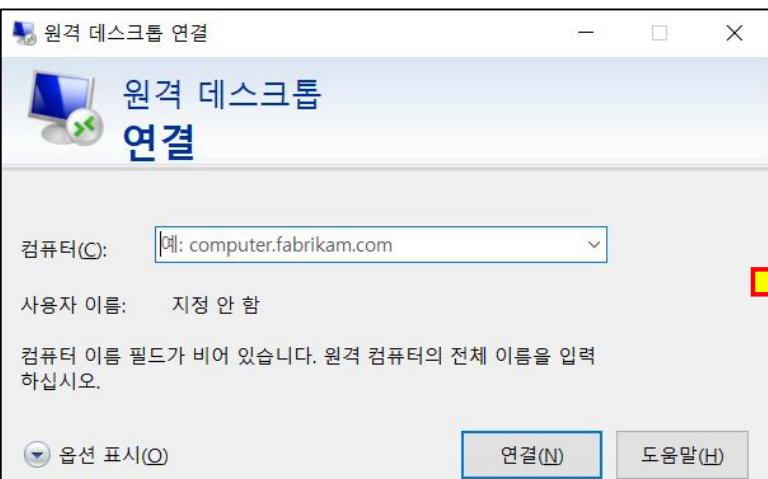


4. ROS 개발환경 구축

Ubuntu 16.04 SKIP



: Windows PC에서 **mstsc** 실행(Win키 + s 놀러서 검색)



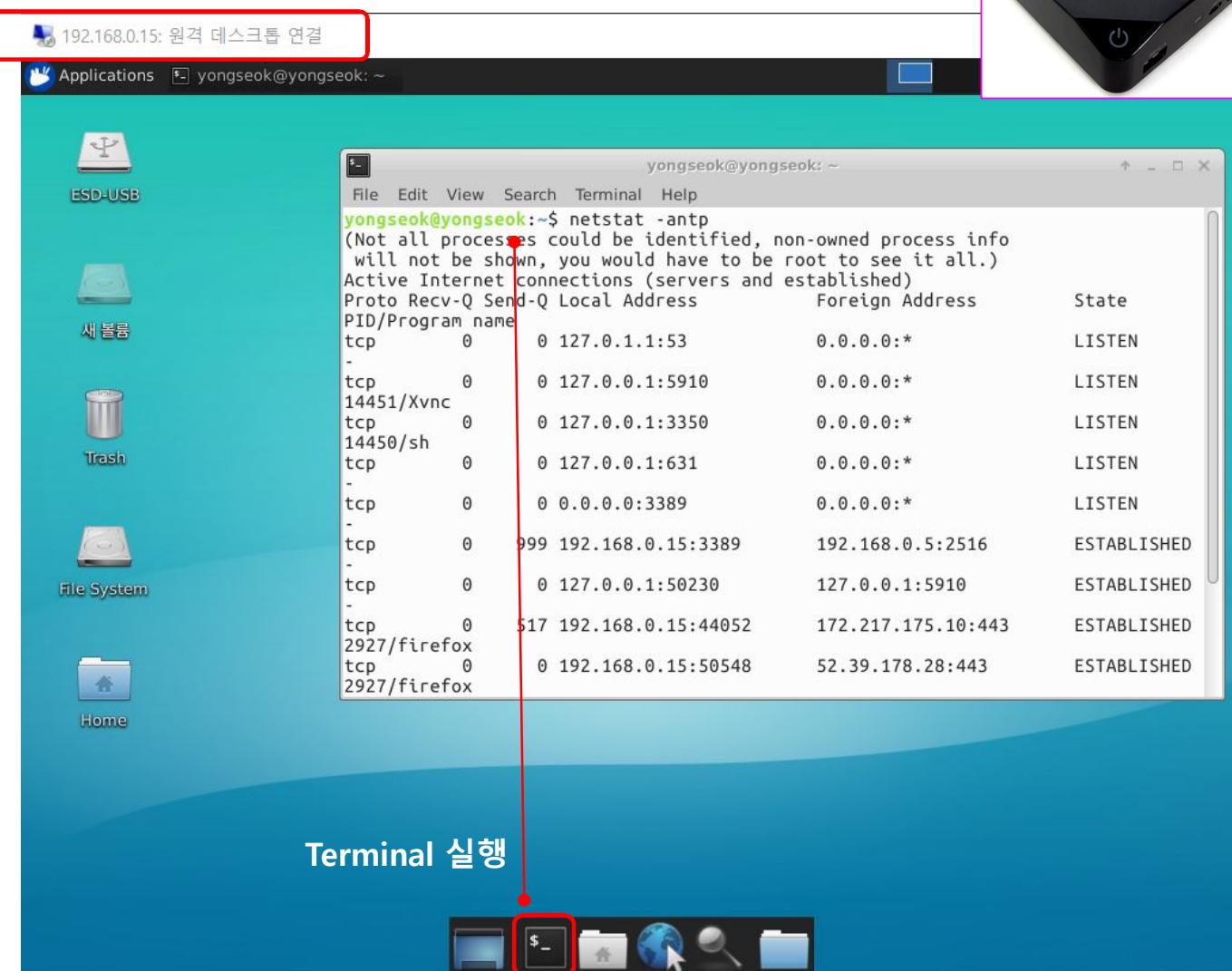
4. ROS 개발환경 구축

: Windows PC에서

ubuntu (Remote PC) 원격제어 화면

ubuntu (Remote PC) 확인

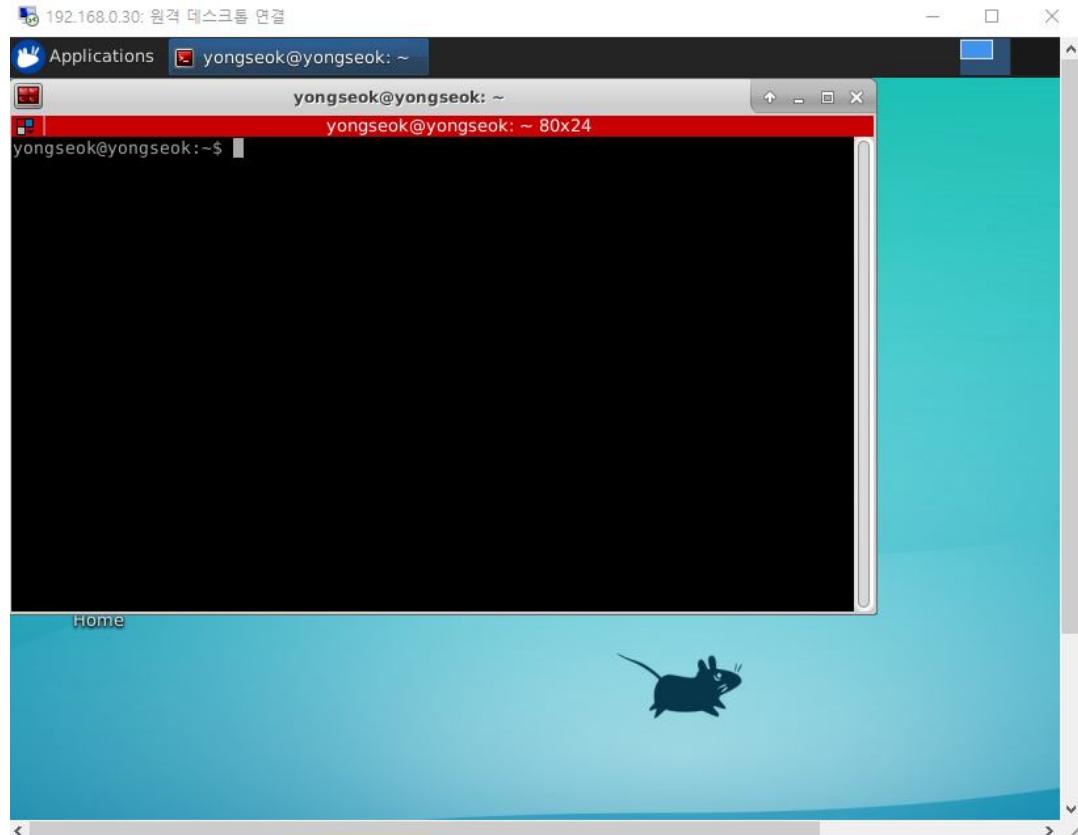
Ubuntu 16.04 SKIP



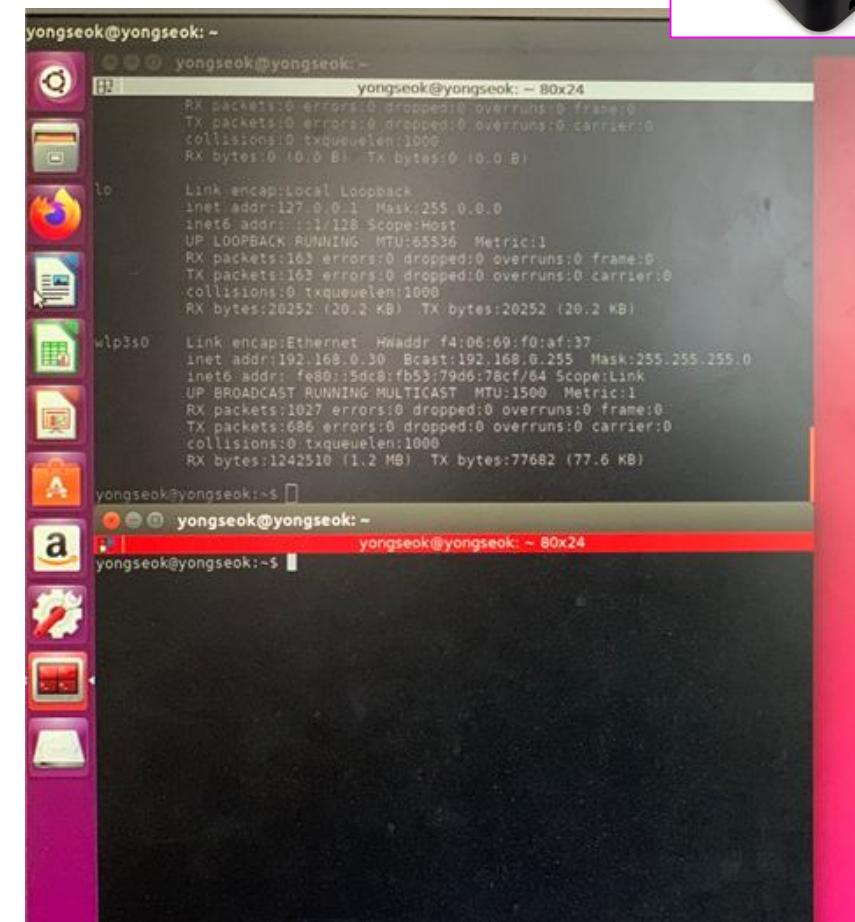
4. ROS 개발환경 구축

: Windows PC에서

ubuntu (Remote PC) 원격제어 화면



: Ubuntu 화면



Ubuntu 16.04 SKIP

4. ROS 개발환경 구축

4-2-2. Linux on Remote PC

(2) Linux 실습

: 무선 네트워크 on, off

→ **ifconfig** 네트워크 설정 확인 (유무선)

→ **iwconfig** 무선랜 설정들을 확인

{ : Windows PC에서 (Remote PC) 원격제어

한다면 명령어 앞에 **sudo** 필요

```
yongseok@yongseok:~$ sudo iwconfig  
enp2s0    no wireless extensions.
```

```
wlp3s0    IEEE 802.11 ESSID:"iptime_star"  
          Mode:Managed Frequency:2.467 GHz Access Point: 88:36:6C:89:8A:28  
          Bit Rate=150 Mb/s Tx-Power=22 dBm  
          Retry short limit:7 RTS thr:off Fragment thr:off  
          Encryption key:off  
          Power Management:on  
          Link Quality=70/70 Signal level=-36 dBm  
          Rx invalid nwid:0 Rx invalid crypt:0 Rx invalid frag:0  
          Tx excessive retries:0 Invalid misc:801 Missed beacon:0
```

```
lo      no wireless extensions.
```

```
yongseok@yongseok:~$ clear Terminal 창 갈무리
```

```
yongseok@yongseok:~$ ifconfig
```

```
Command 'ifconfig' is available in '/sbin/ifconfig'  
The command could not be located because '/sbin' is not included in the PATH environment variable.  
This is most likely caused by the lack of administrative privileges associated with your user account.
```

```
ifconfig: command not found
```

```
yongseok@yongseok:~$ sudo ifconfig
```

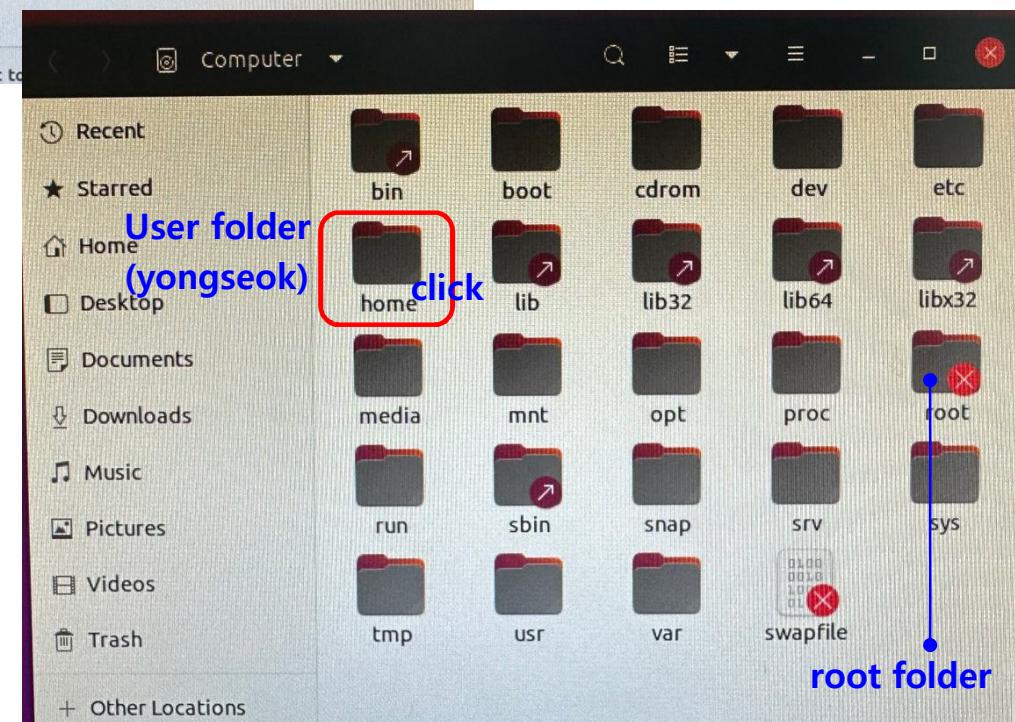
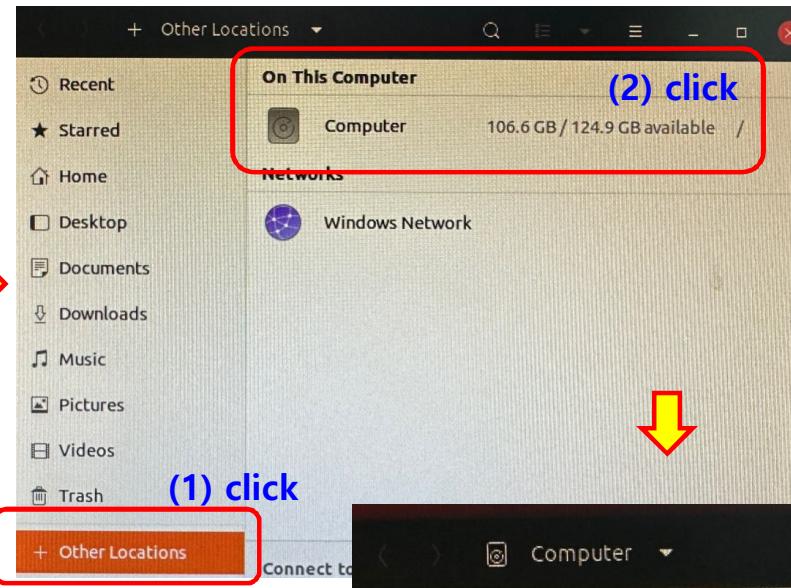
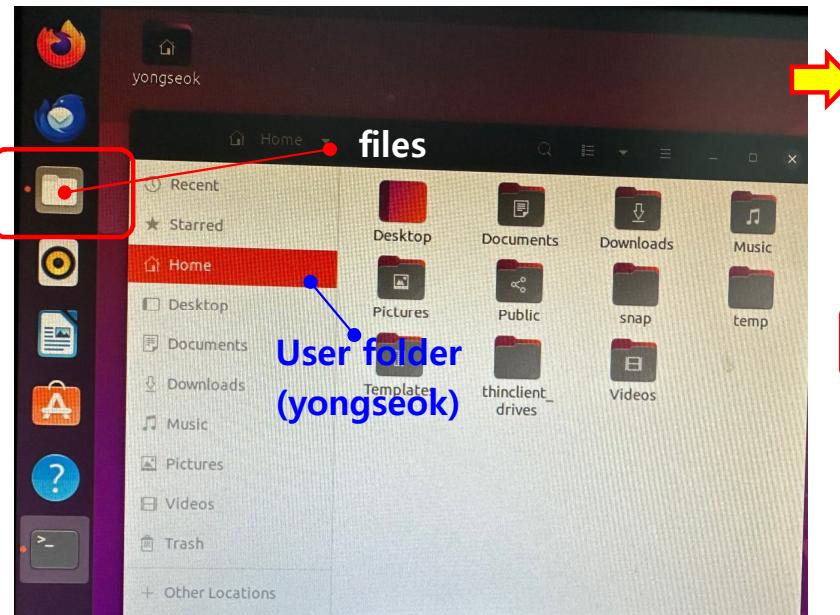
```
[sudo] password for yongseok: _____ 1234 enter  
enp2s0      Link encap:Ethernet HWaddr 4c:cc:6a:9d:47:16  
            UP BROADCAST MULTICAST MTU:1500 Metric:1  
            RX packets:0 errors:0 dropped:0 overruns:0 frame:0  
            TX packets:0 errors:0 dropped:0 overruns:0 carrier:0  
            collisions:0 txqueuelen:1000  
                  .0 B) TX bytes:0 (0.0 B)
```

```
cal Loopback  
.0.0.1 Mask:255.0.0.0  
.1/128 Scope:Host  
UNNING MTU:65536 Metric:1  
740 errors:0 dropped:0 overruns:0 frame:0  
740 errors:0 dropped:0 overruns:0 carrier:0  
txqueuelen:1000  
6715 (74.7 MB) TX bytes:74756715 (74.7 MB)
```

```
hernet HWaddr f4:06:69:f0:af:37  
.168.0.15 Bcast:192.168.0.255 Mask:255.255.255.0  
e80::5dc8:fb53:79d6:78cf/64 Scope:Link  
RUNNING MULTICAST MTU:1500 Metric:1  
RX packets:67312 errors:0 dropped:0 overruns:0 frame:0
```

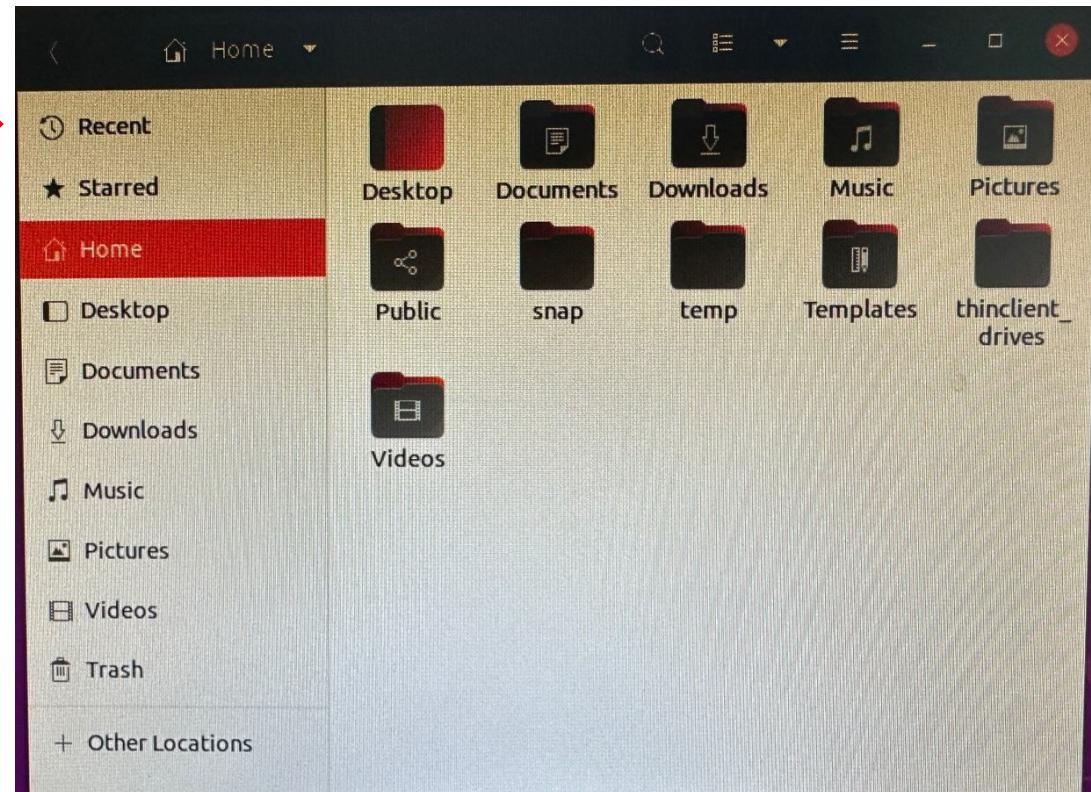
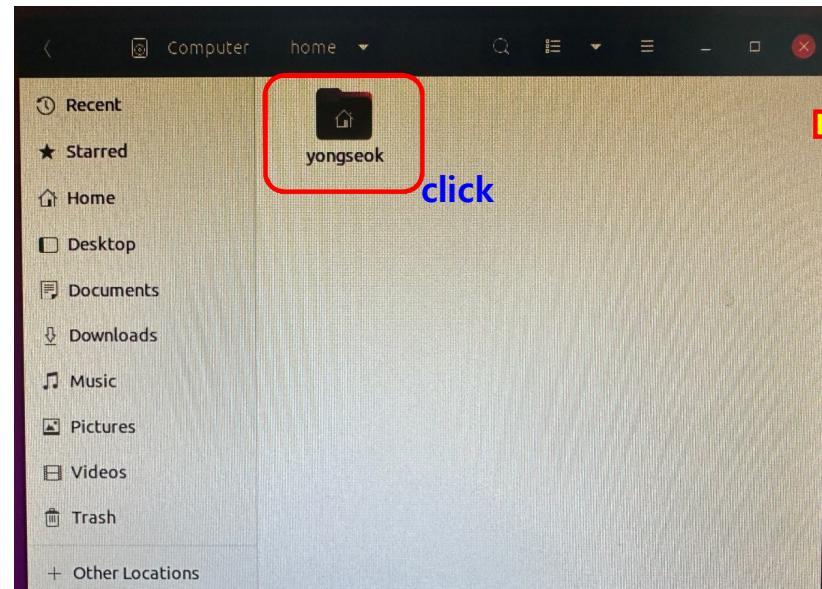
4. ROS 개발환경 구축

: files 경로 확인



4. ROS 개발환경 구축

: files 경로 확인



4. ROS 개발환경 구축

: cd ..

[현재 나의 위치 확인하기]

: cd ~

: ls

The screenshot shows a terminal window with the following session:

```
yongseok@yongseok: ~
yongseok@yongseok: $ ls
Desktop Downloads Pictures snap Templates Videos
Documents Music Public temp thinclient_drives

yongseok@yongseok: ~$ cd ..
yongseok@yongseok: /home$ ls
yongseok
yongseok@yongseok: /home$ cd ..
yongseok@yongseok: /$ ls
bin dev lib lib32 mnt root snap sys var 내 부모 폴더가 속해 있는 폴더 확인
boot etc lib32 lost+found opt run srv tmp
cdrom home lib64 media proc sbin swapfile usr

yongseok@yongseok: /$ cd ~
yongseok@yongseok: ~$ ls
Desktop Downloads Pictures snap Templates Videos
Documents Music Public temp thinclient_drives
```

Annotations in red Korean text are overlaid on the terminal output:

- 내가 갖고 있는 폴더 확인 (highlighted around 'thinclient_drives')
- 부모(?) 위로로 올라가기 (highlighted around 'cd ..' and 'thinclient_drives')
- 내가 속해 있는 폴더 확인 (highlighted around 'yongseok')
- 부모(?) 위로로 올라가기 (highlighted around 'cd ..' and 'yongseok')
- 내 부모 폴더가 속해 있는 폴더 확인 (highlighted around 'var' and the list of system directories)
- 내 고향(원래 위치)로 가기 (highlighted around 'cd ~' and 'thinclient_drives')

4. ROS 개발환경 구축

: uname

```
yongseok@yongseok:~$ uname • 현재 사용하는 운영체제
```

: date

```
yongseok@yongseok:~$ date  
Wed Sep 15 16:22:01 KST 2021
```

: who

```
yongseok@yongseok:~$ who • 현재 로그인 사용자  
yongseok tty7 2021-09-14 11:08 (:0)
```

: ls

```
yongseok@yongseok:~$ ls • 파일 목록  
catkin_ws Documents examples.desktop Music Public Videos  
Desktop Downloads install_ros_kinetic.sh Pictures Templates
```

: hostname

```
yongseok@yongseok:~$ hostname  
yongseok  
yongseok@yongseok:~$ █
```

: pwd

```
yongseok@yongseok:~$ pwd • 현재 작업 디렉토리를 표시  
/home/yongseok
```

: cd

```
yongseok@yongseok:~$ cd • Home 디렉토리로 이동
```

: ls -s

```
yongseok@yongseok:~$ ls -s • 현재 디렉토리, size 표시  
total 56  
4 catkin_ws 4 Downloads 4 Music 4 Templates  
4 Desktop 12 examples.desktop 4 Pictures 4 test  
4 Documents 4 install_ros_kinetic.sh 4 Public 4 Videos
```

: ls -a

```
yongseok@yongseok:~$ ls -a • 현재 디렉토리, all 표시  
. Documents Public  
.. Downloads .ros  
.bash_history examples.desktop .sudo_as_admin_successful  
.bash_logout .gconf Templates  
.bashrc .gnupg test  
.cache .ICEAuthority Videos  
catkin_ws install_ros_kinetic.sh .vnc  
.compiz .local .wget-hsts  
.config .mozilla .Xauthority  
.dbus Music .xsession-errors  
Desktop Pictures .xsession-errors.old  
.dmrc .profile  
yongseok@yongseok:~$ █
```

4. ROS 개발환경 구축

: mkdir 폴더명

```
yongseok@yongseok:~$ cd ~ • Home 디렉토리로 이동(내 고향)
yongseok@yongseok:~$ ls
```

: cd ~

```
catkin_ws  Documents  examples.desktop  Music  Public  test
Desktop   Downloads  install_ros_kinetic.sh  Pictures  Templates  Videos
```

: cd ..

```
yongseok@yongseok:~$ mkdir test_linux • 새로운 디렉토리 만들기
```

: cd 폴더명

```
catkin_ws  Downloads      Music  Templates  Videos
Desktop    examples.desktop  Pictures  test
Documents  install_ros_kinetic.sh  Public  test_linux 생성
```

```
yongseok@yongseok:~$ cd test_linux • 폴더명 디렉토리로 이동하기
```

```
yongseok@yongseok:~/test_linux$ ls
```

```
yongseok@yongseok:~/test_linux$ mkdir test_linux2 • 새로운 디렉토리 만들기
```

```
yongseok@yongseok:~/test_linux$ ls
```

```
test_linux2 생성
```

```
yongseok@yongseok:~/test_linux$ cd ~ • home디렉토리로 이동
```

```
yongseok@yongseok:~/test_linux$ cd test_linux • 폴더명 디렉토리로 이동하기
```

```
yongseok@yongseok:~/test_linux$ cd .. • 부모 디렉토리로 이동
```

```
yongseok@yongseok:~$ █
```

명령어	의미
ls	파일 및 디렉터리 리스트
ls -a	모든 파일과 디렉터리 리스트
ls -asl	모든 파일 자세히 리스트
mkdir	디렉터리 만들기
cd 디렉터리	디렉터리로 이동
cd	홈 디렉터리로 이동
cd ~	홈 디렉터리로 이동
cd ..	부모 디렉터리로 이동
pwd	현재 작업 디렉터리 프린트

