



**ENGINEERING**

DEPT. OF MECHANICAL ENG.  
SEOUL NATIONAL UNIVERSITY  
서울대학교 기계공학부

# 2023-S UROP

ATV depth estimation

# Table of Contents

- Theoretical Backgrounds
- 알루미늄 판
- 판과 ATV의 결합
- PC 및 hub 등 전자장비 컨테이너
- 조립과정
- 데이터 수집 과정

# Theoretical Background

- 영상 Geometry & Camera calibration
- **Textbook**
  - LIE Theory
  - Multiple View Geometry in Computer Vision
- **Paper**
  - A Flexible New Technique for Camera Calibration

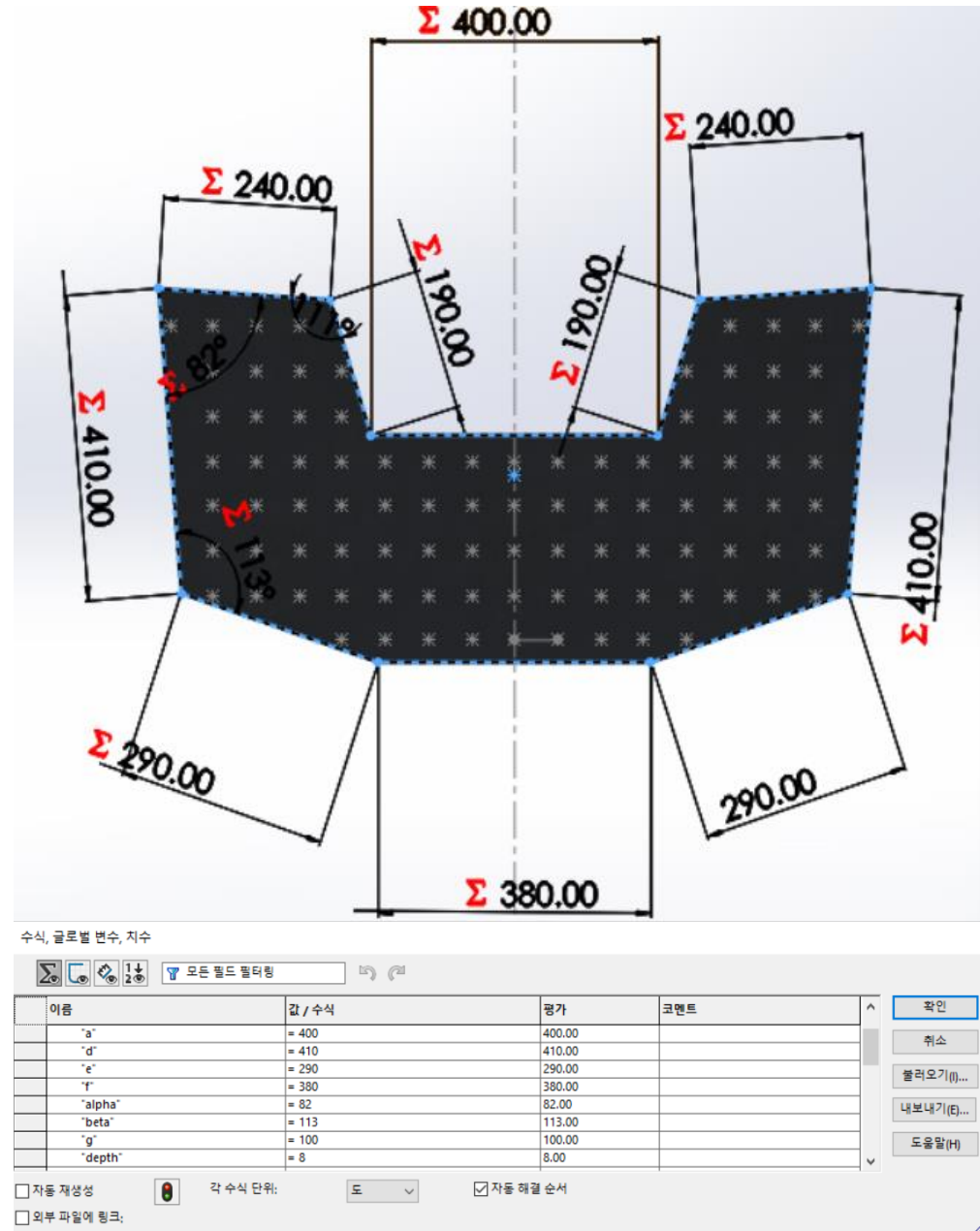
# 판 제작

- NVIDIA® Jetson Orin™
- HUB
- Power Supply
- Camera support structure
- Etc...



