

터널 내 레일 로봇의 실시간 모니터링 및 협동을 통한 자동차 2차 사고 예방 시스템

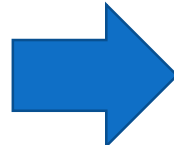
오정 테크(OH Jeong Tec)

2024/09/12

2차 사고 위험성 대비 부속한 후속 조치

전체 교통사고 치사율: 8.4%, 2차 사고 치사율: 54.3%

- 2차 사고의 치사율은 전체 교통사고 대비 **약 7배**[2]
- 터널 2차 사고의 치사율은 전체 터널 사고 대비 **약 4배**[3]



- 감시 공백을 최소화, 실시간 동기화 가능한 시스템 필요
- 후방 운전자가 전방 주시를 유도하는 시스템 필요

현행 사고 알림 시스템 문제점

- 사고를 파악하는 방식은 대부분 "운전자의 신고" 및 "모니터링"에 의존
 - 사고지점, 규모를 파악하는 시간 소요 → VMS 업데이트 지연
 - 일반적인 모니터링 방식의 경우, 신고 및 제보에 의존
- 운전자에게 적합하지 않은 알림 시스템
 - 후방 차량 운전자가 사고 정보를 직관적으로 알 수 없음
 - 운전 환경을 고려하지 않은 SMS, 유선 연락[4]



사고 상황에 VMS 녹색불 점등



개선안: 운전 환경 고려 부족

레일 로봇을 통한 새로운 알림 시스템 특징

터널 내 Rail robot 설치

- 평상시 20 km/h 의 속도로 도로 위험 요소 탐지
- 이벤트 발생시 30 km/h의 속도로 정해진 위치 이동
- 300m 간격으로 설치
- 조명, 부저 등으로 사고 지점에 점점 가까워질수록 발광 및 소리 주기가 짧아짐