4. ROS 개발환경 구축

: ls -asl

```
yongseok@yongseok:~$ ls -asl
total 156
4 drwxr-xr-x 23 yongseok yongseok 4096 Sep 15 16:56 .
4 drwxr-xr-x  3 root    root    4096 Sep 13 23:21 ..
4 -rw-----  1 yongseok yongseok 3494 Sep 14 11:06 .bash_history
4 -rw-r--r--  1 yongseok yongseok 220 Sep 13 23:21 .bash_logout
8 -rw-r--r--  1 yongseok yongseok 4858 Sep 14 01:18 .bashrc
4 drwx----- 16 yongseok yongseok 4096 Sep 14 14:00 .cache
4 drwxrwxr-x  5 yongseok yongseok 4096 Sep 14 01:18 catkin_ws
4 drwx-----  3 yongseok yongseok 4096 Sep 14 11:06 .compiz
4 drwx----- 17 yongseok yongseok 4096 Sep 14 14:00 .config
4 drwx-----  3 yongseok yongseok 4096 Sep 14 11:15 .dbus
4 drwxr-xr-x  2 yongseok yongseok 4096 Sep 13 23:29 Desktop
4 -rw-r--r--  1 yongseok yongseok 25 Sep 13 23:29 .dmrc
4 drwxr-xr-x  2 yongseok yongseok 4096 Sep 13 23:29 Documents
4 drwxr-xr-x  2 yongseok yongseok 4096 Sep 13 23:29 Downloads
12 -rw-r--r--  1 yongseok yongseok 8980 Sep 13 23:21 examples.desktop
4 drwx-----  2 yongseok yongseok 4096 Sep 14 00:54 .gconf
4 drwx-----  3 yongseok yongseok 4096 Sep 14 11:08 .gnupg
4 -rw-----  1 yongseok yongseok 982 Sep 14 14:00 .ICEauthority
4 -rwxr-xr-x  1 yongseok yongseok 2981 Sep 13 23:50 install_ros_kinetic.sh
4 drwx-----  3 yongseok yongseok 4096 Sep 13 23:29 .local
4 drwx-----  5 yongseok yongseok 4096 Sep 13 23:44 .mozilla
4 drwxr-xr-x  2 yongseok yongseok 4096 Sep 13 23:29 Music
4 drwxr-xr-x  2 yongseok yongseok 4096 Sep 13 23:29 Pictures
4 -rw-r--r--  1 yongseok yongseok 655 Sep 13 23:21 .profile
4 drwxr-xr-x  2 yongseok yongseok 4096 Sep 13 23:29 Public
4 drwxrwxr-x  3 yongseok yongseok 4096 Sep 14 00:10 .ros
0 -rw-r--r--  1 yongseok yongseok    0 Sep 13 23:46 .sudo_as_admin_successfu
```

- 모든 디렉토리 자세히 표시

ls -l : 내부에 존재하는 파일과 디렉토리에 대한 상세 정보 표시

ls -al : a옵션은 숨겨진 파일과 디렉토리 정보를 보여줌 + l옵션

4. ROS 개발환경 구축

: cat > sample1.txt

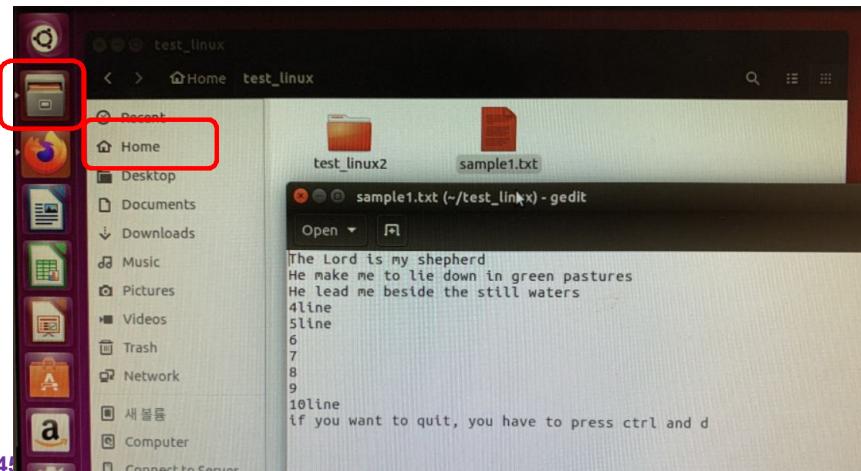
→ cat : concatenate. 여러 개의 텍스트 문자열을 하나의 텍스트 문자열로 연결하는 함수

```
yongseok@yongseok:~$ cd test_linux
yongseok@yongseok:~/test_linux$ cat > sample1.txt
The Lord is my shepherd
He make me to lie down in green pastures
He lead me beside the still waters
4line
5line
6
7
8
9
10line
if you want to quit, you have to press ctrl and d
yongseok@yongseok:~/test_linux$ ls
sample1.txt test_linux2
yongseok@yongseok:~/test_linux$
```

끝내려면, **ctrl + d** 를 누르기

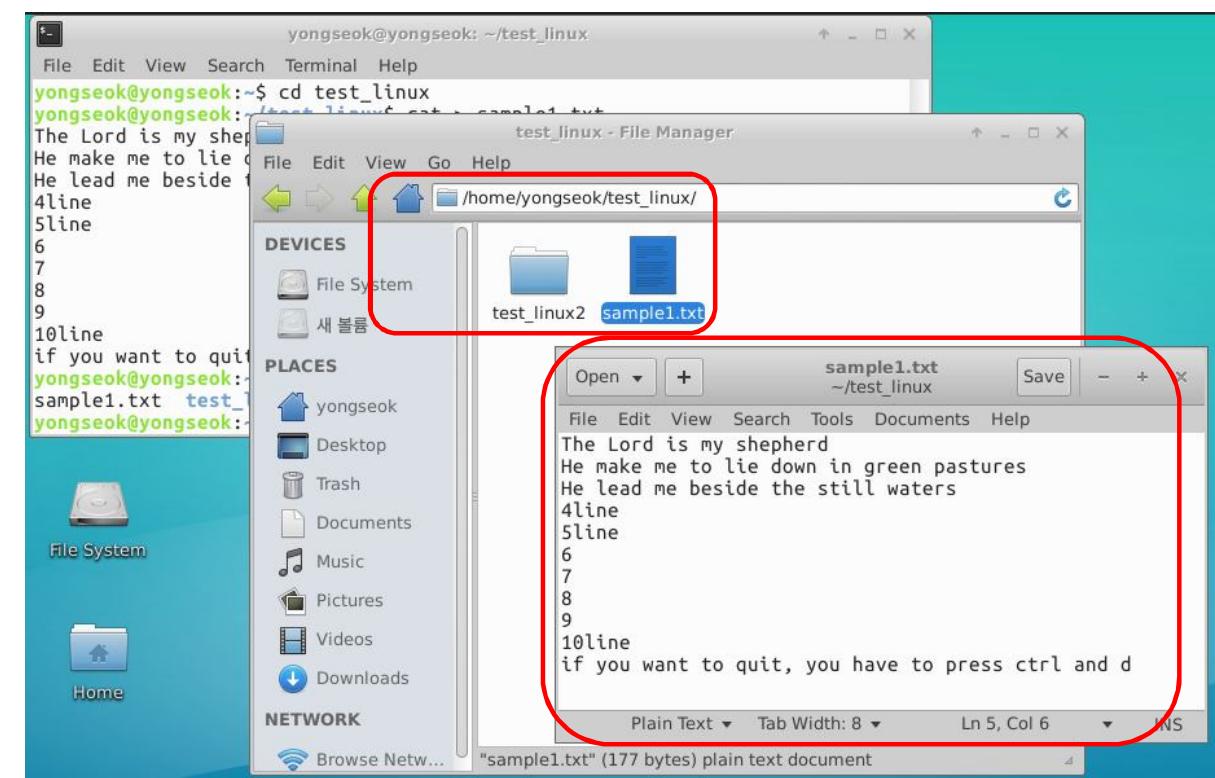


ctrl and d



sample1.txt 파일을 만들어 그 내용을 아래 작성한 내용으로 만듦

• test_linux 폴더로 이동



→ 원격제어 화면

4. ROS 개발환경 구축

: cat sample1.txt

→ terminal 창에서 내용 보기

```
yongseok@yongseok: ~/test_linux
File Edit View Search Terminal Help
yongseok@yongseok:~/test_linux$ cat sample1.txt
The Lord is my shepherd
He make me to lie down in green pastures
He lead me beside the still waters
4line
5line
6
7
8
9
10line
if you want to quit, you have to press ctrl and d
yongseok@yongseok:~/test_linux$
```

내용 보기

: terminal 창을 줄인 후에

: more sample1.txt → terminal 창이 작을 때,

more 명령어를 입력 후 enter마다 내용을 이어서 볼 수 있음

```
yongseok@yongseok: ~/test_linux
File Edit View Search Terminal Help
yongseok@yongseok:~/test_linux$
```

```
yongseok@yongseok: ~/test_linux
File Edit View Search Terminal Help
yongseok@yongseok:~/test_linux$ more sample1.txt
```

```
yongseok@yongseok: ~/tes
File Edit View Search Terminal Help
The Lord is my shepherd
He make me to lie down in green
He lead me beside the still water
4line
5line
6
--More--(64%)
```

enter

enter

```
yongseok@yongseok: ~/test_linux
File Edit View Search Terminal Help
6
7
8
9
10line
if you want to quit, you have to press ctrl and d
yongseok@yongseok:~/test_linux$
```

```
yongseok@yongseok: ~/test_linux
File Edit View Search Terminal Help
He make me to lie down in green pastures
He lead me beside the still waters
4line
5line
6
7
--More--(65%)
```

enter

4. ROS 개발환경 구축

: head sample1.txt

→ 파일의 앞부분 10줄을 출력

• 모든 디렉토리 자세히 표시

: tail sample1.txt

→ 파일의 뒷부분 10줄을 출력

```
yongseok@yongseok: ~/test_linux
File Edit View Search Terminal Help
yongseok@yongseok:~/test_linux$ head sample1.txt
The Lord is my shepherd      1 line
He make me to lie down in green 2 pastures
He lead me beside the still waters 3
4line                         4
5line                         5
6                             6
7                             7
8                             8
9                             9
10line                        10
yongseok@yongseok:~/test_linux$
```

```
yongseok@yongseok: ~/test_linux
File Edit View Search Terminal Help
yongseok@yongseok:~/test_linux$ tail sample1.txt
He make me to lie down in green pastures      1 line
He lead me beside the still waters            2
4line                           3
5line                           4
6                               5
7                               6
8                               7
9                               8
10line                          9
if you want to quit, you have to press ctrl and d 10
yongseok@yongseok:~/test_linux$
```

4. ROS 개발환경 구축

: wc sample1.txt

→ wc : word count 파일 내용의 저장된 줄 수, 단어 수, 문자 수를 출력

```
yongseok@yongseok: ~/test_linux
File Edit View Search Terminal Help
yongseok@yongseok:~/test_linux$ wc sample1.txt
11 40 177 sample1.txt
yongseok@yongseok:~/test_linux$
```

줄 수
단어 수
문자 수
파일 이름

입력

: sudo rm 파일이름

→ 파일을 삭제할 경우

```
yongseok@yongseok:~$ ls
catkin_ws examples.desktop Public
Desktop install_ros_kinetic.sh Templates
Documents Music test
Downloads Pictures test_linux
yongseok@yongseok:~$ cd test_linux
yongseok@yongseok:~/test_linux$ cat > sample2.txt
yongseok@yongseok:~/test_linux$ ls
sample1.txt sample2.txt test_linux2
yongseok@yongseok:~/test_linux$ sudo rm sample2.txt
yongseok@yongseok:~/test_linux$ ls
sample1.txt test_linux2
yongseok@yongseok:~/test_linux$
```

Sample2.txt의 내용을 yongseok 입력 후 ctl + d로 입력을 마침

생성

입력

Sample2.txt 삭제되어 안보임

4. ROS 개발환경 구축

: cat > list1.txt concatenate.

여러 개의 텍스트 문자열을 하나의 텍스트 문자열로 연결하는 함수

cat > list2.txt

```
yongseok@yongseok: ~/test_linux
File Edit View Search Terminal Help
yongseok@yongseok:~/test_linux$ clear
yongseok@yongseok:~/test_linux$ ls
sample1.txt test_linux2
yongseok@yongseok:~/test_linux$ cat > list1.txt
hi my name is yongseok
if you want to quit, press the ctrl + d
yongseok@yongseok:~/test_linux$ ls
list1.txt sample1.txt test_linux2
yongseok@yongseok:~/test_linux$ cat > list2.txt
the eagle does not catch filies
press thr ctrl + d
yongseok@yongseok:~/test_linux$ ls
list1.txt list2.txt sample1.txt test_linux2
yongseok@yongseok:~/test_linux$ ls -sl
total 16
4 -rw-rw-r-- 1 yongseok yongseok 64 Sep 16 11:59 list1.txt
4 -rw-rw-r-- 1 yongseok yongseok 51 Sep 16 12:00 list2.txt
4 -rw-rw-r-- 1 yongseok yongseok 177 Sep 15 17:16 sample1.txt
4 drwxrwxr-x 2 yongseok yongseok 4096 Sep 15 16:56 test_linux2
yongseok@yongseok:~/test_linux$
```

파일 크기

: cat > list1.txt list2.txt > list3.txt

두개의 파일을 하나로 만들어 새로운 파일 만들기

```
yongseok@yongseok: ~/test_linux
File Edit View Search Terminal Help
yongseok@yongseok:~/test_linux$ cat list1.txt list2.txt > list3.txt
yongseok@yongseok:~/test_linux$ ls
list1.txt list2.txt list3.txt sample1.txt test_linux2
yongseok@yongseok:~/test_linux$ ls -sl
total 20
4 -rw-rw-r-- 1 yongseok yongseok 39 Sep 16 12:12 list1.txt
4 -rw-rw-r-- 1 yongseok yongseok 51 Sep 16 12:00 list2.txt
4 -rw-rw-r-- 1 yongseok yongseok 90 Sep 16 12:12 list3.txt
4 -rw-rw-r-- 1 yongseok yongseok 177 Sep 15 17:16 sample1.txt
4 drwxrwxr-x 2 yongseok yongseok 4096 Sep 15 16:56 test_linux2
yongseok@yongseok:~/test_linux$ cat list3.txt
my name is yongseok
press the ctrl + d
the eagle does not catch filies
press thr ctrl + d
yongseok@yongseok:~/test_linux$ cat list1.txt
my name is yongseok
press the ctrl + d
yongseok@yongseok:~/test_linux$ cat list2.txt
the eagle does not catch filies
press thr ctrl + d
yongseok@yongseok:~/test_linux$
```

4. ROS 개발환경 구축

: cat >> list1.txt

→ list1.txt에 내용 추가 하기

```
yongseok@yongseok:~/test_linux$ cat list1.txt  
my name is yongseok  
press the ctrl + d  
yongseok@yongseok:~/test_linux$ cat >> list1.txt  
add the contents  
press thr ctrl and d  
yongseok@yongseok:~/test_linux$ cat list1.txt  
my name is yongseok  
press the ctrl + d  
add the contents  
press thr ctrl and d  
yongseok@yongseok:~/test_linux$
```

내용 추가됨

: cp sample1.txt sample2.txt

→ sample1.txt의 내용을 sample2.txt 만들어 복사하기

```
yongseok@yongseok:~/test_linux$ ls  
list1.txt list2.txt list3.txt sample1.txt test_linux2  
yongseok@yongseok:~/test_linux$ cat sample1.txt  
The Lord is my shepherd  
He make me to lie down in green pastures  
He lead me beside the still waters  
4line  
5line  
6  
7  
8  
9  
10line  
if you want to quit, you have to press ctrl and d  
yongseok@yongseok:~/test_linux$ cp sample1.txt sample2.txt 입력  
yongseok@yongseok:~/test_linux$ ls  
list1.txt list2.txt list3.txt sample1.txt sample2.txt test_linux2  
yongseok@yongseok:~/test_linux$ cat sample2.txt  
The Lord is my shepherd  
He make me to lie down in green pastures  
He lead me beside the still waters  
4line  
5line  
6  
7  
8  
9  
10line  
if you want to quit, you have to press ctrl and d  
yongseok@yongseok:~/test_linux$
```

Sample1.txt

내용 확인

입력

생성 확인

Sample2.txt

내용 확인

4. ROS 개발환경 구축

: mkdir 폴더 이름

→ 폴더 만들기

: rmdir 폴더 이름

→ 폴더 삭제하기

: sudo rm 파일이름

→ 파일을 삭제할 경우

: mv sample2.txt sample3.txt

→ sample2 이름의 파일을 sample3로 변경

```
yongseok@yongseok:~/test_linux$ ls
list1.txt list2.txt list3.txt sample1.txt sample2.txt test_linux2
yongseok@yongseok:~/test_linux$ mkdir making1
yongseok@yongseok:~/test_linux$ ls
list1.txt list3.txt sample1.txt test_linux2
list2.txt making1 sample2.txt
yongseok@yongseok:~/test_linux$ rmdir making1
yongseok@yongseok:~/test_linux$ ls
list1.txt list2.txt list3.txt sample1.txt sample2.txt test_linux2
yongseok@yongseok:~/test_linux$ sudo rm list3.txt
[sudo] password for yongseok:
yongseok@yongseok:~/test_linux$ ls
list1.txt list2.txt sample1.txt sample2.txt test_linux2
yongseok@yongseok:~/test_linux$ mv sample2.txt sample3.txt
yongseok@yongseok:~/test_linux$ ls
list1.txt list2.txt sample1.txt sample3.txt test_linux2
yongseok@yongseok:~/test_linux$ █ sample3.txt test_linux2
```

4. ROS 개발환경 구축

사용권한(permission mode) : 읽기(r), 쓰기(w), 실행(x) 권한

권한	파일	디렉터리
r	파일에 대한 읽기 권한	디렉터리 내에 있는 파일명을 읽을 수 있는 권한
w	파일에 대한 쓰기 권한	디렉터리 내에 파일을 생성하거나 삭제할 수 있는 권한
x	파일에 대한 실행 권한	디렉터리 내로 탐색을 위해 이동할 수 있는 권한

사용 권한은

소유자(owner)/그룹(group)/기타(others)로 구분하여 관리.

d	r	w	x	r	-	x	r	-	x
A: 파일유형	B: 소유자 권한			C: 그룹 소유자 권한			D: 일반 사용자 권한		

d 디렉토리

ls -l : 내부에 존재하는 파일과 디렉토리에 대한 상세 정보 표시

ls -al : a옵션은 숨겨진 파일과 디렉토리 정보를 보여줌 + l옵션

```
yongseok@yongseok:~/test_linux$ ls -al
total 32
drwxrwxr-x  3 yongseok yongseok 4096 Sep 16 14:57 .
drwxr-xr-x 23 yongseok yongseok 4096 Sep 15 16:56 ..
-rw-rw-r--  1 yongseok yongseok    8 Sep 16 12:21 list1.txt
-rw-rw-r--  1 yongseok yongseok   51 Sep 16 12:00 list2.txt
-rw-rw-r--  1 yongseok yongseok   90 Sep 16 12:12 list3.txt
-rw-rw-r--  1 yongseok yongseok  177 Sep 15 17:16 sample1.txt
-rw-rw-r--  1 yongseok yongseok  177 Sep 16 14:57 sample2.txt
drwxrwxr-x  2 yongseok yongseok 4096 Sep 15 16:56 test_linux2
yongseok@yongseok:~/test_linux$
```

owner	group	others	
rw-	rw-	r--	list1.txt
rw-	rw-	r--	list2.txt
rw-	rw-	r--	list3.txt
rw-	rw-	r--	sample1.txt
rw-	rw-	r--	sample2.txt
6	6	4	

4. ROS 개발환경 구축

ls -l : 내부에 존재하는 파일과 디렉토리에 대한 상세 정보 표시

ls -al : a옵션은 숨겨진 파일과 디렉토리 정보를 보여줌 + l옵션

```
yongseok@yongseok:~/test_linux$ ls -sl
total 20
4 -rw-rw-r-- 1 yongseok yongseok 78 Sep 16 12:21 list1.txt
4 -rw-rw-r-- 1 yongseok yongseok 51 Sep 16 12:00 list2.txt
4 -rw-rw-r-- 1 yongseok yongseok 177 Sep 15 17:16 sample1.txt
4 -rw-rw-r-- 1 yongseok yongseok 177 Sep 16 14:57 sample3.txt
4 drwxrwxr-x 2 yongseok yongseok 4096 Sep 15 16:56 test_linux2
yongseok@yongseok:~/test_linux$ ls -al
total 28
drwxrwxr-x 3 yongseok yongseok 4096 Sep 16 16:39 .
drwxr-xr-x 23 yongseok yongseok 4096 Sep 15 16:56 ..
-rw-rw-r-- 1 yongseok yongseok 78 Sep 16 12:21 list1.txt
-rw-rw-r-- 1 yongseok yongseok 51 Sep 16 12:00 list2.txt
-rw-rw-r-- 1 yongseok yongseok 177 Sep 15 17:16 sample1.txt
-rw-rw-r-- 1 yongseok yongseok 177 Sep 16 14:57 sample3.txt
drwxrwxr-x 2 yongseok yongseok 4096 Sep 15 16:56 test_linux2
yongseok@yongseok:~/test_linux$ ls -asl
total 28
4 drwxrwxr-x 3 yongseok yongseok 4096 Sep 16 16:39 .
4 drwxr-xr-x 23 yongseok yongseok 4096 Sep 15 16:56 ..
4 -rw-rw-r-- 1 yongseok yongseok 78 Sep 16 12:21 list1.txt
4 -rw-rw-r-- 1 yongseok yongseok 51 Sep 16 12:00 list2.txt
4 -rw-rw-r-- 1 yongseok yongseok 177 Sep 15 17:16 sample1.txt
4 -rw-rw-r-- 1 yongseok yongseok 177 Sep 16 14:57 sample3.txt
4 drwxrwxr-x 2 yongseok yongseok 4096 Sep 15 16:56 test_linux2
yongseok@yongseok:~/test_linux$ █
```

4. ROS 개발환경 구축

: chmod 8bit값 파일 이름



→ chmod : change mode. 파일의 속성 변경

사용권한 rw- rw- r--
2진수: 110 110 100
8진수: 6 6 4
\$ chmod 664 sample1.txt

```
yongseok@yongseok:~/test_linux$ ls -asl sample1.txt
4 -rw-rw-r-- 1 yongseok yongseok 177 Sep 15 17:16 sample1.txt
yongseok@yongseok:~/test_linux$ chmod 777 sample1.txt
yongseok@yongseok:~/test_linux$ ls -asl sample1.txt
4 -rwxrwxrwx 1 yongseok yongseok 177 Sep 15 17:16 sample1.txt
yongseok@yongseok:~/test_linux$
```

사용권한 rwx rwx rwx
2진수: 111 111 111
8진수: 7 7 7
\$ chmod 777 sample1.txt

: 내가 속한 폴더가 아닌 경우, sudo 명령어와 함께 사용

→ sudo chmod

4. ROS 개발환경 구축

df : disk driver에 있는 사용 가능한 공간을 표시(disk free) Kbyte 단위 df -k Mbyte 단위 df -m

free : 메모리의 양. 시스템의 현재 남은 공간을 표시

```
yongseok@yongseok:~$ df  
Filesystem 1K-blocks Used Available Use% Mounted on  
udev 1928424 0 1928424 0% /dev  
tmpfs 391840 11256 380584 3% /run  
/dev/sda2 121438316 9065028 106181468 8% /  
tmpfs 1959188 612 1958576 1% /dev/shm  
tmpfs 5120 4 5116 1% /run/lock  
tmpfs 1959188 0 1959188 0% /sys/fs/cgroup  
/dev/sda1 523248 5212 518036 1% /boot/efi  
tmpfs 391840 116 391724 1% /run/user/1000
```

```
yongseok@yongseok:~$ free  
total used free shared buff/cache available  
Mem: 3918376 1498028 425788 230524 1994560 1863840  
Swap: 1000444 0 1000444
```

```
yongseok@yongseok:~$ df -k  
Filesystem 1K-blocks Used Available Use% Mounted on  
udev 1928424 0 1928424 0% /dev  
tmpfs 391840 11256 380584 3% /run  
/dev/sda2 121438316 9065040 106181456 8% /  
tmpfs 1959188 612 1958576 1% /dev/shm  
tmpfs 5120 4 5116 1% /run/lock  
tmpfs 1959188 0 1959188 0% /sys/fs/cgroup  
/dev/sda1 523248 5212 518036 1% /boot/efi  
tmpfs 391840 116 391724 1% /run/user/1000
```

```
yongseok@yongseok:~$ df -m  
Filesystem 1M-blocks Used Available Use% Mounted on  
udev 1884 0 1884 0% /dev  
tmpfs 383 11 372 3% /run  
/dev/sda2 118593 8853 103693 8% /  
tmpfs 1914 1 1913 1% /dev/shm  
tmpfs 5 1 5 1% /run/lock  
tmpfs 1914 0 1914 0% /sys/fs/cgroup  
/dev/sda1 511 6 506 1% /boot/efi  
tmpfs 383 1 383 1% /run/user/1000
```

```
yongseok@yongseok:~$
```

4. ROS 개발환경 구축

: Linux

명령어

<code>ls</code>	디렉터리 내용의 목록을 표시한다.
<code>clear</code>	화면을 지운다.
<code>echo</code>	인수로 지정한 문자열을 출력한다.
<code>cp</code>	파일을 복사한다.
<code>rm</code>	파일을 삭제한다.
<code>mv</code>	파일을 이동하거나 이름을 바꾼다.
<code>ln</code>	다른 파일을 가리키는 연결을 생성한다.
<code>cd</code>	현재 디렉터리를 이동한다.
<code>mkdir</code>	디렉터리를 생성한다.
<code>rmdir</code>	디렉터리를 삭제한다.
<code>pwd</code>	현재 작업 중인 디렉터리의 경로를 표시한다.
<code>date</code>	시스템에 설정된 현재의 시간과 날짜를 출력한다.
<code>dirs</code>	디렉터리 스택에 저장된 내용을 표시한다.
<code>pushd</code>	디렉터리 스택에 (현재의) 디렉터리를 저장한다.
<code>popd</code>	디렉터리 스택에서 마지막에 저장한 디렉터리를 삭제하고 그 디렉터리로 이동한다.
<code>cat</code>	파일의 내용을 표시한다.
<code>more</code>	파일의 내용을 페이지 단위로 표시한다.
<code>wc</code>	문서 내에서의 줄(단어, 글자)의 수를 알려준다.
<code>man</code>	유ти리티나 API에 대한 도움말 매뉴얼을 보여준다.
<code>apropos</code>	해당 주제와 관련된 명령어들을 표시한다.
<code>which</code>	명령어의 위치하고 있는 경로를 표시한다.

<code>find</code>	시스템에서 파일이나 명령어를 찾는다.
<code>grep</code>	파일 내의 내용을 패턴을 이용해서 검색한다.
<code>adduser</code>	시스템에 사용자를 추가한다.
<code>sudo</code>	다른 사용자의 보안 권한으로 명령어를 수행한다.
<code>su</code>	root나 다른 사용자로 변경한다.
<code>who</code>	사용자에 대한 정보를 표시한다.
<code>wall</code>	모든 사용자에게 메시지를 표시한다.
<code>logout</code>	현재의 시스템에서 계정을 로그아웃한다.
<code>dmesg</code>	커널의 부팅 로그 및 커널 로그를 출력한다.
<code>chmod</code>	파일의 접근 권한을 변경한다.
<code>chown</code>	파일의 소유자를 변경한다.
<code>chgrp</code>	파일과 관련된 그룹을 변경한다.
<code>passwd</code>	사용자의 비밀번호를 변경한다.
<code>du</code>	디스크 사용량을 검사한다.
<code>df</code>	디스크 남은 공간을 표시한다.
<code>free</code>	시스템의 현재 남은 공간을 표시한다.
<code>mount</code>	디바이스를 시스템에 연결(Mount)한다.
<code>ps</code>	프로세스의 상태를 표시한다.
<code>kill</code>	프로세스에 시그널을 보내며, 주로 프로세스를 죽이는 데 사용한다.
<code>top</code>	시스템에서의 현재 프로세스의 상태를 표시한다.
<code>fg</code>	프로세스를 포그라운드 모드로 실행한다.

4. ROS 개발환경 구축

: Linux

명령어

bg	프로세스를 백그라운드 모드로 실행한다.
sync	현재 캐시된 내용을 저장한다.
cal	달력을 보여준다.
tar	여러 개의 파일을 하나의 파일로 묶는다.
compress	파일을 압축하거나 파일의 압축을 해제한다.
halt	시스템을 종료한다.
reboot	시스템을 재시작한다.
poweroff	시스템 전원을 내린다.
startx	x 윈도우를 시작한다.
telnet	원격 서버 터미널로 접속한다.
ftp	ftp 서버에 접속한다.
apt-get	우분투(Ubuntu)를 포함한 데비안(Debian) 계열에서 소프트웨어 패키지를 설치 및 관리한다.
apt-cache	우분투(Ubuntu)를 포함한 데비안(Debian) 계열에서 소프트웨어 패키지를 검색하여 표시한다.

4. ROS 개발환경 구축

4-2-3. Install ROS on Remote PC 에서 (즉 우분투 환경에서 작업하기, Remote PC에서 작업하기)

(1) Install ROS Noetic (Kinetic) 삭제

: Remote PC 성능에 따라서 ROS 설치가 안정화 되지 않는 현상이 있음

→ ROS 설치가 안되면 down받은 ROS등을 제거하고 재설치 해야 함

: ROS Kinetic 삭제하는 방법

→ 기존에 설치된 ros 삭제

\$ sudo apt-get remove ros-* (패키지만 삭제, 설정 파일은 남겨둠)

\$ sudo apt-get purge ros-* (패키지와 설정파일 함께 삭제)



\$ sudo apt autoremove

→ 삭제 확인

\$ rosversion -d (여기에서 <unknown>이라고 뜨면 삭제가 된 것임)

(만약 기존 ros version이 확인되면, 재부팅 후 다시 확인해 보면, 삭제된 것을 확인할 수 있음)

4. ROS 개발환경 구축

Ref. <https://emanual.robotis.com/docs/en/platform/turtlebot3/quick-start/#pc-setup>

4-2-3. Install ROS on Remote PC에서 (우분투 환경에서 작업하기, Remote PC에서 작업하기)

(2) Install ROS Noetic (emanual.robotis.com 열고 bash 내용을 copy하여 paste)

The screenshot shows the 'Quick Start Guide' section of the website. A red box highlights '3. 1. PC Setup'. Step 1 is labeled '1. click' with a red box around the '3. 1. PC Setup' link. Step 2 is labeled '2. click' with a red box around the 'Noetic' tab in the ROS version menu. Step 3 is labeled 'Ubuntu 20 & Noetic 확인'.

