

Lane Keeper with OpenCV

Date : 2021. 09. 17

Writer : 27기 18이기진



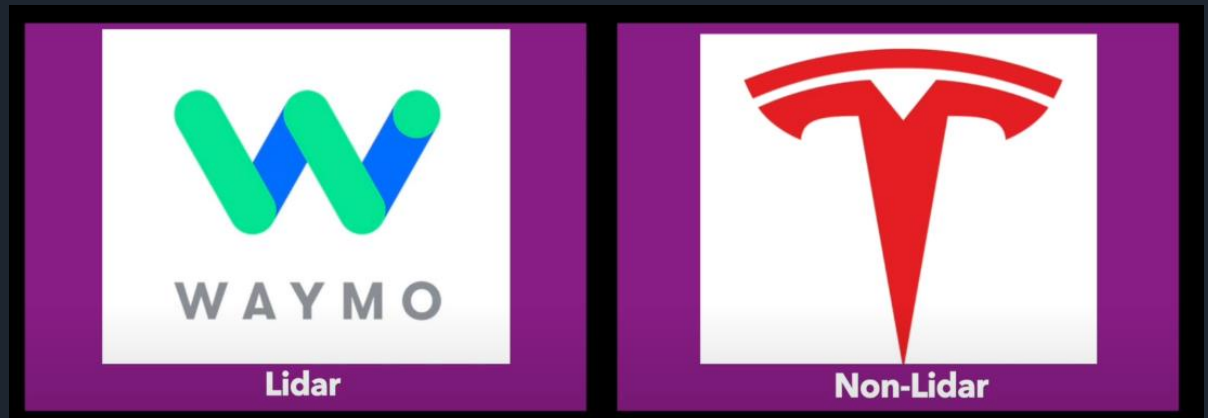
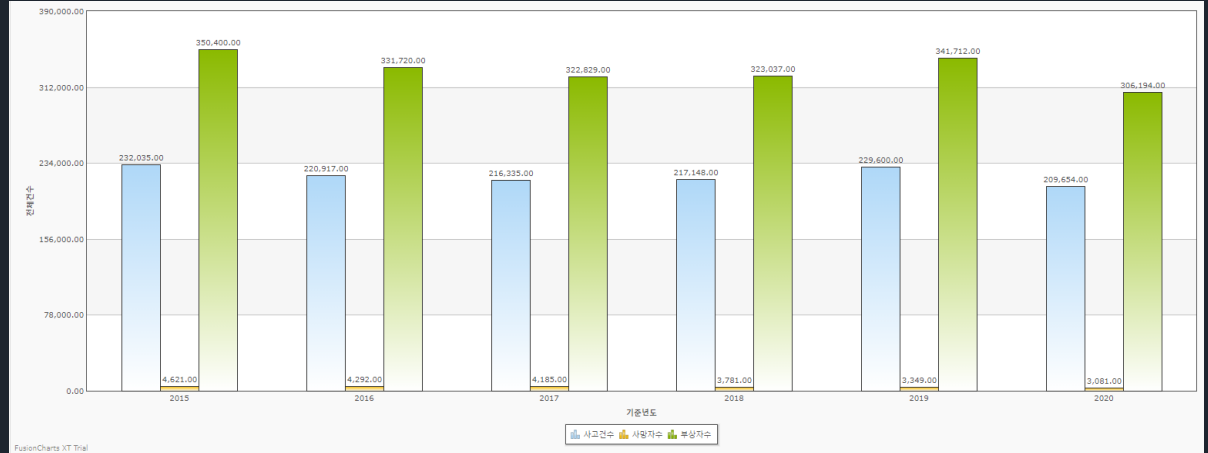
CONTENTS

- Project Introduction
 - Motive for Selection
 - Development Goals
- Project Contents
 - Hardware Architecture
 - System Architecture
 - Project Contents
 - Project Scenarios
- Project Schedule
- Q & A