VMS

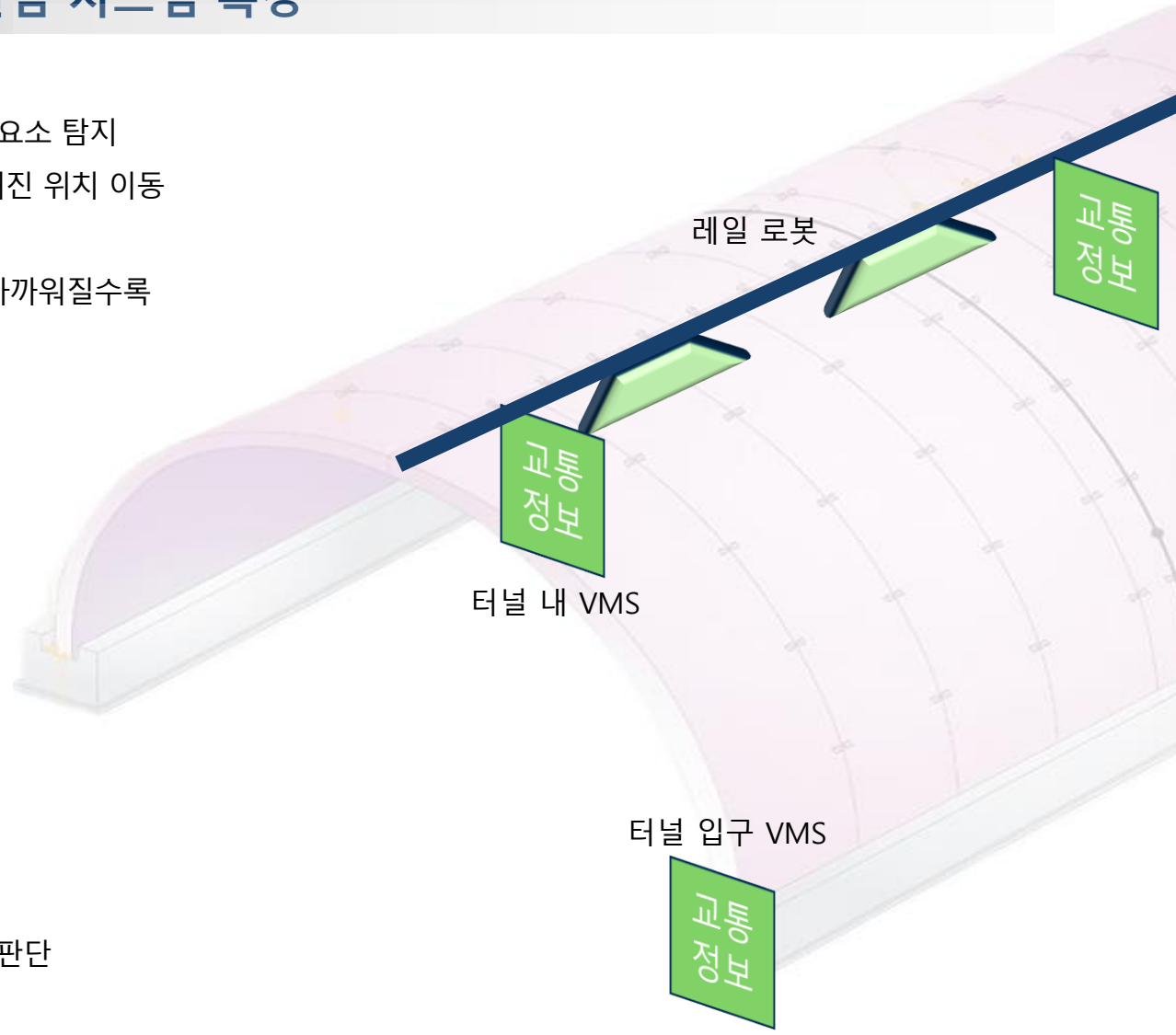
- 서버와의 동기화로, 사고 지점 위치를 명시적으로 제시

기기 간 정보 동기화

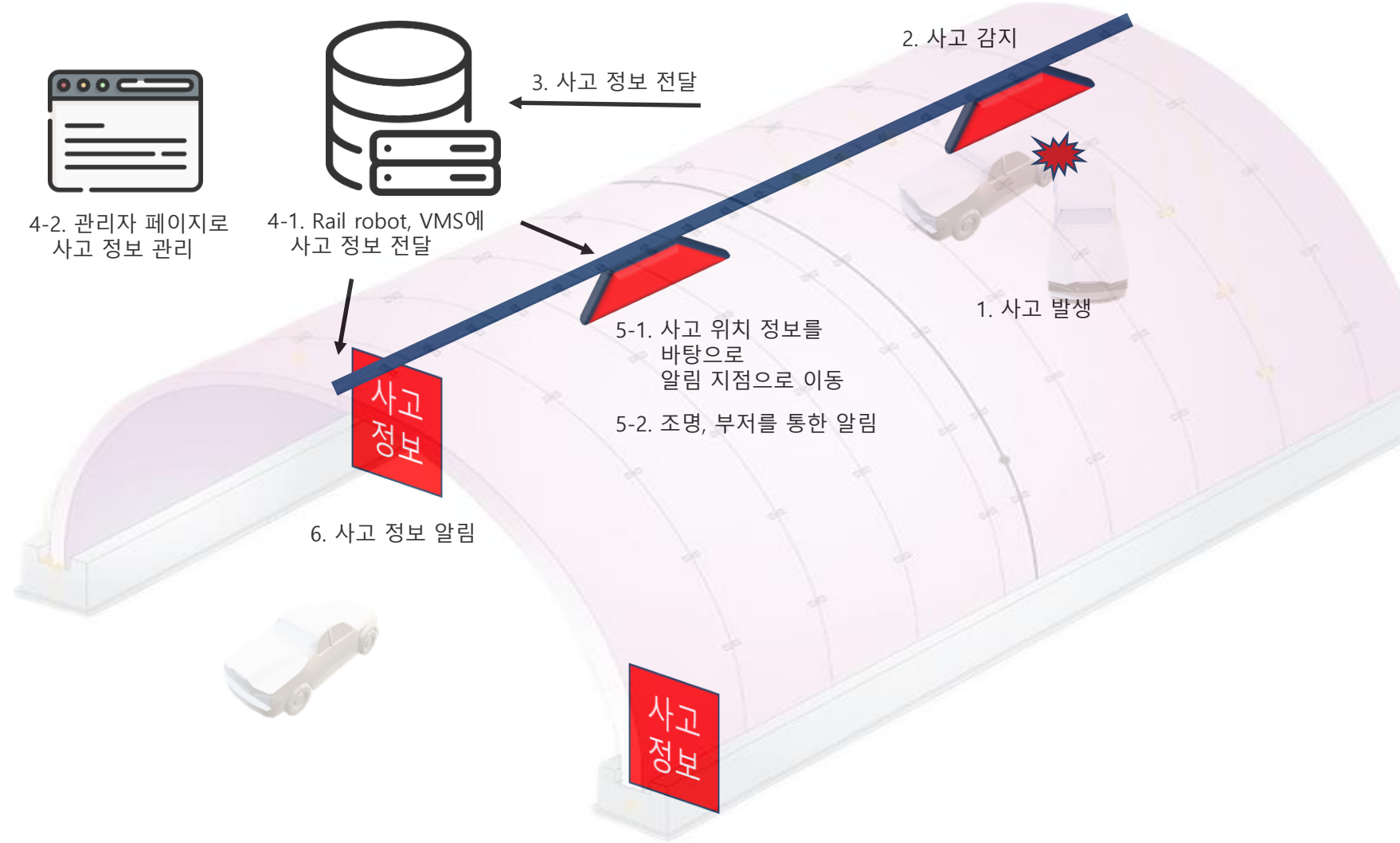
- 모든 Rail robot은 서버와 연결되어, 실시간 위치 및 사고 여부 전송