Ubuntu 20 & Noetic 확인

1. 1. Download and Install Ubuntu on PC

1. Download the proper [Ubuntu 20.04 LTS Desktop](#) image for your PC from the links below.
 - Ubuntu 20.04 LTS Desktop image (64-bit)
2. Follow the instruction below to install Ubuntu on PC.
 - [Install Ubuntu desktop](#)

1. 2. Install ROS on Remote PC

Open the terminal with [Ctrl + Alt + T](#) and enter below commands one at a time.
In order to check the details of the easy installation script, please refer to [the script file](#).

```
$ sudo apt update  
$ sudo apt upgrade  
$ wget https://raw.githubusercontent.com/ROBOTIS-GIT/robotis-tools/master/install_ros_noetic.sh  
$ chmod 755 ./install_ros_noetic.sh  
$ bash ./install_ros_noetic.sh
```

If the above installation fails, please refer to [the official ROS1 Noetic installation guide](#).

1. 3. Install Dependent ROS Packages

```
$ sudo apt-get install ros-noetic-joy ros-noetic-teleop-twist-joy  
ros-noetic-teleop-twist-keyboard ros-noetic-laser-proc  
ros-noetic-rgbd-launch ros-noetic-rosserial-arduino  
ros-noetic-rosserial-python ros-noetic-rosserial-client  
ros-noetic-rosserial-msgs ros-noetic-amcl ros-noetic-map-server  
ros-noetic-move-base ros-noetic-urdf ros-noetic-xacro  
ros-noetic-compressed-image-transport ros-noetic-rat* ros-noetic-rviz  
ros-noetic-gmapping ros-noetic-navigation ros-noetic-interactive-markers
```

2

1. 먼저 무선 연결 반드시 확인
(실제 YOUTUBE 를 실행하여 연결되는지 확인)
2. 터미널 열기 Ctrl + Alt + T
3. Ctrl + Shift + V
4. 운행 종료 Ctrl + C
5. 터미널 닫기 Ctrl + Shift + Q

3 종속성 패키지 설치 후 재부팅
(컴파일 안되는 문제 있음)

4. ROS 개발환경 구축

Ref. <https://emanual.robotis.com/docs/en/platform/turtlebot3/quick-start/#pc-setup>

4-2-3. Install ROS on Remote PC에서

(2) Install ROS Noetic

3. 1. 4. Install TurtleBot3 Packages

Install TurtleBot3 via Debian Packages.

```
$ sudo apt install ros-noetic-dynamixel-sdk  
$ sudo apt install ros-noetic-turtlebot3-msgs  
$ sudo apt install ros-noetic-turtlebot3
```

(컴파일 안되는 문제 있음)

건너뛰고



Click here to expand more details about building TurtleBot3 package from source.

In case you need to download the source codes and build them, please use the
Make sure to remove the identical packages to avoid redundancy.

종속성 패키지 설치 후 재부팅(컴파일 안되는 문제 있음)
: 아래와 같이 진행할 것

4

```
$ sudo apt remove ros-noetic-dynamixel-sdk  
$ sudo apt remove ros-noetic-turtlebot3-msgs  
$ sudo apt remove ros-noetic-turtlebot3  
$ mkdir -p ~/catkin_ws/src  
$ cd ~/catkin_ws/src/  
$ git clone -b noetic-devel https://github.com/ROBOTIS-GIT/DynamixelSDK.git  
$ git clone -b noetic-devel https://github.com/ROBOTIS-GIT/turtlebot3_msgs.git  
$ git clone -b noetic-devel https://github.com/ROBOTIS-GIT/turtlebot3.git  
$ cd ~/catkin_ws && catkin_make  
$ echo "source ~/catkin_ws/devel/setup.bash" >> ~/.bashrc
```

적색 박스 부분 진행



Install ROS Noetic 설치 화면

```
yongseok@yongseok:~$ sudo apt update  
[sudo] password for yongseok:  
Hit:1 http://security.ubuntu.com/ubuntu focal-security InRelease  
Hit:2 http://kr.archive.ubuntu.com/ubuntu focal InRelease  
Get:3 http://kr.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]  
Hit:4 http://kr.archive.ubuntu.com/ubuntu focal-backports InRelease  
Fetched 114 kB in 3s (36.1 kB/s)  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
All packages are up to date.  
yongseok@yongseok:~$ sudo apt upgrade  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
Calculating upgrade... Done  
Get more security updates through Ubuntu Pro with 'esm-apps' enabled:  
  exo-utils tigervnc-common tigervnc-standalone-server libexo-2-0  
  tigervnc-xorg-extension libexo-common xrdp libexo-helpers  
Learn more about Ubuntu Pro at https://ubuntu.com/pro  
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.  
yongseok@yongseok:~$ wget https://raw.githubusercontent.com/ROBOTIS-GIT/robotis_tools/master/install_ros_noetic.sh  
yongseok@yongseok:~$ chmod 755 ./install_ros_noetic.sh  
yongseok@yongseok:~$ ./install_ros_noetic.sh  
yongseok@yongseok:~$
```

수정된 텍스트와 주석:

- sudo apt update → sudo apt update
- 1234 enter(입력 안보임) → 입력 후 enter
- 입력 후 enter → update 화면(완료 확인할 것)
- sudo apt upgrade → sudo apt upgrade
- wget https://raw.githubusercontent.com/ROBOTIS-GIT/robotis_tools/master/install_ros_noetic.sh → 입력 후 enter
- 완료 확인 → 완료 확인
- 권한변경 → 권한변경

Install ROS Noetic 설치 화면

```
yongseok@yongseok:~$ chmod +x ./install_ros_noetic.sh
yongseok@yongseok:~$ bash ./install_ros_noetic.sh

[Note] Target OS version >>> Ubuntu 20.04.x (Focal Fossa) or Linux Mint 21.x
[Note] Target ROS version >>> ROS Noetic Ninjemys
[Note] Catkin workspace >>> /home/yongseok/catkin_ws

PRESS [ENTER] TO CONTINUE THE INSTALLATION
IF YOU WANT TO CANCEL, PRESS [CTRL] + [C]      enter

[Set the target OS, ROS version and name of catkin workspace]
[Update the package lists]
Hit:1 http://kr.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://kr.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://kr.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu focal-security InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
All packages are up to date.
```

입력 후 enter

enter

~ 화면 생략

```
-- BUILD_SHARED_LIBS is on
-- BUILD_SHARED_LIBS is on
-- Configuring done
-- Generating done
-- Build files have been written to: /home/yongseok/catkin_ws/build
#####
##### Running command: "make -j4 -l4" in "/home/yongseok/catkin_ws/build"
#####
[Set the ROS environment]
[Complete!!!]
```

완료 확인 - 매우 중요

```
yongseok@yongseok: ~
 매우 중요 내용 확인하기

/home/yongseok/catkin_ws$ catkin build -j4 -l4
[  0%] Checking for working C compiler: /bin/cc -- works
[  0%] Checking for working CXX compiler: /bin/c++ -- works
[  0%] Detecting C compiler ABI info
[  0%] Detecting C compiler ABI info - done
[  0%] Detecting C compile features
[  0%] Detecting C compile features - done
[  0%] Check for working CXX compiler: /bin/c++ -- works
[  0%] Check for working CXX compiler: /bin/c++ -- works
[  0%] Detecting CXX compiler ABI info
[  0%] Detecting CXX compiler ABI info - done
[  0%] Detecting CXX compile features
[  0%] Detecting CXX compile features - done
[  0%] Using CATKIN_DEVEL_PREFIX: /home/yongseok/catkin_ws/devel
[  0%] Using CMAKE_PREFIX_PATH: /opt/ros/noetic
[  0%] This workspace overlays: /opt/ros/noetic
[  0%] Found PythonInterp: /bin/python3 (found suitable version "3.8.10", minimum required is "3")
[  0%] Using PYTHON_EXECUTABLE: /bin/python3
[  0%] Using Debian Python package layout
[  0%] Found PY_em: /usr/lib/python3/dist-packages/em.py
[  0%] Using empv: /usr/lib/python3/dist-packages/em.py
[  0%] Using CATKIN_ENABLE_TESTING: ON
[  0%] Call enable_testing()
[  0%] Using CATKIN_TEST_RESULTS_DIR: /home/yongseok/catkin_ws/build/test_results
[  0%] Forcing gtest/gmock from source, though one was otherwise available.
[  0%] Found gtest sources under '/usr/src/googletest': gtests will be built
[  0%] Found gmock sources under '/usr/src/googletest': gmock will be built
[  0%] Found PythonInterp: /bin/python3 (found version "3.8.10")
[  0%] Found Threads: TRUE
[  0%] Using Python nosetests: /bin/nosetests3
[  0%] catkin 0.8.10
[  0%] BUILD_SHARED_LIBS is on
[  0%] BUILD_SHARED_LIBS is on
[  0%] Configuring done
[  0%] Generating done
[  0%] Build files have been written to: /home/yongseok/catkin_ws/build
#####
##### Running command: "make -j4 -l4" in "/home/yongseok/catkin_ws/build"
#####
[Set the ROS environment]
```

Install ROS Noetic 설치 화면 : ROS 종속 package

```
yongseok@yongseok:~$ sudo apt-get install ros-noetic-joy ros-noetic-teleop-twist  
-joy \  
> ros-noetic-teleop-twist-keyboard ros-noetic-laser-proc \  
> ros-noetic-rgbd-launch ros-noetic-rosserial-arduino \  
> ros-noetic-rosserial-python ros-noetic-rosserial-client \  
> ros-noetic-rosserial-msgs ros-noetic-amcl ros-noetic-map-server \  
> ros-noetic-move-base ros-noetic-urdf ros-noetic-xacro \  
> ros-noetic-compressed-image-transport ros-noetic-rqt* ros-noetic-rviz \  
> ros-noetic-gmapping ros-noetic-navigation ros-noetic-interactive-markers  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
Note, selecting 'ros-noetic-rqt-common-plugins' for glob 'ros-noetic-rqt*'  
Note, selecting 'ros-noetic-rqt-ground-robot-teleop' for glob 'ros-noetic-rqt*'  
Done 확인
```

입력 후 enter

~ 화면 생략

```
Setting up ros-noetic-carrot-planner (1.17.3-1focal.20231013.192333) ...  
Setting up ros-noetic-move-base (1.17.3-1focal.20231013.192946) ...  
Setting up ros-noetic-global-planner (1.17.3-1focal.20231013.192945) ...  
Setting up arduino-core (2:1.0.5+dfsg2-4.1) ...  
Setting up ros-noetic-rosserial-arduino (0.9.2-1focal.20231013.201734) ...  
Setting up ros-noetic-dwa-local-planner (1.17.3-1focal.20231013.192337) ...  
Setting up ros-noetic-navigation (1.17.3-1focal.20231013.195837) ...  
Processing triggers for man-db (2.9.1-1) ...  
Processing triggers for libc-bin (2.31-0ubuntu9.12) ...  
yongseok@yongseok:~$
```

Install ROS Noetic 설치 화면

Install TurtleBot3 Package from source : 주의

종속성 패키지 설치 후 재부팅(컴파일 안되는 문제 있음)
: 아래와 같이 진행할 것

```
yongseok@yongseok: $ cd ~/catkin_ws/src/  
yongseok@yongseok: $ git clone -b noetic-devel https://github.com/  
ROBOTIS-GIT/DynamixelSDK.git  
Cloning into 'DynamixelSDK'...  
remote: Enumerating objects: 13861, done.  
remote: Counting objects: 100% (1235/1235), done.  
remote: Compressing objects: 100% (553/553), done.  
remote: Total 13861 (delta 629), reused 1131 (delta 580), pack-reused 12626  
Receiving objects: 100% (13861/13861), 28.58 MiB | 3.99 MiB/s, done. Done 확인  
Resolving deltas: 100% (7877/7877), done.  
yongseok@yongseok:~/catkin_ws/src$ git clone -b noetic-devel https://github.com/  
ROBOTIS-GIT/turtlebot3_msgs.git  
Cloning into 'turtlebot3_msgs'...  
remote: Enumerating objects: 415, done.  
remote: Counting objects: 100% (173/173), done.  
remote: Compressing objects: 100% (58/58), done.  
remote: Total 415 (delta 72), reused 156 (delta 61), pack-reused 242  
Receiving objects: 100% (415/415), 91.62 KiB | 1.95 MiB/s, done. Done 확인  
Resolving deltas: 100% (173/173), done.  
  
username for 'https://github.com':  
yongseok@yongseok:~/catkin_ws/src$ git clone -b noetic-devel https://github.com/  
ROBOTIS-GIT/turtlebot3.git  
Cloning into 'turtlebot3'...  
remote: Enumerating objects: 6636, done.  
remote: Counting objects: 100% (1255/1255), done.  
remote: Compressing objects: 100% (207/207), done.  
remote: Total 6636 (delta 1096), reused 1097 (delta 1048), pack-reused 5381  
Receiving objects: 100% (6636/6636), 119.99 MiB | 3.92 MiB/s, done.  
Resolving deltas: 100% (4149/4149), done. Done 확인
```

Install ROS Noetic 설치 화면

Build - catkin_make

```
Resolving dependencies: 100% (1/1), done.  
yongseok@yongseok:~/catkin_ws/src$ cd ~/catkin_ws && catkin_make  
Base path: /home/yongseok/catkin_ws  
Source space: /home/yongseok/catkin_ws/src  
Build space: /home/yongseok/catkin_ws/build  
Devel space: /home/yongseok/catkin_ws/devel  
Install space: /home/yongseok/catkin_ws/install  
####  
#### Running command: "cmake /home/yongseok/catkin_ws/src -DCATKIN_DEVEL_PREFIX=/home/yongseok/catkin_ws/devel -DCMAKE_INSTALL_PREFIX=/home/yongseok/catkin_ws/install -G Unix Makefiles" in "/home/yongseok/catkin_ws/build"  
####  
-- Using CATKIN_DEVEL_PREFIX: /home/yongseok/catkin_ws/devel  
-- Using CMAKE_PREFIX_PATH: /home/yongseok/catkin_ws/devel;/opt/ros/noetic  
-- This workspace overlays: /home/yongseok/catkin_ws/devel;/opt/ros/noetic  
-- Found PythonInterp: /bin/python3 (found suitable version "3.8.10", minimum re
```

```
-- Using Python nosetests: /bin/nosetests3  
-- catkin 0.8.10  
-- BUILD_SHARED_LIBS is on  
-- BUILD_SHARED_LIBS is on  
--  
-- ~~~ traversing 10 packages in topological order:  
-- ~~~ - turtlebot3 (metapackage)  
-- ~~~ - turtlebot3_msgs  
-- ~~~ - turtlebot3_navigation  
-- ~~~ - dynamixel_sdk  
-- ~~~ - dynamixel_sdk_examples  
-- ~~~ - turtlebot3_bringup  
-- ~~~ - turtlebot3_example  
-- ~~~ - turtlebot3_slam  
-- ~~~ - turtlebot3_teleop  
-- ~~~ - turtlebot3_description  
--  
-- +++ processing catkin metapackage: 'turtlebot3'  
-- Build files have been written to: /home/yongseok/catkin_ws/build
```

```
####  
#### Running command: "make -j4 -l4" in "/home/yongseok/catkin_ws/build"  
####  
Scanning dependencies of target _turtlebot3_msgs_generate_messages_check_deps_VersionInfo  
Scanning dependencies of target _turtlebot3_msgs_generate_messages_check_deps_Sound  
Scanning dependencies of target std_msgs_generate_messages_cpp  
Scanning dependencies of target _turtlebot3_msgs_generate_messages_check_deps_SensorState  
[ 0%] Built target std_msgs_generate_messages_cpp  
Scanning dependencies of target std_msgs_generate_messages_py
```

경로& build

: 경로 확인 (build 경로)

```
$ cd ~/catkin_ws && catkin_make  
→ $ cd ~/catkin_ws  
→ $ catkin_make
```

: 우리가 만든 src 폴더 확인

그리고 build 와 devel 폴더 생성

: catkin 빌드 파일은 build 폴더 저장

: 빌드 후 실행 파일은 devel 폴더 저장

만일 이 내용이 없다면 catkin_make가 불안전
(source code와 환경설정 build 안 된 것임)

→ ubuntu 재설치 (roscore는 설치 되었어도 문제가 내재)

```
[ 98%] Built target indirect_address_node  
[100%] Linking CXX executable /home/yongseok/catkin_ws/devel/lib/dynamixel_sdk_examples/sync_read_write_node  
[100%] Built target sync_read_write_node  
yongseok@yongseok:~/catkin_ws$
```

100% build 확인

환경변수에 IP 반영하기 : emanual.robotis 3.1.5 Network Configuration

```
yongseok@yongseok: $ ifconfig  
enp2s0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500  
ether 04:7c:16:ca:1e:b5 txqueuelen 1000 (Ethernet)  
RX packets 0 bytes 0 (0.0 B)  
RX errors 0 dropped 0 overruns 0 frame 0  
TX packets 0 bytes 0 (0.0 B)  
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
enp3s0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500  
ether 04:7c:16:ca:1e:b6 txqueuelen 1000 (Ethernet)  
RX packets 0 bytes 0 (0.0 B)  
RX errors 0 dropped 0 overruns 0 frame 0  
TX packets 0 bytes 0 (0.0 B)  
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536  
inet 127.0.0.1 netmask 255.0.0.0  
inet6 ::1 prefixlen 128 scopeid 0x10<host>  
loop txqueuelen 1000 (Local Loopback)  
RX packets 2000 bytes 212766 (212.7 KB)  
RX errors 0 dropped 0 overruns 0 frame 0  
TX packets 2000 bytes 212766 (212.7 KB)  
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
wlo1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
inet 192.168.1.141 netmask 255.255.255.0 broadcast 192.168.1.255  
inet6 fe80::2b82:96d6:a3ce:da15 prefixlen 64 scopeid 0x10<link>  
ether 00:d4:9e:f7:ba:c0 txqueuelen 1000 (Ethernet)  
RX packets 71348 bytes 100165169 (100.1 MB)
```

\$ ifconfig

Ip기억

: emanual.robotis 3.1.5 에서는 nano 편집기 사용하였으나,

```
yongseok@yongseok:~$ gedit ~/.bashrc
```

\$ gedit ~/.bashrc



ROS 환경변수 설정 :

```
$ source /opt/ros/noetic/setup.bash  
$ source ~/catkin_ws/devel/setup.bash
```

: 위와 같은 ROS 설치 과정에서 사용된 환경설정 파일을
매번 불러오기 번거롭기에 bashrc 파일에 환경변수 설정

```
Open + .bashrc Save  
1 # ~/.bashrc: executed by bash(1) for non-login shells.  
2 # see /usr/share/doc/bash/examples/startup-files (in the package bash-doc)  
3 # for examples  
4  
5 # If not running interactively, don't do anything  
6 case $- in  
7     *i*) ;;  
8     *) return;;  
9 esac  
10  
11 # don't put duplicate lines or lines starting with space in the history.  
12 # See bash(1) for more options  
13 HISTCONTROL=ignoreboth  
14  
15  
16 alias eb='nano ~/.bashrc'  
17 alias sb='source ~/.bashrc'  
18 alias gs='git status'  
19 alias gp='git pull'  
20 alias cw='cd ~/catkin_ws'  
21 alias cs='cd ~/catkin_ws/src'  
22 alias cm='cd ~/catkin_ws && catkin make'  
23  
24 source /opt/ros/noetic/setup.bash  
25 source ~/catkin_ws/devel/setup.bash  
26 export ROS_MASTER_URI=http://localhost:11311  
27 export ROS_HOSTNAME=localhost
```

~ 화면 생략

환경변수에 IP 반영하기 : emanual.robotis 3.1.5 Network Configuration

```
118 alias eb='nano ~/.bashrc'  
119 alias sb='source ~/.bashrc'  
120 alias gs='git status'  
121 alias gp='git pull'  
122 alias cw='cd ~/catkin_ws'  
123 alias cs='cd ~/catkin_ws/src'  
124 alias cm='cd ~/catkin_ws && catkin_make'  
125 source /opt/ros/noetic/setup.bash  
126 source ~/catkin_ws/devel/setup.bash  
127 export ROS_MASTER_URI=http://localhost:11311  
128 export ROS_HOSTNAME=localhost
```



```
117 '*  
118 alias eb='nano ~/.bashrc'  
119 alias sb='source ~/.bashrc'  
120 alias gs='git status'  
121 alias gp='git pull'  
122 alias cw='cd ~/catkin_ws'  
123 alias cs='cd ~/catkin_ws/src'  
124 alias cm='cd ~/catkin_ws && catkin_make'  
125 source /opt/ros/noetic/setup.bash  
126 source ~/catkin_ws/devel/setup.bash  
127 export ROS_MASTER_URI=http://192.168.1.141:11311  
128 export ROS_HOSTNAME=192.168.1.141
```

IP 변경 후, SAVE, 닫기



ROS 환경변수 반영하기

```
yongseok@yongseok:~$ gedit ~/.bashrc  
yongseok@yongseok:~$ source ~/.bashrc  
yongseok@yongseok:~$ █  
$ source ~/.bashrc
```

[alias 단축 명령어]

alias **eb**'nano ~/.bashrc'

alias **sb**'source ~/.bashrc'

alias **gs**'git status'

alias **gp**'git pull'

alias **cw**'cd ~/catkin_ws'

alias **cs**'cd ~/catkin_ws/src'

alias **cm**'cd ~/catkin_ws && catkin_make'

→ **eb** nano 편집기로 bashrc 파일을 열기 nano ~/.bashrc

→ **sb** 환경변수 설정을 반영 source ~/.bashrc

→ **gs** git status 현재상태를 확인

→ **gp** git pull fetch하고 + merge하는 명령어

→ **cw** cd ~/catkin_ws build 디렉토리 이동

→ **cs** cd ~/catkin_ws/src 이동

→ **cm** cd ~/catkin_ws && catkin_make

4. ROS 개발환경 구축

절대 X 누르지 않기 

★ 최종 테스트

→ \$ roscore

: ROS master 구동하는 명령어



오류가 발생한다면

ROS 설치에 문제 있음

재설치 필요

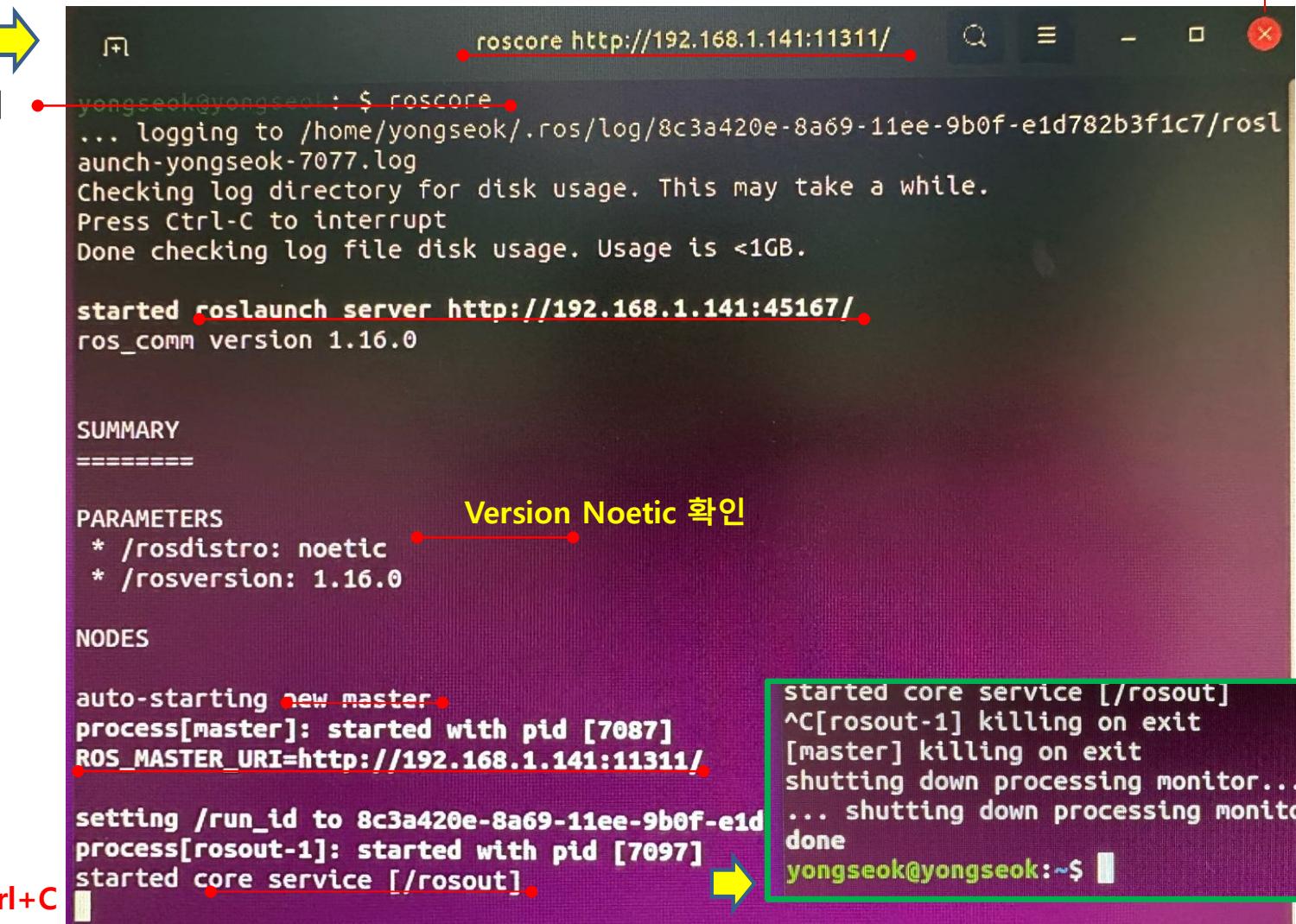


문제가 지속 발생 경우

UBUNTU 재설치



이 상태에서 끝내려면 Ctrl+C



```
roscore http://192.168.1.141:11311/
yongseok@yongseok: ~$ roscore
... logging to /home/yongseok/.ros/log/8c3a420e-8a69-11ee-9b0f-e1d782b3f1c7/roslog-yongseok-7077.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://192.168.1.141:45167/
ros_comm version 1.16.0

SUMMARY
=====

PARAMETERS
* /rosdistro: noetic
* /rosversion: 1.16.0          Version Noetic 확인

NODES

auto-starting new master
process[master]: started with pid [7087]
ROS_MASTER_URI=http://192.168.1.141:11311/                                ROS_MASTER_URI 설정

setting /run_id to 8c3a420e-8a69-11ee-9b0f-e1d
process[rosout-1]: started with pid [7097]                                     process[rosout-1] 시작
started core service [/rosout]                                                 started core service [/rosout]

started core service [/rosout]
^C[rosout-1] killing on exit
[master] killing on exit
shutting down processing monitor...
... shutting down processing monitor complete
done
yongseok@yongseok: ~$
```

4. ROS 개발환경 구축

Ubuntu 16.04 SKIP

Ref. ROS 로봇 프로그래밍(Ruby paper) 3-1 ROS 설치

4-2-3. Install ROS on Remote PC에서 (즉 우분투 환경에서 작업하기, Remote PC에서 작업하기)

(2) Install ROS Kinetic

: 방법 1(수동설치방법, 방법2 자동설치. 방법2로 하세요) - ROS 로봇 프로그래밍(Ruby paper) 3-1 ROS 설치

ⓐ 다른 PC간 통신(server, client)에서 ROS Time 오차를 줄일 수 있도록 NTP(Network Time Protocol)를 설정(시간 동기화)

- 설정 : **chrony** (Linux time synchronizing)를 설치한 후에 **ntpdate** 명령어로 NTP서버를 지정
- **CTRL + ALT + T** 를 사용해 터미널 창 열기

```
$ sudo apt-get install -y chrony ntpdate
```

```
$ sudo ntpdate -q ntp.ubuntu.com
```

The terminal window shows the following command being run:

```
yongseok@yongseok:~$ sudo apt-get install -y chrony ntpdate
```

Followed by the output of the command:

```
[sudo] password for yongseok:  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
chrony is already the newest version (2.1.1-1ubuntu0.1).  
ntpdate is already the newest version (1:4.2.8p4+dfsg-3ubuntu5.10).  
0 upgraded, 0 newly installed, 0 to remove and 9 not upgraded.
```

Then, the command:

```
yongseok@yongseok:~$ sudo ntpdate -q ntp.ubuntu.com
```

Followed by the output:

```
server 91.189.89.199, stratum 2, offset 0.000245, delay 0.28320  
server 91.189.94.4, stratum 2, offset 0.003806, delay 0.28764  
server 91.189.91.157, stratum 2, offset -0.008543, delay 0.22823  
server 91.189.89.198, stratum 2, offset -0.017860, delay 0.29843  
26 Sep 16:15:52 ntpdate[5117]: adjust time server 91.189.91.157 offset -0.008543  
sec
```

