

Week1

1장 ROS2 소개

ROS

ROS는 Robot Operating System의 줄임말로 로봇을 위한 오픈소스, 메타운영 시스템이다.

ROS는 윈도우, 리눅스와 같은 독립된 운영체제가 아니고, 기존 운영체제 위에 설치하여 운영체제에서 제공하는 프로세스 관리 시스템이나 UI, 프로그램 유틸리티를 사용하여 로봇 응용프로그램에 필요한 기능들을 지원한다. 이러한 개념을 미들웨어(Middleware) 또는 소프트웨어 프레임워크(Software framework)라고 한다.

ROS의 목적

ROS의 목적은 “로보틱스 소프트웨어 개발을 전 세계 레벨에서 공동 작업이 가능하도록 환경을 구축하는 것”이다. ROS는 로봇 소프트웨어 플랫폼만을 지향하기 보다는 로보틱스 연구, 개발에서의 코드 재사용을 극대화하는 것에 초점이 잡혀있다. 이를 위해 ROS는 노드단위의 분산 프로세스, 공유 및 재배포를 쉽게 하기위한 패키지 단위 관리, 다양한 프로그래밍 언어 지원 기능을 갖추고 있다.

ROS 버전

ROS가 상업 서비스 로봇 시장으로 진입하면서 여러 보안이나 통신과 같은 문제들에 봉착하게 되면서 ROS1은 2020년 ROS Noetic를 마지막 ROS1 버전으로 릴리즈되었다. 이후 여러 문제점을 개선하여 상업 서비스 로봇 시장에서도 견실하게 작동할 ROS2 버전을 릴리즈하였고 현재 ROS2 humble 버전까지 릴리즈되었다.

ROS는 버전에 따라 기본 운영체제가 정해져 있어 사용할 ROS 버전에 따라 기본 운영체제를 설치하여야 한다. 본인은 이번에 ROS2 foxy를 사용하기 때문에 아래의 운영체제를 설치하였다.

- 운영체제 : Ubuntu 20.04.x Focal Fossa

2장 ROS2 기반 로봇 개발에 필요한 정보

ROS는 커뮤니티 중심의 개발문화가 정착된 소프트웨어 플랫폼이기 때문에 쉽게 참여할 수 있고 개방적이라는 장점이 있지만, 처음 접할 때 흩어져 있는 코드와 문서, 자료로 인해 학습하기가 어렵다는 단점도 존재한다.

ROS2 정보 및 개발 자료 사이트

ROS2를 사용하면서 자주 찾게될 사이트를 몇가지 나열해 보았다.

1. ROS 커뮤니티 게시판 <https://discourse.ros.org/>
2. ROS2 공식 문서 <https://docs.ros.org/>
3. ROS2 디자인 문서 <https://design.ros2.org/>
4. ROS2 리포지터리 <https://github.com/ros2>

3장 ROS2 개발환경 구축

ROS2 개발환경을 구축해보자.


기본 운영체제 설치

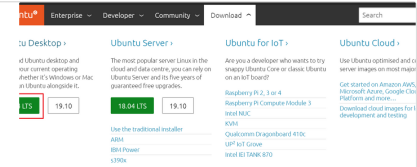
기본 운영체제는 Linux Ubuntu 20.04를 설치하였다. 현재 사용중인 Window가 설치되어 있는 노트북에 설치하기 위해 D drive에 파티션을 분리하여 Grub를 사용하여 멀티 부트가 되는 방식으로 설치를 진행하였다.

설치는 아래의 링크를 이용하여 설치를 진행하였다.

[Linux] 리눅스 우분투(Ubuntu) 설치하기 for PC (듀얼 운영체제)

PC의 운영체제는 윈도우가 거의 필수적으로 자리를 잡았습니다. 하지만 PC 운영체제가 윈도우만 있는 것은 아니고 리눅스라는 운영체제도 있습니다. 개인 PC에서 주 운영체제로 리눅스를 사용하기는 불편한 점이 많지만 실무에서 서버를 구성하고 있는 OS로 리눅스를 많이 채택하고 있으므로 공부 목적으로라도 리눅스라는 플랫폼의 운영체제를 사