사고 정보 관리

- 관리자 페이지에서 사고 여부를 직접 판단
- 특정 위치로 로봇을 보내 감시 가능



사고 발생시 구성 요소 간 동작 흐름



기존 시스템과 2차 사고 방지 시스템

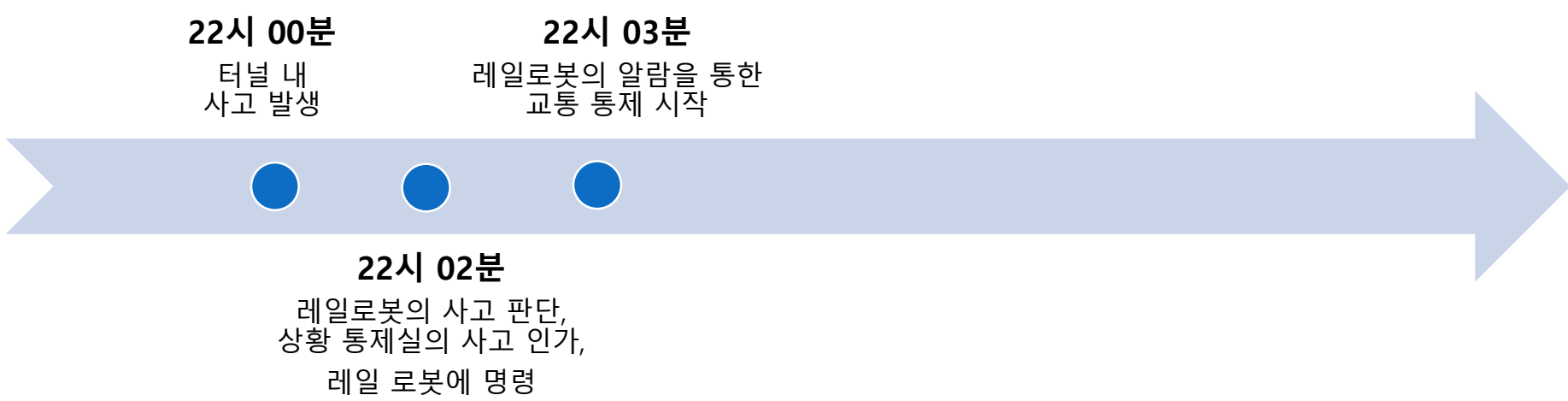
가상 시나리오 설정 및 기존 시스템 타임라인

[기존 시스템]



* 도로교통공사(동서울)를 인터뷰 기준

[개선 시스템]



기존 시스템과 개선 시스템의 비교

기존 시스템 대비 개선 시스템의 주요 지표 비교

대응 시간 비교 (H: 00_ 사고 발생)

	*기존 시스템	개선 시스템
상황실 사고 파악	H: 05	H: 01
VMS 업데이트	H: 05~06	H: 01~02
최종 교통 통제	H: 25	H: 03

* 도로교통공사(동서울)를 인터뷰 기준

- 현행 대비 8배 빠른 교통 통제
→ **최종 교통통제까지 최소 22분 단축**

CCTV 설치 간격 및 추가 기능 비교

	기존 시스템	개선 시스템	*지능형 CCTV
CCTV 설치 간격	200 m	300 m	50 m
사고 알림 시스템	터널 입구 진입차단 설비 VMS	레일 로봇 협동 통한 사고 지점 1km 전 알림 제공 실시간 VMS 터널 입구 진입차단 설비	터널 입구 진입차단 설비 VMS

* 감시 공백 없이 설치한다는 가정 하

- 운전자의 단기 기억 특성(22초~30초)을 고려한 직관적 사고 알림 제공
→ **운전자의 전방 주시를 효과적으로 유도**

Rail robot을 통한 터널 2차 사고 문제 완화

사고 모니터링 실시간성 확보

- Rail robot의 AI가 모니터링을 자동화하여 관리자의 컨디션에 따른 사고 탐지 시간을 고정
- 현행 대비 빠른 탐지 및 대응으로 2차 사고 예방 극대화

