

Alcohol Watcher

음주 운전 방지 시스템

팀 3조 : 민경서, 오동훈, 윤용성, 이지호, 장승원



Table of contents

01

프로젝트 개요

02

프로젝트
일정

03

요구사항 분석

04

아키텍처 / 하드웨어

05

개발

06

테스팅

07

업무 분장 / 협업

08

시연연상 / 회고





01

프로젝트 개요





프로젝트 개요

프로젝트 배경

음주 운전 교통사고 현황



뉴스1

“최근 5년간 총 **75,950** 건 발생”





프로젝트 개요

프로젝트 주제 및 기대효과

음주 운전 방지 서비스



음주운전자에게 경고 알림



주변 보행자에게 경고 알림



다른 차량에게 음주운전 차량
알림



음주운전 사고 발생률
감소



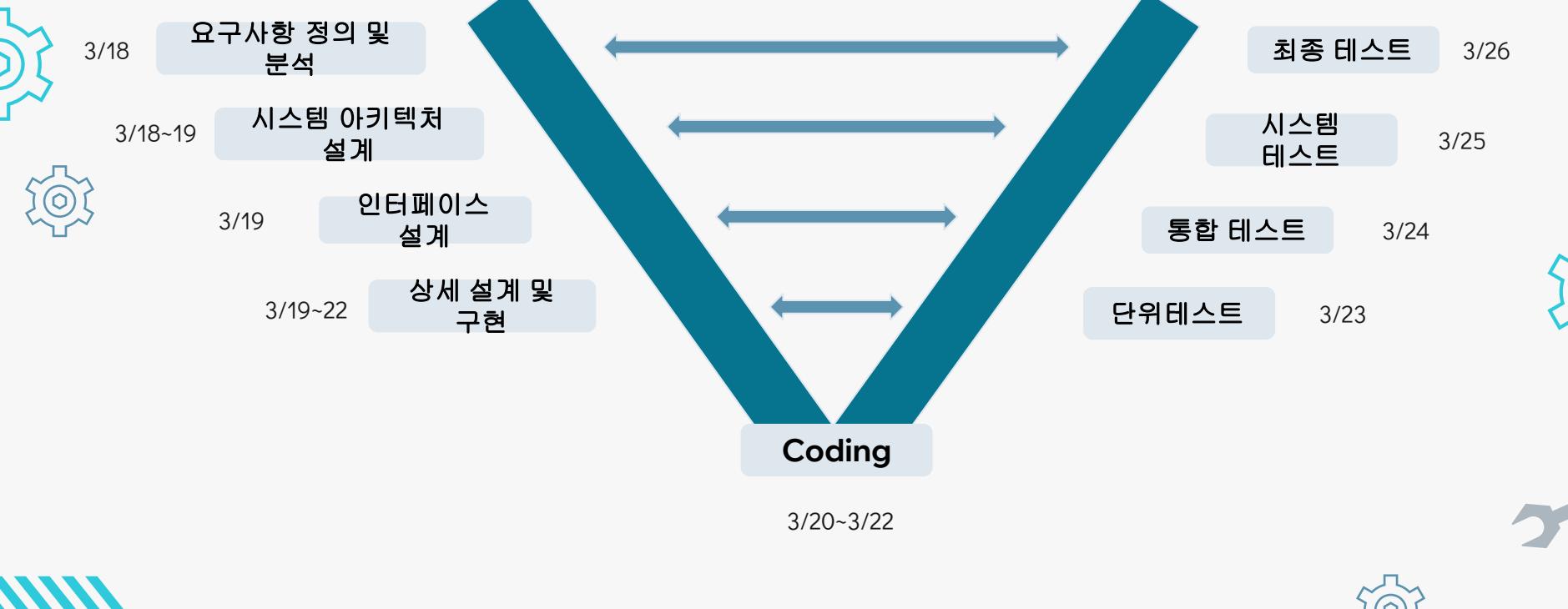
02

프로젝트 일정



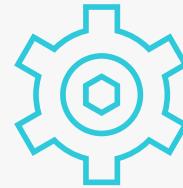
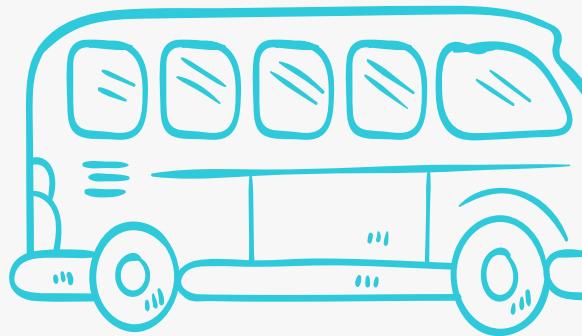
프로젝트 일정