Motive for Selection

- 교통수단은 인류에게 있어서 불가피한 요소이다.
- 인간의 실수로 인해 그동안 많은 교통 사고 및 피해가 발생해 왔다.
- 자율주행의 구현을 통해 인명 사고 및 경제적 피해를 줄일 수 있다.
- 라이다를 사용하지 않고, 영상처리를 통해 효율적인 자율주행을 구현 하고자 한다.



| Development Goals

1. 차선 인식 및 지키기

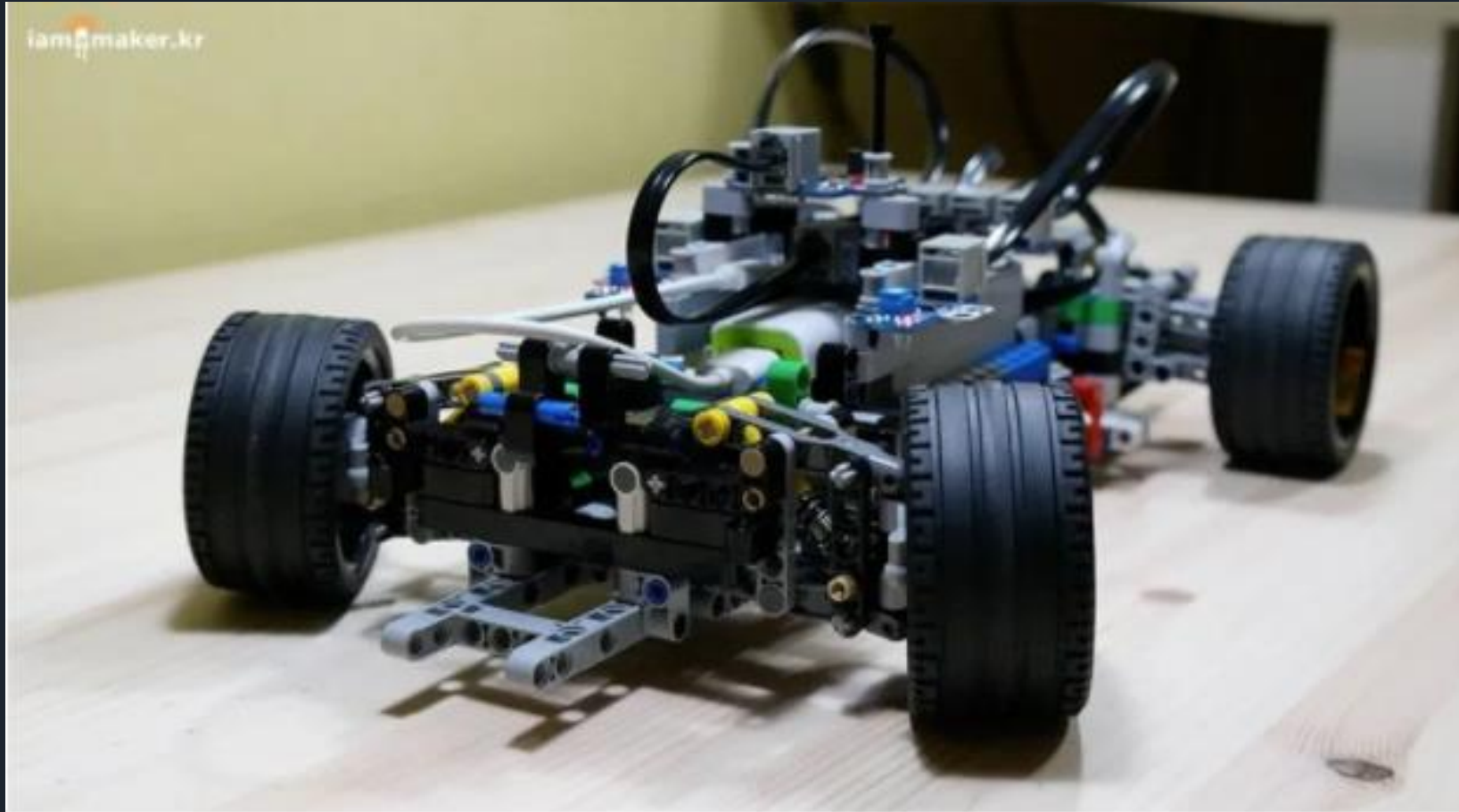
- OpenCV로 양쪽 차선을 인식, 좌표를 추출하여 곡률 및 이탈정보 계산
- 감지한 차선에 맞춰서 조향 및 주행

