

# 프로젝트 개요 및 환경세팅

2023.딥러닝시스템

# 목차

## 0. 프로젝트 개요

- 프로젝트 목적
- 프로젝트 과정 소개

## 1. ROS란?

- 로봇의 필수 기능 3가지
- ROS 등장

## 2. 임베디드 보드

- 임베디드 보드란?
- Jetson Nano

## 3. Jetson Nano 환경 세팅

- 사용하는 환경
- Ubuntu 설치
- ROS2 설치



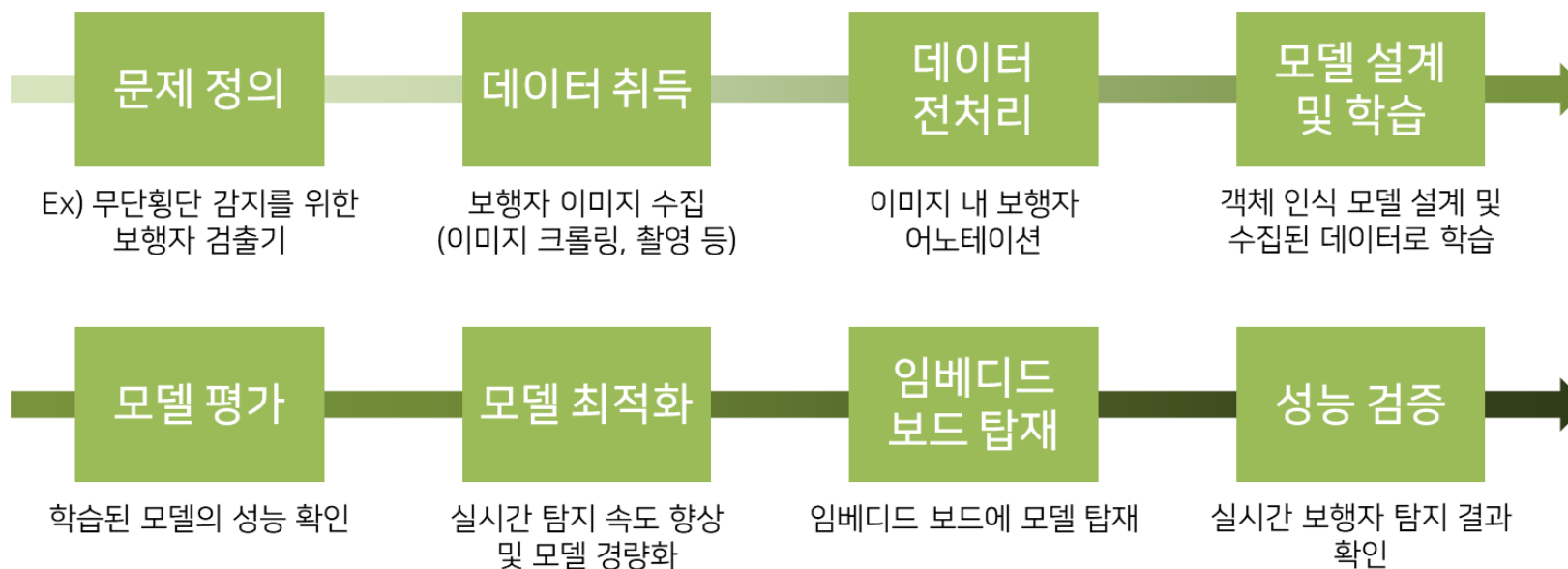
## 프로젝트 개요

# 프로젝트 목적

## 목적

- 인공지능을 활용한 서비스 설계 및 구현 과정을 경험해보기 위한 프로젝트
- 임베디드 보드를 활용하여 시각인지 SW

### 인공지능이 산업 현장에 활용되는 과정



# 프로젝트 과정

- ROS 기초 이론 및 실습

- ROS란 무엇인가
- 메시지별 통신

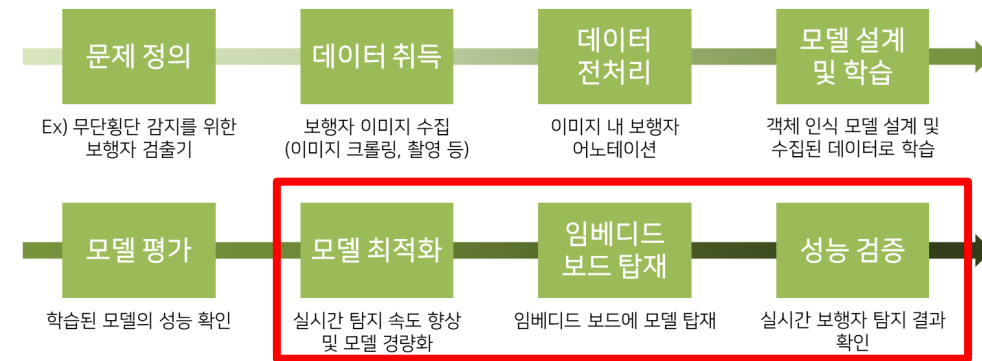
- ROS 응용

- SSD 모델 관련 코드 설명 및 최적화 관련 설명
- 사전 학습된 Detection 모델(SSD)을 이용한 실시간 객체 탐지 실습

- 프로젝트 진행

- 문제 정의 및 임베디드 보드를 활용한 SW 구현
- 목적에 맞는 데이터 셋 수집 및 모델 학습

## 인공지능이 산업 현장에 활용되는 과정



---

**ROS란?**

---

# 로봇의 필수 기능 3가지

- **인지** - 센서
  - 주변 환경과 상황을 인지하기 위해 오감의 역할을 하는 센서
- **판단** - 소프트웨어
  - 인지를 통해 입력된 정보에 대해 판단하기 위한 소프트웨어
- **행동** - 액츄에이터
  - 판단을 기반으로 직접적인 행동을 취하는 액츄에이터



# 로봇의 필수 기능 3가지

- **인지** - 센서
  - 주변 환경과 상황을 인지하기 위해 오감의 역할을 하는 센서
- **판단** - 소프트웨어
  - 인지를 통해 입력된 정보에 대해 판단하기 위한 소프트웨어
- **행동** - 액츄에이터
  - 판단을 기반으로 직접적인 행동을 취하는 액츄에이터

모든 것 개발 능력

Or

각 분야 전문가들과  
긴밀한 협력





# ROS 등장

- ROS란?

- "ROS는 로봇을 위한 오픈소스, 메타 운영 시스템이며, 하드웨어 추상화, 하위 디바이스 제어, 로봇틱스에서 많이 사용되는 센싱, 인식, 지도 작성, 모션 플래닝 등의 기능 구현과 프로세스 사이의 메시지 패싱, 패키지 관리, 개발환경에 필요한 라이브러리와 다양한 개발, 디버깅 도구를 제공한다."

- ROS 위키

- ROS의 목적

- "로보틱스 소프트웨어 개발을 전 세계 레벨에서 공동 작업이 가능하도록 하는 환경을 구축하는 것"