Finally, the command:

```
yongseok@yongseok:~$
```

Annotations on the right side of the terminal window:

- y library 설치 시 yes
- q : ntpdate 설정 옵션
- Time synchronizing

1. 제일먼저

무선 연결 반드시 확인

(실제 YOUTUBE 를 실행하여 연결되는지 확인)

4. ROS 개발환경 구축

Ubuntu 16.04 SKIP

Ref. ROS 로봇 프로그래밍(Ruby paper) 3-1 ROS 설치

ⓑ ros-latest.list에 ROS 저장 주소 추가

→ \$ sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu \$(lsb_release -sc) main" > /etc/apt/sources.list.d/ros-latest.list'

: \$(lsb_release -sc)는 리눅스 배포 판의 정보 중 코드 네임을 가져오는 것

```
sec
yongseok@yongseok:~$ sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release -sc) main" > /etc/apt/sources.list.d/ros-latest.list'
sh: 1: cannot create /etc/apt/: Is a directory
sh: 2: sources.list.d/ros-latest.list: not found
yongseok@yongseok:~$
```

ⓒ ROS 저장소로부터 패키지를 받기 위해 공개 키 추가 (아래 old key) ⭐

→ \$ sudo apt-key adv --keyserver hkp://ha.pool.sks-keyservers.net:80 --recv-key

421C365BD9FF1F717815A3895523BAEEB01FA116

```
yongseok@yongseok:~$ sudo apt-key adv --keyserver hkp://ha.pool.sks-keyservers.net:80 --recv-key 421C365BD9FF1F717815A3895523BAEEB01FA116
Executing: /tmp/tmp.oK6zdGGNCd/gpg.1.sh --keyserver hkp://ha.pool.sks-keyservers.net:80 --recv-key 421C365BD9FF1F717815A3895523BAEEB01FA116
usage: gpg [options] [filename]
yongseok@yongseok:~$
```

한 칸 띄기

4. ROS 개발환경 구축

Ubuntu 16.04 SKIP

Ref. ROS 로봇 프로그래밍(Ruby paper) 3-1 ROS 설치

④ 현재 설치된 Ubuntu package index 업데이트

➔ \$ sudo apt-get update && sudo apt-get upgrade -y

```
yongseok@yongseok:~$ sudo apt-get update && sudo apt-get upgrade -y
[youtube] http://kr.archive.ubuntu.com/ubuntu xenial InRelease
[youtube] http://packages.ros.org/ubuntu xenial InRelease
[get] http://kr.archive.ubuntu.com/ubuntu xenial-updates InRelease [109 kB]
[get] http://security.ubuntu.com/ubuntu xenial-security InRelease [109 kB]
[youtube] http://archive.ubuntu.com/ubuntu xenial-backports InRelease [107 kB]
```

-Y : YES

-설치 내용 생략

```
Unpacking libgnutls30:amd64 (3.4.10-4ubuntu1.9) over (3.4.10-4ubuntu1.8) ...
Processing triggers for libc-bin (2.23-0ubuntu11.3) ...
Setting up libgnutls30:amd64 (3.4.10-4ubuntu1.9) ...
Setting up libgnutls-openssl27:amd64 (3.4.10-4ubuntu1.9) ...
Processing triggers for libc-bin (2.23-0ubuntu11.3) ...
```

4. ROS 개발환경 구축

Ubuntu 16.04 SKIP

Ref. ROS 로봇 프로그래밍(Ruby paper) 3-1 ROS 설치

④ ROS Kinetic Kame 설치

: ROS 패키지(ROS, rqt, RViz, 로봇 관련 라이브러리, 시뮬레이션, 내비게이션 등)

→ \$ sudo apt-get install ros-kinetic-desktop-full

```
yongseok@yongseok:~$ sudo apt-get install ros-kinetic-desktop-full
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed: 시간 소요
  cmake cmake-data fltk1.3-doc fluid fonts-lato freeglut3 freeglut3-dev
  gazebo7 gazebo7-common gazebo7-plugins
```

시간 소요

-설치 내용 생략

: rqt 관련 모든 패키지를 설치(rqt 플러그인 사용 위함)

→ \$ sudo apt-get install ros-kinetic-rqt*

```
yongseok@yongseok:~$ sudo apt-get install ros-kinetic-rqt*
Reading package lists... Done
Building dependency tree
Reading state information... Done
Note, selecting 'ros-kinetic-rqt-drone-teleop' for glob 'ros-kinetic-rqt*'
Note, selecting 'ros-kinetic-rqt-gui-py' for glob 'ros-kinetic-rqt*'
Note, selecting 'ros-kinetic-rqt-tf-tree' for glob 'ros-kinetic-rqt*'
```

-설치 내용 생략

4. ROS 개발환경 구축

Ubuntu 16.04 SKIP

Ref. ROS 로봇 프로그래밍(Ruby paper) 3-1 ROS 설치

① rosdep 초기화

: rosdep은 ros의 핵심 컴포넌트들을 사용하거나

컴파일할 때 의존성 패키지를 설치하기 위해(초기화하고, 업데이트를 진행) 사용자 편의성을 높여주는 기능

→ \$ sudo rosdep init
\$ rosdep update

```
Processing triggers for libc-bin (2.23-0ubuntu11.3) ...
yongseok@yongseok:~$ sudo rosdep init 초기화
ERROR: default sources list file already exists:
/etc/ros/rosdep/sources.list.d/20-default.list
Please delete if you wish to re-initialize
yongseok@yongseok:~$ rosdep update 업데이트
reading in sources list data from /etc/ros/rosdep/sources.list.d
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/osx-homebrew.
aml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/base.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/python.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/ruby.yaml
```

-설치 내용 생략

② rosinstall 설치

: ROS의 다양한 패키지를 인스톨 프로그램

→ \$ sudo apt-get install python-rosinstall

```
updated cache in /home/yongseok/.ros/rosdep/sources.list.d
yongseok@yongseok:~$ sudo apt-get install python-rosinstall
Reading package lists... Done
Building dependency tree
Reading state information... Done
python-rosinstall is already the newest version (0.7.8-1).
0 upgraded, 0 newly installed, 0 to remove and 7 not upgrad
```

4. ROS 개발환경 구축

방법 2에서도 동일

Ubuntu 16.04 SKIP

Ref. ROS 로봇 프로그래밍(Ruby paper) 3-1 ROS 설치

④ 환경설정 파일 불러오기

: ROS_ROOT, ROS_PACKAGE_PATH 등의 환경변수 정의된 환경설정 파일 불러오기

→ \$ source /opt/ros/kinetic/setup.bash

```
yongseok@yongseok:~$ source /opt/ros/kinetic/setup.bash
```

bash : 파일을 읽고 실행 명령어

⑤ 작업 폴더 생성 및 초기화

: ROS에서는 catkin이라는 ROS 전용 빌드 시스템 사용

: catkin 사용하려면 catkin 작업 폴더를 생성하고 초기화해야 함

→ \$ mkdir -p ~/catkin_ws/src

\$ cd ~/catkin_ws/src

: mkdir -p : -parent 필요한 경우 상위경로까지 생성

```
yongseok@yongseok:~$ mkdir -p ~/catkin_ws/src
```

```
yongseok@yongseok:~$ cd ~/catkin_ws/src
```

```
yongseok@yongseok:~/catkin_ws/src$
```

directory 확인

\$ catkin_init_workspace

● CMakeLists.txt 파일 생성

```
yongseok@yongseok:~/catkin_ws/src$ catkin_init_workspace
```

```
File "/home/yongseok/catkin_ws/src/CMakeLists.txt" already exists
```

```
yongseok@yongseok:~/catkin_ws/src$
```

CMakeLists.txt 파일 확인

4. ROS 개발환경 구축

Ubuntu 16.04 SKIP

Ref. ROS 로봇 프로그래밍(Ruby paper) 3-1 ROS 설치

① build

: 경로 확인 (build 경로)

→ \$ cd ~/catkin_ws/

\$ catkin_make



\$ ls

```
[100%] Built target turtlebot3_example_generate_messages
yongseok@yongseok:~/catkin_ws$ ls
build  devel  install_ros_kinetic.sh  src
yongseok@yongseok:~/catkin_ws$
```



우리가 만든 src 폴더 확인

그리고 build 와 devel 폴더 생성

: catkin 빌드 파일은 build 폴더에 저장

: 빌드 후 실행 파일은 devel 폴더에 저장

```
yongseok@yongseok:~/catkin_ws/src$ catkin_init_workspace
File "/home/yongseok/catkin_ws/src/CMakeLists.txt" already existsyongseok@yongseok:~/catkin_ws$ cd ~ /catkin_ws/
yongseok@yongseok:~/catkin_ws$ catkin_make
Base path: /home/yongseok/catkin_ws
Source space: /home/yongseok/catkin_ws/src
Build space: /home/yongseok/catkin_ws/build
Devel space: /home/yongseok/catkin_ws/devel
Install space: /home/yongseok/catkin_ws/install
#####
##### Running command: "make cmake_check_build_system" in "/home/yongseok/catkin_ws/build"
```

만일 이 내용이 없다면 catkin_make가 불안전
(source code와 환경설정 build 안 된 것임)
→ ubuntu 재설치
(roscore는 설치되었어도 문제가 내재)

```
0%] Built target std_msgs_generate_messages_py
0%] Built target _turtlebot3_msgs_generate_messages_check_deps_SensorState
0%] Built target _turtlebot3_msgs_generate_messages_check_deps_Sound
0%] Built target std_msgs_generate_messages_cpp
0%] Built target turtlebot3_msgs_generate_messages_check_deps_VersionInfo
```

-실행 내용 생략

```
[ 84%] Built target turtlebot3_example_generate_messages_nodejs
[ 96%] Built target turtlebot3_example_generate_messages_lisp
[ 96%] Built target turtlebot3_msgs_generate_messages
[100%] Built target turtlebot3_diagnostics
[100%] Built target turtlebot3_example_generate_messages 100% build 확인
yongseok@yongseok:~/catkin_ws$
```



4. ROS 개발환경 구축

Ubuntu 16.04 SKIP

Ref. ROS 로봇 프로그래밍(Ruby paper) 3-1 ROS 설치

⑤ catkin 빌드 시스템과 관련된 환경 파일 적용하기

→ \$ source ~/catkin_ws/devel/setup.bash ★

```
yongseok@yongseok:~/catkin_ws$ ls  
build  devel  install_ros_kinetic.sh  src  
yongseok@yongseok:~/catkin_ws$ source ~/catkin_ws/devel/setup.bash  
yongseok@yongseok:~/catkin_ws$ █
```

테스트를 위해

CTRL + Shift +Q 열려있는 Terminal 창 닫기

다시 새로운 창 열기

CTRL + ALT + T 를 사용해 터미널 창 열기

build과정으로 생성된 devel파일 안의 setup.bash를 실행하여
그 작업공간이 ros 환경의 최상위에 overlay되도록 쉘에 등록
→ setup.bash를 실행하면 environment variables가 setting되고
이것이 시스템이 필요한 모든 package와 library 등을 찾게 도와줌

4. ROS 개발환경 구축

Ubuntu 16.04 SKIP

Ref. ROS 로봇 프로그래밍(Ruby paper) 3-1 ROS 설치

★ 최종 테스트

→ \$ roscore

: ROS master 구동하는 명령어



오류가 발생한다면



ROS 설치에 문제 있음

재설치 필요



문제가 지속 발생 경우

UBUNTU 재설치

```
yongseok@yongseok:~$ roscore
... logging to /home/yongseok/.ros/log/83cce24c-1e9c-11ec-b5be-f40669f0af37/roslaunch-yongseok-21558.log
Checking log directory for disk usage. This may take awhile.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://localhost:36691/
ros_comm version 1.12.17

SUMMARY
=====

PARAMETERS
* /rosdistro: kinetic
* /rosversion: 1.12.17 → Version Kinetic 확인

NODES

auto-starting new master
process[master]: started with pid [21569]
ROS_MASTER_URI=http://localhost:11311/

setting /run_id to 83cce24c-1e9c-11ec-b5be-f40669f0af37
process[rosout-1]: started with pid [21582]
started core service [/rosout] → 이 상태에서 끝내려면 Ctrl+c ★
```

4. ROS 개발환경 구축

방법 2 참고

Ubuntu 16.04 SKIP

Ref. <https://emanual.robotis.com/docs/en/platform/turtlebot3/quick-start/>

4-2-3. Install ROS on Remote PC에서 (즉 우분투 환경에서 작업하기, Remote PC에서 작업하기)

(3) Install ROS Kinetic

: 방법 2(자동설치) <https://emanual.robotis.com/docs/en/platform/turtlebot3/quick-start>

1.1.4까지 진행

→ Firefox web browser에서 위 주소 열기

drag 하여 터미널 창
에 붙여 넣기 실행

drag 하여 실행하기

drag 하고
TXT 파일을 만들어 붙여 넣고
다시 COPY후 터미널 창에 붙이기

The screenshot shows the Robotis E-Manual website with the URL <https://emanual.robotis.com/docs/en/platform/turtlebot3/quick-start/>. The page title is "4-2-3. Install ROS on Remote PC". A red box highlights the "방법 2 참고" (Method 2 Reference) button. Another red box highlights the "Ubuntu 16.04 SKIP" button. The main content area is titled "3. 1. 2. Install ROS on Remote PC". It contains instructions to open the terminal with $Ctrl + Alt + T$ and enter commands one at a time. It also provides a link to the official ROS Kinetic installation guide if the above fails. Below this, there are sections for "3. 1. 3. Install Dependent ROS Packages" and "3. 1. 4. Install TurtleBot3 Packages", each with a code block showing the terminal commands.

3. 1. 2. Install ROS on Remote PC

Open the terminal with $Ctrl + Alt + T$ and enter below commands one at a time.
In order to check the details of the easy installation script, please refer to the [script file](#).

```
$ sudo apt-get update  
$ sudo apt-get upgrade  
$ wget https://raw.githubusercontent.com/ROBOTIS-GIT/robotis_tools/master/install_ros_kinetic.sh  
$ chmod 755 ./install_ros_kinetic.sh  
$ bash ./install_ros_kinetic.sh
```

If the above installation fails, please refer to the [official ROS Kinetic installation guide](#).

3. 1. 3. Install Dependent ROS Packages

```
$ sudo apt-get install ros-kinetic-joy ros-kinetic-teleop-twist-joy  
ros-kinetic-teleop-twist-keyboard ros-kinetic-laser-proc  
ros-kinetic-rgbd-launch ros-kinetic-depthimage-to-laserscan  
ros-kinetic-rosserial-arduino ros-kinetic-rosserial-python  
ros-kinetic-rosserial-server ros-kinetic-rosserial-client  
ros-kinetic-rosserial-msgs ros-kinetic-amcl ros-kinetic-map-server  
ros-kinetic-move-base ros-kinetic-urdf ros-kinetic-xacro  
ros-kinetic-compressed-image-transport ros-kinetic-rqt  
ros-kinetic-gmapping ros-kinetic-navigation ros-kinetic-interactive-markers
```

3. 1. 4. Install TurtleBot3 Packages

Install TurtleBot3 via Debian Packages.

```
$ sudo apt-get install ros-kinetic-dynamixel-sdk  
$ sudo apt-get install ros-kinetic-turtlebot3-msgs  
$ sudo apt-get install ros-kinetic-turtlebot3
```

4. ROS 개발환경 구축

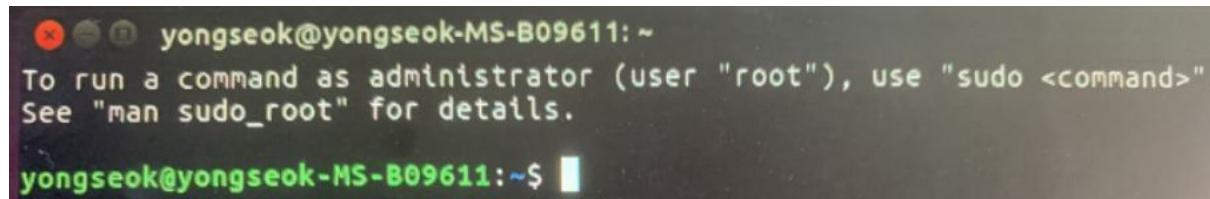
방법 2 참고

Ubuntu 16.04 SKIP

Ref. <https://emanual.robotis.com/docs/en/platform/turtlebot3/quick-start/>

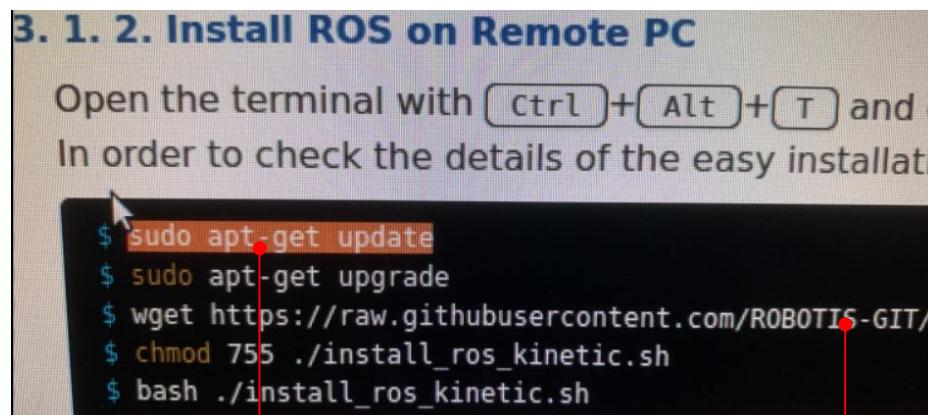
방법 2 참고: Install ROS Kinetic

: Open the terminal with **CTRL + ALT + T**



```
yongseok@yongseok-MS-B09611:~  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
yongseok@yongseok-MS-B09611:~$
```

: <https://emanual.robotis.com/docs/en/platform/turtlebot3/quick-start>에서



3. 1. 2. Install ROS on Remote PC

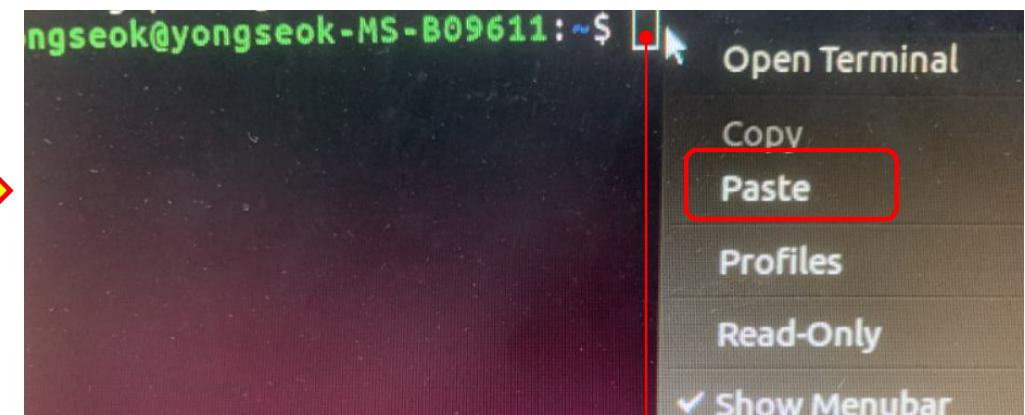
Open the terminal with **ctrl + Alt + T** and enter the following command.

In order to check the details of the easy installation, click **here**.

```
$ sudo apt-get update  
$ sudo apt-get upgrade  
$ wget https://raw.githubusercontent.com/ROBOTIS-GIT/  
$ chmod 755 ./install_ros_kinetic.sh  
$ bash ./install_ros_kinetic.sh
```

drag 하고 copy

Copy시 주의 : 전체 copy할것



우 click → paste click

또는 Ctl + Shift + V

4. ROS 개발환경 구축

방법 2 참고

Ubuntu 16.04 SKIP

Ref. <https://emanual.robotis.com/docs/en/platform/turtlebot3/quick-start/>

: sudo apt-get update

```
yongseok@yongseok-MS-B09611:~  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
  
yongseok@yongseok-MS-B09611:~$ sudo apt-get update  
[sudo] password for yongseok:  
Hit:1 http://kr.archive.ubuntu.com/ubuntu xenial InRelease  
Reading package lists... Done  
yongseok@yongseok-MS-B09611:~$
```

- 입력 후 enter
- 1234 enter(입력 안보임)
- update 화면(완료 확인할 것)

: sudo apt-get upgrade

```
yongseok@yongseok-MS-B09611:~  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
  
yongseok@yongseok-MS-B09611:~$ sudo apt-get update  
[sudo] password for yongseok:  
Hit:1 http://kr.archive.ubuntu.com/ubuntu xenial InRelease  
Reading package lists... Done  
yongseok@yongseok-MS-B09611:~$ sudo apt-get upgrade  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
Calculating upgrade... Done  
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.  
yongseok@yongseok-MS-B09611:~$
```

- 입력 후 enter
- upgrade 화면(완료 확인할 것)

4. ROS 개발환경 구축

방법 2 참고

Ubuntu 16.04 SKIP

Ref. <https://emanual.robotis.com/docs/en/platform/turtlebot3/quick-start/>

: 위에 동일한 방법으로

: wget https://raw.githubusercontent.com/ROBOTIS-GIT/robotis_tools/master/install_ros_kinetic.sh

```
yongseok@yongseok-MS-B09611:~$ wget https://raw.githubusercontent.com/ROBOTIS-GIT/robotis_tools/master/install_ros_kinetic.sh
--2021-09-13 19:25:06-- https://raw.githubusercontent.com/ROBOTIS-GIT/robotis_tools/master/install_ros_kinetic.sh
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.108.133, 185.199.110.133, 185.199.111.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.108.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2981 (2.9K) [text/plain]
Saving to: 'install_ros_kinetic.sh'

install_ros_kinetic 100%[=====] 2.91K --.-KB/s in 0s

2021-09-13 19:25:06 (15.2 MB/s) - 'install_ros_kinetic.sh' saved [2981/2981]
```

입력 후
enter

완료 확인

→ 주의: robotis emanual에서 제대로 copy 하기

```
$ apt-get upgrade
$ wget https://raw.githubusercontent.com/ROBOTIS-GIT/robotis_tools/master/install_ros_kinetic.sh
$ chmod 755 ./install_ros_kinetic.sh
```

화면에 보이는 부분 외에 우측까지
주소가 있음

: chmod 755 ./install_ros_kinetic.sh

```
yongseok@yongseok-MS-B09611:~$ chmod 755 ./install_ros_kinetic.sh
```

입력 후
enter

4. ROS 개발환경 구축

방법 2 참고

Ubuntu 16.04 SKIP

Ref. <https://emanual.robotis.com/docs/en/platform/turtlebot3/quick-start/>

: 위에 동일한 방법으로

: bash ./install_ros_kinetic.sh

```
yongseok@yongseok-MS-B09611:~$ bash ./install_ros_kinetic.sh
[Note] Target OS version >> Ubuntu 16.04.x (xenial) or Linux Mint 18.x
[Note] Target ROS version >> ROS Kinetic Kame
[Note] Catkin workspace >> /home/yongseok/catkin_ws
PRESS [ENTER] TO CONTINUE THE INSTALLATION
IF YOU WANT TO CANCEL, PRESS [CTRL] + [C]
[Set the target OS, ROS version and name of catkin workspace]
[Update the package lists and upgrade them]
Hit:1 http://kr.archive.ubuntu.com/ubuntu xenial InRelease
Reading package lists... Done
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

~ 화면 생략

```
[Add the ROS repository]
[Download the ROS keys]
./install_ros_kinetic.sh: line 35: curl: command not found
gpg: no valid OpenPGP data found.
[Check the ROS keys]
[Failed to receive the ROS key, aborts the installation]
yongseok@yongseok-MS-B09611:~$
```

• 입력 후 enter

version 확인

• enter

우분투 재설치 하기

ros 설치 완료_20211007_debugging.TXT 처
럼 하기



C:\2021_lecture\W
ROS 프로그래밍

ROS Key 오류

ROS Key 성공 화면

```
-- Generating done
-- Build files have been written to: /home/dabin/catkin_ws/build
#####
##### Running command: "make -j2 -l2" in "/home/dabin/catkin_ws/build"
#####
[Set the ROS environment]
[Complete!!!]
dabin@dabin:~$
```

→ ROS Key(new version)를 설치하지 못했다면, 아래 실행

```
sudo apt-key adv --keyserver 'hkp://keyserver.ubuntu.com:80' --recv-key
C1CF6E31E6BADE8868B172B4F42ED6FBAB17C654
```

4. ROS 개발환경 구축

Ubuntu 16.04 SKIP

Ref. <https://emanual.robotis.com/docs/en/platform/turtlebot3/quick-start/>

(4) Install Dependent ROS Packages (종속 package)

: <https://emanual.robotis.com/docs/en/platform/turtlebot3/quick-start>에서

3. 1. 3. Install Dependent ROS Packages

```
$ sudo apt-get install ros-kinetic-joy ros-kinetic-teleop-twist-joy  
ros-kinetic-teleop-twist-keyboard ros-kinetic-laser-proc  
ros-kinetic-rgbd-launch ros-kinetic-depthimage-to-laserscan  
ros-kinetic-rosserial-arduino ros-kinetic-rosserial-python  
ros-kinetic-rosserial-server ros-kinetic-rosserial-client  
ros-kinetic-rosserial-msgs ros-kinetic-amcl ros-kinetic-map-server  
ros-kinetic-move-base ros-kinetic-urdf ros-kinetic-xacro  
ros-kinetic-compressed-image-transport ros-kinetic-rqt*  
ros-kinetic-gmapping ros-kinetic-navigation ros-kinetic-interactive-markers
```

Copy후 terminal창에서 실행

한 칸 띄기

```
$ sudo apt-get install ros-kinetic-joy ros-kinetic-teleop-twist-joy ros-kinetic-teleop-twist-keyboard ros-kinetic-laser-proc  
ros-kinetic-rgbd-launch ros-kinetic-depthimage-to-laserscan ros-kinetic-rosserial-arduino ros-kinetic-rosserial-python ros-  
kinetic-rosserial-server ros-kinetic-rosserial-client ros-kinetic-rosserial-msg ros-kinetic-amcl ros-kinetic-map-server ros-  
kinetic-move-base ros-kinetic-urdf ros-kinetic-xacro ros-kinetic-compressed-image-transport ros-kinetic-rqt-image-view  
ros-kinetic-gmapping ros-kinetic-navigation ros-kinetic-interactive-markers
```

4. ROS 개발환경 구축

Ubuntu 16.04 SKIP

Ref. <https://emanual.robotis.com/docs/en/platform/turtlebot3/quick-start/>



```
yongseok@yongseok:~$ gedit ~/.bashrc
yongseok@yongseok:~$ sudo apt-get install ros-kinetic-joy ros-kinetic-teleop-twist-joy ros-kinetic-teleop-twist-keyboard ros-kinetic-laser-proc ros-kinetic-rgbd-launch ros-kinetic-depthimage-to-laserscan ros-kinetic-rosserial-arduino ros-kinetic-rosserial-python ros-kinetic-rosserial-server ros-kinetic-rosserial-client ros-kinetic-rosserial-msg ros-kinetic-amcl ros-kinetic-map-server ros-kinetic-move-base ros-kinetic-urdf ros-kinetic-xacro ros-kinetic-compressed-image-transport ros-kinetic-rqt-image-view ros-kinetic-gmapping ros-kinetic-navigation ros-kinetic-interactive-markers
[sudo] password for yongseok:
Reading package lists... Done
Building dependency tree
Reading state information... Done
E: Unable to locate package ros-kinetic-rosserial-msg
yongseok@yongseok:~$
```

} 입력 후 실행

4. ROS 개발환경 구축

Ubuntu 16.04 SKIP

Ref. <https://emanual.robotis.com/docs/en/platform/turtlebot3/quick-start/>

(5) Install TurtleBot3 Packages

: Install TurtleBot3 via Debian Package

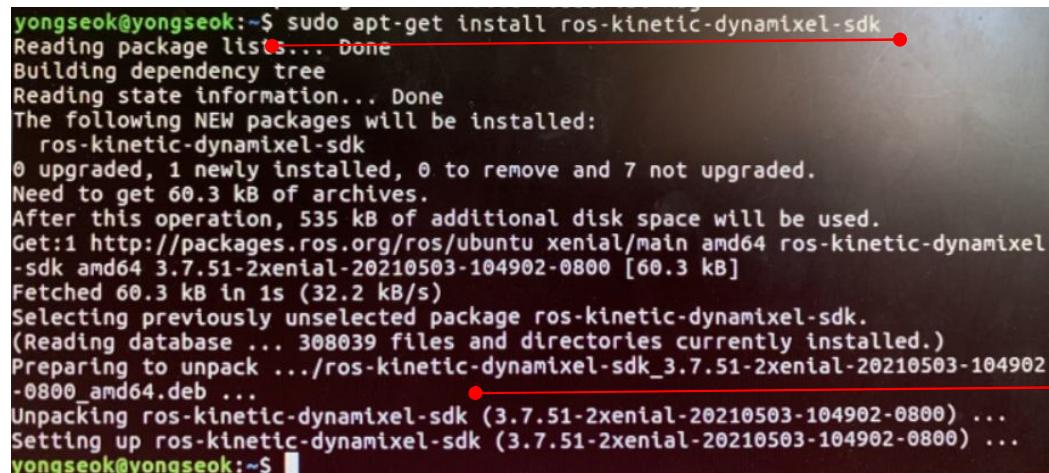
: <https://emanual.robotis.com/docs/en/platform/turtlebot3/quick-start>에서

3. 1. 4. Install TurtleBot3 Packages

Install TurtleBot3 via Debian Packages.