 <https://coding-factory.tistory.com/494>

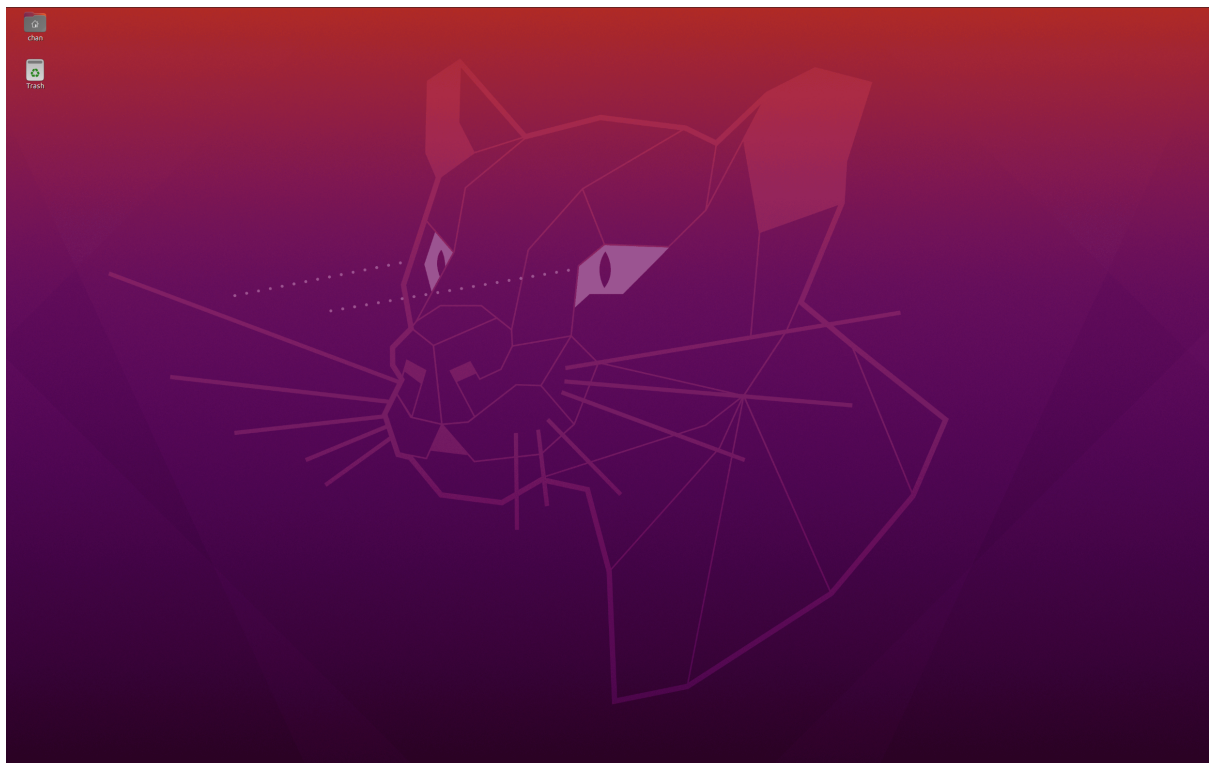


설치를 진행할때 우분투 설치용 USB를 만들어야 했는데 아래의 프로그램을 사용하였다.

[Linux] 리눅스 우분투(Ubuntu) 설치하기 for PC (듀얼 운영체제)

PC의 운영체제는 윈도우가 거의 필수적으로 자리를 잡았습니다. 하지만 PC 운영체제가 윈도우만 있는 것은 아니고 리눅스라는 운영체제도 있습니다. 개인 PC에서 주 운영체제로 리눅스를 사용하기는 불편한 점이 많지만 실무에서 서버를 구성하고 있는 OS로 리눅스를 많이 채택하고 있으므로 공부 목적으로라도 리눅스라는 플랫폼의 운영체제를 사


 <https://coding-factory.tistory.com/494>



ROS2 설치

다음으로 ROS2 foxy를 설치하였다.

ROS2 설치는 기본적으로 아래의 링크를 따라 설치하였다.