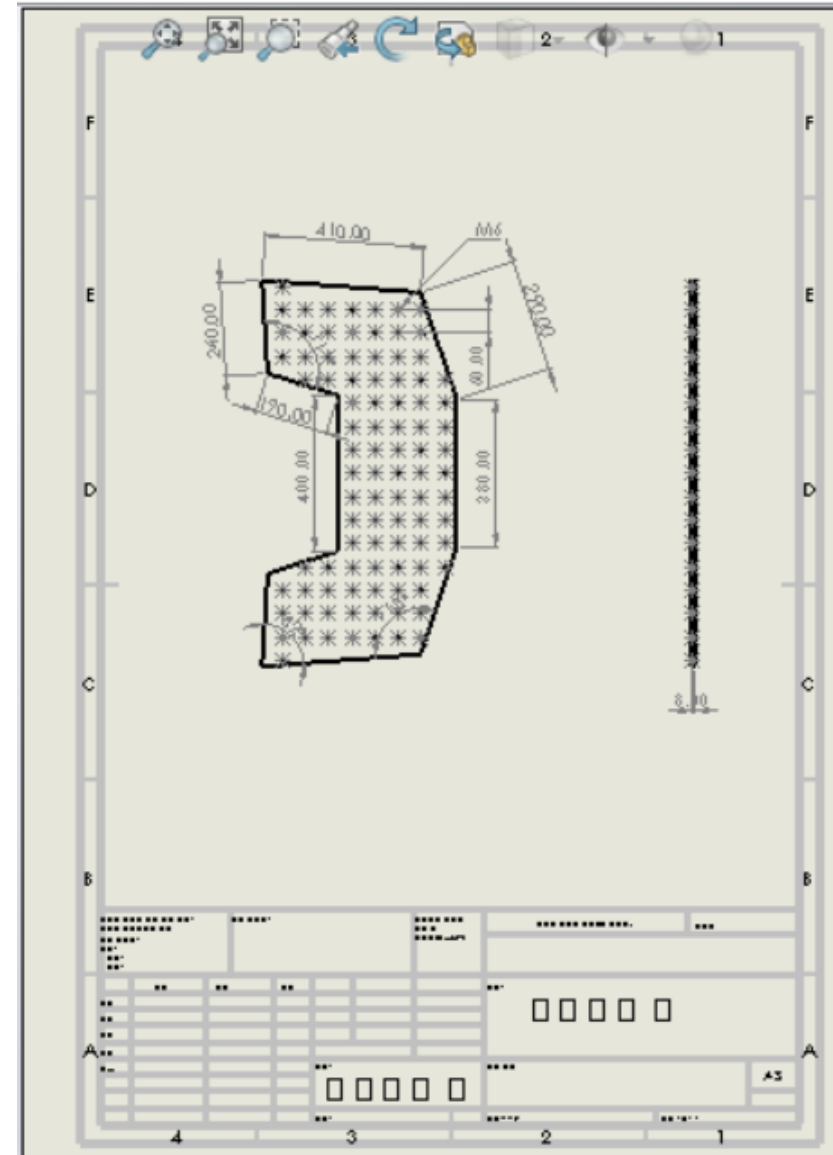
# 판 제작

- SOLIDWORKS2018 사용
- Equation 활용으로 수정 용이
- 0.8 cm의 두께로 가벼운 무게
- M6 구멍 추가하여 조립 용이



# 판 제작

- SOLIDWORKS2018 사용
  - Equation 활용으로 수정 용이
  - 0.8 cm의 두께로 가벼운 무게
  - M6 구멍 추가하여 조립 용이
- 
- **철물점에 주문 제작**