→ 로보틱스 연구, 개발에서 코드의 재사용 극대화에 초점을 둔 SDK

→ 이를 위해 노드단위의 분산 프로세스, 쉬운 공유 및 재배포를 위한 패키지단위 관리, 다양한 프로그래밍 언어 지원 기능을 갖추

# ROS 등장

---

- ROS의 장점
  - Linux, Mac, Window 다양한 운영체제 지원
  - Python, C 등 다양한 프로그래밍 언어 지원
  - 패키지 지향으로 인한 높은 재사용성
  - 활발한 커뮤니티 교류
  - 오픈 소스

# 임베디드 보드

# 임베디드 보드란?

- 용어 정의

- **Embedded Board**: PC 이외의 장비에 사용되는 칩이 내장된 PCB 등의 메인보드
- **=Single Board**: CPU, 램 등의 PC의 기본적 부품이 한 보드에 내장된 형태의 메인보드
- **= Development Board**: 센서와 구동계 등을 연결하고 코딩을 하는 등의 특정 시스템 개발을 목적으로 만든 메인보드

- 예시



아두이노 우노



임베디드 보드를 이용한 가전 제품

# Jetson Nano

- NVIDIA
  - GPU, 임베디드 보드 등을 디자인하는 반도체 회사
  - GPU 분야를 선도
  - 임베디드

<https://www.nvidia.com/ko-kr/autonomous-machines/embedded-systems/>

- Jetson Nano

- <https://developer.nvidia.com/embedded/jetson-nano-developer-kit>



GPU	128-core Maxwell
CPU	Quad-core ARM A57 @ 1.43 GHz
Memory	4 GB 64-bit LPDDR4 25.6 GB/s
Storage	microSD (not included)
Video Encode	4K @ 30   4x 1080p @ 30   9x 720p @ 30 (H.264/H.265)
Video Decode	4K @ 60   2x 4K @ 30   8x 1080p @ 30   18x 720p @ 30 (H.264/H.265)
Camera	2x MIPI CSI-2 DPHY lanes
Connectivity	Gigabit Ethernet, M.2 Key E
Display	HDMI and display port
USB	4x USB 3.0, USB 2.0 Micro-B
Others	GPIO, I <sup>2</sup> C, I <sup>2</sup> S, SPI, UART
Mechanical	69 mm x 45 mm, 260-pin edge connector