2. 전방에 갑자기 물체 출현 시 정지

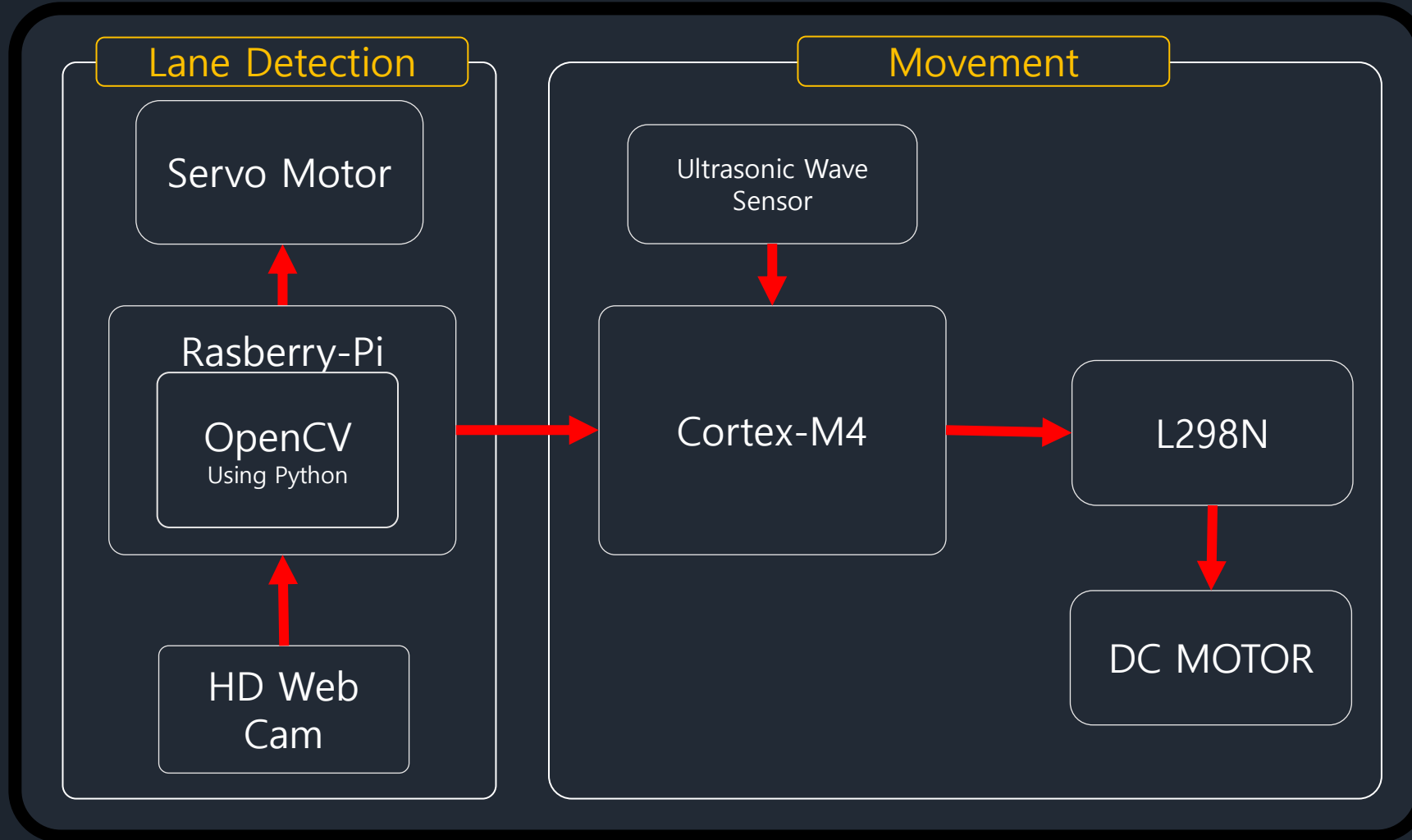
- 초음파 센서로 물체와의 거리를 측정해 돌발 출현하는 물체를 감지 및 차량 정지