# ATV와 판 결합 (Considering)

## 1. 판의 수평 맞추기

- 카메라의 시야 확보
- 모든 기기들의 수평 확보
- 안정성

## 2. 견고한 고정

- 카메라의 흔들림 방지
- 안정적이고 지속 가능한 연구를 위한 시스템 설계

# ATV와 판 결합 (초기 의도)



## Problem

- ATV 전면부의 경사
- 견고한 고정 불가

## Solution

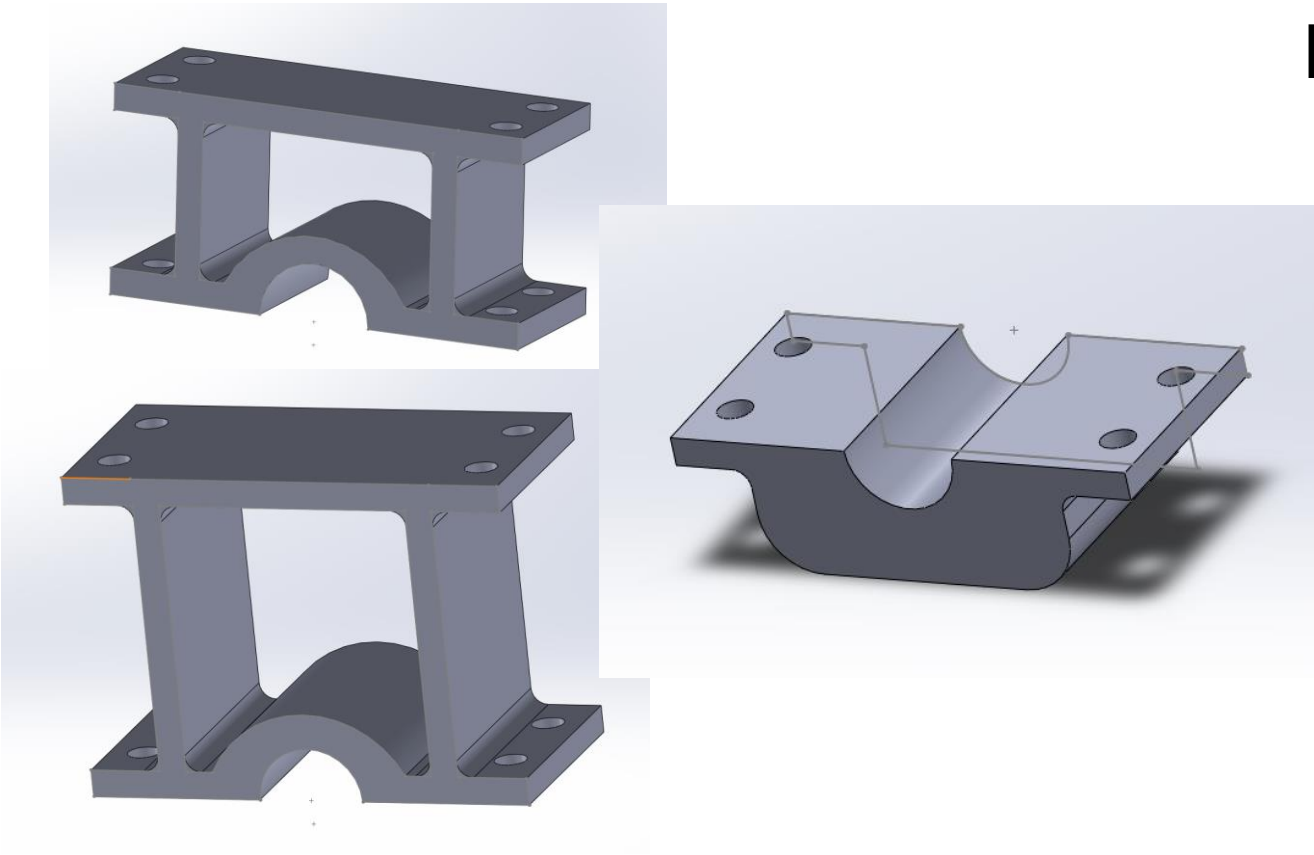
- 판과 ATV를 결합하는 PART 설계



# ATV와 판 결합 (결합부 prototype)

## Feature

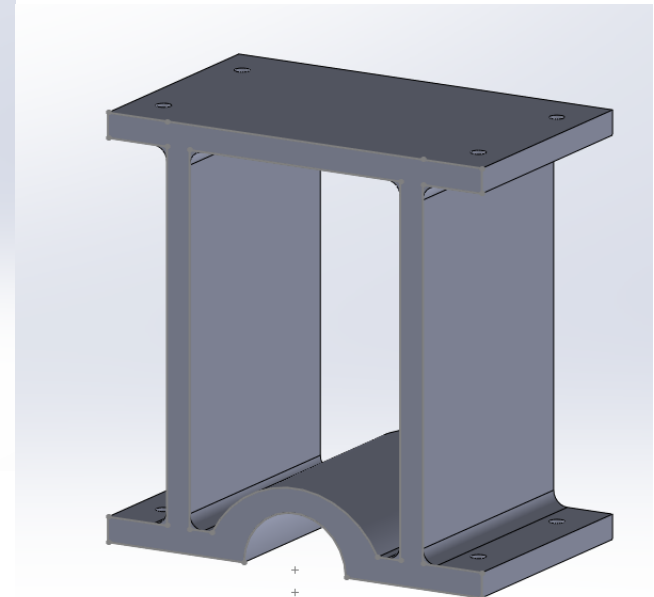
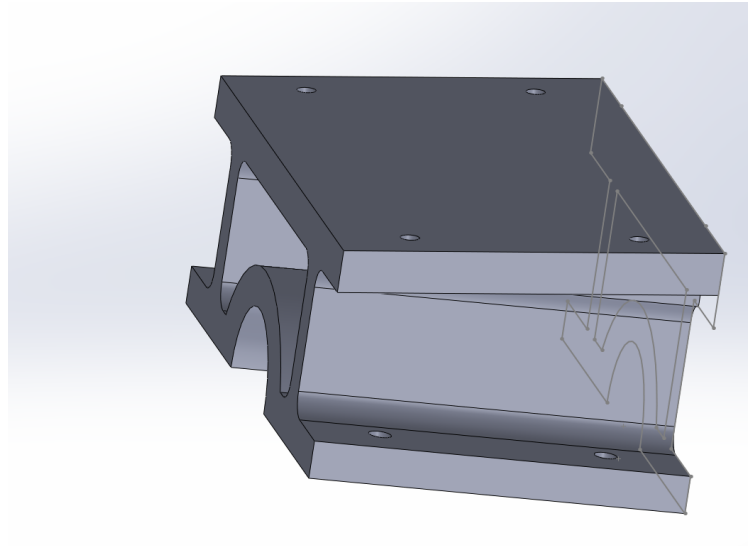
- ATV의 결합 부위마다 다른 설계
- 무게중심을 아래에 위치시키기 위한 hollow 형태로 설계
- 봉과 판에 각각 고정시키기 위한 설계