```
$ sudo apt-get install ros-kinetic-dynamixel-sdk  
$ sudo apt-get install ros-kinetic-turtlebot3-msgs  
$ sudo apt-get install ros-kinetic-turtlebot3
```

\$ sudo apt-get install ros-kinetic-dynamixel-sdk



```
yongseok@yongseok:~$ sudo apt-get install ros-kinetic-dynamixel-sdk  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following NEW packages will be installed:  
  ros-kinetic-dynamixel-sdk  
0 upgraded, 1 newly installed, 0 to remove and 7 not upgraded.  
Need to get 60.3 kB of archives.  
After this operation, 535 kB of additional disk space will be used.  
Get:1 http://packages.ros.org/ros/ubuntu xenial/main amd64 ros-kinetic-dynamixel-  
-sdk amd64 3.7.51-2xenial-20210503-104902-0800 [60.3 kB]  
Fetched 60.3 kB in 1s (32.2 kB/s)  
Selecting previously unselected package ros-kinetic-dynamixel-sdk.  
(Reading database ... 308039 files and directories currently installed.)  
Preparing to unpack .../ros-kinetic-dynamixel-sdk_3.7.51-2xenial-20210503-104902-  
-0800_amd64.deb ...  
Unpacking ros-kinetic-dynamixel-sdk (3.7.51-2xenial-20210503-104902-0800) ...  
Setting up ros-kinetic-dynamixel-sdk (3.7.51-2xenial-20210503-104902-0800) ...  
yongseok@yongseok:~$
```

● kinetic-dynamixel 확인

4. ROS 개발환경 구축

Ubuntu 16.04 SKIP

Ref. <https://emanual.robotis.com/docs/en/platform/turtlebot3/quick-start/>

\$ sudo apt-get install ros-kinetic-turtlebot3-msgs

```
Setting up ros_kinetic_dynamixel-sok (3.7.51-2xenial-20210503-104902-0800) ...
yongseok@yongseok:~$ sudo apt-get install ros-kinetic-turtlebot3-msgs
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  ros-kinetic-turtlebot3-msgs
0 upgraded, 1 newly installed, 0 to remove and 7 not upgraded.
Need to get 22.3 kB of archives.
After this operation, 200 kB of additional disk space will be used.
Get:1 http://packages.ros.org/ros/ubuntu xenial/main amd64 ros-kinetic-turtlebot
3-msgs amd64 1.0.1-1xenial-20210503-100300-0800 [22.3 kB]
Fetched 22.3 kB in 3s (6,939 B/s)
Selecting previously unselected package ros-kinetic-turtlebot3-msgs.
(Reading database ... 308094 files and directories currently installed.)
Preparing to unpack .../ros-kinetic-turtlebot3-msgs_1.0.1-1xenial-20210503-10030
0-0800_amd64.deb ...
Unpacking ros-kinetic-turtlebot3-msgs (1.0.1-1xenial-20210503-100300-0800) ...
Setting up ros-kinetic-turtlebot3-msgs (1.0.1-1xenial-20210503-100300-0800) ...
yongseok@yongseok:~$ █
```

● kinetic-turtlebot3-msgs

확인

4. ROS 개발환경 구축

Ubuntu 16.04 SKIP

Ref. <https://emanual.robotis.com/docs/en/platform/turtlebot3/quick-start/>

```
$ sudo apt-get install ros-kinetic-turtlebot3
```

```
yongseok@yongseok:~$ sudo apt-get install ros-kinetic-turtlebot3
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
 libasound2-dev libbullet-dev libbulletsoftbody2.83.6 libcaca-dev
 libpulse-dev libsdl-image1.2 libsdl-image1.2-dev libsdl1.2-dev
 libsdl1.2debian libssl2-dev ros-kinetic-amcl
 ros-kinetic-base-local-planner ros-kinetic-clear-costmap-recovery
 ros-kinetic-costmap-2d ros-kinetic-hls-lfcd-lds-driver
 ros-kinetic-map-server ros-kinetic-navigation
0 upgraded, 14 newly installed, 0 to remove and 0 not upgraded.
```

~ 화면 생략

```
) ...  
Setting up ros-kinetic-turtlebot3-slam (1.2.5-1xenial-20210503-155605-0800) ...  
Setting up ros-kinetic-turtlebot3 (1.2.5-1xenial-20210503-160735-0800) ... I  
Processing triggers for libc-bin (2.23-0ubuntu11.3) ...  
yongseok@yongseok:~$ █
```

emanual.robotis.com/docs/en/platform/turtlebot3/quick-start/ 에서 1.1.4 까지 진행

4. ROS 개발환경 구축

Ubuntu 16.04 SKIP

Ref. <https://emanual.robotis.com/docs/en/platform/turtlebot3/quick-start/>

(6) ROS 환경변수 설정 :

```
$ source /opt/ros/kinetic/setup.bash
```

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

: 위와 같은 ROS 설치 과정에서 사용된 환경설정 파일을

매번 불러오기 번거롭기에 bashrc 파일에 환경변수 설정

```
$ gedit ~/.bashrc
```



```
# ~/.bashrc: executed by bash(1) for non-login shells.
# see /usr/share/doc/bash/examples/startup-files (in the package bash-doc)
# for examples

# If not running interactively, don't do anything
case $- in
    *t* );
    *) return;;
esac

# don't put duplicate lines or lines starting with space in the history.
# See bash(1) for more options
HISTCONTROL=ignoreboth

# append to the history file, don't overwrite it
shopt -s histappend

# for setting history length see HISTSIZE and HISTFILESIZE in bash(1)
HISTSIZE=1000
HISTFILESIZE=2000

# check the window size after each command and, if necessary,
# update the values of LINES and COLUMNS.
shopt -s checkwinsize

# If set, the pattern "*" used in a pathname expansion context will
# match all files and zero or more directories and subdirectories.
#shopt -s globstar

# make less more friendly for non-text input files, see lesspipe(1)
[ -x /usr/bin/lesspipe ] && eval "$(SHELL=/bin/sh lesspipe)"

# set variable identifying the chroot you work in (used in the prompt below)
if [ -z "$debian_chroot" ] && [ -r /etc/debian_chroot ]; then
    debian_chroot=$(cat /etc/debian_chroot)
fi

# set a fancy prompt (non-color, unless we know we "want" color)
case "$TERM" in
    xterm-color|*-256color) color_prompt=yes;;
esac

# uncomment for a colored prompt, if the terminal has the capability; turned
# off by default to not distract the user: the focus is in a terminal window
# should be on the output of commands, not on the prompt
#force_color_prompt=yes
```

```
# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile).
# sources /etc/bash.bashrc.

if ! shopt -oq posix; then
    if [ -f /usr/share/bash-completion/bash_completion ]; then
        . /usr/share/bash-completion/bash_completion
    elif [ -f /etc/bash_completion ]; then
        . /etc/bash_completion
    fi
fi

alias eb='nano ~/.bashrc'
alias sb='source ~/.bashrc'
alias gs='git status'
alias gp='git pull'
alias cw='cd ~/catkin_ws'
alias cs='cd ~/catkin_ws/src'
alias cm='cd ~/catkin_ws && catkin_make'
source /opt/ros/kinetic/setup.bash
source ~/catkin_ws/devel/setup.bash
export ROS_MASTER_URI=http://localhost:11311
export ROS_HOSTNAME=localhost
source /opt/ros/kinetic/setup.bash
source /opt/ros/kinetic/setup.bash
alias eb='nano ~/.bashrc'
alias sb='source ~/.bashrc'
alias gs='git status'
alias gp='git pull'
alias cw='cd ~/catkin_ws'
alias cs='cd ~/catkin_ws/src'
alias cm='cd ~/catkin_ws && catkin_make'
source /opt/ros/kinetic/setup.bash
source ~/catkin_ws/devel/setup.bash
export ROS_MASTER_URI=http://localhost:11311
export ROS_HOSTNAME=localhost
alias eb='nano ~/.bashrc'
alias sb='source ~/.bashrc'
alias gs='git status'
alias gp='git pull'
alias cw='cd ~/catkin_ws'
alias cs='cd ~/catkin_ws/src'
alias cm='cd ~/catkin_ws && catkin_make'
source /opt/ros/kinetic/setup.bash
source ~/catkin_ws/devel/setup.bash
export ROS_MASTER_URI=http://localhost:11311
export ROS_HOSTNAME=localhost
alias eb='nano ~/.bashrc'
alias sb='source ~/.bashrc'
alias gs='git status'
alias gp='git pull'
alias cw='cd ~/catkin_ws'
alias cs='cd ~/catkin_ws/src'
alias cm='cd ~/catkin_ws && catkin_make'
source /opt/ros/kinetic/setup.bash
source ~/catkin_ws/devel/setup.bash
export ROS_MASTER_URI=http://localhost:11311
export ROS_HOSTNAME=localhost
```

확인

sh ▾ Tab Width: 8

4. ROS 개발환경 구축

방법 2에서도 동일

Ubuntu 16.04 SKIP

Ref. ROS 로봇 프로그래밍(Ruby paper) 3-1 ROS 설치

환경설정 파일 불러오기

: ROS_ROOT, ROS_PACKAGE_PATH 등의 환경변수 정의된 환경설정 파일 불러오기

→ \$ source /opt/ros/kinetic/setup.bash

```
yongseok@yongseok:~$ source /opt/ros/kinetic/setup.bash
```

bash : 파일을 읽고 실행 명령어

작업 폴더 생성 및 초기화

: ROS에서는 catkin이라는 ROS 전용 빌드 시스템 사용

: catkin 사용하려면 catkin 작업 폴더를 **생성하고 초기화해야 함**

→ \$ mkdir -p ~/catkin_ws/src

\$ cd ~/catkin_ws/src

\$ catkin_init_workspace

```
yongseok@yongseok:~$ mkdir -p ~/catkin_ws/src
yongseok@yongseok:~$ cd ~/catkin_ws/src
yongseok@yongseok:~/catkin_ws/src$
```

• CMakeLists.txt 파일 생성

• directory 확인

```
yongseok@yongseok:~/catkin_ws/src$ catkin_init_workspace
File "/home/yongseok/catkin_ws/src/CMakeLists.txt" already exists
yongseok@yongseok:~/catkin_ws/src$
```

-CMakeLists.txt 파일 확인

catkin_init_workspace : 작업공간 초기화를 의미.

위의 명령들을 실행하고 난 뒤에 작업공간이 비어 있음(src 폴더 안에 CMakeLists.txt만 존재)

4. ROS 개발환경 구축

Ubuntu 16.04 SKIP

Ref. ROS 로봇 프로그래밍(Ruby paper) 3-1 ROS 설치

build

: 경로 확인 (build 경로)

→ \$ cd ~/catkin_ws/

\$ catkin_make



\$ ls

```
[100%] Built target turtlebot3_example_generate_messages
yongseok@yongseok:~/catkin_ws$ ls
build  devel  install_ros_kinetic.sh  src
yongseok@yongseok:~/catkin_ws$
```



우리가 만든 src 폴더 확인

그리고 build 와 devel 폴더 생성

→ catkin 빌드 파일은 build 폴더에 저장

→ 빌드 후 실행 파일은 devel 폴더에 저장

```
yongseok@yongseok:~/catkin_ws/src$ catkin_init_workspace
File "/home/yongseok/catkin_ws/src/CMakeLists.txt" already existsyongseok@yongseok:~/catkin_ws$ cd ~ /catkin_ws/
yongseok@yongseok:~/catkin_ws$ catkin_make
Base path: /home/yongseok/catkin_ws
Source space: /home/yongseok/catkin_ws/src
Build space: /home/yongseok/catkin_ws/build
Devel space: /home/yongseok/catkin_ws/devel
Install space: /home/yongseok/catkin_ws/install
#####
##### Running command: "make cmake_check_build_system" in "/home/yongseok/catkin_ws/build"
```

만일 이 내용이 없다면 catkin_make가 불안전
(source code와 환경설정 build 안 된 것임)
→ ubuntu 재설치
(roscore는 설치되었어도 문제가 내재)

```
0%] Built target std_msgs_generate_messages_py
0%] Built target _turtlebot3_msgs_generate_messages_check_deps_SensorState
0%] Built target _turtlebot3_msgs_generate_messages_check_deps_Sound
0%] Built target std_msgs_generate_messages_cpp
0%] Built target turtlebot3_msgs_generate_messages_check_deps_VersionInfo
```

-실행 내용 생략

```
[ 84%] Built target turtlebot3_example_generate_messages_nodejs
[ 96%] Built target turtlebot3_example_generate_messages_lisp
[ 96%] Built target turtlebot3_msgs_generate_messages
[100%] Built target turtlebot3_diagnostics
[100%] Built target turtlebot3_example_generate_messages 100% build 확인
yongseok@yongseok:~/catkin_ws$
```



4. ROS 개발환경 구축

Ubuntu 16.04 SKIP

Ref. ROS 로봇 프로그래밍(Ruby paper) 3-1 ROS 설치

catkin 빌드 시스템과 관련된 환경 파일 적용하기

→ \$ source ~/catkin_ws/devel/setup.bash ★

```
yongseok@yongseok:~/catkin_ws$ ls  
build  devel  install_ros_kinetic.sh  src  
yongseok@yongseok:~/catkin_ws$ source ~/catkin_ws/devel/setup.bash  
yongseok@yongseok:~/catkin_ws$ █
```



build과정으로 생성된 devel파일 안의 setup.bash를 실행하여

- 그 작업공간이 ros 환경의 최상위에 overlay되도록 쉘에 등록
- setup.bash를 실행하면 environment variables가 setting되고
이것이 시스템이 필요한 모든 package와 library 등을 찾게 도와줌

4. ROS 개발환경 구축

Ubuntu 16.04 SKIP

Ref. ROS 로봇 프로그래밍(Ruby paper) 3-1 ROS 설치

★ 최종 테스트

→ \$ roscore

: ROS master 구동하는 명령어



오류가 발생한다면



ROS 설치에 문제 있음

재설치 필요



문제가 지속 발생 경우

UBUNTU 재설치

```
yongseok@yongseok:~$ roscore
... logging to /home/yongseok/.ros/log/83cce24c-1e9c-11ec-b5be-f40669f0af37/roslaunch-yongseok-21558.log
Checking log directory for disk usage. This may take awhile.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://localhost:36691/
ros_comm version 1.12.17

SUMMARY
=====

PARAMETERS
* /rosdistro: kinetic
* /rosversion: 1.12.17 → Version Kinetic 확인

NODES

auto-starting new master
process[master]: started with pid [21569]
ROS_MASTER_URI=http://localhost:11311/

setting /run_id to 83cce24c-1e9c-11ec-b5be-f40669f0af37
process[rosout-1]: started with pid [21582]
started core service [/rosout] → 이 상태에서 끝내려면 Ctrl+c ★
```

4. ROS 개발환경 구축

Ubuntu 16.04 SKIP

Ref. <https://emanual.robotis.com/docs/en/platform/turtlebot3/quick-start/>



최종 정리 ROS 설치

ros 설치 완료_20211007_debugging.txt

설치시 와이파이 상태 반드시 지속적으로 확인하기
(youtube 등을 들어가면서)



build

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



CTRL + Shift +Q 열려있는 Terminal 창 닫기

다시 새로운 창 열기

CTRL + ALT + T 를 사용해 터미널 창 열기



ROS master 구동

```
$ roscore  
: kinetic version 확인하기  
: 끝내려면 Ctrl+c
```

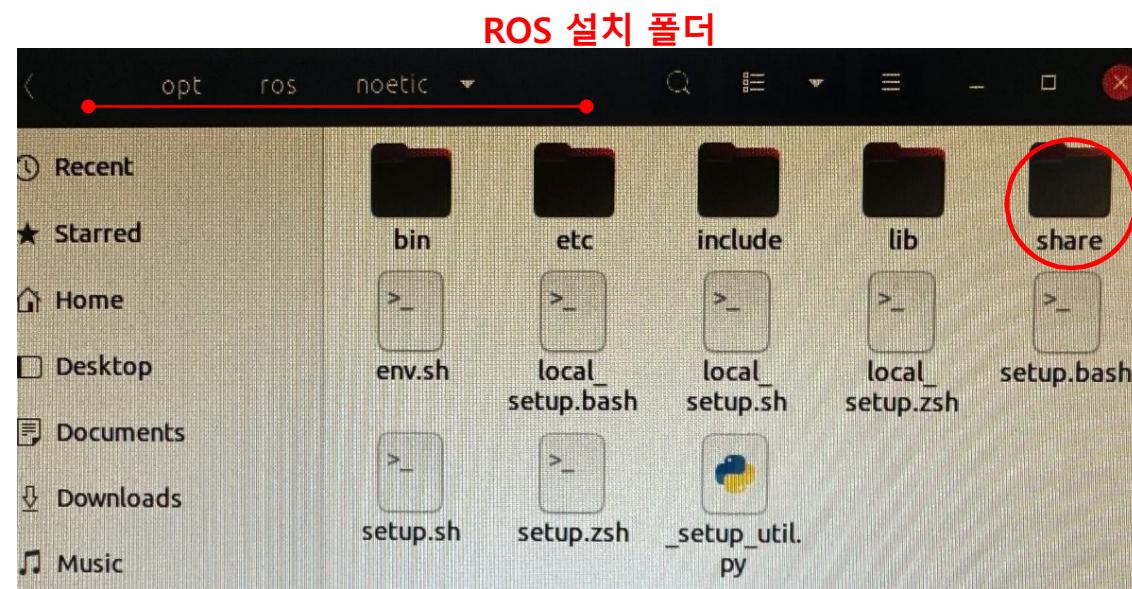
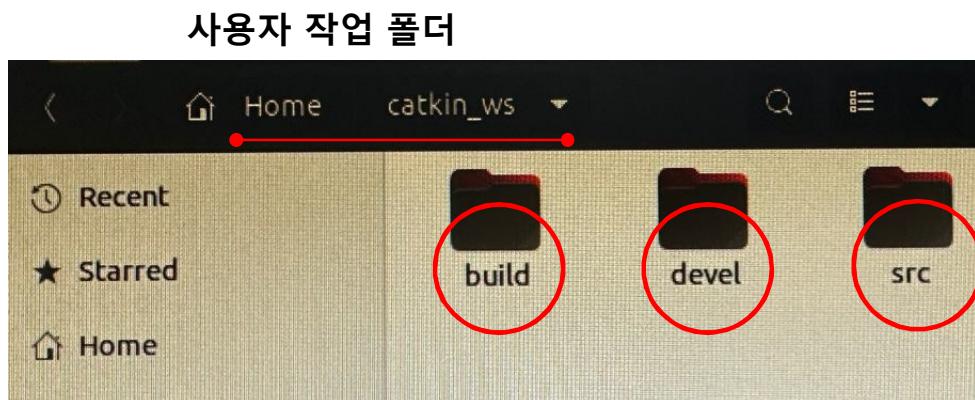


오류가 발생한다면
ROS 설치에 문제 있음
재설치 필요

4. ROS 개발환경 구축

[파일 구성]

- : ROS 파일 시스템은 설치 폴더와 사용자 작업 폴더로 구분
 - ROS 설치 폴더는 ROS를 설치하면 /opt 폴더에 ros 폴더 생성
→ ROS 설치 폴더 경로 /opt/ros/noetic
 - ros 폴더에 roscore를 포함한 핵심 유ти리티와 rqt, Rviz, 로봇 관련 라이브러리, 시뮬레이션, 내비게이션 등 설치
- 사용자 작업 폴더는 사용자 폴더인 ~/catkin_ws 사용

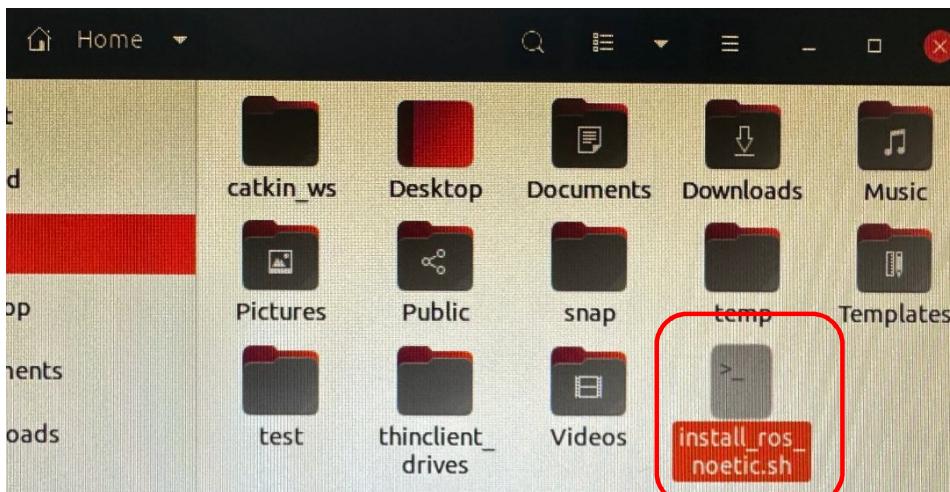


4. ROS 개발환경 구축

[파일 구성]

: 사용자 작업 폴더는 사용자 폴더인

: ~/catkin_ws == Home/catkin_ws



Install_ros_noetic.sh

```
1#!/bin/bash
2# Apache License 2.0
3# Copyright (c) 2020, ROBOTIS CO., LTD.
4
5echo ""
6echo "[Note] Target OS version >>> Ubuntu 20.04.x (Focal Fossa) or Linux Mint 21.x"
7echo "[Note] Target ROS version >>> ROS Noetic Nijjemys"
8echo "[Note] Catkin workspace >>> $HOME/catkin_ws"
9echo ""
10echo "PRESS [ENTER] TO CONTINUE THE INSTALLATION"
11echo "IF YOU WANT TO CANCEL, PRESS [CTRL] + [C]"
12read
13
14echo "[Set the target OS, ROS version and name of catkin workspace]"
15name_os_version=${name_os_version:="focal"}
16name_ros_version=${name_ros_version:="noetic"}
17name_catkin_workspace=${name_catkin_workspace:="catkin_ws"}
18
19echo "[Update the package lists]"
20sudo apt update -y
21
22echo "[Install build environment, the chrony, ntpdate and set the ntpdate]"
23sudo apt install -y chrony ntpdate curl build-essential
24sudo ntpdate ntp.ubuntu.com
25
26echo "[Add the ROS repository]"
27if [ ! -e /etc/apt/sources.list.d/ros-latest.list ]; then
28    sudo sh -c "echo \\"deb http://packages.ros.org/ros/ubuntu ${name_os_version} main\\" > /etc/
sources.list.d/ros-latest.list"
29fi
30
31echo "[Download the ROS keys]"
32roskey= apt-key list | grep "Open Robotics"
33if [ -z "$roskey" ]; then
34    curl -s https://raw.githubusercontent.com/ros/rosdistro/master/ros.asc | sudo apt-key add -
35
36
37echo "[Check the ROS keys]"
38roskey=apt-key list | grep "Open Robotics"
39if [ -n "$roskey" ]; then
40    echo "[ROS key exists in the list]"
41else
```

4. ROS 개발환경 구축

[파일 구성]

: 사용자 작업 폴더는 사용자 폴더인

: ~/catkin_ws == Home/catkin_ws

```
48 echo "[Install ros-desktop-full version of Noetic"
49 sudo apt install -y ros-$name_ros_version-desktop-full
51
52 echo "[Install RQT & Gazebo]"
53 sudo apt install -y ros-$name_ros_version-rqt-* ros-$name_ros_version-gazebo-*
54
55 echo "[Environment setup and getting rosinstall]"
56 source /opt/ros/$name_ros_version/setup.sh
57 sudo apt install -y python3-rosinstall python3-rosinstall-generator python3-wstool build-essential git
58
59 echo "[Install rosdep and Update]"
60 sudo apt install python3-rosdep
61
62 echo "[Initialize rosdep and Update]"
63 sudo sh -c "rosdep init"
64 rosdep update
65
66 echo "[Make the catkin workspace and test the catkin_make]"
67 mkdir -p $HOME/$name_catkin_workspace/src
68 cd $HOME/$name_catkin_workspace/src
69 catkin_init_workspace
70 cd $HOME/$name_catkin_workspace
71 catkin_make
72
73 echo "[Set the ROS environment]"
74 sh -c "echo \"alias eb='nano ~/.bashrc'\" >> ~/.bashrc"
75 sh -c "echo \"alias sb='source ~/.bashrc'\" >> ~/.bashrc"
76 sh -c "echo \"alias gs='git status'\" >> ~/.bashrc"
77 sh -c "echo \"alias gp='git pull'\" >> ~/.bashrc"
78 sh -c "echo \"alias cw='cd ~/$name_catkin_workspace'\" >> ~/.bashrc"
79 sh -c "echo \"alias cs='cd ~/$name_catkin_workspace/src'\" >> ~/.bashrc"
80 sh -c "echo \"alias cm='cd ~/$name_catkin_workspace && catkin_make'\" >> ~/.bashrc"
81
82 sh -c "echo \"source /opt/ros/$name_ros_version/setup.bash\" >> ~/.bashrc"
83 sh -c "echo \"source ~/catkin_ws/devel/setup.bash\" >> ~/.bashrc"
84
85 sh -c "echo \"export ROS_MASTER_URI=http://localhost:11311\" >> ~/.bashrc"
86 sh -c "echo \"export ROS_HOSTNAME=localhost\" >> ~/.bashrc"
87
88 source $HOME/.bashrc
89
90 echo "[Complete!!!]"
```

Install_ros_noetic.sh

4. ROS 개발환경 구축

[파일 구성]

: Home/catkin_ws/devel

: 우리가 만든 src 폴더 확인

그리고 build 와 devel 폴더 생성

: catkin 빌드 파일은 build 폴더 저장

: 빌드 후 실행 파일은 devel 폴더 저장

The terminal window shows the contents of the setup.bash script:

```
1#!/usr/bin/env bash
2# generated from catkin/cmake/templates/setup.bash.in
3
4CATKIN_SHELL=bash
5
6# source setup.sh from same directory as this file
7_CATKIN_SETUP_DIR=$(builtin cd "`dirname "${BASH_SOURCE[0]}`" > /dev/null && pwd)
8 . "$_CATKIN_SETUP_DIR/setup.sh"
```

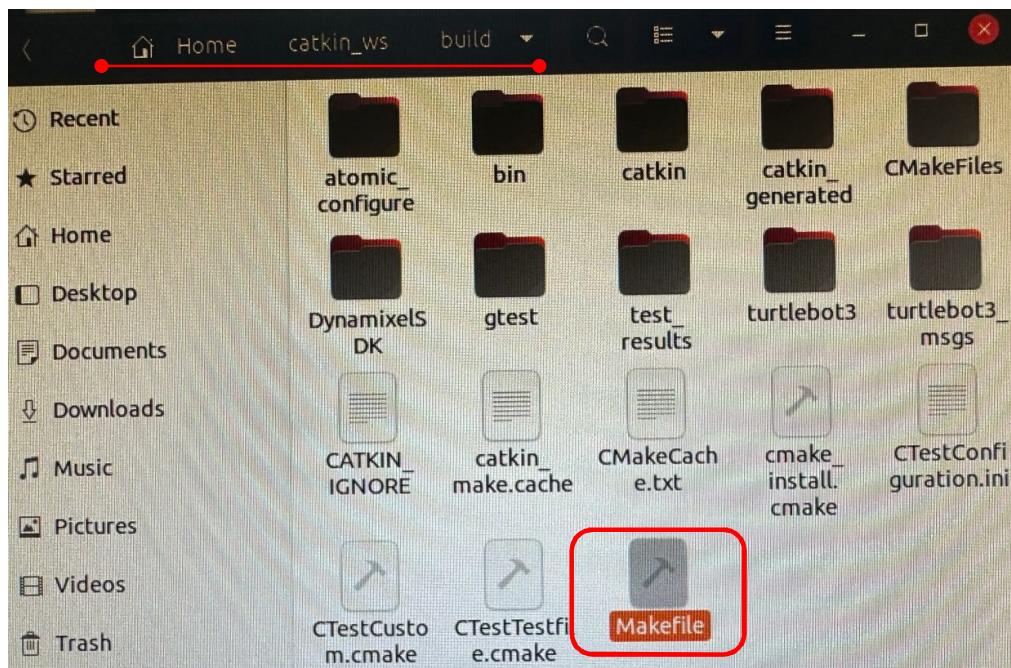
The file explorer shows the directory structure under 'catkin_ws/devel':

- Recent
- Starred
- Home
- Desktop
- Documents
- Downloads
- Music
- include
- lib
- share
- cmake.lock
- env.sh
- local_setup.bash
- local_setup.sh
- local_setup.zsh
- setup.bash
- setup.sh
- Setup.bash
- setup.zsh
- _setup_util.py

4. ROS 개발환경 구축

[파일 구성]

: Home/catkin_ws/build



Makefile

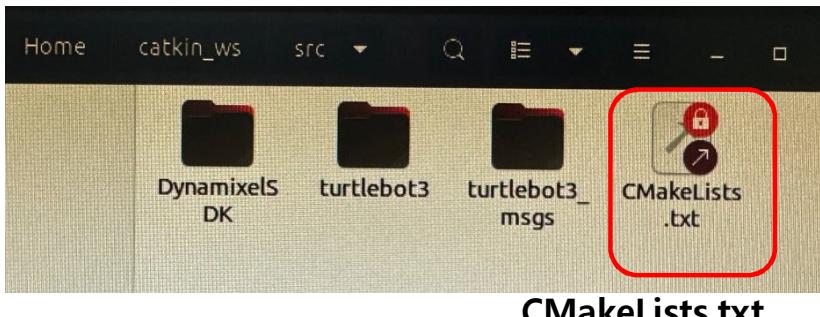
The screenshot shows a code editor window titled 'Makefile'. The content of the file is as follows:

```
1 # CMAKE generated file: DO NOT EDIT!
2 # Generated by "Unix Makefiles" Generator, CMake Version 3.16
3
4 # Default target executed when no arguments are given to make.
5 default_target: all
6
7 .PHONY : default_target
8
9 # Allow only one "make -f Makefile2" at a time, but pass parallelism.
10 .NOTPARALLEL:
11
12
13 =====
14 # Special targets provided by cmake.
15
16 # Disable implicit rules so canonical targets will work.
17 .SUFFIXES:
18
19
20 # Remove some rules from gmake that .SUFFIXES does not remove.
21 SUFFIXES =
22
23 .SUFFIXES: .hpx_make_needs_suffix_list
24
25
26 # Suppress display of executed commands.
27 $(VERBOSE).SILENT:
28
29
30 # A target that is always out of date.
31 cmake_force:
32
33 .PHONY : cmake_force
34
35 =====
36 # Set environment variables for the build.
37
38 # The shell in which to execute make rules.
39 SHELL = /bin/sh
40
41 # The CMake executable.
42 CMAKE_COMMAND = /usr/bin/cmake
43
44 # The command to remove a file.
45 RM = /usr/bin/cmake -E remove -f
46
47 # Escaping for special characters.
```

4. ROS 개발환경 구축

[파일 구성]

: Home/catkin_ws/src



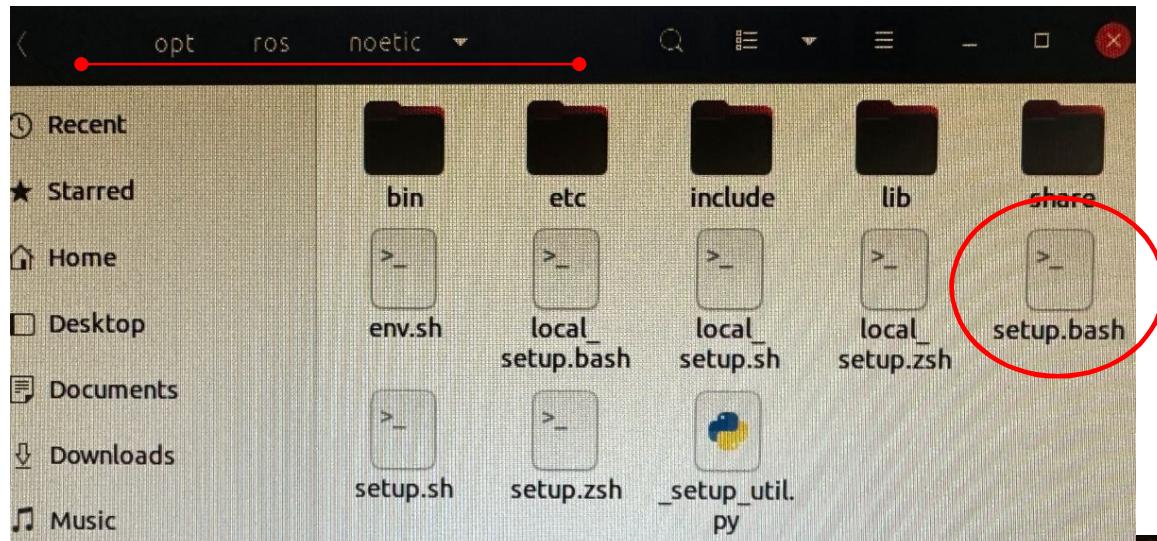
```
Open ▾ CMakeLists.txt [Read-Only]

1 # toplevel CMakeLists.txt for a catkin workspace
2 # catkin/cmake/toplevel.cmake
3
4 cmake_minimum_required(VERSION 3.0.2)
5
6 project(Project)
7
8 set(CATKIN_TOPLEVEL TRUE)
9
10 # search for catkin within the workspace
11 set( _cmd "catkin find pkg" "catkin" "${CMAKE_SOURCE_DIR}")
12 execute_process(COMMAND ${_cmd})
13 RESULT_VARIABLE _res
14 OUTPUT_VARIABLE _out
15 ERROR_VARIABLE _err
16 OUTPUT_STRIP_TRAILING_WHITESPACE
17 ERROR_STRIP_TRAILING_WHITESPACE
18 )
19 if(NOT _res EQUAL 0 AND NOT _res EQUAL 2)
20 # searching for catkin resulted in an error
21 string(REPLACE ";" " " _cmd_str "${_cmd}")
22 message(FATAL_ERROR "Search for 'catkin' in workspace failed (${_cmd_s
23 endif()
24
25 # include catkin from workspace or via find_package()
26 if(_res EQUAL 0)
27 set(catkin_EXTRAS_DIR "${CMAKE_SOURCE_DIR}/${_out}/cmake")
28 # include all.cmake without add_subdirectory to let it operate in same
29 include(${catkin_EXTRAS_DIR}/all.cmake NO_POLICY_SCOPE)
30 add_subdirectory("${_out}")
31
32 else()
33 # use either CMAKE_PREFIX_PATH explicitly passed to CMake as a command
34 # or CMAKE_PREFIX_PATH from the environment
35 if(NOT DEFINED CMAKE_PREFIX_PATH)
36 if(NOT "$ENV{CMAKE_PREFIX_PATH}" STREQUAL "")
37 if(NOT WIN32)
38 string(REPLACE ":" ";" CMAKE_PREFIX_PATH $ENV{CMAKE_PREFIX_PATH}
39 else()
40 set(CMAKE_PREFIX_PATH $ENV{CMAKE_PREFIX_PATH})
41 endif()
42 endif()
43 endif()
44
45 # list of catkin workspaces
46 set(catkin_search_path "")
47 foreach(path ${CMAKE_PREFIX_PATH})
48 if(EXISTS "${path}/.catkin")
```

4. ROS 개발환경 구축

[파일 구성]

: ROS 설치 폴더 경로 /opt/ros/noetic



The screenshot shows a gedit editor window titled 'setup.bash [Read-Only] (/opt/ros/kinetic)'. The file content is as follows:

```
#!/usr/bin/env bash
# generated from catkin/cmake/templates/setup.bash.in

CATKIN_SHELL=bash

# source setup.sh from same directory as this file
_CATKIN_SETUP_DIR=$(builtin cd `dirname "${BASH_SOURCE[0]}` > /dev/null && pwd)
. "$_CATKIN_SETUP_DIR/setup.sh"
```

4. ROS 개발환경 구축

(7) 네트워크 구성

: turtlebot은 Remote PC(미니PC)와 SBC(Single Board Computer : Raspberry Pi 4B가 원격으로 연결되어야 함

: 원격으로 연결을 위해 Wifi를 사용하여 IP 주소를 맞춰 주어야 함

: Remote PC의 IP 주소 확인

\$ ifconfig

```
yongseok@yongseok:~$ ifconfig
enp2s0    Link encap:Ethernet HWaddr 4c:cc:6a:9d:47:16
          UP BROADCAST MULTICAST MTU:1500 Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)

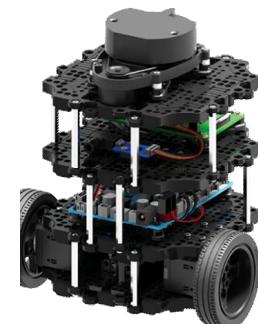
lo        Link encap:Local Loopback
          inet addr:127.0.0.1 Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
            UP LOOPBACK RUNNING MTU:65536 Metric:1
            RX packets:3638 errors:0 dropped:0 overruns:0 frame:0
            TX packets:3638 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:713589 (713.5 KB) TX bytes:713589 (713.5 KB)

wlp3s0    Link encap:Ethernet HWaddr f4:06:69:f0:af:37
         inet addr:192.168.0.15 Bcast:192.168.0.255 Mask:255.255.255.0
          inet6 addr: fe80::f406:69ff:fe0af:37 Scope:Link
            UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
            RX packets:553870 errors:0 dropped:0 overruns:0 frame:0
            TX packets:224046 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:810194447 (810.1 MB) TX bytes:23074618 (23.0 MB)
```

Remote PC (미니PC)
: ROS Master



SBC(Single Board Computer)
: Raspberry Pi 4B



WiFi

실습 환경이 변경될 때마다,
네트워크 구성은 지속적으로 변경됨

4. ROS 개발환경 구축

(7) 네트워크 구성

: 환경설정 파일에서 Remote PC IP 입력하기

\$ gedit ~/.bashrc



모든 ROS_MASTER_URI과 ROS_HOSTNAME에서

localhost를 본인 Remote PC IP로 변경하기

Export path = 새로 등록할 환경변수 주소

export ROS_MASTER_URI=http://localhost:11311

export ROS_HOSTNAME= localhost



export ROS_MASTER_URI=http://192.168.1.141:11311

export ROS_HOSTNAME=192.168.1.141



: 저장하기

: 설정 반영하기

\$ source ~/.bashrc

yongseok@yongseok:~\$ source ~/.bashrc
yongseok@yongseok:~\$

```
1 # ~/.bashrc: executed by bash(1) for non-login shells.
2 # see /usr/share/doc/bash/examples/startup-files (in the package bash-doc)
3 # for examples
4
5 # If not running interactively, don't do anything
6 case $- in
7     *i*) ;;
8     *) return;;
9 esac
10
11 # don't put duplicate lines or lines starting with space in the history.
12 # See bash(1) for more options
13 HISTCONTROL=ignoreboth
14
```

~ 화면 생략

```
118 alias eb='nano ~/.bashrc'
119 alias sb='source ~/.bashrc'
120 alias gs='git status'
121 alias gp='git pull'
122 alias cw='cd ~/catkin_ws'
123 alias cs='cd ~/catkin_ws/src'
124 alias cm='cd ~/catkin_ws && catkin_make'
125 source /opt/ros/noetic/setup.bash
126 source ~/catkin_ws/devel/setup.bash
127 export ROS_MASTER_URI=http://localhost:11311
128 export ROS_HOSTNAME=localhost
```

4. ROS 개발환경 구축

(8) Set TurtleBot3 Model Name

```
$ echo "export TURTLEBOT3_MODEL=burger" >> ~/.bashrc
```

\$ source ~/.bashrc : 환경변수 설정 반영하기

```
yongseok@yongseok:~$ echo "export TURTLEBOT3_MODEL=burger" >> ~/.bashrc
yongseok@yongseok:~$ gedit ~/.bashrc
yongseok@yongseok:~$ source ~/.bashrc
```

gedit ~/.bashrc를 실행하여

echo 명령어로 입력된 내용을 확인하기



```
source /opt/ros/kinetic/setup.bash
source ~/catkin_ws/devel/setup.bash
export ROS_MASTER_URI=http://192.168.0.15:11311
export ROS_HOSTNAME=192.168.0.15
export TURTLEBOT3_MODEL=burger
```

(10) ROS 등작 테스트 (turtlesim package)

: 새 터미널에서

\$ roscore

: 새 터미널에서

\$ rosrun turtlesim

turtlesim_node

→ turtlesim 열림

: 새 터미널에서

\$ rosrun turtlesim

turtle_teleop_key

→ → 우측으로 회전

←좌측으로 회전

↑ 전진 ↓ 후진

③ turtlesim package의 turtlesim_node 실행결과



⑤ 실행결과

→ 우측으로 회전 ←좌측으로 회전
↑ 전진 ↓ 후진

```
roscore http://192.168.0.15:11311
① roscore 실행
yongseok@yongseok:~$ roscore
... logging to /home/yongseok/.ros/log/65949316-1f81-11ec-b5be-f40669f0af37/r
oslaunch-yongseok-6033.log
Checking log directory for disk usage. This may take awhile.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://192.168.0.15:39841/
ros_comm version 1.12.17

SUMMARY
=====

PARAMETERS
* /rosdistro: kinetic
* /rosversion: 1.12.17

NODES
auto-starting new master
process[master]: started with pid [6043]
ROS_MASTER_URI=http://192.168.0.15:11311

setting /run_id to 65949316-1f81-11ec-b5be-f40669f0af37
process[rosout-1]: started with pid [6056]
started core service [/rosout]
```



② 새 터미널에서 rosrun turtlesim turtlesim_node 실행

```
yongseok@yongseok:~$ rosrun turtlesim turtlesim_node
[ INFO] [1632740216.631981920]: Starting turtlesim with node name /turtlesim
[ INFO] [1632740216.654538701]: Spawning turtle [turtle1] at x=[5.544445], y=[5.544445], theta=[0.000000]
```

```
yongseok@yongseok:~$
```

```
rosrun turtlesim turtle_teleop_key
Reading from keyboard
-----
Use arrow keys to move the turtle.
```

④ 새 터미널에서

rosrun turtlesim turtle_teleop_key 실행



[tab] key 사용

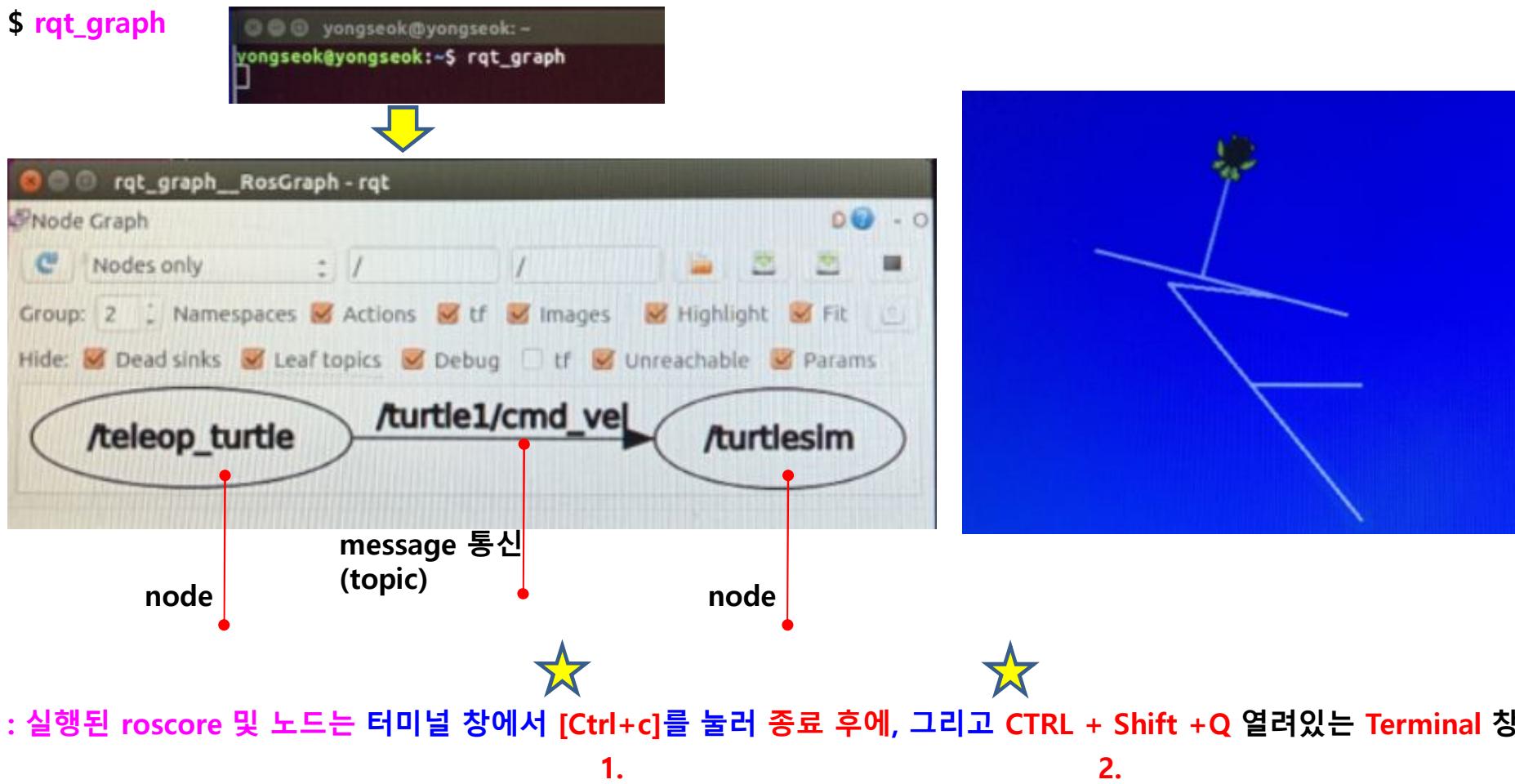
: 명령어 자동완성

(10) ROS 동작 테스트 (rqt_graph 노드)

: rqt_graph 노드는 현재 실행 중인 노드들의 정보를 GUI 형태로 시각화

: 새 터미널에서

\$ rqt_graph



: 실행된 roscore 및 노드는 터미널 창에서 [Ctrl+c]를 눌러 종료 후에, 그리고 CTRL + Shift +Q 열려있는 Terminal 창 닫기

1.

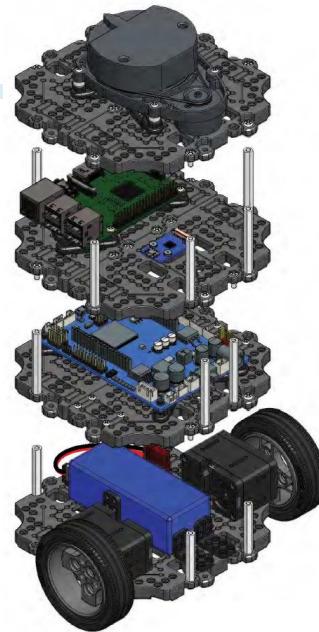
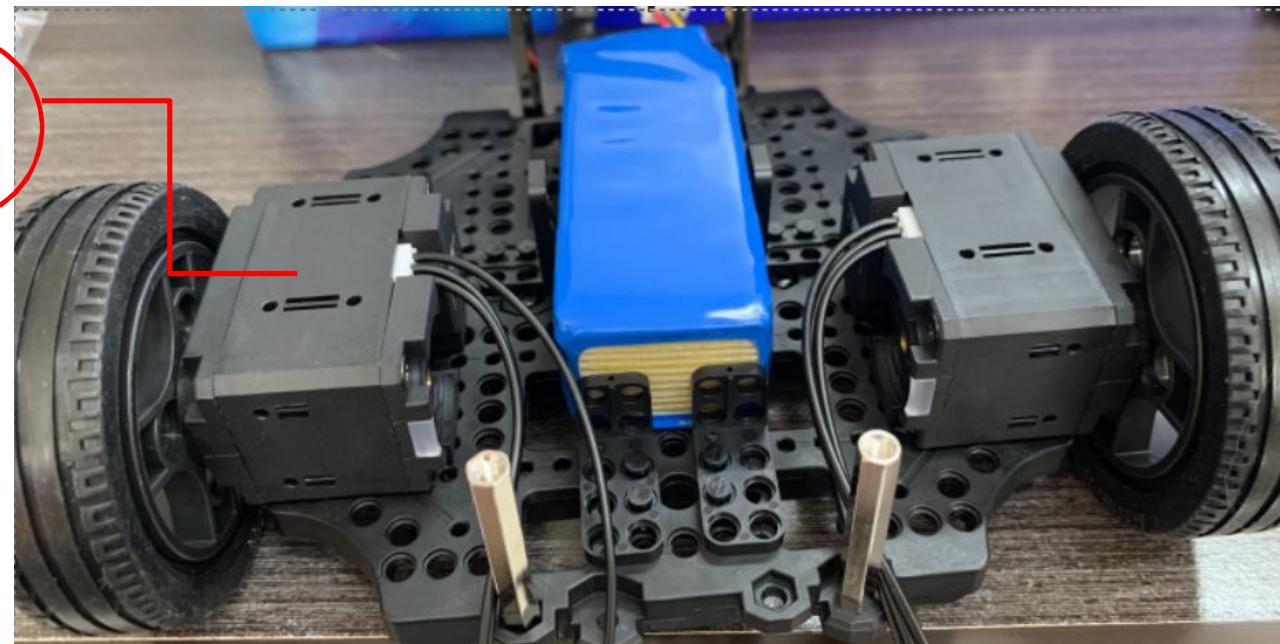
2.

4. ROS 개발환경 구축

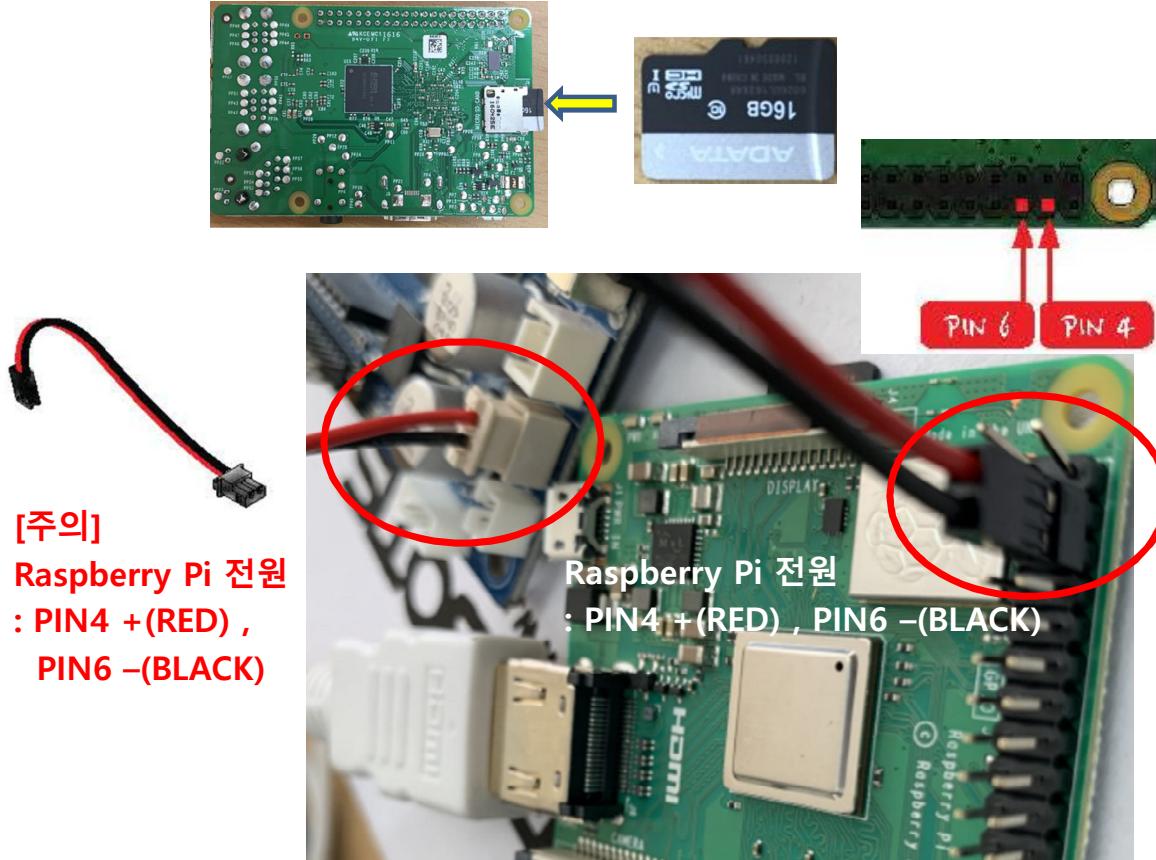
TurtleBot3 burger 조립하기

- : <https://youtu.be/rvm-m2ogrLA> 조립 동영상 참고하여 조립하기
- : 조립 pdf 파일 참고하여 조립하기
- : https://emanual.robotis.com/docs/en/platform/turtlebot3/hardware_setup/#hardware-assembly

TurtleBot3 burger 조립 주의 사항 : 왼쪽2 오른쪽1

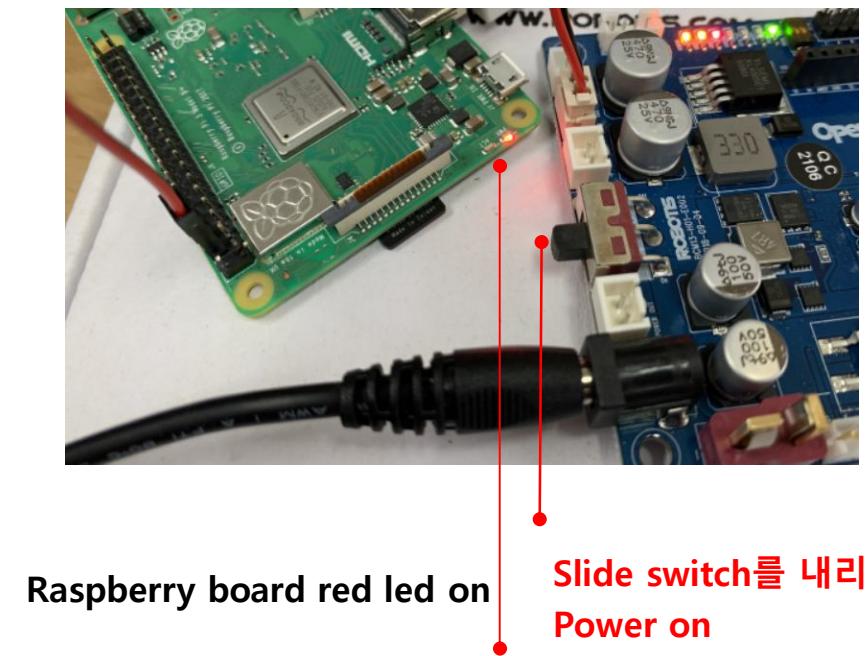


4. ROS 개발환경 구축



[주의]

Raspberry Pi 전원
: PIN4 +(RED) ,
PIN6 -(BLACK)



Raspberry board red led on

Slide switch를 내리면
Power on

4. ROS 개발환경 구축

실험 환경 2 : Install ROS SBC(Single Board Computer) - Raspberry Pi 4B 에서 작업하기

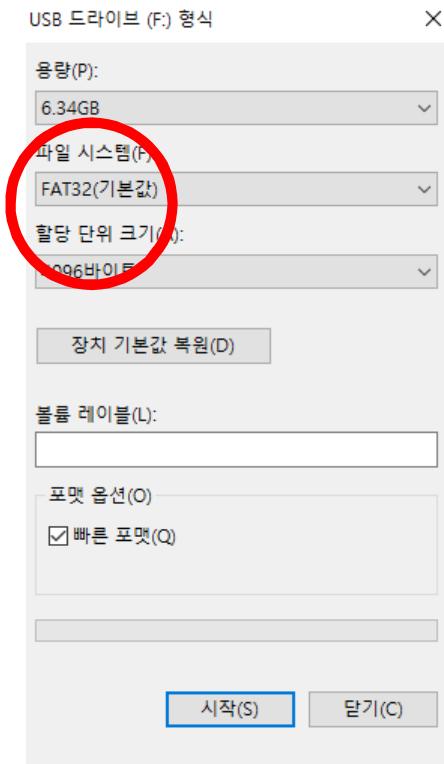
- ① TurtleBot3 burge의 SBC에서 MicroSD 카드 빼기



- ② MicroSD를 USB socket에 넣기



- ③ [option] 강의용 PC에서 Format 진행하기



먼저 :

MicroSD 에 Raspberry Pi OS(Raspbian OS),
ROS를 구울 것

4. ROS 개발환경 구축

Ref. https://emanual.robotis.com/docs/en/platform/turtlebot3/sbc_setup/#sbc-setup

4-2-4. Install ROS SBC(Single Board Computer) - Raspberry Pi 4B 작업하기

(1) Raspbian & ROS Noetic 다운받기

: TurtleBot3 burger에 조립된 Raspberry Pi 4B에 Linux 계열의 Raspbian OS와 ROS 를 설치

: Raspbian 파일을 www.raspberrypi.org/software 에서 받을 수 있지만

→ robotis에서 제공한 raspbian은 ROS package 포함되어 효과적이며 호환성으로 인해

The screenshot shows the '3.2. SBC Setup' section of the TurtleBot3 documentation. It includes a search bar with 'Enter Search Terms' and a dropdown menu with options like Kinetic, Melodic, Noetic, Dashing, Foxy, Humble, and Windows. The main content area has a 'WARNING' section with two bullet points: 'This process may take long time. Please do not use battery while following this section.' and 'An HDMI monitor and input devices such as a keyboard and a mouse will be required to complete this section.' Below this is a '3.2. 1. Prepare microSD Card and Reader' section with an image of a microSD card reader and text: 'If your PC does not have a microSD slot, please use a microSD card reader to burn the image onto a microSD card.' There is also a '3.2. 2. Download TurtleBot3 SBC Image' section with two download links: one for 'Raspberry Pi 3B+' ROS Noetic image (SHA256: a7c57e20f2ee4204c95315866f4a2748860947c63ed390b6d06d95074830309) and another for 'Raspberry Pi 4B (2GB or 4GB) ROS Noetic image' (SHA256: 9d48925a78381885916a6f3bb77891adbfaf2b271b05fe2ae9a9b7ebd12c46cc). A red arrow points from the second download link to the 'tb3_rpi4+_noetic_20220224.zip' file mentioned in the text.

SBC(Single Board Computer)
: Raspberry Pi 4B



① windows 기반의 개인 PC에서 Download
→ (tb3_rpi4+_noetic_20220224.zip)

② Download 파일을 압축을 풀면 .img 생성
(tb3_rpi4+_noetic_20220224.img)

③ microSD에 img 굽기 위해 tool 준비



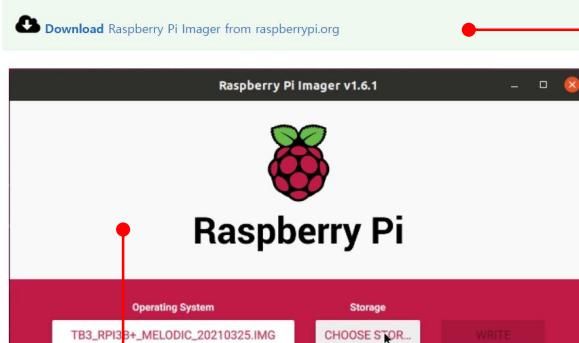
개인 PC에
삽입하기

4. ROS 개발환경 구축

3. Quick Start Guide
3.1. PC Setup
3.2. SBC Setup
3.3. OpenCR Setup
3.4. Hardware Assembly
3.5. Bringup
3.6. Basic Operation
4. SLAM
5. Navigation
6. Simulation
7. Manipulation
8. Autonomous Driving
9. Machine Learning
10. Examples
11. Friends(Locomotion)
12. Learn
13. More Info

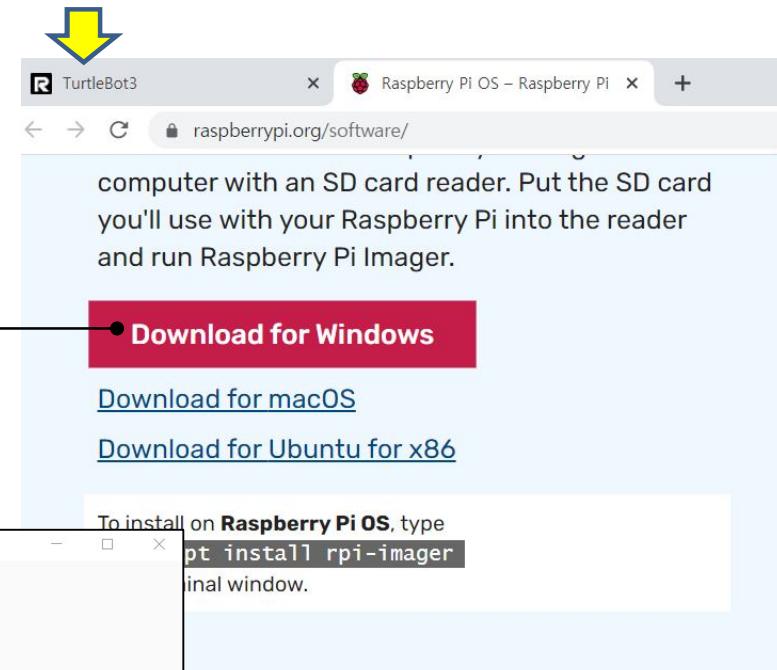
3. 2. 4. Burn the image file
You can use various image burning tools.
For example, [Raspberry Pi Imager](#) or Linux [Disk Utility](#) can be used.
Choose your preferred tool to burn the image to microSD.

3. 2. 4. 1. Raspberry Pi Imager
Please refer to this article to find more information about Raspberry Pi Imager.



microSD에 img 굽기 위해 tool 준비

④ click

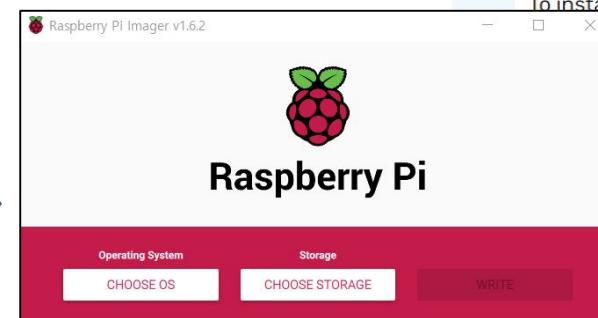


⑤ imager download
(imager_1.8.1.exe)

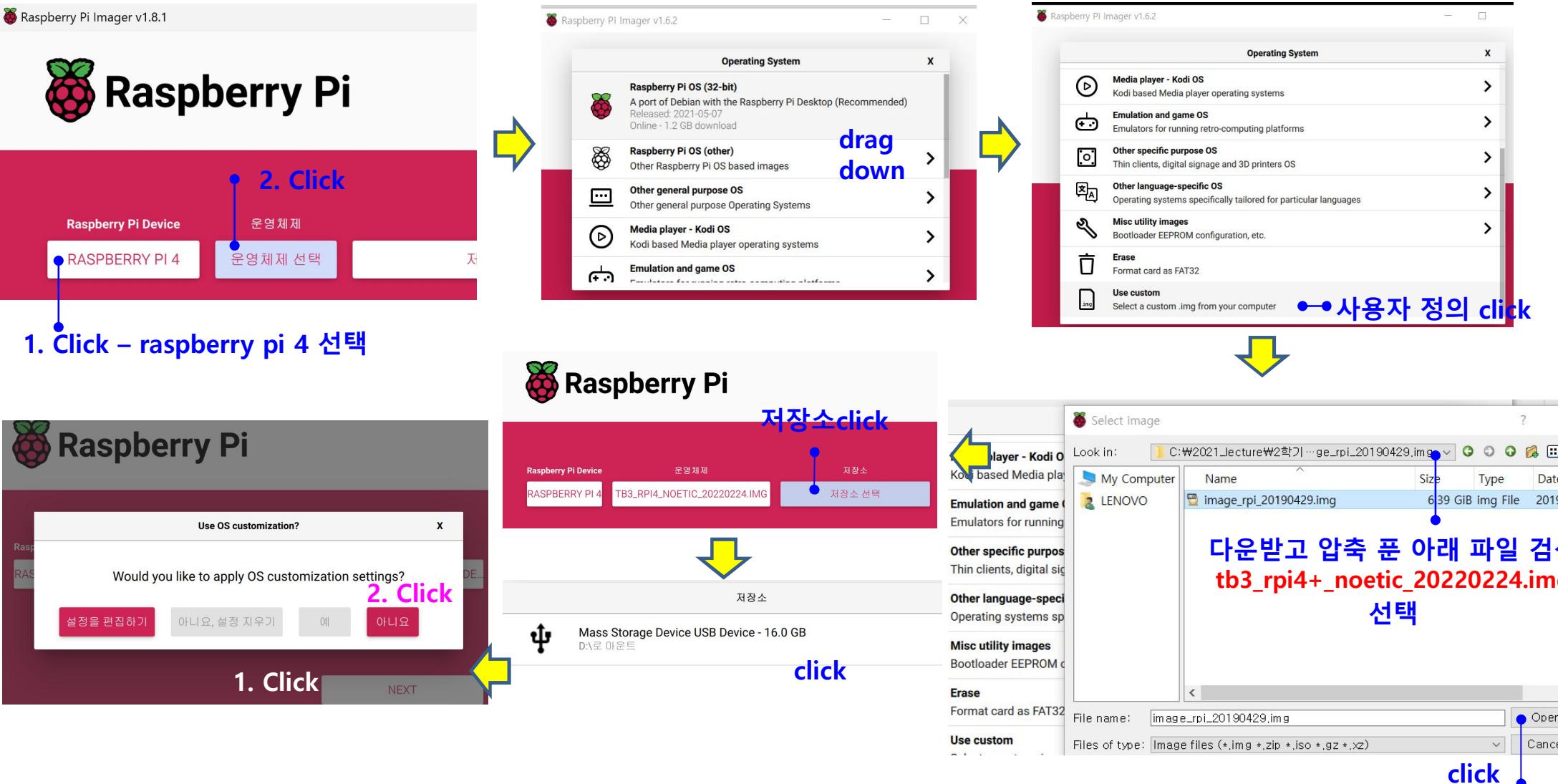
⑥ 안내 youtube 동영상을 보면서
microSD에 img 굽기



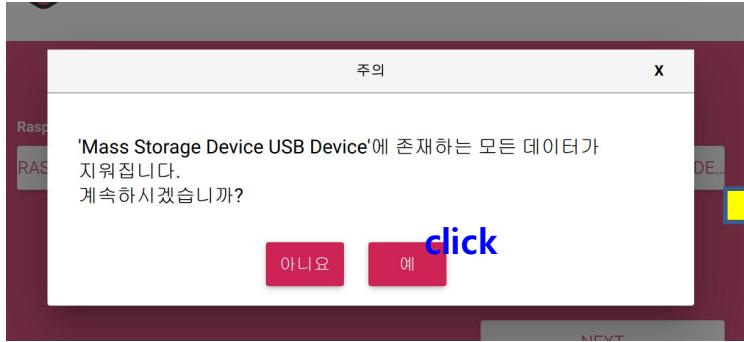
→ imager_1.8.1.exe 관리자 권한으로 실행



4. ROS 개발환경 구축



4. ROS 개발환경 구축



교수용 : IP 분배

4-2-4. Install ROS SBC(Single Board Computer) - Raspberry Pi 4B 작업하기

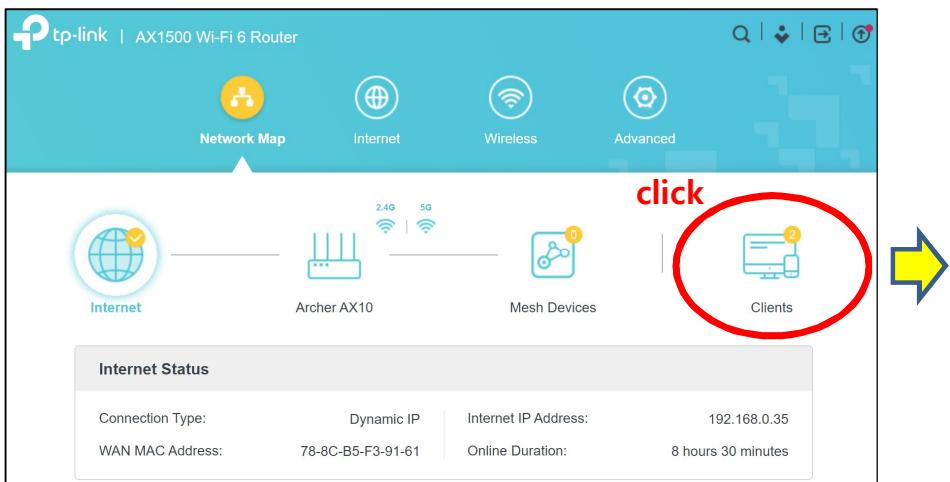
(2) Configure the WiFi Network Setting : Ref 3.2.6

[스마트 팩토리실 TP-Link 9160]

: AX10 AX1500 WIFI MODEL

Window pc에서

<http://tplinkwifi.net> (admin, pwd a123456) 접속



Connected Clients							View Blacklist
Type	Information	Real-time Rate	Interface	Tx/Rx Rate(Mbps)	Duration	Block	
star	8C-17-59-3A-F0-A7 192.168.1.151	↑ 0.11 KB/s ↓ 0.13 KB/s	2.4G	144 / 144	3 h 31 min	<input checked="" type="checkbox"/>	
yongseok	00-D4-9E-F7-BA-C0 192.168.1.141	↑ 0 KB/s ↓ 0 KB/s	5G	433 / 6.0	3 min	<input checked="" type="checkbox"/>	

접속한 id를 확인할 수 있음

4. ROS 개발환경 구축

Ref. https://emanual.robotis.com/docs/en/platform/turtlebot3/sbc_setup/#sbc-setup

4-2-4. Install ROS SBC(Single Board Computer) - Raspberry Pi 4B 작업하기

(2) Configure the WiFi Network Setting : Ref 3.2.6

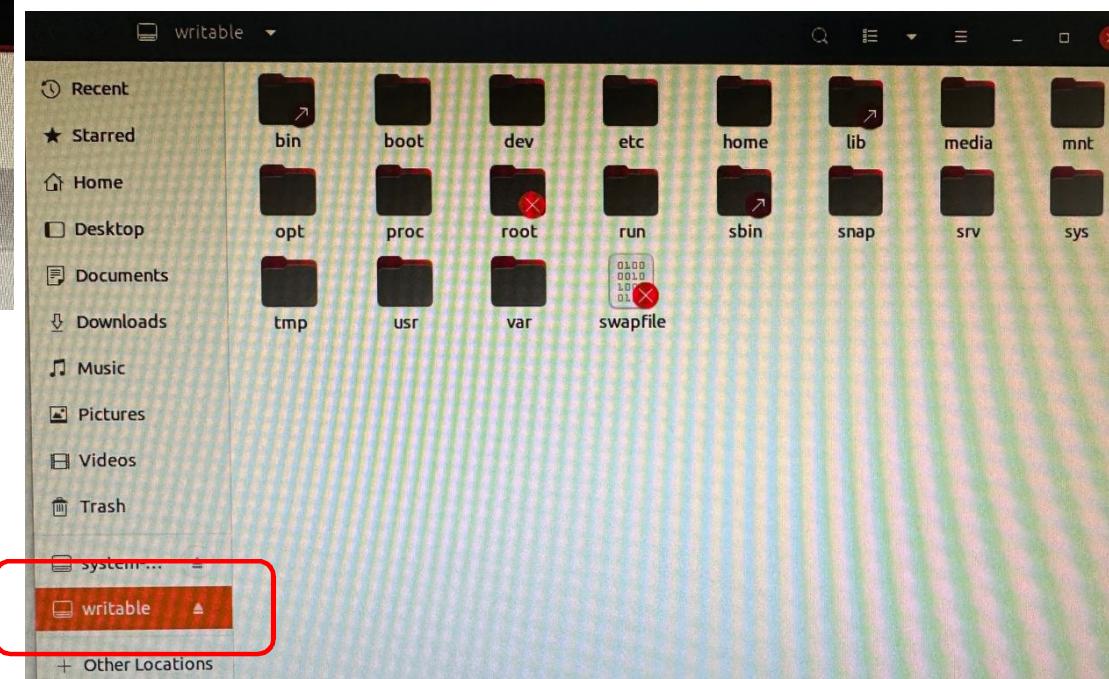
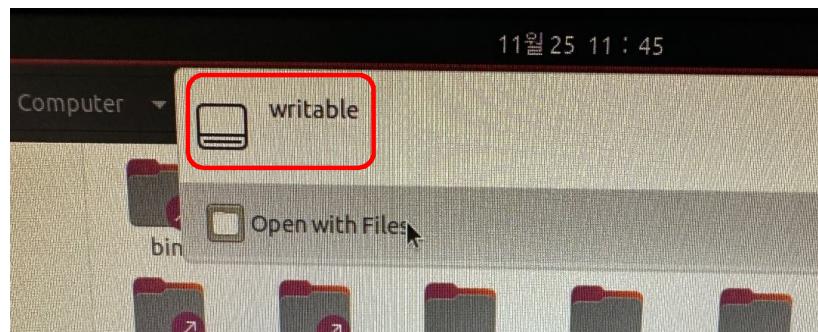
- ⓐ tb3_rpi4+_noetic_20220224.img 가 구워진 Micro SD 카드를 미니PC(Master)에 삽입

: 정상적인 연결의 경우 아래와 같이 **writable** folder 열림

(열리지 않는 경우 : usb reader 교체)



Raspberry Pi 4B
HDMI -MICRO



4. ROS 개발환경 구축

Ref. https://emanual.robotis.com/docs/en/platform/turtlebot3/sbc_setup/#sbc-setup

(2) Configure the WiFi Network Setting : Ref 3.2.6

- ⑤ 미니PC(Master)에서: tb3_rpi4+_noetic_20220224.img 가 구워진 Micro SD 카드를 미니PC(Master) 삽입 상태

\$ cd /media/\$USER/writable/etc/netplan