프로젝트 일정



03

요구사항 분석





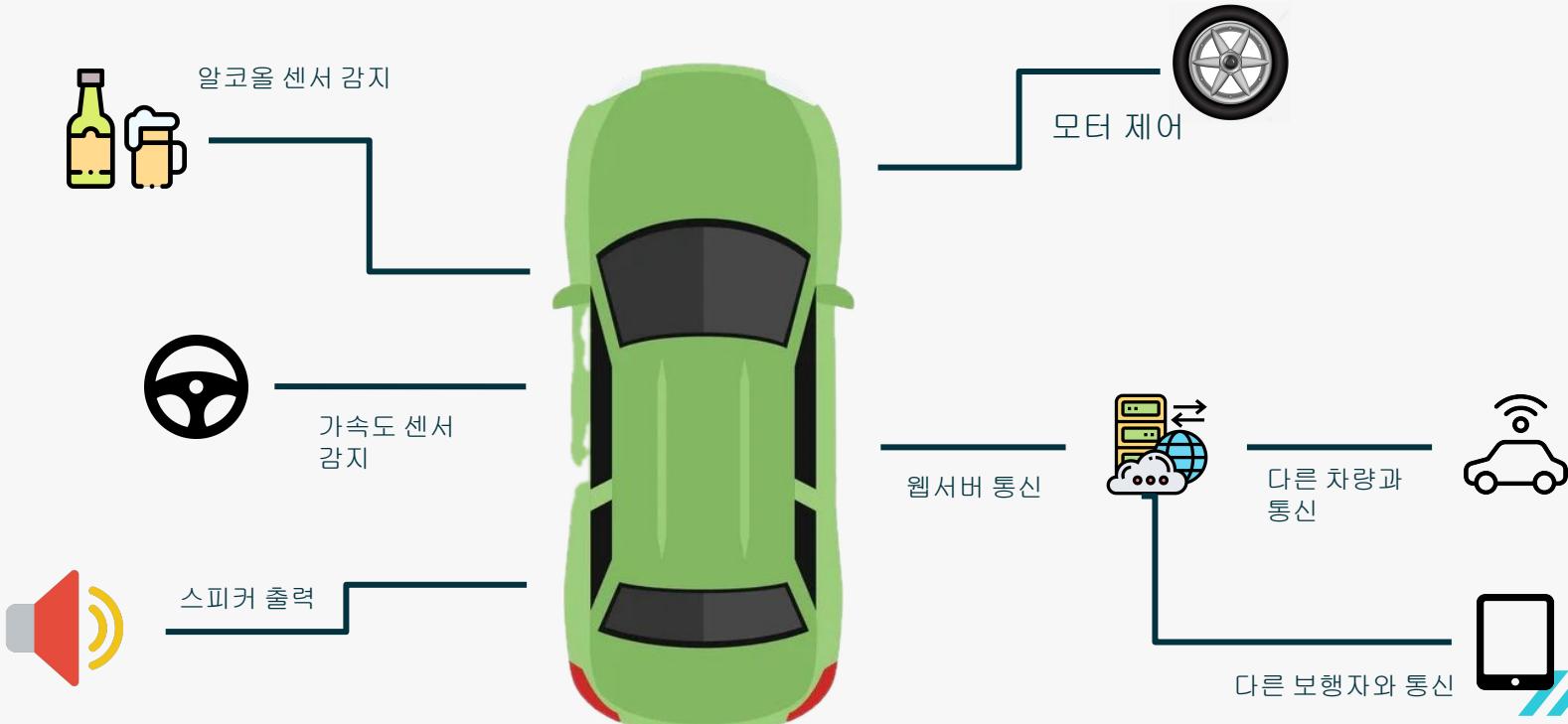
요구사항 분석

요구사항 정의 및 분석

ID	요구사항 정의	ID	세부 요구사항
FR_01	1차 감지: 차량은 알코올 농도 센서를 통해 음주 상태를 감지해야 함.	FR_01.1	차량은 운전자의 호흡을 분석하여 혈중 알코올 농도를 측정할 수 있어야 함.
		FR_01.2	차량은 알코올 감지 센서는 1초 주기로 측정하고, 알코올 농도 수치가 100을 넘을 경우 1차 음주 상태로 판단해야 함.
FR_02	2차 감지: 차량은 운전 패턴을 모니터링해야 함.	FR_02.1	차량은 자이로 센서를 통해 조향 및 가감속 데이터를 측정하고 운전 패턴을 분석할 수 있어야 함.
		FR_02.2	차량은 자이로 센서를 통해 1초 주기로 측정하고, 20초 동안 50% 이상 자이로 센서 값이 1을 넘어갈 경우 2차 음주 상태로 판단함.
FR_03	차량은 단계에 따라 음주운전에 대한 경고 및 대응을 해야 함.	FR_03.1	차량은 1차 음주 상태 판단 시 1차 경고 음성 메세지를 출력함.
		FR_03.2	차량은 2차 음주 상태 판단 시 감속 및 경고 음성 메시지를 출력하고, 감지 데이터를 웹 서버로 전송해야 함.
		FR_03.3	차량은 2차 음주 상태 판단 시 모터 PWM 값을 500에서 10초 간격으로 100씩 감속시킴.
		FR_03.4	차량은 2차 음주 상태 판단 시 주변 차량에서 2차 경고 음성 메시지 출력함.
		FR_03.5	차량은 2차 음주 상태 판단 시 주변 보행자의 핸드폰에 경고 메시지 출력함.
FR_04	웹 서버는 차량의 데이터를 수신하고 데이터를 분석해야 함.	FR_04.1	웹 서버는 차량에서 보내는 알코올 농도, 자이로 센서, 모터 PWM 데이터를 수신함.
		FR_04.2	웹 서버는 수신한 알코올 농도, 자이로 센서, 모터 PWM 데이터를 수치값 표기와 그래프로 시각화함.
		FR_04.3	웹 서버는 수신한 알코올 농도, 자이로 센서, 모터 PWM 데이터를 바탕으로 음주운전 상태를 판단함.
FR_05	웹 서버는 다른 차량과의 통신을 통해 주변에 음주운전 차량이 있음을 알려야 함.	FR_05.1	웹 서버는 차량이 2차 음주 상태로 판단 시 주변 차량으로 차량 A의 음주운전 상태를 전송함.
FR_06	주변 차량은 차량A의 음주운전 상태를 인지해야 함.	FR_06.1	주변 차량은 웹 서버 상에서 주변에 음주운전 차량이 있음을 인지하고 경고 음성 메세지를 출력함.
FR_07	클라우드는 보행자의 핸드폰과 통신을 통해 주변에 음주운전 차량이 있음을 알려야 함.	FR_07.1	웹 서버는 차량이 2차 음주 상태로 판단 시 주변 보행자의 핸드폰으로 음주운전 상태를 전송함.
FR_08	주변 보행자는 차량 A의 음주운전 상태를 인지해야 함.	FR_08.1	주변 보행자의 핸드폰은 웹 서버 상에서 음주운전 차량이 있음을 인지하고 어플리케이션으로 경고 음성 메세지를 출력함.
FR_09	주변 차량은 차량 A의 음주운전 상태를 인지해야 함.	FR_09.1	주변 차량은 웹 서버 상에서 음주운전 차량이 있음을 인지하고 경고 음성 메세지를 출력함.