# ATV와 판 결합 (시행착오)

## 1. 결합부의 개수

- 초기에 결합의 견고함을 위하여 양쪽 부분, 아래 부분을 위한 결합부도 설계하였음



# ATV와 판 결합 (시행착오)

## 2. 높이조절

- 수평을 맞추기 위해서
- 안정성을 확보하고자 ATV와 맞닿게 하기 위해서



## Solution

- 결합부 상단을 다양한 조합으로 Prototype을 제작하여 조립해보면서 가장 적절한 높이를 찾았음

# ATV와 판 결합 (시행착오)

## 3. 실제 결합

- 판의 hole이 일정하게 뚫려 있어 prototype을 고정한 후 구멍과 맞도록 맞추어야 함



## Solution

- 먼저 결합부를 모두 고정시킨 뒤 Prototype에 표시하면서 구멍을 맞춤

# ATV와 판 결합 (실제 제작)

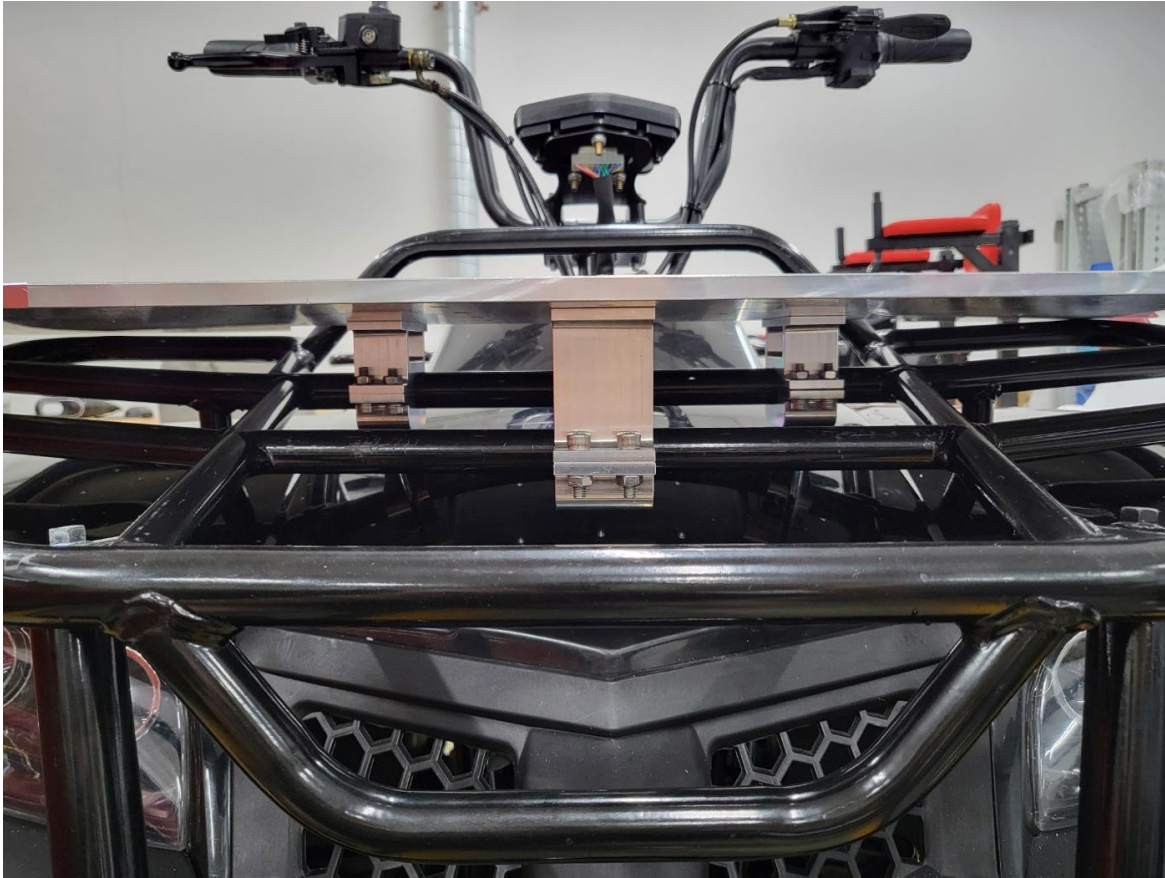


## Reason

- 3D printing 방식에 의한 재료의 약한 강성
- 더욱 견고한 고정
- 더욱 긴 기간 동안의 지속성 확보



# 조립 과정 - 판과 ATV의 결합

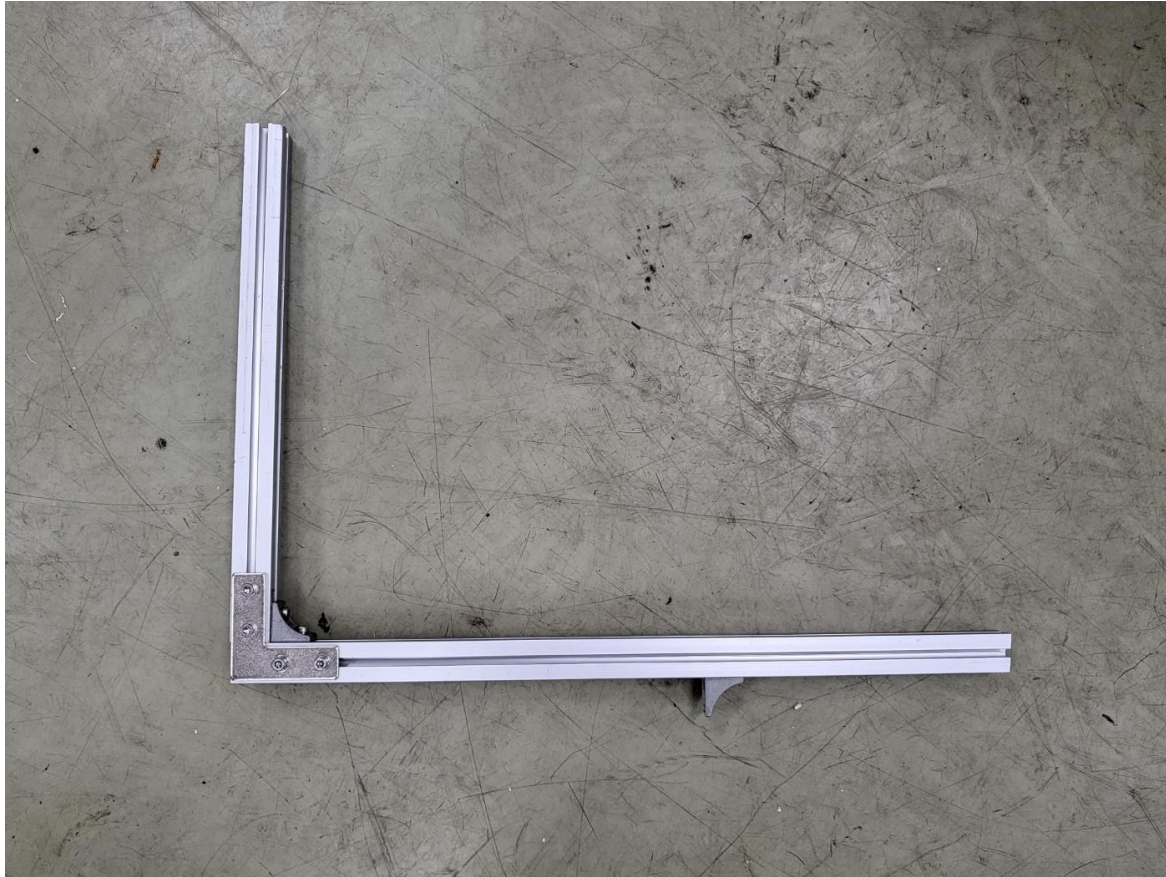