```
yongseok@yongseok: /media/yongseok/writable/etc/netplan
yongseok@yongseok:~$ cd /media/$USER/writable/etc/netplan
yongseok@yongseok: /media/yongseok/writable/etc/netplan$
```

\$ sudo nano 50-cloud-init.yaml

```
e-gvfs-metadata.
yongseok@yongseok: /media/yongseok/writable/etc/netplan$ sudo nano 50-cloud-init.yaml
yongseok@yongseok: /media/yongseok/writable/etc/netplan$
```

또는 \$ sudo gedit 50-cloud-init.yaml

```
yongseok@yongseok: /media/yongseok/writable/etc/netplan
yongseok@yongseok:~$ cd /media/$USER/writable/etc/netplan
yongseok@yongseok: /media/yongseok/writable/etc/netplan$ sudo gedit 50-cloud-init.yaml
[sudo] password for yongseok:
```

*50-cloud-init.yaml
writable /media/yongseok/writable/etc/netplan

```
1# This file is generated from information provided by the datasource. Changes
2# to it will not persist across an instance reboot. To disable cloud-init's
3# network configuration capabilities, write a file
4# /etc/cloud/cloud.cfg.d/99-disable-network-config.cfg with the following:
```

gedit 오류 발생할 수 있음

```
yongseok@yongseok: /media/yongseok/writable/etc/netplan$ sudo gedit 50-cloud-init.yaml
[sudo] password for yongseok:
(gedit:5964): Tepl-WARNING **: 11:51:55.873: GVfs metadata is not supported. Fallback
to TeplMetadataManager. Either GVfs is not correctly installed or GVfs metadata are not
supported on this platform. In the latter case, you should configure Tepl with --disab
e-gvfs-metadata.
yongseok@yongseok: /media/yongseok/writable/etc/netplan$ sudo nano 50-cloud-init.yaml
yongseok@yongseok: /media/yongseok/writable/etc/netplan$
```

4. ROS 개발환경 구축

Ref. https://emanual.robotis.com/docs/en/platform/turtlebot3/sbc_setup/#sbc-setup

(2) Configure the WiFi Network Setting : Ref 3.2.6

⑤ WiFi Network Setting을 위한 yaml(parameter file)변경 (매우 주의)

```
wifis:  
  wlan0:  
    dhcp4: yes  
    dhcp6: yes  
    access-points:  
      WIFI_SSID:  
        password: WIFI_PASSWORD
```

초록색 : wifi 명과 pwd입력(띄어쓰기 주의)

```
network:  
  version: 2  
  renderer: networkd  
  ethernets:  
    eth0:  
      dhcp4: yes  
      dhcp6: yes  
      optional: true  
  wifis:  
    wlan0:  
      dhcp4: yes  
      dhcp6: yes  
      optional: true  
      access-points:  
        TP-Link_9160_5G:  
          password: 91919533
```

2칸 띄어쓰기

2칸 띄어쓰기

1칸 띄어쓰기

Nano 편집기

저장 : Ctrl + S
나가기 : Ctrl + X

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify
^X Exit ^R Read File ^\ Replace ^U Paste Text ^T To Spell

```
6 network:  
7   version: 2  
8   renderer: networkd  
9   ethernets:  
10  eth0:  
11    dhcp4: yes  
12    dhcp6: yes  
13    optional: true  
14  wifis:  
15    wlan0:  
16    dhcp4: yes  
17    dhcp6: yes  
18    access-points:  
19      TP-Link_9160_5G:  
20        password: 91919533
```

4. ROS 개발환경 구축

Ref. https://emanual.robotis.com/docs/en/platform/turtlebot3/sbc_setup/#sbc-setup

(2) Configure the WiFi Network Setting : Ref 3.2.6

ⓐ Raspberry Pi 4B 의 host name 변경하기

: WiFi Network Setting을 위한 **yaml**(parameter file)변경후

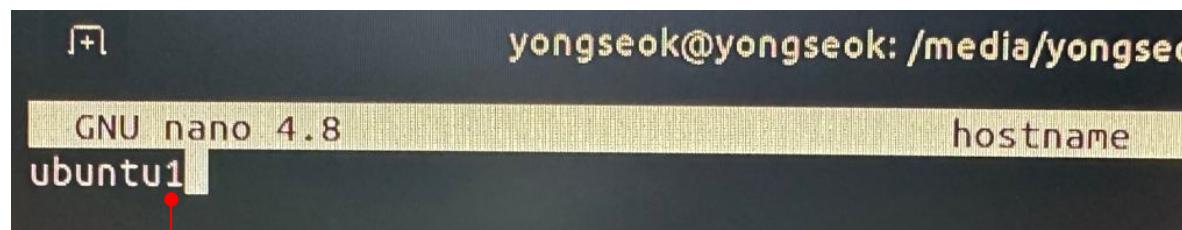
동일한 이름의 네트워크 연결 문제(학생 수 多)를 해결하기

```
$ cd ..
```

```
$ sudo nano hostname
```

```
yongseok@yongseok:/media/yongseok/writable/etc/netplan$ sudo nano 50-cloud-init.yaml
yongseok@yongseok:/media/yongseok/writable/etc/netplan$ cd ..
yongseok@yongseok:/media/yongseok/writable/etc$ sudo nano hostname
```

etc 경로



```
yongseok@yongseok: /media/yongseok
GNU nano 4.8
hostname
```

: 숫자는 서로 다르게 교수 지정

숫자 추가 저장 : Ctrl + S
나가기 : Ctrl + X

4. ROS 개발환경 구축

Ref. https://emanual.robotis.com/docs/en/platform/turtlebot3/sbc_setup/#sbc-setup

(2) Configure the WiFi Network Setting : Ref 3.2.6

④ Raspberry Pi 4B에 tb3_rpi4+_noetic_20220224.img 가 구워진 Micro SD 카드 삽입후 전원 ON

<http://tplinkwifi.net>



전원 ON 후
30초~5분내



Connected Clients						
Type	Information	Real-time Rate	Interface	Tx/Rx Rate(Mbps)	Duration	Block
UNKNOWN	9E-C9-8D-0B-23-D2 192.168.1.144	↑ 0.26 KB/s ↓ 0.36 KB/s	2.4G	144 / 24	1 h 31 min	🚫
star	8C-17-59-3A-F0-A7 192.168.1.151	↑ 0 KB/s ↓ 0.06 KB/s	5G	1201 / 1201	1 h 17 min	🚫
ubuntu1	E4-5F-01-DB-E3-19 192.168.1.145	↑ 0 KB/s ↓ 0 KB/s	5G	390 / 24	0 min	🚫
yongseok	00-D4-9E-F7-BA-C0 192.168.1.141	↑ 0 KB/s ↓ 0 KB/s	5G	433 / 6.0	3 h 19 min	🚫

설정한 ubuntu1 확인됨



Id 학생 공지

4. ROS 개발환경 구축

Ref. https://emanual.robotis.com/docs/en/platform/turtlebot3/sbc_setup/#sbc-setup

(3) 원격 제어 및 ROS Network Configuration : Ref 3.2.7

① Remote PC(미니 PC)에서 SBC(Raspberry Pi 4B)를 원격 제어 및 ROS Network Configuration

→ Remote PC에 아래와 같이 SBC(Raspberry Pi 3 4B) IP 주소를 입력하면 연결됨

(ssh : Secure Shell 명령어를 통한 원격 제어함)

→ : Remote PC(미니 PC)에서 새 터미널(Ctrl + Alt + T)에서

\$ ssh ubuntu@192.168.xxx.xxx (ID : ubuntu, PWD : turtlebot)

```
yongseok@yongseok:~$ ssh ubuntu@192.168.1.145
ubuntu@192.168.1.145's password:
Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 5.4.0-1053-raspi aarch64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

 System information as of Sat 25 Nov 2023 04:10:49 AM UTC

 System load:  0.08      Temperature:        39.4  C
 Usage of /:   59.5% of 14.33GB  Processes:          136
 Memory usage: 9%           Users logged in:     0
 Swap usage:   0%           IPv4 address for wlan0: 192.168.1.145

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
 just raised the bar for easy, resilient and secure K8s cluster deployment.

 https://ubuntu.com/engage/secure-kubernetes-at-the-edge

1 update can be applied immediately.
To see these additional updates run: apt list --upgradable

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Thu Feb 24 07:08:10 2022 from 192.168.0.7
ubuntu@ubuntu1:~$
```

● remote connection

turtlebot 입력하지만 안보임



shell의 user name 변경됨(원격으로 turtlebot 접속함)

4. ROS 개발환경 구축

Ref. https://emanual.robotis.com/docs/en/platform/turtlebot3/sbc_setup/#sbc-setup

(3) 원격 제어 및 ROS Network Configuration : Ref 3.2.7

⑨ \$ ifconfig

```
Last login: Sat Nov 25 04:10:52 2023 from 192.168.1.141
ubuntu@ubuntu1:~$ ifconfig
eth0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    ether e4:5f:01:db:e3:18 txqueuelen 1000 (Ethernet)
      RX packets 0 bytes 0 (0.0 B)
      RX errors 0 dropped 0 overruns 0 frame 0
      TX packets 0 bytes 0 (0.0 B)
      TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
        loop txqueuelen 1000 (Local Loopback)
      RX packets 108 bytes 8744 (8.7 KB)
      RX errors 0 dropped 0 overruns 0 frame 0
      TX packets 108 bytes 8744 (8.7 KB)
      TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.145 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::e45f:1ff:fedb:e319 prefixlen 64 scopeid 0x20<link>
        ether e4:5f:01:db:e3:19 txqueuelen 1000 (Ethernet)
      RX packets 822 bytes 1055678 (1.0 MB)
      RX errors 0 dropped 0 overruns 0 frame 0
      TX packets 457 bytes 53033 (53.0 KB)
      TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
ubuntu@ubuntu1:~$
```

⑩ \$ nano ~/.bashrc

```
ubuntu@ubuntu1:~$ nano ~/.bashrc
ubuntu@ubuntu1:~$ nano ~/.bashrc
```

환경변수 열기(nano 편집기)

id 확인



```
ubuntu@ubuntu1:~ 80x38
GNU nano 4.8
# ~./.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
    if [ -f /usr/share/bash-completion/bash_completion ]; then
        . /usr/share/bash-completion/bash_completion
    elif [ -f /etc/bash_completion ]; then
        . /etc/bash_completion
    fi
fi
source /opt/ros/noetic/setup.bash
source ~/catkin_ws/devel/setup.bash

alias cs='cd ~/catkin_ws/src'
alias cm='cd ~/catkin_ws && catkin_make'
alias eb='nano ~/.bashrc'
alias nb='nano ~/.bashrc'
alias sb='source ~/.bashrc'

# Replace {IP_ADDRESS_OF_REMOTE_PC} with the IP address of remote
# Both Remote PC and Raspberry Pi should be connected in the same
# network
export ROS_MASTER_URI=http://192.168.1.141:11311
# Replace {IP_ADDRESS_OF_RASPBERRY_PI} with the IP address of Raspberry
# Pi
export ROS_HOSTNAME=192.168.1.145
export LDS_MODEL=LDS-02

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify
^X Exit ^R Read File ^Y Replace ^U Paste Text ^T To Spell
```

: 아래 방향키로 밑으로

master id 설정

raspberry id 설정

LDS 설정

Ctrl + S 저장

Ctrl + X 나가기

4. ROS 개발환경 구축

Ref. https://emanual.robotis.com/docs/en/platform/turtlebot3/sbc_setup/#sbc-setup

(3) 원격 제어 및 ROS Network Configuration : Ref 3.2.7

① \$ source ~/.bashrc

```
ubuntu@ubuntu1:~$ nano ~/.bashrc
ubuntu@ubuntu1:~$ source ~/.bashrc
ubuntu@ubuntu1:~$
```

: 환경변수 반영



(4) 원격 제어 및 LDS-02 Configuration : Ref 3.2.8

① \$ sudo apt update

```
ubuntu@ubuntu1:~$ sudo apt update
Get:1 http://packages.ros.org/ros/ubuntu focal In
Hit:2 http://ports.ubuntu.com/ubuntu-ports focal 
Get:3 http://ports.ubuntu.com/ubuntu-ports focal-u
Get:4 http://packages.ros.org/ros/ubuntu focal/las
```

\$ sudo apt upgrade

```
ubuntu@ubuntu1:~$ sudo apt upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following packages were automatically
  libfwupdplugin1 libxmlb1
Use 'sudo apt autoremove' to remove them.
```

생략가능함

시간 소요

```
Taking backup of spi2-1cs.dtbo.
Installing new spi2-1cs.dtbo.
Taking backup of w5500.dtbo.
Installing new w5500.dtbo.
ubuntu@ubuntu1:~$
```

4. ROS 개발환경 구축

Ref. https://emanual.robotis.com/docs/en/platform/turtlebot3/sbc_setup/#sbc-setup

LDS -02를 위한 update 작업

```
$ sudo apt install libudev-dev  
$ cd ~/catkin_ws/src  
$ git clone -b develop https://github.com/ROBOTIS-GIT/lid08_driver.git  
$ cd ~/catkin_ws/src/turtlebot3 && git pull  
$ rm -r turtlebot3_description/ turtlebot3_teleop/ turtlebot3_navigation/ turtlebot3_slam/ turtlebot3_example/  
$ cd ~/catkin_ws && catkin_make
```



```
$ echo 'export LDS_MODEL=LDS-02' >> ~/.bashrc  
$ source ~/.bashrc
```

앞에서 반영함



export path = 새로 등록할 환경변수 주소