후방 차량 운전자가 직관적으로 사고 여부 판단

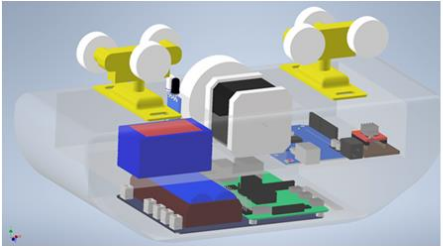
- Rail robot에 조명, 스피커를 장착하여 사고 발생 지점 100m 전부터 500m의 간격으로 시각적/청각적 알림 제공
- 사고 상황에만 움직이는 사고 알림 프로토콜로 낯선 환경에 운전자의 전방 주시 및 줄음 운전 예방 효과

위치 유동적인 시각적 알림 제공 가능

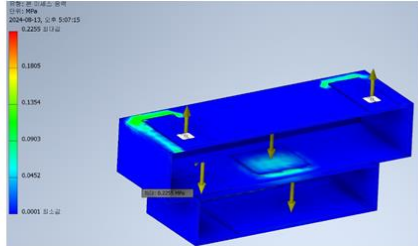
- 사고지점 "1km 전에 정보를 제공"하고 "터널 입구로부터 약 650 m 지점"에 알림을 제공할 경우 2차 사고 예방 효과[5]
 - 기존 VMS만 사용한 시스템은 고정적임
 - 반면, Rail robot은 **위치 유동적으로 알림 제공 가능**
 - 어떤 곳에서 사고가 발생하든, 운전자로 하여금 쉽게 사고 인지

하드웨어_레일로봇조립

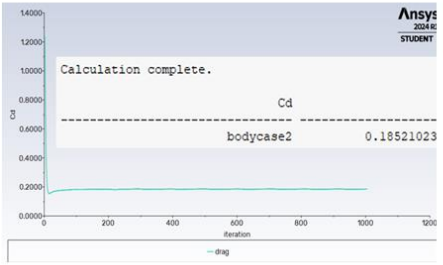
Rail robot 내/외부 구성요소



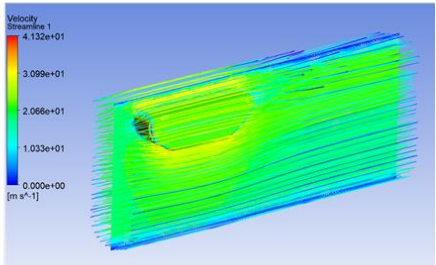
레일 로봇 3D 구조도



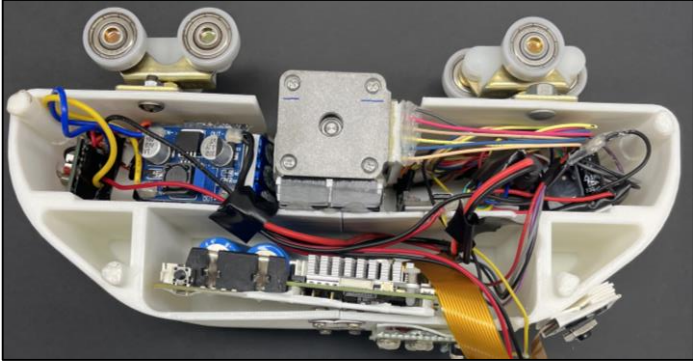
응력 해석 결과



외형 항력 계수 해석 그래프



CFD 해석 결과



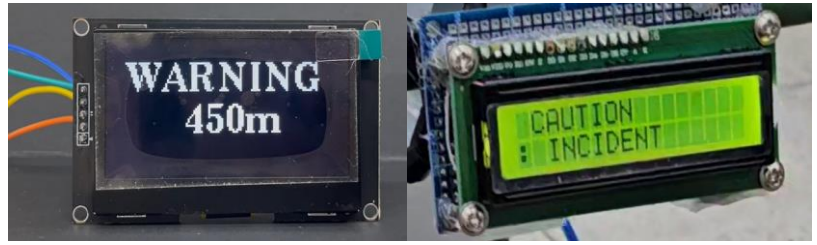
Rail robot 내부 사진

외부	
로봇 외형 틀	3D 프린터 출력
베어링	레일 및 로봇 지지
내부	
스텝 모터	동력 제공
리튬-폴리머 배터리	7.4V * 3(직렬) 전원 공급
모터 드라이버	스텝 모터 제어
라즈베리파이5	서버와 통신, AI 학습 기반 사고 탐지
아두이노 UNO R4	스텝 모터 제어, 현재 위치 계산
카메라 모듈	터널 내부 영상 촬영
IR 센서	로봇 위치 인식
LED	사고 시 알람 발생
레귤레이터	모듈 간 안정화를 위한 전압 조정