전면

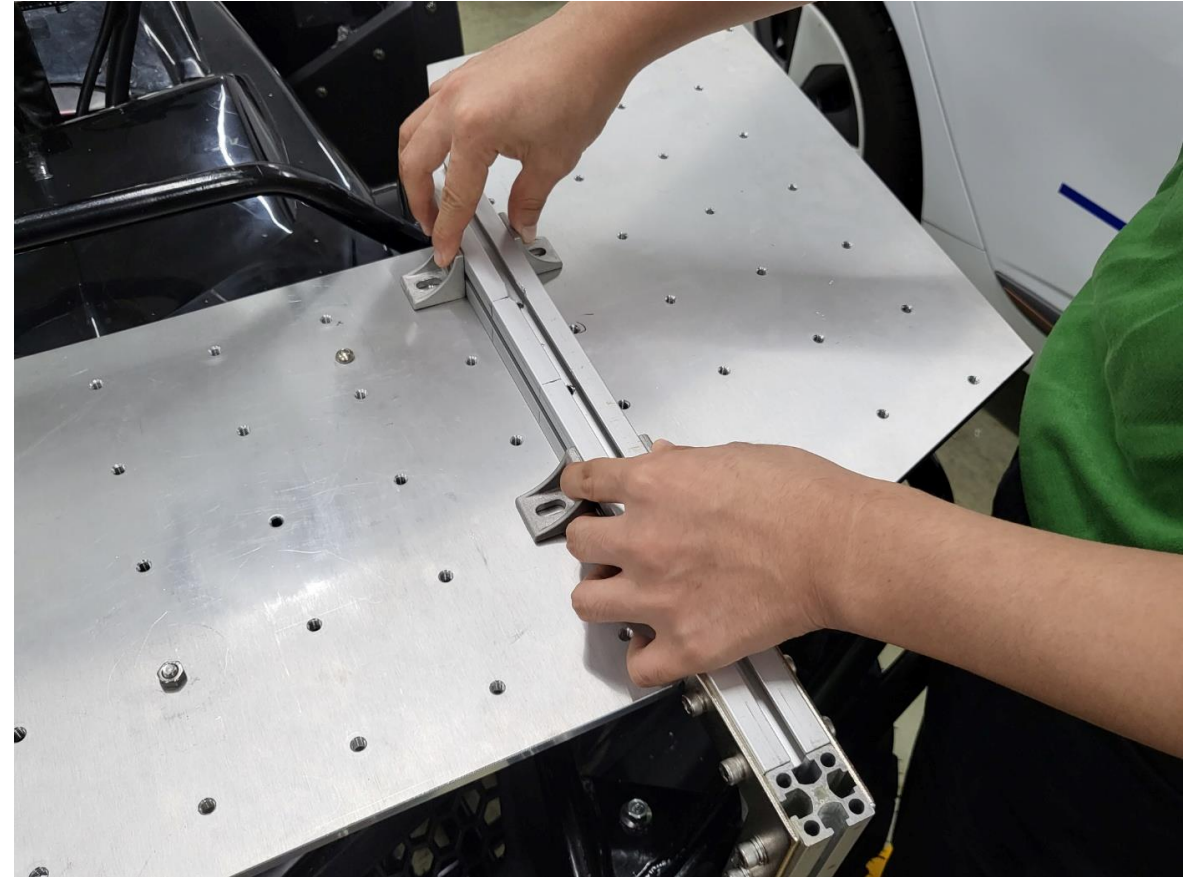


측면

# 조립 과정 - 카메라 결합



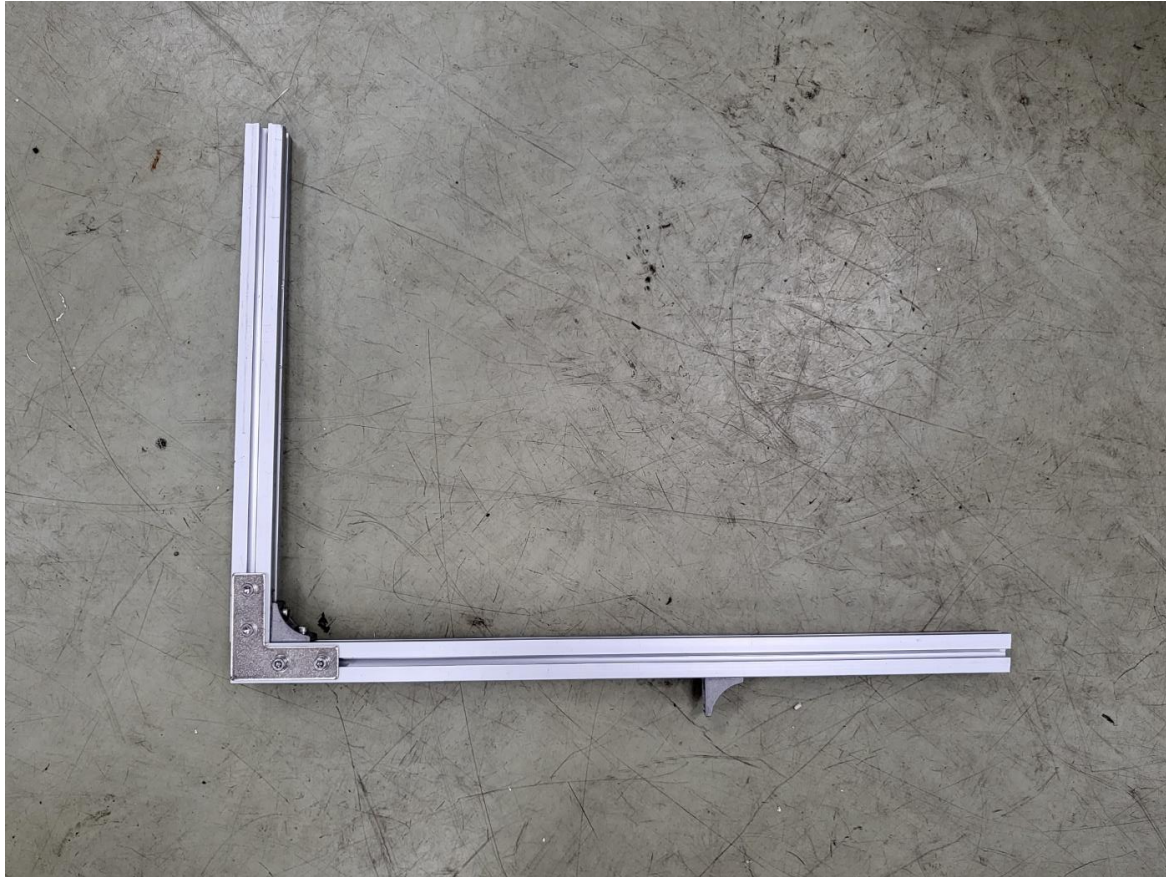
카메라 지지 구조



판과 결합



# 조립 과정 - 카메라 결합



카메라 지지 구조



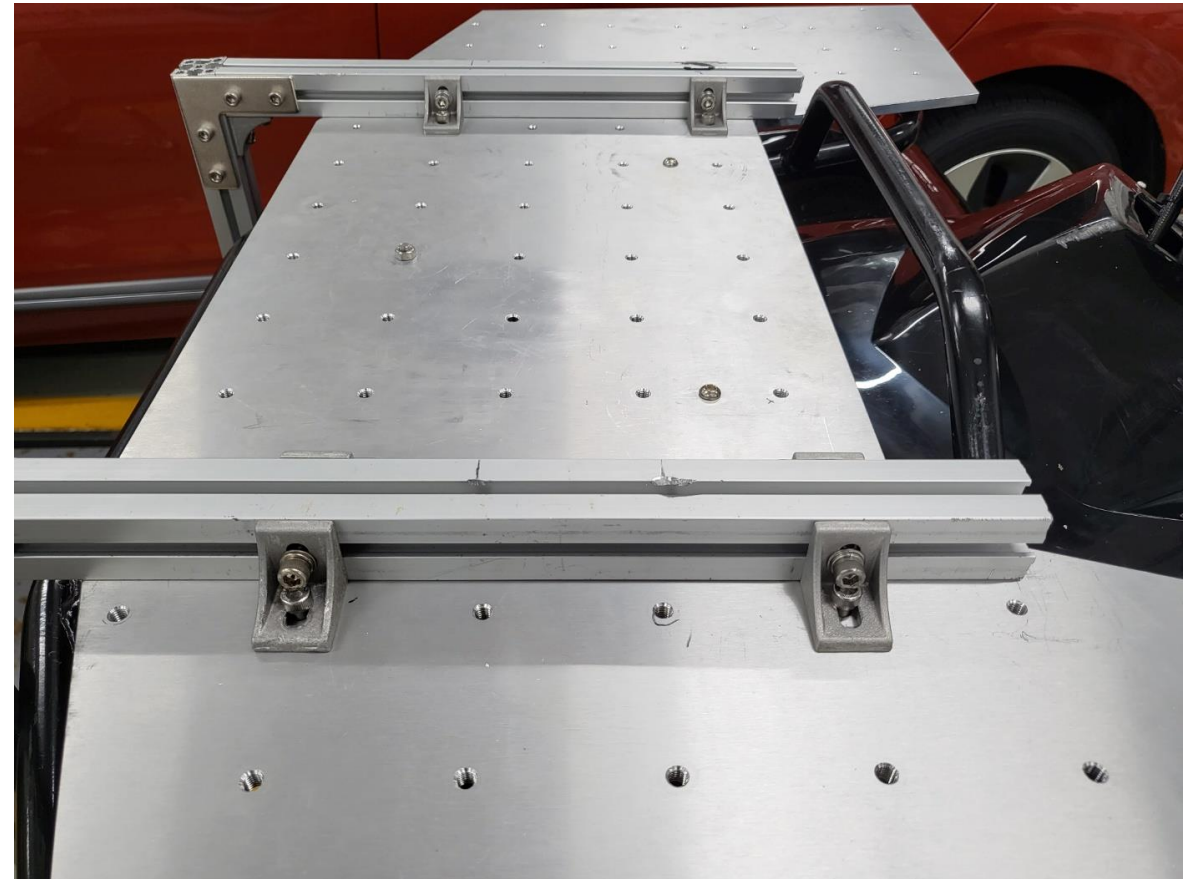
판과 결합



# 조립 과정 - 카메라 결합



전면 뷰



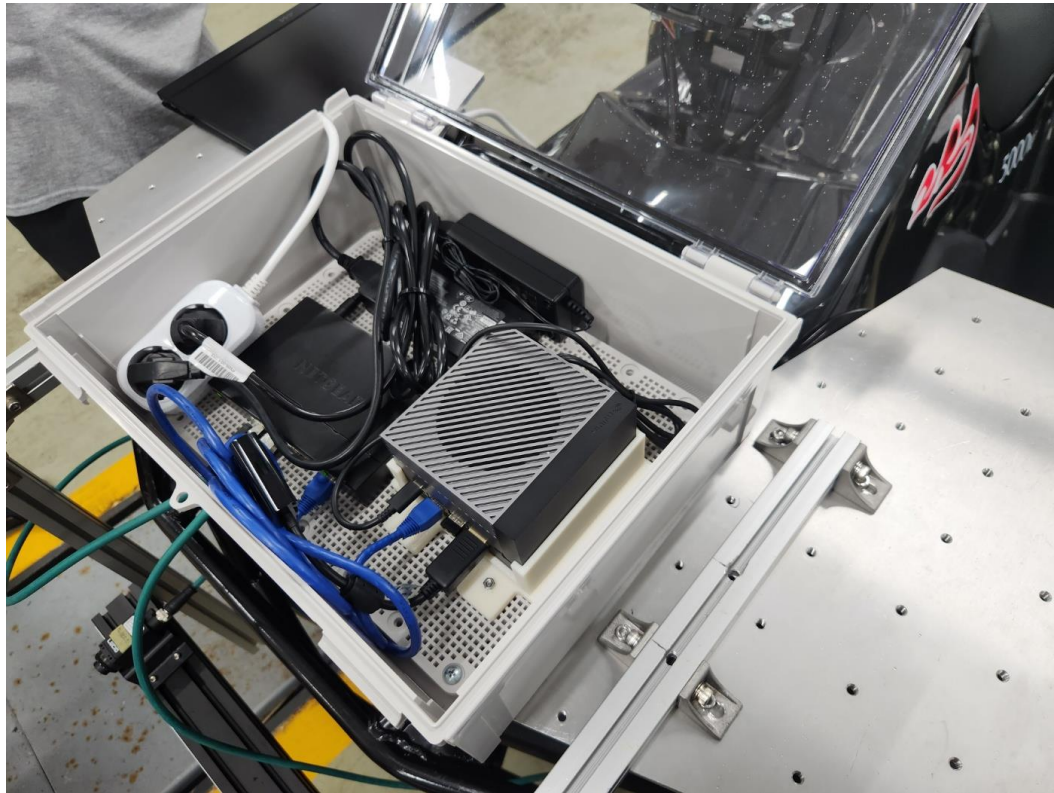
측면 뷰

# 조립 과정 - 완성





# 전자장비 상자



## Object

- 주행에도 기기들이 흔들리지 않도록 상자에 고정

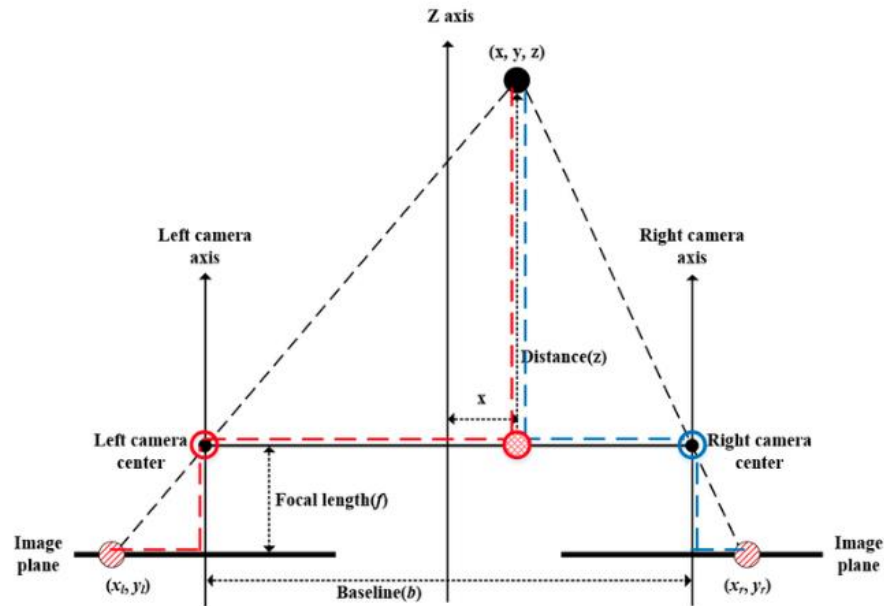
# 전자장비 상자



# NVIDIA® Jetson Orin™ Setup

- Linux 설치 및 각종 드라이버 setup
- Jetson Orin Module Developer Kits 설치
- ROS2 설치
  - 초기에 ROS2 foxy 설치
  - Lucid Vision Labs' ArenaSDK와 호환 문제 발생