 [ROS 2 Foxy 설치 - Linux20.04](#)

```

$ locale # check for UTF-8

$ sudo apt update && sudo apt install locales
$ sudo locale-gen en_US en_US.UTF-8
$ sudo update-locale LC_ALL=en_US.UTF-8 LANG=en_US.UTF-8
$ export LANG=en_US.UTF-8

locale # verify settings

$ sudo apt update && sudo apt install curl gnupg2 lsb-release -y
$ sudo curl -sSL https://raw.githubusercontent.com/ros/rosdistro/master/ros.key -o /usr/share/keyrings/ros-archive-keyring.gpg

$ sudo sh -c 'echo "deb [arch=$(dpkg --print-architecture)] http://packages.ros.org/ros2/ubuntu $(lsb_release -cs) main" > /etc/apt/sources.list.d/ros2.list'
$ sudo apt update

# Desktop Install (Recommended): ROS, RViz, demos, tutorials.
$ sudo apt install ros-foxy-desktop -y

# Install argcomplete (optional)
$ sudo apt install -y python3-pip
$ pip3 install -U argcomplete

```

ROS2 워크스페이스 빌드

ROS2 설치를 완료하고 “ros2_ws”라는 이름의 워크스페이스 폴더를 생성한 후 빌드를 진행해보았다.

워크스페이스 안에는 “ros_tutorials”라는 예시 패키지를 git에서 가져와서 실제로 빌드가 잘되는지 확인해보았다.

```

$ source /opt/ros/foxy/setup.bash

$ mkdir -p ~/ros2_ws/src
$ cd ~/ros2_ws/src

$ git clone https://github.com/ros/ros_tutorials.git -b foxy-devel
$ ls ros_tutorials
> roscpp_tutorials  rospy_tutorials  ros_tutorials  turtlesim

$ cd ../
$ rosdep install -i --from-path src --rosdistro foxy -y
> All required rosdeps installed successfully

$ colcon build --symlink-install
$ ls
> build  install  log  src

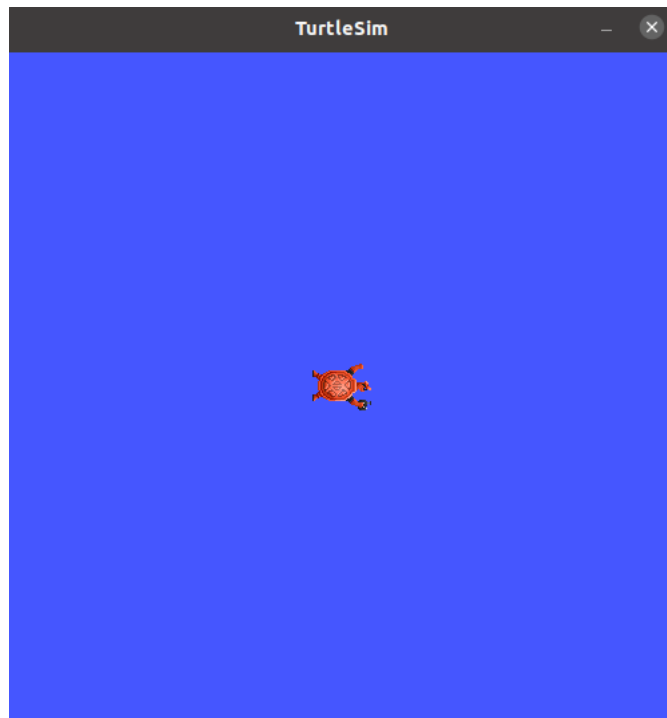
```

```

source /opt/ros/foxy/setup.bash
ros2 run turtlesim turtlesim_node

# new terminal
source /opt/ros/foxy/setup.bash
ros2 run turtlesim turtle_teleop_key

```



```
Reading from keyboard
-----
Use arrow keys to move the turtle.
Use G|B|V|C|D|E|R|T keys to rotate to absolute orientations. 'F' to cancel a rotation.
'Q' to quit.
```

Run commands 설정

\$ `source /opt/ros/foxy/setup.bash` 와 같은 반복적으로 사용할 running code(rc)를 편리하게 사용하기 위해 rc를 모아두는 `bashrc` 파일에 ROS2에서 많이 사용할 rc를 넣어 주었다.