VMS, 전송 선로, Power Tree

VMS의 구현

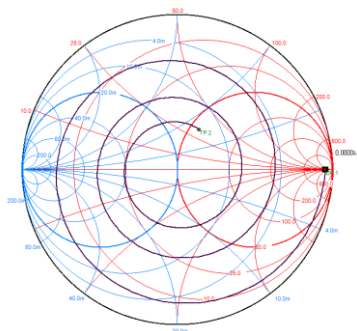
- LCD와 OLED 디스플레이를 이용하여 VMS 구현
- RC 필터와 CAT7 도선을 응용하여 장거리 송신 파동의 viscosity 및 노이즈 영향 최소화



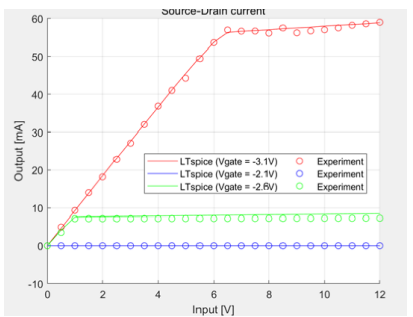
OLED, LCD를 이용한 VMS 구현

MOSFET LED 스위치 회로

- FET(Field Effect Transistor)를 이용한 즉각적인 LED 점등
- Saturation current를 이용한 최대 광속(lumen) 전달