| Hardware Architecture



System Architecture

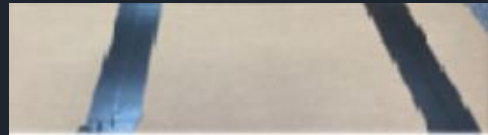


| Project Contents

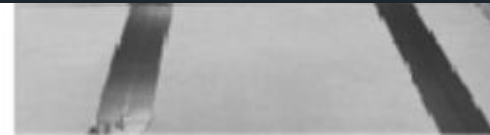
1. 차선 인식 알고리즘 [Lane Detection Algorithm]-1

서킷은 검은 테이프를 바닥에 붙여 만든다.

라즈베리파이에 웹캠을 연결하고, 받아온 영상으로 Python의 OpenCV를 통해 바닥의 차선을 검출한다.



Crop



Gray Scale



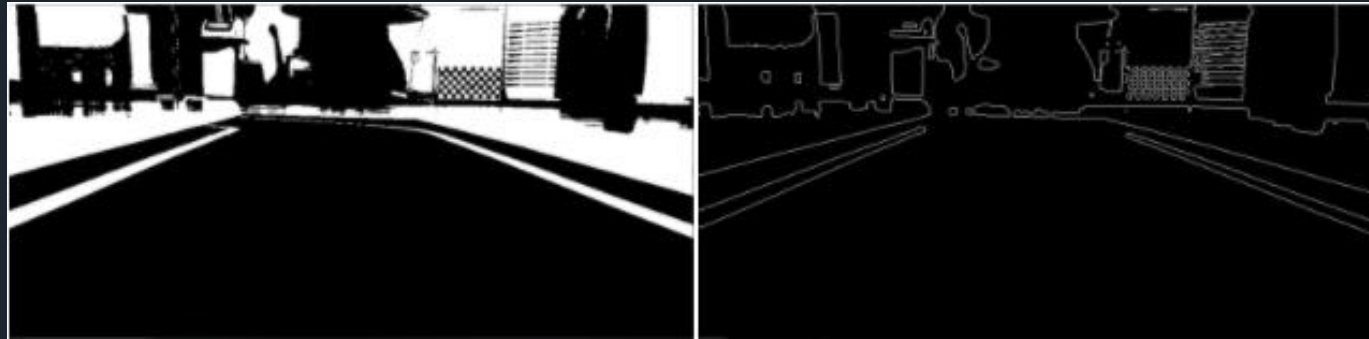
Gaussian Blur



| Project Contents

1. 차선 인식 알고리즘 [Lane Detection Algorithm]-2

Gaussian Blur를 통해 노이즈가 제거된 이미지에 OpenCV의 Canny 함수를 적용한다.
이를 통해 아래처럼 차선의 엣지를 검출한다.