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

```
alias eb='gedit ~/.bashrc'
alias sb='source ~/.bashrc'

alias cba='colcon build --symlink-install'
alias cbp='colcon build --symlink-install --packages-select'
alias killg='killall -9 gzserver && killall -9 gzclient && killall -9 rosmaster'

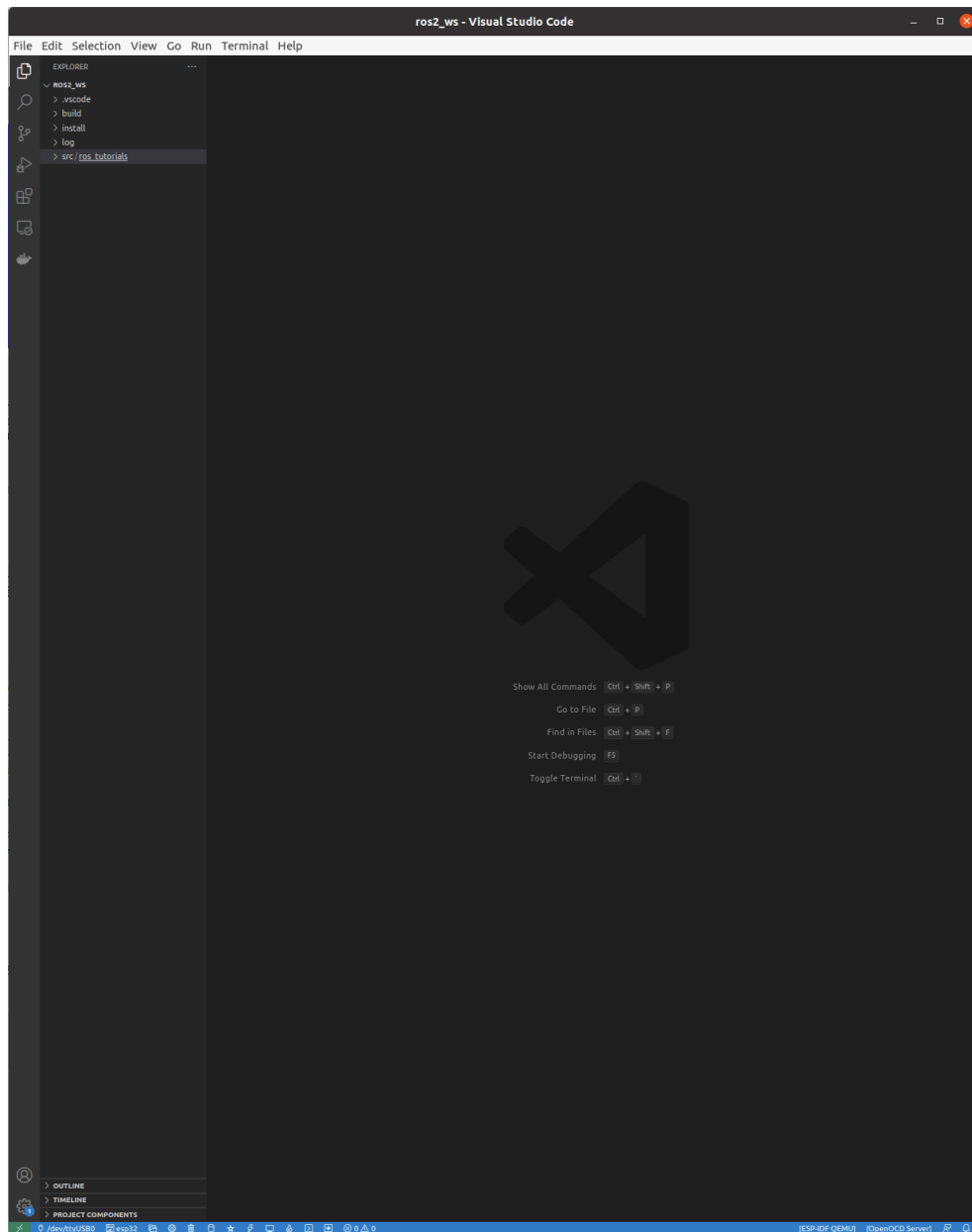
alias rosfoxy='source /opt/ros/foxy/setup.bash && source ~/ros2_ws/install/local_setup.bash'

source /usr/share/colcon_cd/function/colcon_cd.sh
export _colcon_cd_root=~/.ros2_ws
```

통합 개발환경(IDE) 설치

ROS2 개발환경에서 IDE는 Visual Studio Code를 사용하였다.

<https://code.visualstudio.com/>



4장 왜? ROS2로 가야 하는가?

현재 ROS1은 Noetic을 마지막으로 개발이 종료되고 유지 보수만 진행중이고 사용자 커뮤니티 중심인 ROS는 현재 많은 사람들이 ROS2로 개발을 진행 중이기 때문에 ROS1 보다는 ROS2를 사용하는 것이 바람직하다.