---

# **Jetson Nano 환경 세팅**

---

# 사용하는 환경


---

- 리눅스(우분투) & ROS(Foxy)
- why? <https://youtu.be/57DpwVfvLBo>  
**리눅스**
  - 무료 공개 운영체제
  - 대부분의 하드웨어를 지원(이식성이 좋음)
  - 많은 사용자

## **ROS**

- 개발시간 단축
  - **노드** 간의 프로토콜 정의
  - 모듈화 ➔ 이식성 높음, 디버깅 유리
- 편리한 디버깅 도구 및 시각화 도구
- 많은 사용자

# 환경 세팅 정보



ROS 2 Documentation: Foxy

Search docs

Installation

Ubuntu (Debian)  
Windows (binary)

Alternatives

- Maintain source checkout
- Testing with pre-release binaries

DDS implementations

Distributions

Tutorials

How-to Guides

Concepts

Contact

[Home](#) / Installation[Edit on GitHub](#)

You're reading the documentation for an older, but still supported, version of ROS 2. For information on the latest version, please have a look at [Humble](#).

## Installation

Options for installing ROS 2 Foxy Fitzroy:

### Binary packages

Binaries are only created for the Tier 1 operating systems listed in [REP-2000](#). If you are not running any of the following operating systems you may need to build from source or use a [container solution](#) to run ROS 2 on your platform.

We provide ROS 2 binary packages for the following platforms:

- Ubuntu Linux - Focal Fossa (20.04)
  - [Debian packages](#) (recommended)
  - ["fat" archive](#)
- [macOS Mojave \(10.14\)](#)
- [Windows \(VS 2019\)](#)

### Building from source



# 환경 세팅

---

## 1. Nano에 Ubuntu20.04 설치하기

1. Nano 보드에 넣을 SD카드에 Ubuntu20.04 이미지 올리기
2. Nano 보드에 넣고 부팅하기

## 2. Ubuntu20.04에 ROS2 설치하기

- 방법1: ROS2 매뉴얼을 따라 ROS2설치
  - ROS2 Documentation
    -  <https://docs.ros.org/en/foxy/Installation/Ubuntu-Install-Debians.html>
- 방법2: 제공된 Github를 이용하여 ROS2 설치
  - Github
    -  <https://github.com/xown3197/JetsonNano-ROS2>

# Ubuntu 20.04 설치

---

## 1. Ubuntu 20.04 image 준비

- 아래의 링크에서 **JetsonNanoUb20\_2.img.xz** (7.9GByte) 다운
- [🔗 https://github.com/Qengineering/Jetson-Nano-Ubuntu-20-image](https://github.com/Qengineering/Jetson-Nano-Ubuntu-20-image)

## 2. 필요한 프로그램 설치

- a. SD Card 포맷을 위한 SD Card Formatter 프로그램 설치

[🔗 https://www.sdcard.org/downloads/formatter/eula\\_windows/](https://www.sdcard.org/downloads/formatter/eula_windows/)