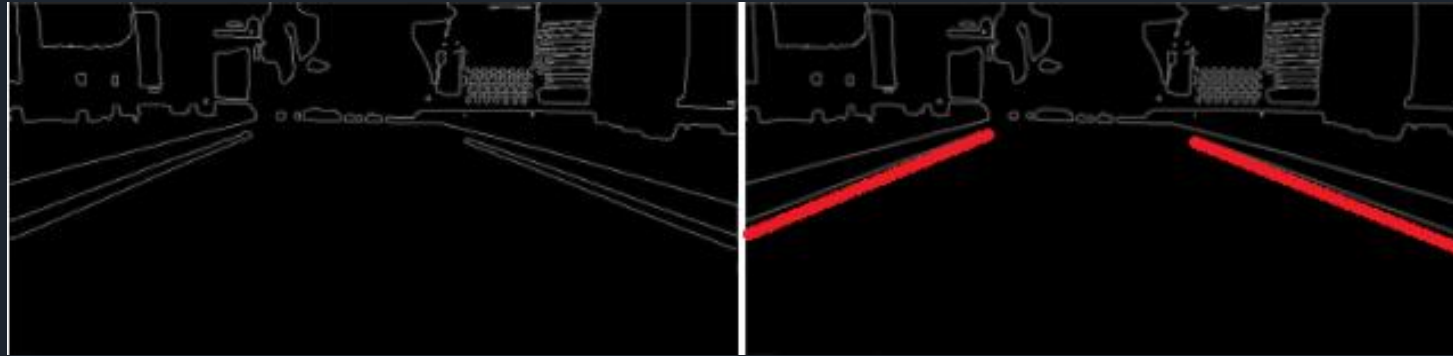
```
Edges = cv2.Canny(gray, 50, 150, apertureSize = 3)
```



Project Contents

1. 차선 인식 알고리즘 [Lane Detection Algorithm]-3

Canny Detection을 통해 검출된 엣지들 중에서 차선만을 골라내기 위해 HoughLines[] 함수를 사용한다.



골라진 차선을 빨간색으로 표시하고, 그 중심직선을 구해 영상에 표시한다. 이때 얻은 중심 직선의 끝점을 PurePursuit 알고리즘에서 사용한다.