```
Installing new w5500.dtbo.  
ubuntu@ubuntu1:~$ sudo apt install libudev-dev  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
libudev-dev is already the newest version (245.4-4ubuntu3.22).  
The following packages were automatically installed and are no  
    libfwupdplugin1 libxmlb1 linux-headers-5.4.0-1043-raspi linux-  
    linux-modules-5.4.0-1043-raspi linux-raspi-headers-5.4.0-1043  
Use 'sudo apt autoremove' to remove them.  
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

```
ubuntu@ubuntu1:~$ cd ~/catkin_ws/src  
ubuntu@ubuntu1:~/catkin_ws/src$ git clone -b develop https://github.com/ROBOTIS-GIT/ld08_driver.git  
fatal: destination path 'ld08_driver' already exists and is not an empty directory.  
ubuntu@ubuntu1:~/catkin_ws/src$ cd ~/catkin_ws/src/turtlebot3 && git pull  
remote: Enumerating objects: 476, done.  
remote: Counting objects: 100% (323/323), done.  
remote: Compressing objects: 100% (95/95), done.  
remote: Total 476 (delta 250), reused 277 (delta 228), pack-reused 153  
Receiving objects: 100% (476/476), 119.15 KiB | 3.22 MiB/s, done.  
Resolving deltas: 100% (280/280), completed with 40 local objects.  
From https://github.com/ROBOTIS-GIT/turtlebot3  
  beaf02d..f8631cf  develop          -> origin/develop  
* [new branch]  feature-burger-camera -> origin/feature-burger-camera  
* [new branch]  feature-camera       -> origin/feature-camera  
* [new branch]  feature-humble        -> origin/feature-humble  
* [new branch]  hotfix-humble         -> origin/hotfix-humble  
* [new branch]  hotfix-humble-example -> origin/hotfix-humble-example  
* [new branch]  hotfix-melodic-cartographer -> origin/hotfix-melodic-cartographer  
* [new branch]  humble-devel         -> origin/humble-devel  
b4b18ef..27166af  ros2             -> origin/ros2  
7dde7f0..6f393f5  ros2-devel        -> origin/ros2-devel  
* [new tag]      2.1.5             -> 2.1.5  
Already up to date.  
ubuntu@ubuntu1:~/catkin_ws/src/turtlebot3$ rm -r turtlebot3_description/ turtlebot3_teleop/ turtlebot3_navigation/ turtlebot3_slam/ turtlebot3_example/  
rm: cannot remove 'turtlebot3_description/': No such file or directory  
rm: cannot remove 'turtlebot3_teleop/': No such file or directory  
rm: cannot remove 'turtlebot3_navigation/': No such file or directory  
rm: cannot remove 'turtlebot3_slam/': No such file or directory  
rm: cannot remove 'turtlebot3_example/': No such file or directory  
ubuntu@ubuntu1:~/catkin_ws/src/turtlebot3$
```

```
ubuntu@ubuntu1:~/catkin_ws/src/turtlebot3$ cd ~/catkin_ws && catkin_make
Base path: /home/ubuntu/catkin_ws
Source space: /home/ubuntu/catkin_ws/src
Build space: /home/ubuntu/catkin_ws/build
Devel space: /home/ubuntu/catkin_ws/devel
Install space: /home/ubuntu/catkin_ws/install
#####
##### Running command: "make cmake_check_build_system" in "/home/ubuntu/catkin_ws"
#####
-- Using CATKIN_DEVEL_PREFIX: /home/ubuntu/catkin_ws/devel
-- Using CMAKE_PREFIX_PATH: /home/ubuntu/catkin_ws/devel;/opt/ros/noetic
-- This workspace overlays: /home/ubuntu/catkin_ws/devel;/opt/ros/noetic
-- Found PythonInterp: /usr/bin/python3 (found suitable version "3.8.10")
-- BUILD_SHARED_LIBS is on
-- ~~~ traversing 3 packages in topological order:
-- ~~~ - turtlebot3 (metapackage)
-- ~~~ - ld08_driver
-- ~~~ - turtlebot3_bringup
-- ~~~
-- +++ processing catkin metapackage: 'turtlebot3'
-- ==> add_subdirectory(turtlebot3/turtlebot3)
-- +++ processing catkin package: 'ld08_driver'
-- ==> add_subdirectory(ld08_driver)
-- +++ processing catkin package: 'turtlebot3_bringup'
-- ==> add_subdirectory(turtlebot3/turtlebot3_bringup)
-- Configuring done
-- Generating done
-- Build files have been written to: /home/ubuntu/catkin_ws/build
#####
##### Running command: "make -j4 -l4" in "/home/ubuntu/catkin_ws/build"
#####
Scanning dependencies of target ld08_driver
[ 0%] Built target turtlebot3_msgs_generate_messages_py
[ 0%] Built target rosgraph_msgs_generate_messages_py
[ 0%] Built target roscpp_generate_messages_lisp
[ 0%] Built target geometry_msgs_generate_messages_nodejs
[ 0%] Built target geometry_msgs_generate_messages_cpp
[ 0%] Built target rosgraph_msgs_generate_messages_lisp
[ 12%] Building CXX object ld08_driver/CMakeFiles/ld08_driver.dir/src/
[ 12%] Built target rosgraph_msgs_generate_messages_eus
[ 12%] Built target turtlebot3_bringup
```

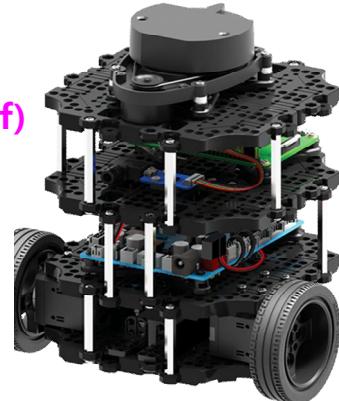
```
[ 12%] Built target diagnostic_msgs_generate_messages_lisp
[ 12%] Built target diagnostic_msgs_generate_messages_py
[ 12%] Built target turtlebot3_msgs_generate_messages_cpp
[ 12%] Built target turtlebot3_msgs_generate_messages_nodejs
Scanning dependencies of target turtlebot3_diagnostics
[ 25%] Building CXX object turtlebot3/turtlebot3_bringup/CMakeFiles/turtlebot3_diagnostics.dir/turtlebot3_diagnostics.cpp.o
[ 37%] Linking CXX executable /home/ubuntu/catkin_ws/devel/lib/ld08_driver/ld08_driver
[ 87%] Built target ld08_driver
[100%] Linking CXX executable /home/ubuntu/catkin_ws/devel/lib/turtlebot3_bringup/turtlebot3_bringup
[100%] Built target turtlebot3_diagnostics
ubuntu@ubuntu1:~/catkin_ws$
```

4. ROS 개발환경 구축

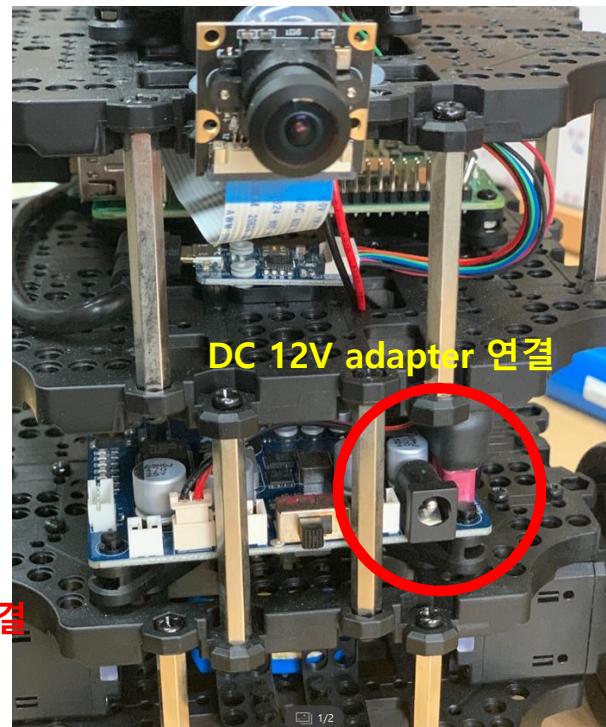
실험 환경 1 : 만일 TurtleBot3 burger 조립되어 있지 않다면,

: <https://youtu.be/rvm-m2ogrLA> 조립 동영상 보고 조립 과정 이해하기

: https://emanual.robotis.com/docs/en/platform/turtlebot3/hardware_setup/#hardware-assembly (조립 pdf)



전원 인가 및 충전 방법



Battery 사전에 충전하기
(미충전 경우 Buzzer 소리 발생)
→ 방치 시 완전 방전됨



4. ROS 개발환경 구축

4-2-5. OpenCR setup

: OpenCR(Control module for ROS)

- STM32F746 (ARM Cortex-M7 32bits RISC core)
- 216MHz frequency

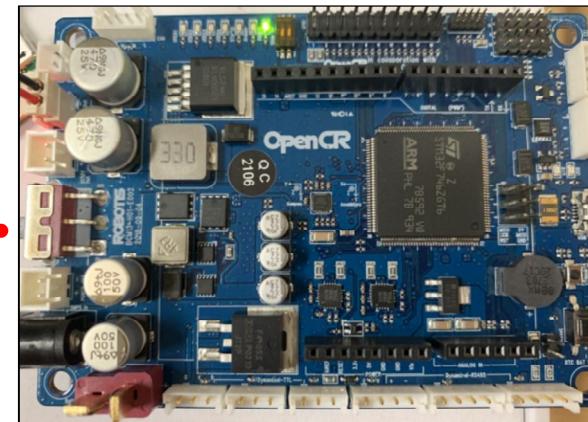
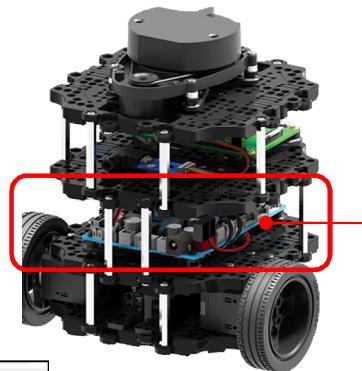
: <https://emanual.robotis.com/docs/en/parts/controller/opencr10/>

The screenshot shows the 'OpenCR 1.0' documentation page. The left sidebar contains a navigation menu with the following items:

- 1. Introduction
- 2. Specifications
- 3. Layout / Pin Map
- 4. Arduino IDE
- 5. Examples
- 6. Bootloader
- 7. Downloader
- 8. Reference

The main content area displays two sections: '1. Introduction' and '2. Specifications'. The 'Introduction' section provides an overview of the OpenCR1.0 board, its purpose, and its compatibility with various development environments. The 'Specifications' section is a table detailing the board's hardware components and capabilities.

Items	Specifications
Microcontroller	STM32F746ZG76 / 32-bit ARM Cortex®-M7 with FPU (216MHz, 462DMIPS) Reference Manual, Datasheet
Sensors	(Discontinued) Gyroscope 3Axis, Accelerometer 3Axis, Magnetometer 3Axis (MPU9250) (New) 3-axis Gyroscope, 3-Axis Accelerometer, A Digital Motion Processor™ (ICM-20648)
Programmer	ARM Cortex 10pin JTAG/SWD connector USB Device Firmware Upgrade (DFU) Serial
Digital I/O	32 pins (L 14, R 18) *Arduino connectivity 5Pin OLLO x 4 GPIO x 18 pins PWM x 6 I2C x 1 SPI x 1
Analog INPUT	ADC Channels (Max 12bit) x 6 USB x 1 (Micro-B USB connector/USB 2.0/Host/Peripheral/OTG)



OpenCR에 대해 학습하기

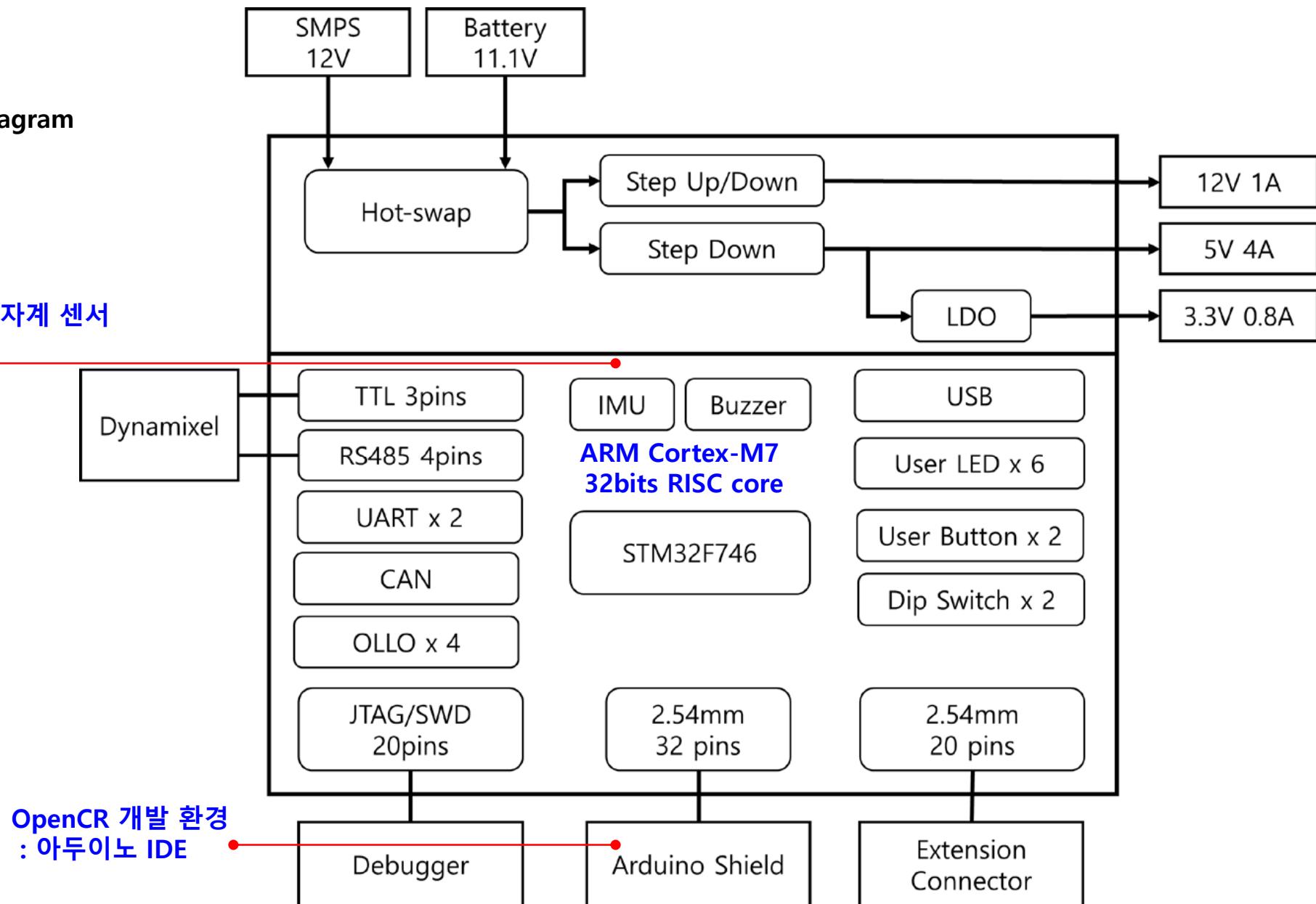
4-2-5. OpenCR setup

: OpenCR block diagram

IMU

: 자이로/가속도/지자계 센서

: MPU925010



4. ROS 개발환경 구축

4-2-5. OpenCR setup

: https://emanual.robotis.com/docs/en/platform/turtlebot3/opencr_setup/#opencr-setup

E-manual 3.3과 같이
openCR 설정하기



전원이 인가되면
USB 초록 LED 동작됨



3. 3. OpenCR Setup

1. Connect the OpenCR to the Raspberry Pi using the micro USB cable.
2. Install required packages on the Raspberry Pi to upload the OpenCR firmware.

```
$ sudo dpkg --add-architecture armhf  
$ sudo apt-get update  
$ sudo apt-get install libc6:armhf
```

3. Depending on the platform, use either `burger` or `waffle` for the **OPENCR_MODEL** name.

```
$ export OPENCR_PORT=/dev/ttyACM0  
$ export OPENCR_MODEL=burger_noetic  
$ rm -rf ./opencr_update.tar.bz2
```

4. Download the firmware and loader, then extract the file.

```
$ wget https://github.com/ROBOTIS-GIT/OpenCR-Binaries/raw/master/turtlebot3/ROS1/latest/opencr_update.tar.bz2  
$ tar -xvf opencr_update.tar.bz2
```

5. Upload firmware to the OpenCR.

```
$ cd ./opencr_update  
$ ./update.sh $OPENCR_PORT $OPENCR_MODEL.opencr
```

4. ROS 개발환경 구축

: openCR firmware 설정

Remote PC에서 원격으로 Raspbian에서 CTRL + ALT + T를 사용해 터미널 창 열기

\$ ssh ubuntu@192.168.xxx.xxx (ID : **ubuntu**, PWD : **turtlebot**)

← 이미 진입했으면 생략 (Path주의)

Install packages on the Raspberry Pi to upload the OpenCR firmware

\$ sudo dpkg --add-architecture armhf

\$ sudo apt-get update

```
ubuntu@ubuntu1:~$ sudo dpkg --add-architecture armhf
ubuntu@ubuntu1:~$ sudo apt-get update
Hit:1 http://ports.ubuntu.com/ubuntu-ports focal InRelease
Hit:2 http://packages.ros.org/ros/ubuntu focal InRelease
Get:3 http://ports.ubuntu.com/ubuntu-ports focal-updates InRelease [114 kB]
Hit:4 http://ports.ubuntu.com/ubuntu-ports focal-backports InRelease
Hit:5 http://ports.ubuntu.com/ubuntu-ports focal-security InRelease
Get:6 http://ports.ubuntu.com/ubuntu-ports focal-proposed InRelease [267 kB]
Get:7 http://ports.ubuntu.com/ubuntu-ports focal-updates/main arm64 Packages [2,256 kB]
Get:8 http://ports.ubuntu.com/ubuntu-ports focal-updates/main armhf Packages [1,450 kB]
Get:9 http://ports.ubuntu.com/ubuntu-ports focal-updates/universe armhf Packages [904 kB]
Get:10 http://ports.ubuntu.com/ubuntu-ports focal-updates/universe arm64 Packages [1,049 kB]
Get:11 http://ports.ubuntu.com/ubuntu-ports focal-proposed/universe armhf Packages [46.3 kB]
Get:12 http://ports.ubuntu.com/ubuntu-ports focal-proposed/universe arm64 Packages [51.3 kB]
Get:13 http://ports.ubuntu.com/ubuntu-ports focal-proposed/universe Translation-en [24.4 kB]
Fetched 6,161 kB in 6s (999 kB/s)
Reading package lists... Done
```

\$ sudo apt-get install libc6:armhf

```
ubuntu@ubuntu1:~$ sudo apt-get install libc6:armhf
Reading package lists... Done
Building dependency tree
Reading state information... Done
libc6:armhf is already the newest version (2.31-0ubuntu9.12).
The following packages were automatically installed and are no longer required:
  libfwupdplugin1 libxml2 linux-headers-5.4.0-1043-raspi linux-image-5.4.0-1043-raspi
  linux-modules-5.4.0-1043-raspi linux-raspi-headers-5.4.0-1043
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
ubuntu@ubuntu1:~$
```

dpkg

: debian package 관리 시스템의 소프트웨어. dpkg 명령어
armhf

: arm core Hardware FPU 구조의 debian 명칭

libc6:armhf

: C library compiler

4. ROS 개발환경 구축

- ④ OPENCR_MODEL name burger 로 변경하기

```
$ export OPENCR_PORT=/dev/ttyACM0
```

```
$ export OPENCR_MODEL=burger_noetic
```

```
$ rm -rf ./opencr_update.tar.bz2
```

- ⑤ firmware와 loader 내려 받고, 압축

```
$ wget https://github.com/ROBOTIS-GIT/OpenCR-Binaries/raw/master/turtlebot3/ROS1/latest/opencr_update.tar.bz2
```

```
ubuntu@ubuntu1:~$ export OPENCR_PORT=/dev/ttyACM0
ubuntu@ubuntu1:~$ export OPENCR_MODEL=burger_noetic
ubuntu@ubuntu1:~$ rm -rf ./opencr_update.tar.bz2
ubuntu@ubuntu1:~$ wget https://github.com/ROBOTIS-GIT/OpenCR-Binaries/raw/master/turtlebot3/ROS1/latest/opencr_update.tar.bz2
--2023-11-27 18:46:55-- https://github.com/ROBOTIS-GIT/OpenCR-Binaries/raw/master/turtlebot3/ROS1/latest/opencr_update.tar.bz2
Resolving github.com (github.com)... 20.200.245.247
Connecting to github.com (github.com)|20.200.245.247|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://raw.githubusercontent.com/ROBOTIS-GIT/OpenCR-Binaries/master/turtlebot3/ROS1/latest/opencr_update.tar.bz2 [following]
--2023-11-27 18:46:55-- https://raw.githubusercontent.com/ROBOTIS-GIT/OpenCR-Binaries/master/turtlebot3/ROS1/latest/opencr_update.tar.bz2
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.108.133, 185.199.111.133, 185.199.110.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.108.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 507379 (495K) [application/octet-stream]
Saving to: 'opencr_update.tar.bz2'

opencr_update.tar.bz2      100%[=====] 495.49K  2.93MB/s   in 0.2s
```

```
$ tar -xvf opencr_update.tar.bz2
```

export

: shell 변수를 환경 변수로 저장
: OPENCR_PORT= 는 이후의 폴더로 저장

ttyACM0

: 리눅스에서 포트에 장치를 연결할 때,
특히 Arduino-uno를 연결할 때 부여되는 이름

rm -rf

: directory 삭제. -rf 파일 권한에 관계없이 제거

tar.bz2

: tar 여러 개의 파일을 압축하기.
: bz2 압축풀기. tar.bz2 압축하거나 풀거나
: -x, -v, -f 등 옵션

wget

: Web Get'의 약어로 웹 상의 파일을 다운로드

```
ubuntu@ubuntu1:~$ tar -xvf opencr_update.tar.bz2
opencr_update/
opencr_update/waffle.opencr
opencr_update/released_1.2.6.txt
opencr_update/burger_noetic.opencr
opencr_update/om_with_tb3_noetic.opencr
opencr_update/open_manipulator.opencr
opencr_update/opencr_ld_shell_arm
opencr_update/burger.opencr
opencr_update/update.sh
opencr_update/waffle_noetic.opencr
opencr_update/om_with_tb3.opencr
opencr_update/opencr_ld_shell_x86
ubuntu@ubuntu1:~$
```

4. ROS 개발환경 구축

⑥ Upload firmware to the OpenCR

```
$ cd ./opencr_update
```

```
$ ./update.sh $OPENCR_PORT $OPENCR_MODEL.opencr
```

```
opencr_update/opencr_ld_shell_x86
ubuntu@ubuntu1:~$ cd ./opencr_update
ubuntu@ubuntu1:~/opencr_update$ ./update.sh $OPENCR_PORT $OPENCR_MODEL.opencr
aarch64
arm
OpenCR Update Start..
opencr_ld_shell ver 1.0.0
opencr_ld_main
[ ] file name      : burger_noetic.opencr
[ ] file size       : 183 KB
[ ] fw_name         : burger_noetic
[ ] fw_ver          : 1.2.6
[OK] Open port      : /dev/ttyACM0
[ ]
[ ] Board Name      : OpenCR R1.0
[ ] Board Ver        : 0x17020800
[ ] Board Rev         : 0x00000000
[OK] flash_erase     : 0.92s
[OK] flash_write     : 1.59s
[OK] CRC Check       : 12A5C20 12A5C20 , 0.005000 sec
[OK] Download
[OK] jump_to_fw
ubuntu@ubuntu1:~/opencr_update$
```

이름 확인

이름 확인

Firmware 완료가 되면 출력됨

4. ROS 개발환경 구축

Remote PC에서 원격으로 Raspbian 연결중에 off 방법

\$ ssh ubuntu@192.168.xxx.xxx (ID : **ubuntu**, PWD : **turtlebot**)

← 이미 진입했으면 생략 (**Path주의**)

\$ sudo shutdown now



```
last login: Mon Nov 27 08:18:25 2023 from 192.168.1.141
ubuntu@ubuntu1:~$ sudo shutdown now
Connection to 192.168.1.145 closed by remote host.
Connection to 192.168.1.145 closed.
yongseok@yongseok:~$
```

Ubuntu에서 yongseok 변경됨



초록색 LED 점멸후 꺼짐
LDS 회전 → 멈춤



OpenCR switch off

Remote PC에서 원격 Raspbian 연결에서 끊는 방법

\$ exit

```
ubuntu@ubuntu1:~/opencr_update$ exit
logout
Connection to 192.168.1.145 closed.
yongseok@yongseok:~$
```

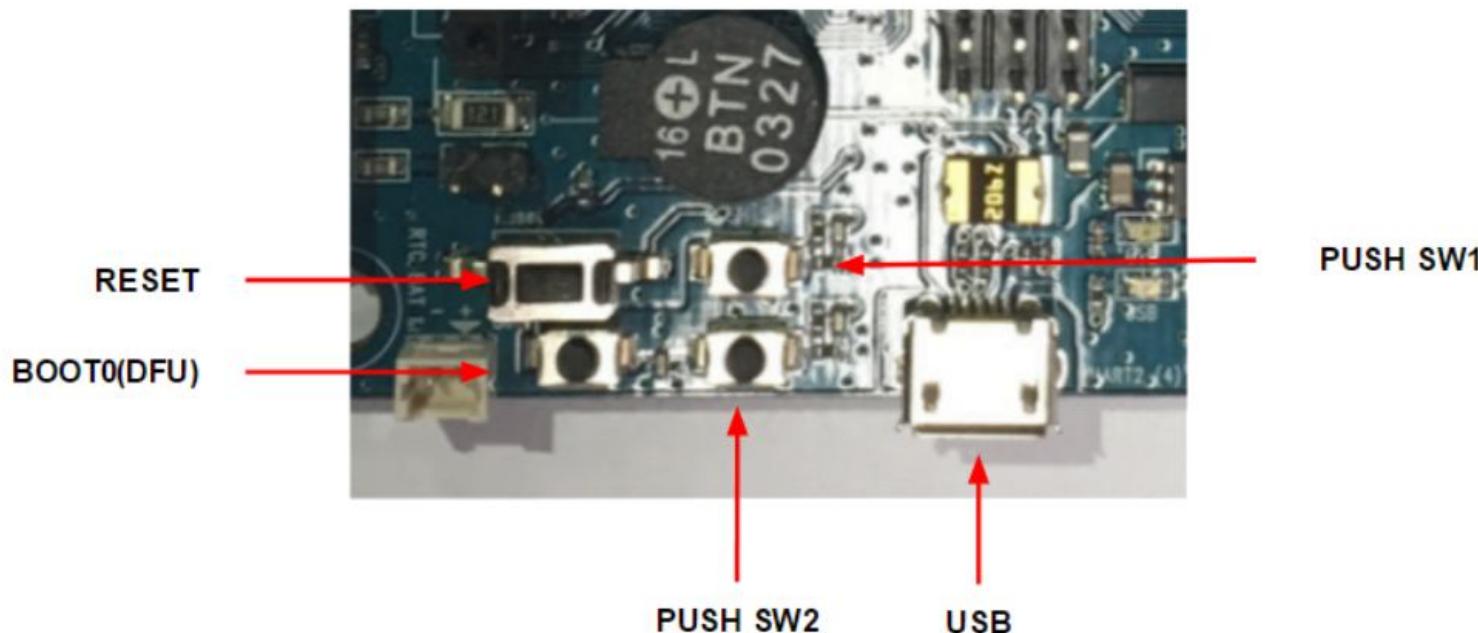
4. ROS 개발환경 구축

⑦ 만일 firmware 업로드 실패하면, (jump to fw message 출력 안되면)

: https://emanual.robotis.com/docs/en/platform/turtlebot3/opencr_setup/#opencr-setup 의 아래 실행하기

If firmware upload fails, try uploading with the recovery mode. Below sequence activates the recovery mode of OpenCR. Under the recovery mode, the **STATUS** led of OpenCR will blink periodically.

- Hold down the **PUSH SW2** button.
- Press the **Reset** button.
- Release the **Reset** button.
- Release the **PUSH SW2** button.



4. ROS 개발환경 구축

4-2-6. Online Competition on ROS Development Studio (RDS)

: <https://emanual.robotis.com/docs/en/platform/turtlebot3/challenges/#events>

(1) Online Competition using TurtleBot3

: are preparing an online competition on **ROS Development Studio (RDS)** with **TurtleBot3 AutoRace and Task Mission** using **TurtleBot3 and OpenManipulator**.
: can participate free of charge in this online competition and learn about **SLAM, Navigatin, Autonomous driving, Manipulation** in a defined rule.

(2) TurtleBot3 AutoRace on RDS

(3) Task Mission using TurtleBot3 and OpenManipulator on RDS

→ 대회 사진 공유



TIP

: 터미널 창 나누기

```
sudo apt-get install terminator
```

- * 수평 분할 : Ctrl + Shift + O
 - * 수직 분할 : Ctrl + Shift + E
 - * 다음 창 활성화 : Ctrl + Tab or Ctrl + Shift + N
 - * 이전 창 활성화 : Ctrl + Shift + Tab or Ctrl + Shift + P
 - * 현재 활성화 된 창 닫기 : Ctrl + Shift + W
 - * 터미네이터 실행 : Ctrl + Alt + T
 - * 터미네이터 종료 : Ctrl + Shift + Q