요구사항 분석

기능 소개



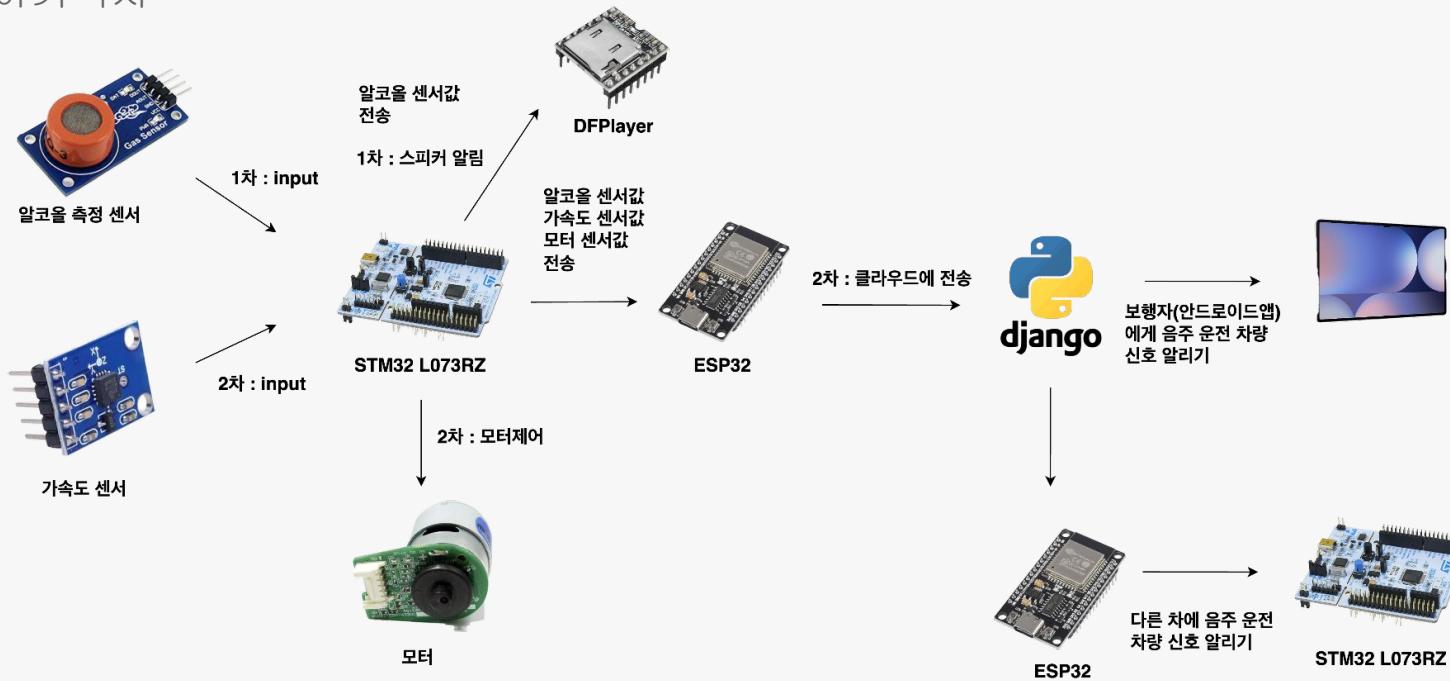
04

아키텍처 / 하드웨어



아키텍처 / 하드웨어

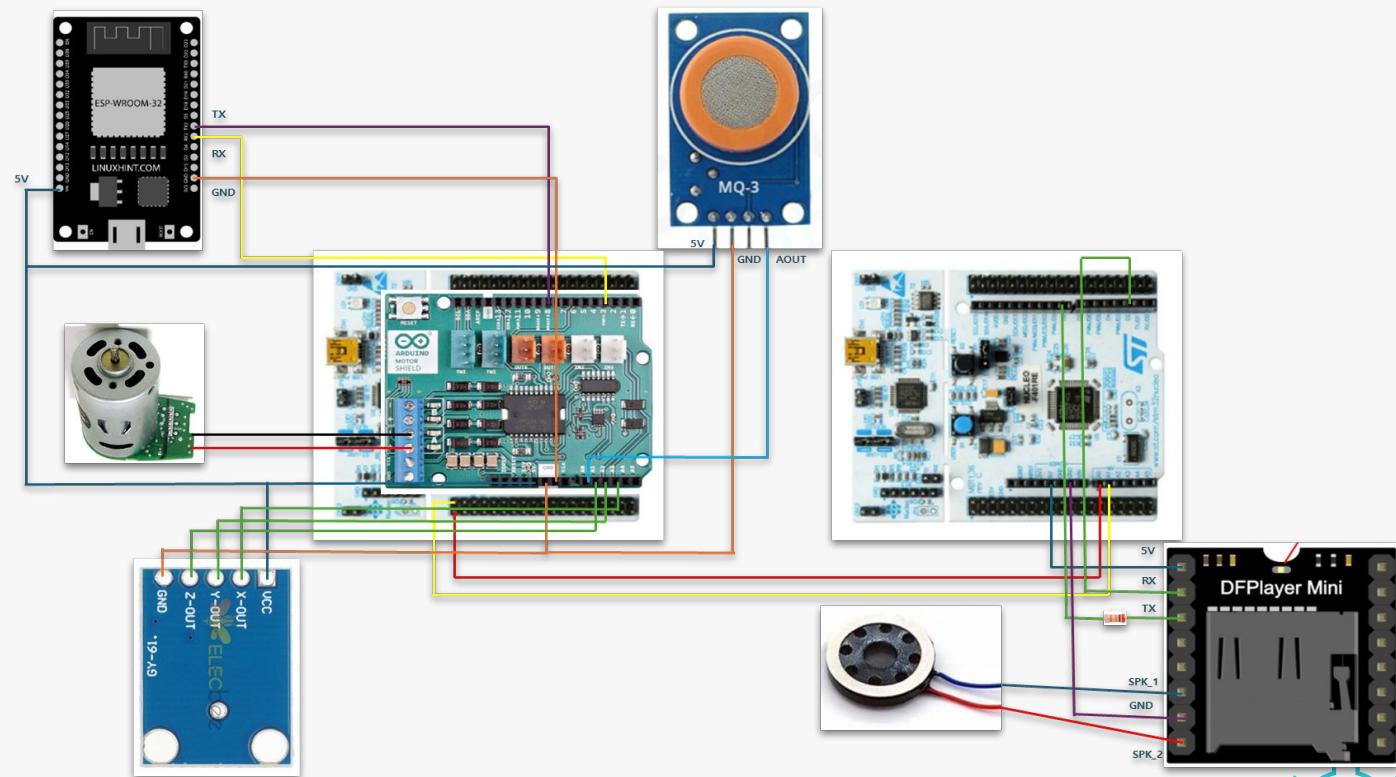
시스템 아키텍처





아키텍처 / 하드웨어

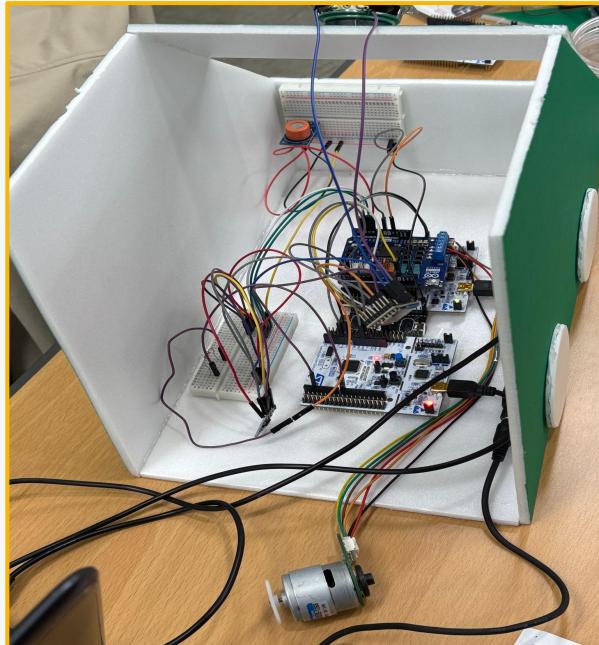
회로 설계



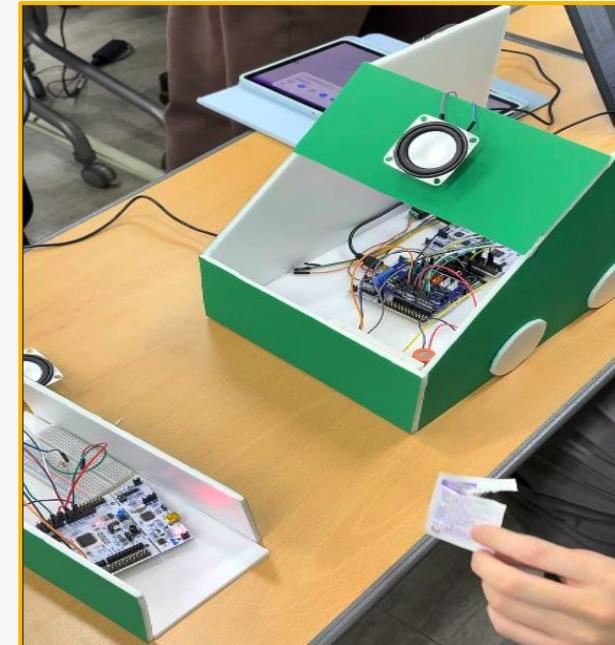


아키텍처 / 하드웨어

하드웨어 제작



음주운전
차량

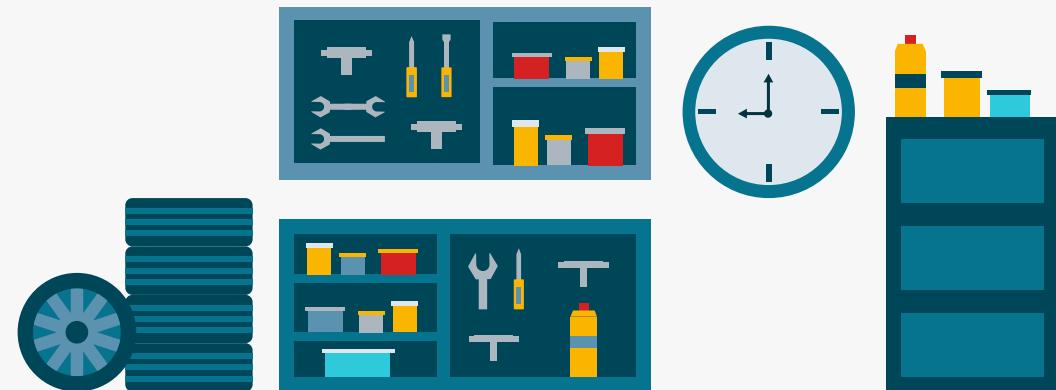


다른 차량



05

개발





개발

상세 설계 및 구현 - 스피커 출력