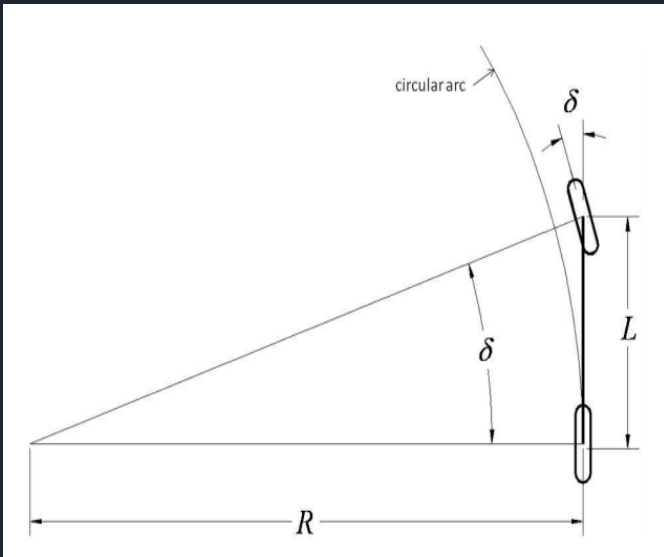


Project Contents

2. 움직임 알고리즘 [Movement Algorithm] - 1. 차선 인식에 따른 조향 및 구동

1) 전륜 조향각-조향각 도출 알고리즘

Ackermann's bicycle steering Algorithm

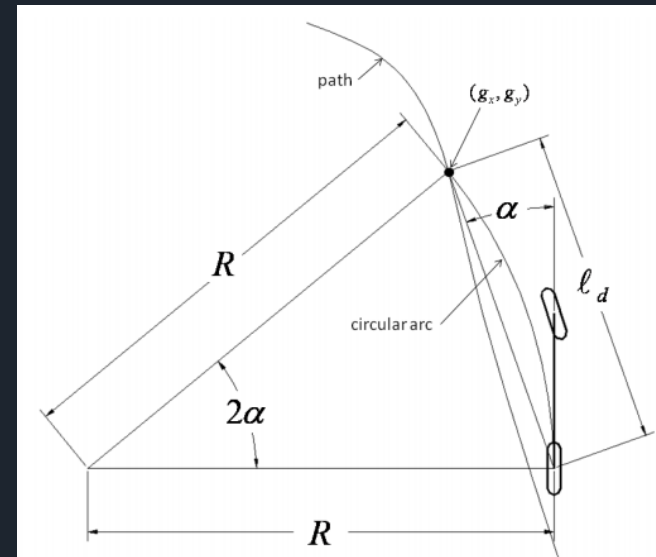


$$\frac{l_d}{\sin(2\alpha)} = \frac{R}{\sin\left(\frac{\pi}{2} - \alpha\right)}$$

$$\frac{l_d}{2 \sin(\alpha) \cos(\alpha)} = \frac{R}{\cos(\alpha)}$$

$$\frac{l_d}{\sin(\alpha)} = 2R$$

Pure Pursuit Algorithm



$$K = \frac{2 \sin(\alpha)}{l_d}$$

$$K = \frac{2}{l_d^2} e_{l_d}$$

$$\delta(t) = \tan^{-1} \left(\frac{2L \sin(\alpha)}{kv_x(t)} \right)$$



Project Contents

2. 움직임 알고리즘 [Movement Algorithm] - 1. 차선 인식에 따른 조향 및 구동

1)전륜 조향각-서보모터 제어

Pure pursuit 알고리즘을 통해 구한 목표 조향각 θ_1 , 피니언 기어의 회전각(=서보모터의 회전각)을 θ_2 라고 하면, θ_1 과 θ_2 의 관계식은 아래와 같다.

$$\theta_2 = \frac{1152}{\pi} \times \arctan(\theta_1)$$

이 관계식으로 pwm 신호의 펄스 폭(duty)를 조절하여 서보모터를 제어한다.

서보모터의 제어에는 랙과 피니언 기어를 사용한다.