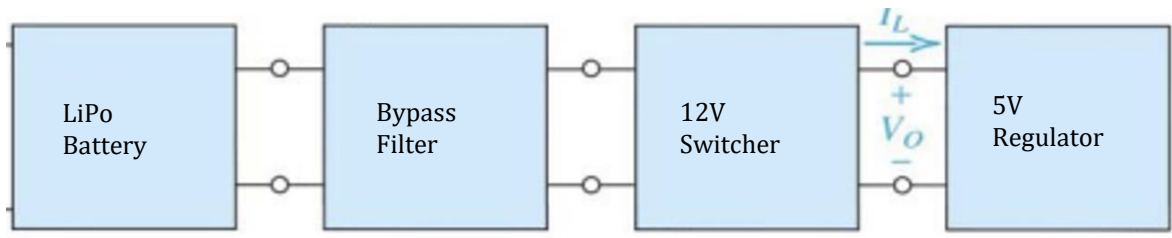
전송선의 감쇄항 Smith Chart 시뮬레이션



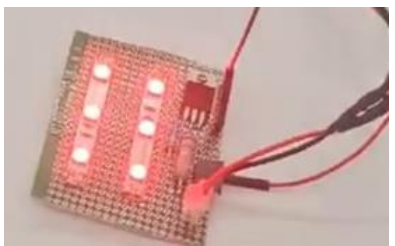
LED 스위치 회로의 동작 영역 시뮬레이션

POWER TREE

- 배터리와 regulator의 cascaded system
- 최대 전류를 감당 가능한 AWG의 도선 분배



전체 system의 power tree 모식도



NMOS LED switch

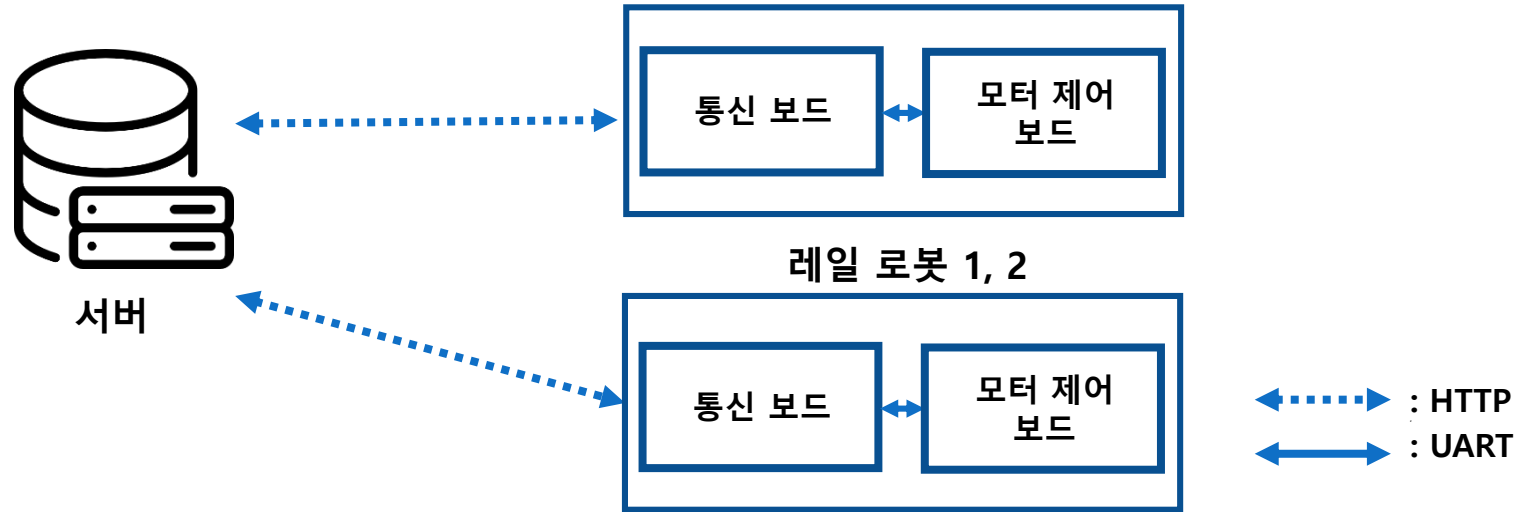
(통신 보드 – 모터 제어 보드/서버) 간 통신

(통신 보드 – 모터 제어 보드) – UART

- 두 종류의 통신 보드(ESP32 D32, RaspberryP_5)와 모터 제어 보드인 Arduino 간의 UART 통신
- 통신보드 → 모터 제어 보드 : **"Command / Patrol_Section / (Blink_Period) / END"**
- 모터제어보드 → 통신보드 : **"Current_Location"**

(통신 보드 – 서버) – HTTP

- 두 종류의 통신 보드(ESP32 D32, RaspberryP_5)와 서버 간의 HTTP 메소드 활용한 무선 통신
- 통신보드 → 서버 : **"Update_Current_Location", "Accident_Sensing"**
- 서버 → 통신보드 : **"Move_To_Target_Location", "Patrol_Section", "Go/Stop_Flag", "Alarm_Start/End"**
- 1초 마다 Command 반영, 특정 거리에 도달할 때마다 로봇 위치 업데이트



Server

- ```

(base) j93es
❏ k6 run test.js

 /_/\
 / _ \
 /_____\ \
 / \
 / \
 / \
/ \
\ /
 \ /
 \ /
 \ /
 _____/

execution: local
script: test.js
output: -

scenarios: (100.00%) 1 scenario, 100 max VUs, 1m30s max duration (incl. graceful stop):
 * default: 100 looping VUs for 1m0s (gracefulStop: 30s)

data_received.....: 4.4 MB 73 kB/s
data_sent.....: 1.1 MB 18 kB/s
http_req_blocked.....: avg=8.72ms min=0s med=4µs max=656.2ms p(90)=10µs p(95)=15µs
http_req_connecting.....: avg=1.42ms min=0s med=0s max=105.79ms p(90)=0s p(95)=0s
http_req_duration.....: avg=47.28ms min=12.93ms med=29.63ms max=10.28s p(90)=71.55ms p(95)=121.31ms
{ expected_response:true }.: avg=47.28ms min=12.93ms med=29.63ms max=10.28s p(90)=71.55ms p(95)=121.31ms
http_req_failed.....: 0.00% ✓ 0 ✗ 5713
http_req_receiving.....: avg=59.91µs min=9µs med=38µs max=5.67ms p(90)=102µs p(95)=162µs
http_req_sending.....: avg=25.03µs min=3µs med=15µs max=3.78ms p(90)=37µs p(95)=60µs
http_req_tls_handshaking.....: avg=3.05ms min=0s med=0s max=330.88ms p(90)=0s p(95)=0s
http_req_waiting.....: avg=47.2ms min=12.82ms med=29.53ms max=10.28s p(90)=71.5ms p(95)=121.27ms
http_reqs.....: 5713 93.60803/s
iteration_duration.....: avg=1.05s min=1.01s med=1.03s max=11.28s p(90)=1.07s p(95)=1.14s
iterations.....: 5713 93.60803/s
vus.....: 19 min=19 max=100
vus_max.....: 100 min=100 max=100

running (1m01.0s), 000/100 VUs, 5713 complete and 0 interrupted iterations
default ✓ [=====] 100 VUs 1m0s