스피커 출력

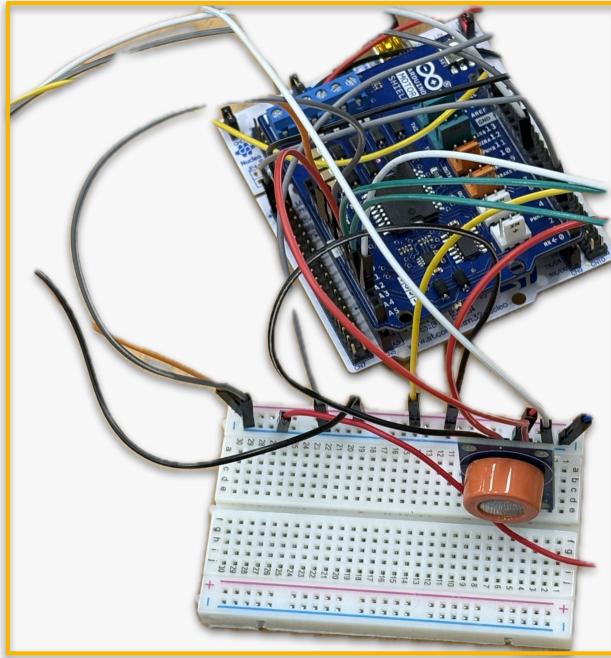
- 0단계 : 차량이 주행 중입니다. (When pressing reset button)
- 1단계 : 음주 상태가 감지되었습니다. 차량이 제한될 수 있습니다.
- 2단계 : 즉시 차량을 멈추십시오. 차량의 속도가 감속됩니다.
- 3단계 : 10초 후 차량이 정지됩니다.





개발

상세 설계 및 구현 - 알코올 센서 감지



알코올 감지 전 : 10~20 사이 유지

알코올 감지 후(1cm 이내) :

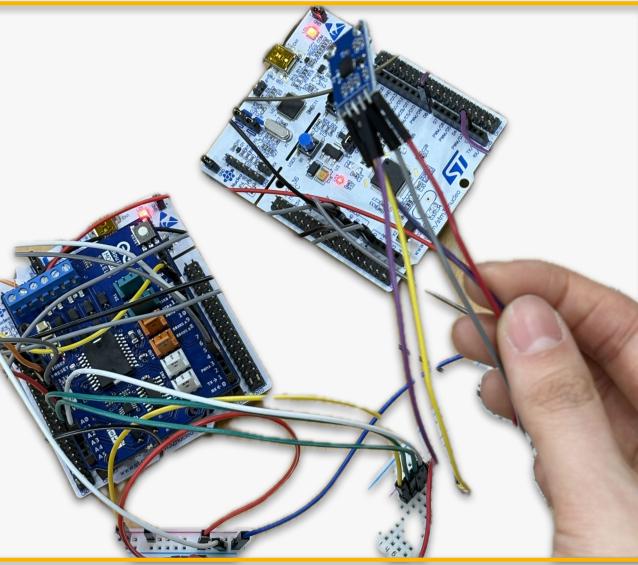
- 초당 30씩 증가, 100 이상 감지 시 알코올 센서 음주 판단