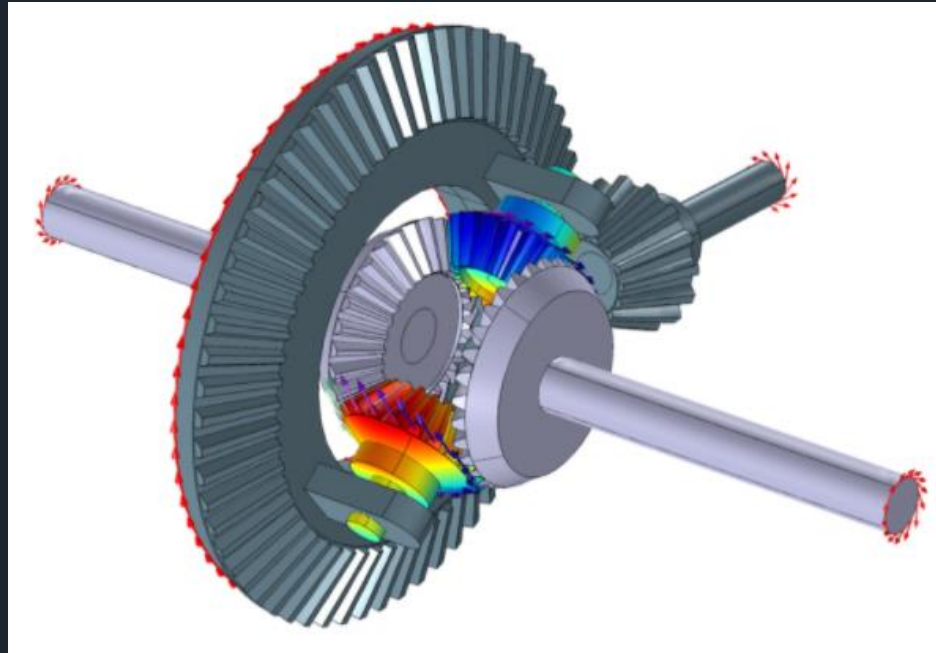
| Project Scenarios

2. 움직임 알고리즘 [Movement Algorithm] - 1. 차선 인식에 따른 조향 및 구동

2)후륜동력 전달을 위한 모터 구동

-dc모터 사용

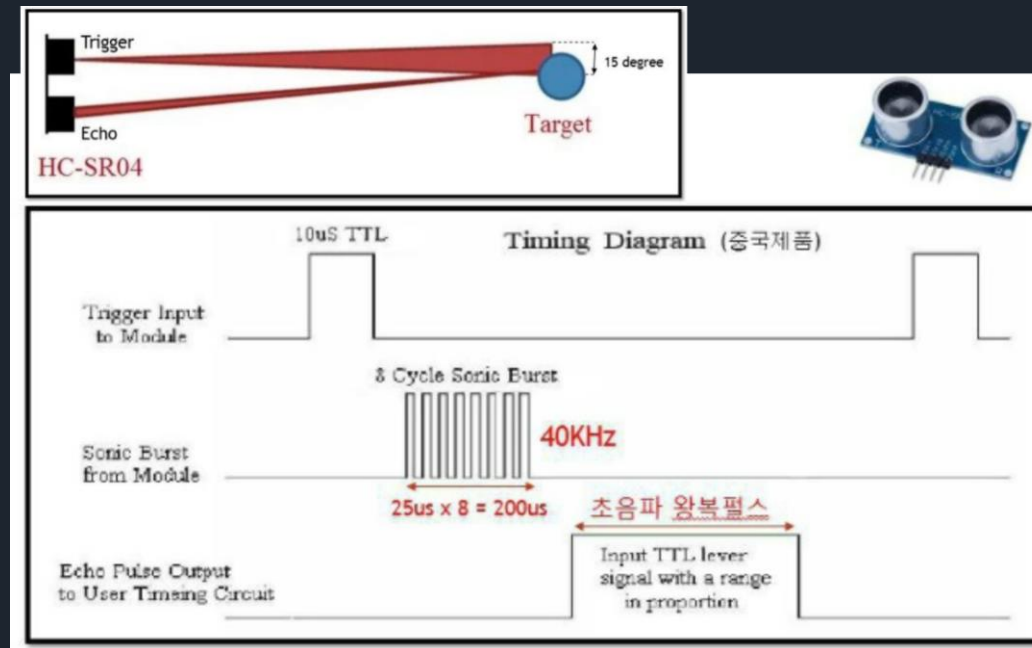
-조향 시, 양쪽 바퀴가 다른 속도로 돌아야 하므로 차동[differential] 기어 사용



Project Scenarios

2. 움직임 알고리즘 [Movement Algorithm] - 2. 물체 출현시 급정지

- 전후방에 초음파 센서 장착
- 일정 거리 내의 노상에 물체 출현 감지시, 코어텍스로 신호 전송
- 전송받은 신호로 차량 정지!



| Project Scenarios

1. 정해진 트랙 위에 자동차를 놓는다.
2. 양쪽 차선을 감지하여 벗어나지 않고 자동주행을 한다.
 - 웹캠 영상으로 차선 감지 후 차 전방 목표지점을 계속 업데이트
3. 노상에 물체가 갑자기 출현하면 급정지한다.
 - 초음파 센서를 사용해 10~20cm 전방에 물체 감지시 정지



Project Schedule

진행 목표	9M		10M					11M	
	4w	5w	1w	2w	3w	4w	5w	1w	2w
라즈베리파이 환경 구축					중 간 고 사				
재료 구매 및 하드웨어 제작									
OpenCV로 차선 인식 코딩									
모터와 서보모터 제어 코딩									
차선 인식 알고리즘과 제어 코드 연동									
테스트 및 디버깅, 코드 개선									



Q&A

Thank you for Listening