```

## API Documentation

**Accident Code**

```
export const enum AccidentCode {
 CAR_CRASH = "0",
 FIRE = "1",
 FLOOD = "2",
}
```

```
export const enum AccidentStatus {
 DETECTED = "0",
 IGNORED = "1",
 ALARMING = "2",
 END = "3",
}
```

```
export const enum RailRobotCommand {
 PATROL = "0",
 ALARMING = "1",
 CHARGE = "3",
 MOVE_TO_TARGET_LOCATION = "4",
 STOP = "5",
}
```

## 11

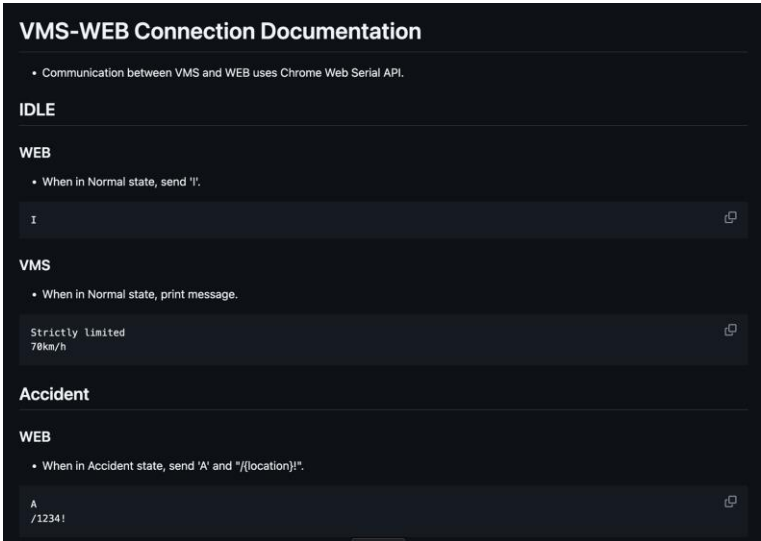
## 관리자 페이지 구현

### 터널 관제 시스템 웹사이트

- <https://cedc.webserialmonitor.com> 에 터널 관제 시스템 웹사이트 배포하여 완성도를 높임
  - React를 통한 기능 구현
  - Chrome Web Serial Api를 통하여 WEB-VMS 간의 실시간 통신 구현
  - WEB-VMS 간의 통신 명세를 통한 체계적인 협업 추진
- (<https://github.com/J93es/2024-creative-engineering-design-competition/blob/main/frontend/README.md>)



터널 관제 시스템 웹페이지



WEB-VMS 간 통신 명세

## TensorFlow Lite 기반의 AI 탑재

### 실시간 영상 분석

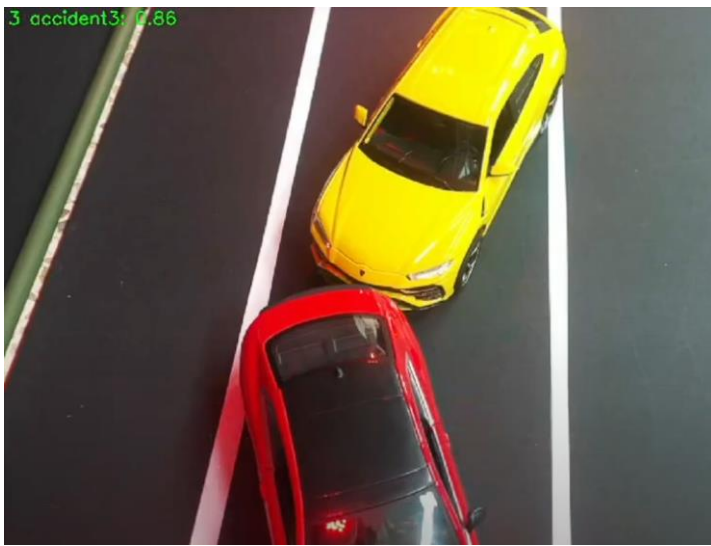
- 카메라에서 들어오는 영상을 실시간으로 받아, AI 모델에 입력하여 사고 발생 여부를 분석
- 카메라 스트리밍 → 이미지 프레임 → AI 모델 분석
- TensorFlow Lite 모델이며, 부동 소수점 모델

### 역할



- 실시간으로 사고 징후가 발견되면 화면에 경고 메시지를 표시, 필요하면 서버에 사고 발생 보고
- 판단 여부를 기준으로 HTTP 통신으로 원격 제어 및 명령 수행, 모터 제어 보드와 UART로 명령 및 상태 정보 전송

### 학습 데이터

|            | 설명              | 수량(장) |
|------------|-----------------|-------|
| Normal     | 충돌하지 않은 보통 상황   | 1000  |
| Accident 1 | 차 손상(범퍼 및 문 열림) | 500   |
| Accident 2 | 차선을 넘어 정지       | 500   |
| Accident 3 | 이중 차량 충돌        | 500   |
| Accident 4 | 삼중 차량 충돌 혹은 정체  | 500   |