- 1초마다 센서값을 ESP32를 통해 웹 서버에 전송





개발

상세 설계 및 구현 - 자이로 센서 감지



자이로 센서 감지 전 : 0.0~0.2 사이

자이로 센서 감지 후 :

1초 주기로 측정하고, Threshold 100 이상의 값이
10회 이상 감지될 경우 2차 로직 실행

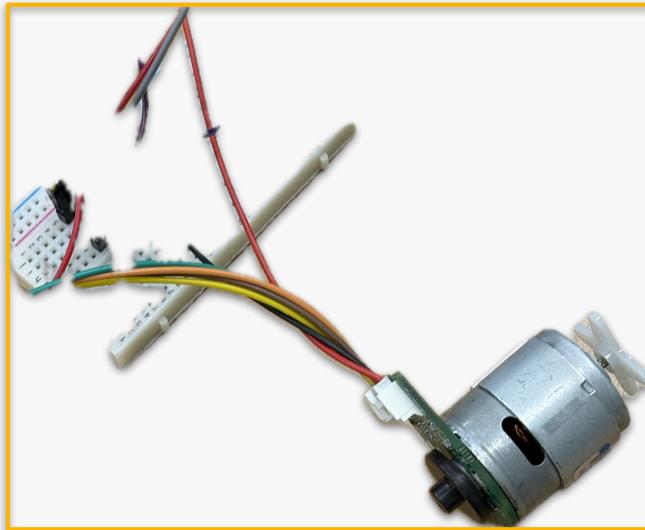




개발



상세 설계 및 구현 - 모터 제어



모터스피드(PWM) 출력

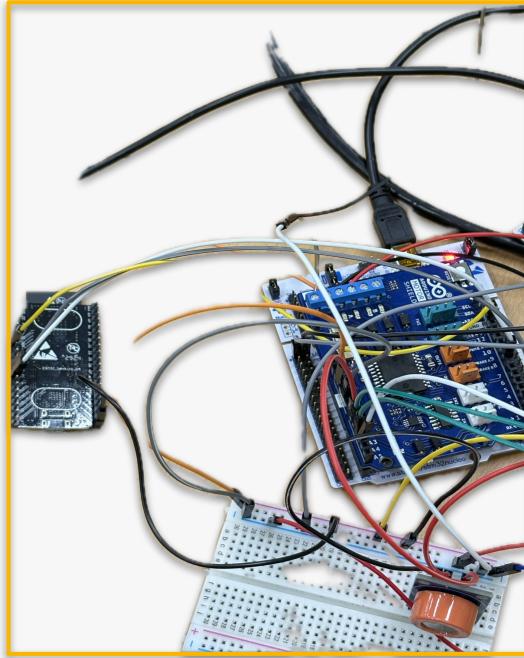
- 2차 로직 이후 PWM 값 500에서 10초당 100씩 감소
- 3차 스피커 출력 이후 200에서 0으로 모터 정지