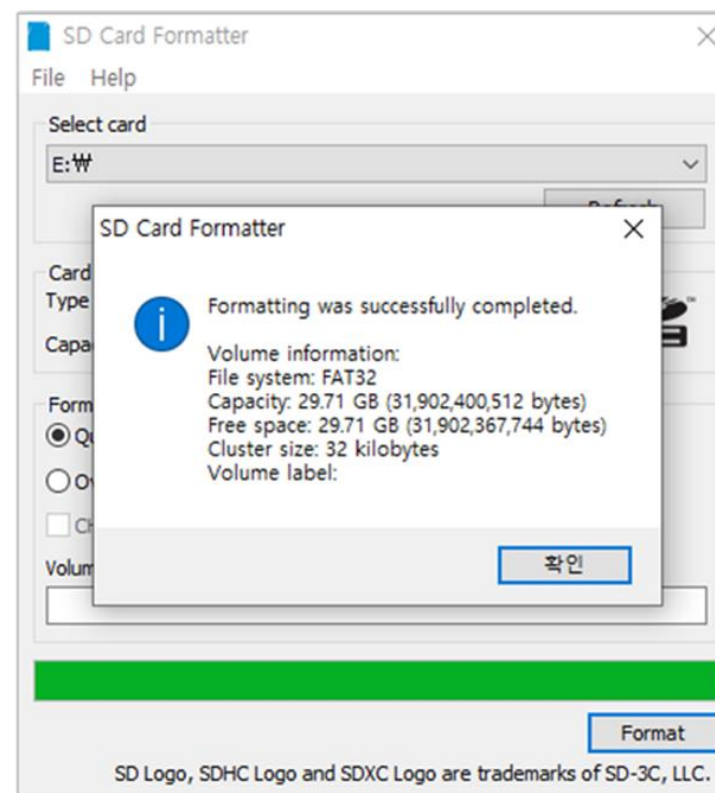
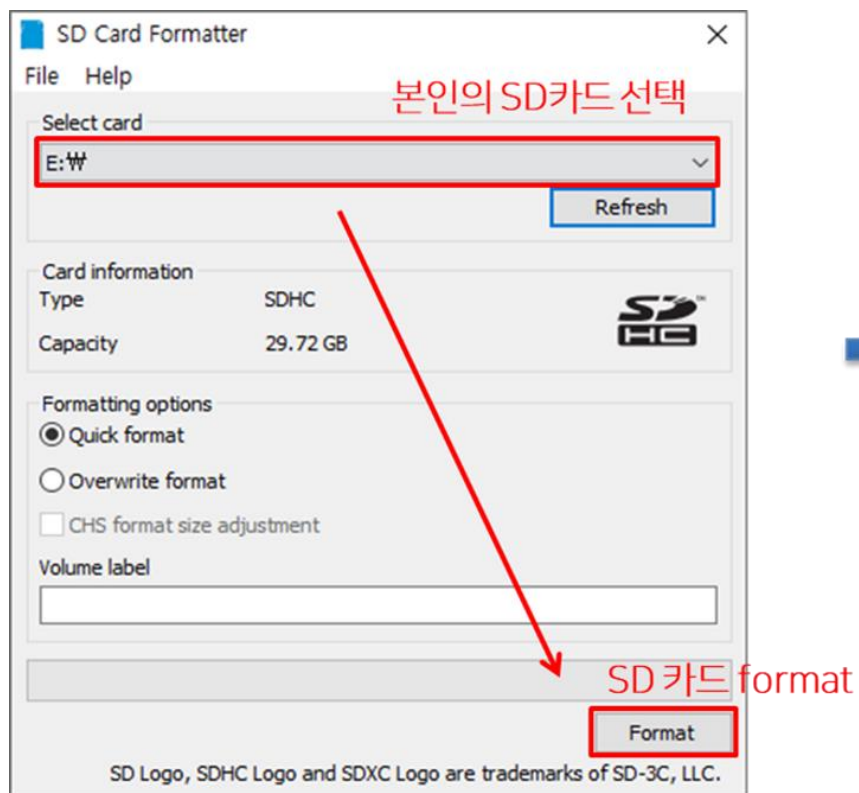
- b. 이미지를 Flash하기 위한 Balena Etcher 프로그램 설치

[🔗 http://www.balena.io/etcher/](http://www.balena.io/etcher/)