  - Camera Driver와 호환되는 ROS2 galactic 재설치(및 Linux 재설치)

# Depth Estimation



1. Matching

2. Computation of Essential Matrix(Camera Calibration)

3. Depth computation

# Camera Calibration

$$s \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{bmatrix} f_x & \gamma & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{matrix} \downarrow \\ \text{Intrinsic} \\ \text{parameter} \end{matrix} [R|t] \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

Homogeneous coordinate(plane)      Extrinsic parameter      Homogeneous coordinate(space)

The diagram illustrates the camera calibration equation. It shows the mapping from a 3D point in space to a 2D point on the image plane. The equation is:  $s \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{bmatrix} f_x & \gamma & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} [R|t] \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$ . Below the equation, four labels are provided with arrows pointing to their respective parts: 'Homogeneous coordinate(plane)' points to the vector  $\begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$ ; 'Intrinsic parameter' points to the  $3 \times 3$  matrix; 'Extrinsic parameter' points to the  $[R|t]$  matrix; and 'Homogeneous coordinate(space)' points to the vector  $\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$ .

# Camera Calibration(Using Ros2)

- Exposure time, gain 등의 설정에서의 어려움
- Ros2의 camera calibration node를 이용한 하나의 카메라에 대해서 Intrinsic parameter 획득
- 두 카메라를 동시에 calibration하여 각각의 Intrinsic parameter와 두 camera의 상대 위치(Extrinsic parameter를 구해보아야 함)



# 데이터 수집 과정(Using Ros2)

- 카메라의 촬영 정보를 Topic으로 받아 bag record를 이용하여 적절한 확장자로 저장
- Topic을 받아 png로 변환하는 역할을 하는 Python source file(Ros2 Node)을 이용하여 이미지 파일을 획득



**Thank you.**