개발

상세 설계 및 구현 - **STM32 ↔ ESP32** 통신



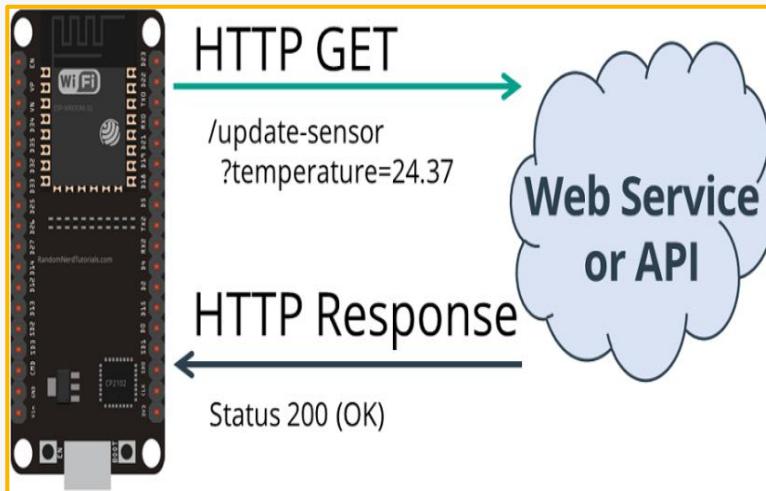
보드 통신 :

STM32 ↔ ESP32 UART1 통신, 센서 데이터 전송



개발

상세 설계 및 구현 - **ESP32 ↔ Django** 웹 서버 통신



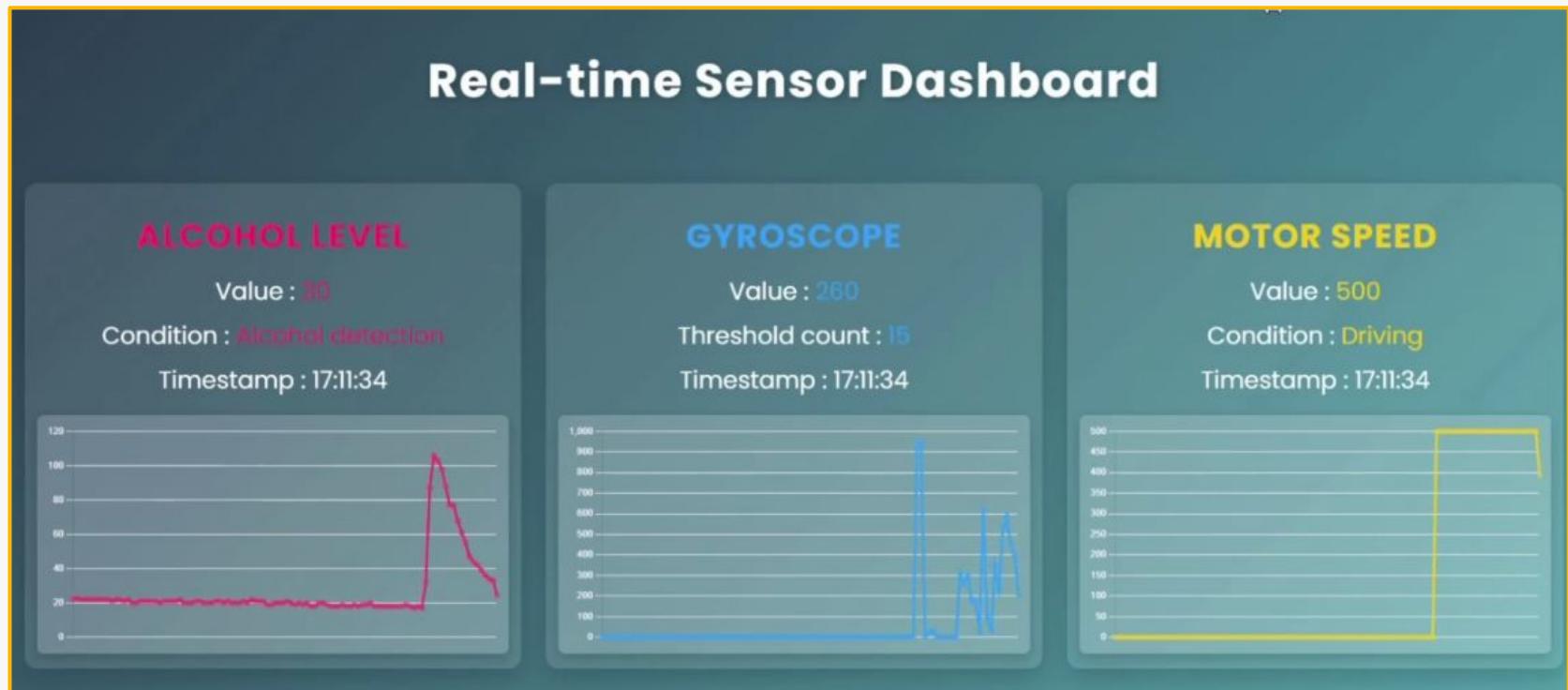
웹 서버 통신 :