세부 기능의 원활한 연계



터널 관제 시스템

레일 로봇 관리

레일 로봇 위치(km)

0 20 40 60 80 100 120 140 160 180 200

레일 로봇 정보

| ID | 행방 | 현재 위치 | 순환 지점       | 이동할 위치 |
|----|----|-------|-------------|--------|
| 0  | 정지 | 0m    | 0m ~ 100m   | 50m    |
| 1  | 행진 | 100m  | 100m ~ 200m | 150m   |

위치를 입력하세요.

VMS 관리

VMS 정보

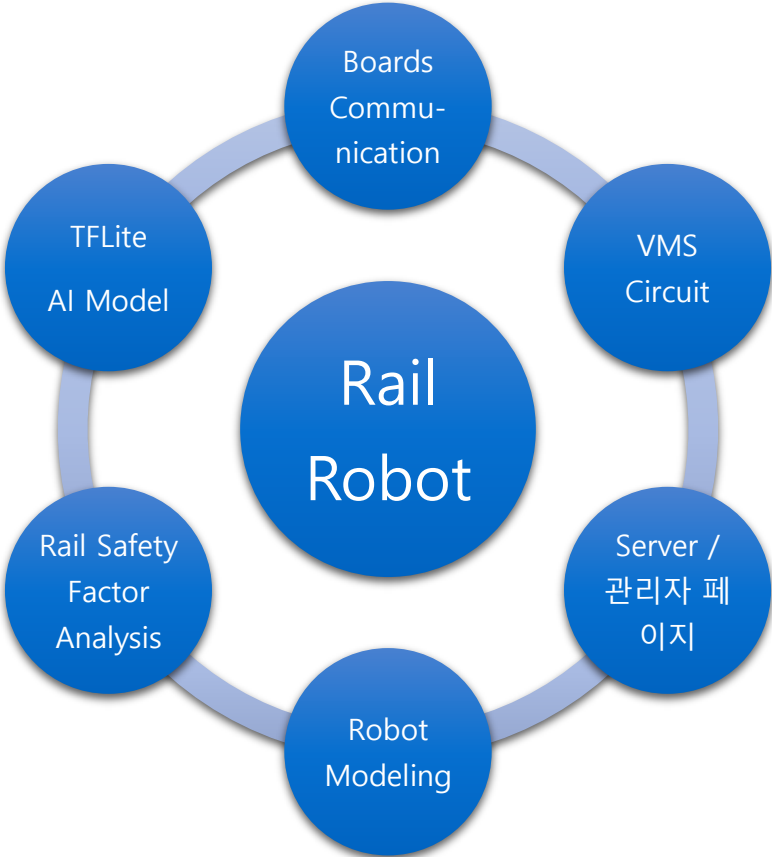
VMS 연결 상태 연결 끊김

VMS 연결 VMS 연결 해제 영상방해 수동 전송 사고감지 수동 전송

사고 정보 관리

사고 정보

발생한 사고가 없습니다.



- [1] 유튜브 채널 한문철의 블랙박스 리뷰, '졸음운전으로 터널에서 추돌? 아수라장 된 터널 사고 모음 30분', JTBC 221020 방송 외
- [2] 윤보람, '자칫하면 대형참사...터널 교통사고로 5년간 128명 사망', 연합뉴스, 2024.02.09., <https://www.yna.co.kr/view/AKR20240208112400004>
- [3] 강세훈, '고속도로 터널 2차 사고 사망률, 전체 터널 사고의 4배', 뉴시스-서울, 2022.10.07., [https://www.newsis.com/view/NISX20221007\\_0002040731#](https://www.newsis.com/view/NISX20221007_0002040731#)
- [4] 이재영, '차량침수·고속도로 2차사고 위험 이젠 사전예방하세요', 이투데이, 2024.01.15., <https://www.etoday.co.kr/news/view/2321232>
- [5] 신소명, 이수범, 김형규, 박민재, 김경태. (2019). 동영상식 VMS로 사전정보제공시 터널 내 2차사고 감소효과에 관한 연구. 한국안전학회지, 34(2), 56-62.
- [6] 박승원, "사고현장 5분 내 도착 '고속주행 로봇' 개발", 토목신문, 2024.03.07., <http://m.cenews.co.kr/news/articleView.html?idxno=14868>
- [7] Basilico, Nicola. "Recent trends in robotic patrolling." Current Robotics Reports 3.2 (2022): 65-76

끝