# Ubuntu 20.04 설치

## 3. SD Card 포맷

- ① SD 카드와 컴퓨터 연결
- ② SD CardFormatter 프로그램 실행
- ③ 아래 이미지와 같이 포맷 수행

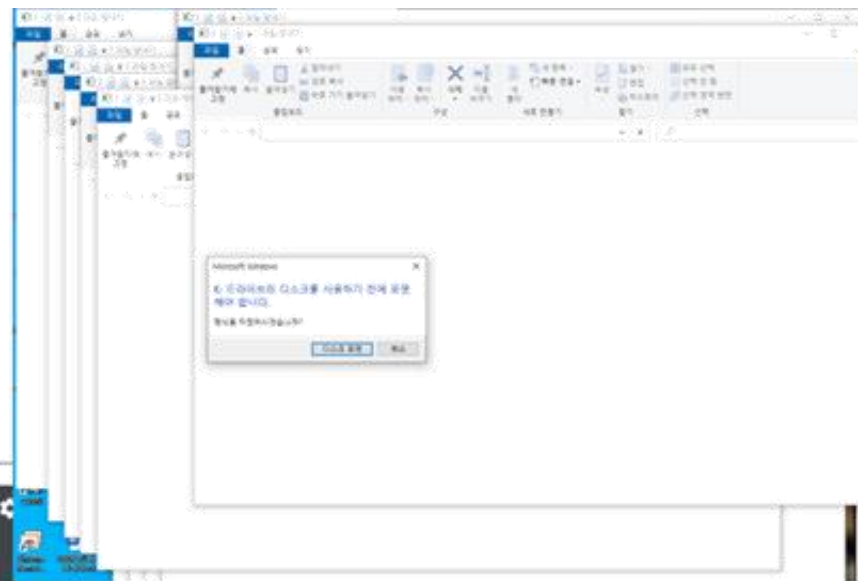
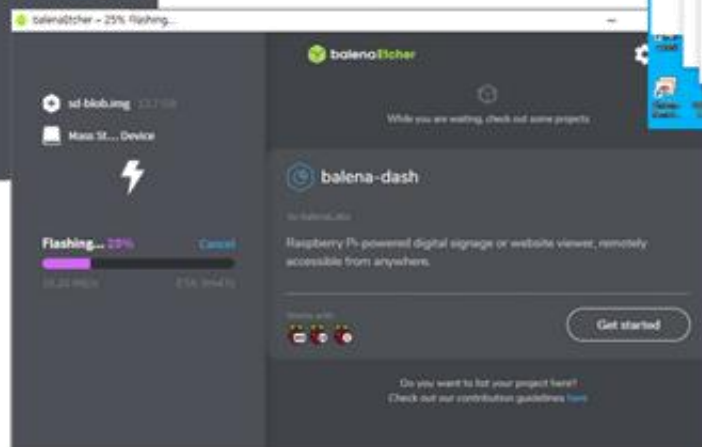


Format 완료 시 화면

# Ubuntu 20.04 설치

## 4. SD Card에 이미지 담기

- ① Balena Etcher 프로그램 실행
- ② 'Flash from file'에 앞서 다운받은 **JetsonNanoUb20\_2.img.xz** 이미지 파일 선택
- ③ 'Select target'에서 SD 카드 선택
- ④ 'Flash' 선택



flash 완료 화면

\* 이때 많은 창이 떠도  
모두 취소하면 됨.

# ROS 2 설치 - (1) Documentation 이용

---


- **ROS2 Foxy Documentation**

-  <https://docs.ros.org/en/foxy/Installation/Ubuntu-Install-Debians.html>

- **Gazebo install**

-  [https://classic.gazebosim.org/tutorials?tut=install\\_ubuntu](https://classic.gazebosim.org/tutorials?tut=install_ubuntu)

- **Colcon install**

-  <https://docs.ros.org/en/foxy/Tutorials/Beginner-Client-Libraries/Colcon-Tutorial.html#install-colcon>

# ROS 2 설치 - (2) Github 이용

```
#!/bin/bash
```

```
install_ros2_jetson_nano.sh
```

```
sudo apt update
```

```
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-latest.list'
```

```
sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys F42ED6FBAB17C654
```

```
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 install

```
# Install Gazebo
```

```
sudo sh -c 'echo "deb http://packages.osrfoundation.org/gazebo/ubuntu-stable `lsb_release -cs` main" > /etc/apt/sources.list.d/gazebo-stable.list'
```

```
wget https://packages.osrfoundation.org/gazebo.key -O - | sudo apt-key add -
```

```
sudo apt update
```

```
$ sudo apt install gazebo11 libgazebo11-dev -y
```

```
# Gazebo ROS 패키지들도 설치해줍니다.
```

```
sudo apt install ros-foxy-gazebo-ros-pkgs -y
```

Gazebo install

```
# Install Colcon Build system
```

```
sudo apt install python3-colcon-common-extensions
```

Colcon install

# ROS 2 설치 - (2) Github 이용

plus.sh

```
#!/bin/bash

## VSCODE
git clone https://github.com/JetsonHacksNano/installVSCode.git & cd installVSCode & sh ./installVSCode.sh

cd .. & rm -r installVSCode

# [Jetson Nano] cv_bridge 에러 이슈
echo "export LD_PRELOAD=/usr/lib/aarch64-linux-gnu/libgomp.so.1" >> ~/.bashrc
source ~/.bashrc

# Vision task messages package
sudo apt install ros-foxy-vision-msgs

## optional
sudo apt install gedit
```