ESP32 ↔ 웹 서버 HTTP 통신, 센서 데이터 전송



개발

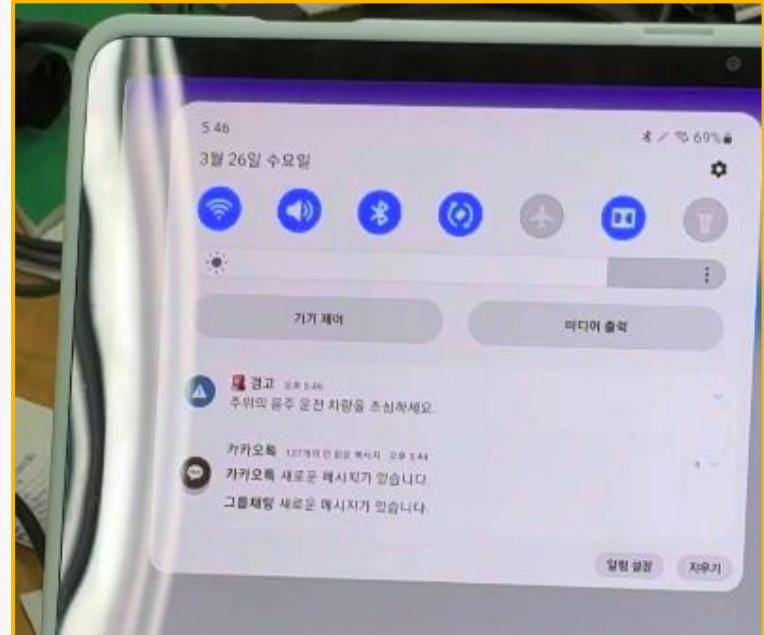
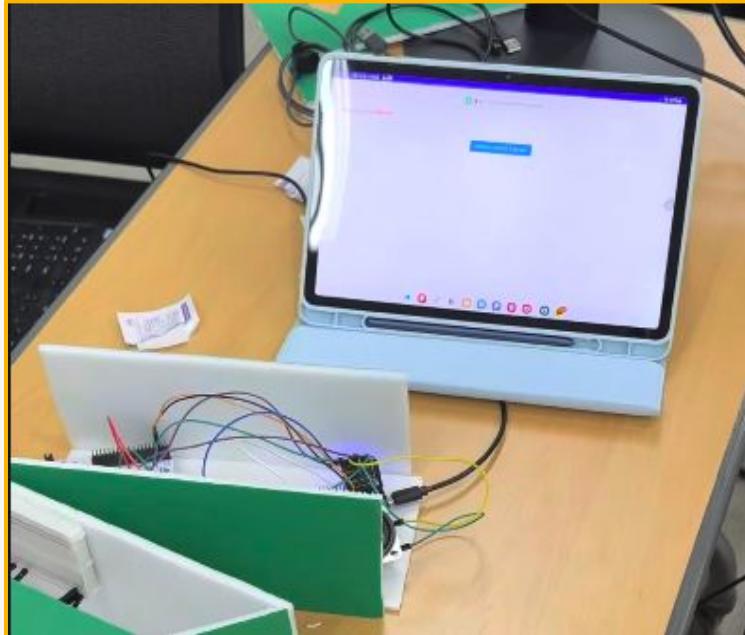
상세 설계 및 구현 - 센서 데이터 모니터링 대시보드 (**Django** 웹 서버)





개발

상세 설계 및 구현 - 다른 차량 스피커 알림 & 보행자 핸드폰 알림



06

테스팅





테스팅

단위 테스트

테스트 ID	테스트 명칭	테스트 목적	입력	예상 결과	테스트 결과
TC-Gyro					
전체조건					
- 자이로 센서가 정상 작동 상태일 것					
- 진동 입력이 가능한 상태일 것					
TC-G-01	정지 상태 유지	자이로 변화량 0에 근접	정지 상태	자이로 변화량 0에 근접	자이로 변화량 0에 근접
TC-G-02	1초간 작은 진동 입력	자이로 변화량 미미	1초간 작은 진동 입력	자이로 변화량 미미	자이로 변화량 미미
TC-G-03	1초간 큰 진동 입력	자이로 변화량 급증	1초간 큰 진동 입력	자이로 변화량 급증	자이로 변화량 급증
TC-G-04	10초간 5회 진동	누적 감지 수 5회 기록	10초간 5회 진동	누적 감지 수 5회 기록	누적 감지 수 5회 기록
TC-G-05	10초간 20회 진동	누적 감지 수 20회 초과, 2단계 진입	10초간 20회 진동	누적 감지 수 20회 초과, 2단계 진입	누적 감지 수 20회 초과, 2단계 진입
TC-G-06	자이로 센서 탈거	자이로 값 0 또는 비정상	자이로 센서 탈거	자이로 값 0 또는 비정상	자이로 값 0 또는 비정상
TC-G-07	자이로 센서 고정값 입력	자이로 변화 없음으로 판단	고정값 입력	자이로 변화 없음으로 판단	자이로 변화 없음으로 판단
TC-Alcohol					
전체조건					
- 음주 감지 센서가 정상 작동 상태일 것					
- 테스트용 알코올 표준 샘플(알코올 송)이 준비되어 있을 것					
- 알코올 레벨 단위(거리에 따른 일의의 기준 적용)					
TC-A-01	공기 중 노출	ADC 값 범위 검증	알코올 송과 센서 간 거리 5cm 테스트	ADC 값 10~20 유지	ADC 값 10~20 유지
TC-A-02	소량 알코올 노출	ADC 값 상승 검증	알코올 송과 센서 간 거리 2cm 테스트	ADC 값 50~80 범위 상승	ADC 값 50~80 범위 상승
TC-A-03	알코올 근접	ADC 값 100 이상 검증	알코올 송과 센서 간 거리 0.5cm 테스트	ADC 값 100 이상	ADC 값 100 이상
TC-A-04	알코올 농도 300 이상 유지	1단계 진입 트리거	알코올 농도 300 이상 유지	1단계 진입 트리거	1단계 진입 트리거
TC-A-05	알코올 감지 후 제거	1단계 종료 없음	알코올 송 제거 후 측정	값 감소, 1단계 종료 없음	값 감소, 1단계 종료 없음
TC-A-06	센서 탈거	센서 탈거 시 동작 확인	센서 탈거	ADC 값 0 또는 포화	ADC 값 0 또는 포화
TC-A-07	센서 고정 전압 입력	센서 고정 전압으로 상태 판단 불가	고정 전압 입력	값 고정으로 상태 판단 불가	값 고정으로 상태 판단 불가
TC-Motor					
전체조건					
- 모터가 정상 작동 상태일 것					
- PWM 제어 신호가 정상적으로 입력될 것					
- 테스트용 알코올 표준 샘플(알코올 송)이 준비되어 있을 것					
TC-M-01	초기 PWM 500 설정	모터 고속 회전	PWM 500 설정	모터 고속 회전	모터 고속 회전
TC-M-02	1단계 진입 전 유지	속도 유지	PWM 500 설정, 1단계 진입 전 유지	속도 유지	속도 유지
TC-M-03	2단계 진입 후 10초	PWM 400 설정	2단계 진입 후 10초, PWM 400 설정	PWM 400 설정 후 정상 동작	PWM 400 설정 후 정상 동작
TC-M-04	2단계 후 20초	PWM 300 설정	2단계 후 20초, PWM 300 설정	PWM 300 설정 후 동작	PWM 300 설정 후 동작
TC-M-05	2단계 후 50초	PWM 0까지 감속 완료	2단계 후 50초, PWM 0까지 감속 설정	PWM 0까지 감속 완료	PWM 0까지 감속 완료
TC-M-06	기초 주 액션을 제거	기초 기초	기초 주 액션을 제거 후 모터 드라이브 헤이 기초 기초 모터 드라이브 오기	기초 기초 모터 드라이브 오기	기초 기초 모터 드라이브 오기

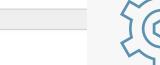




테스팅

통합 테스트

테스트 ID	테스트 명칭	테스트 목적	입력	예상 결과
TC-Server to ESP32 to STM32				
전체조건				
- 서버에서 고정된 포맷을 가지고 있을 것				
- 서버 URL 을 들어갔을 때 값이 들어와 있을 것				
TC-SES-01	정상 음주 감지 시 경고 송신	서버에서 음주운전 감지 값을 받아 STM32에 경고 신호를 전송하는지 확인	is_drunk_driving: true 포함된 JSON 수신	STM32가 "ALERT" 신호 수신 및 스피커 작동
TC-SES-02	정상 상태 수신 시 무반응 확인	서버에서 음주 아님 값을 받아 STM32가 무반응으로 처리하는지 확인	is_drunk_driving: false 포함된 JSON 수신	STM32 무반응, 스피커 OFF
TC-SES-03	서버 미응답 처리	서버 연결 실패 또는 응답 없을 상황에서 STM32에 잘못된 신호가 전송되지 않도록 확인	서버 오프라인 상태 또는 잘못된 URL	STM32 무반응, 예외 처리 로그
TC-SES-04	잘못된 JSON 포맷 수신 처리	필드 누락이나 오류 등 비정상 JSON이 수신될 경우 시스템이 오류를 방지하고 무시하는지 확인	{"is_drunk_drive": true} 또는 필드 누락 JSON	파싱 실패 → STM32 무반응
TC-SES-05	연속 음주 감지 처리	연속적으로 감지되는 음주 신호에 대해 STM32가 충복 작동 없이 알림을 유지하는지 확인	is_drunk_driving: true 값 10초간 반복 수신	알림 1회 유지 또는 일정 주기로 반복
TC-SES-06	음주 → 정상 전환 처리	음주 상태 감지 후 정상 상태로 전환 시 STM32가 알림을 중지하는지 확인	true 수신 후 false 수신	스피커 OFF, 상태 초기화
TC-SES-07	STM32 미연결 상태 확인	서버에서 음주 감지를 보내도 STM32가 오프라인일 경우, 시스템이 예외를 잘 처리하는지 확인	STM32 전원 OFF 상태, is_drunk_driving: true 수신	전송 실패 로그 출력, 오류 발생 안 함
TC-SES-08	스피커 연결 누락 시 반응 확인	STM32에서 신호 전송은 되지만 스피커 미연결 시 하드웨어 오류 없이 넘어가는지 확인	스피커 제거 상태에서 is_drunk_driving: true 수신	경고음 없음, STM32 명령 전송 OK
TC-SES-09	오작동 방지 위한 디버깅 테스트	서버 JSON이 0 또는 고정값일 때 비정상 작동이 없는지 확인	alcohol=0, gyro=0, motor_speed=0 상태 반복 수신	"정상"으로 판단, 경고 없음
TC-Server to App				
전체조건				
- 서버에서 고정된 포맷을 가지고 있을 것				
- 서버 URL 을 들어갔을 때 값이 들어와 있을 것				
TC-SA-01	정상 응답 수신	앱이 서버로부터 is_drunk_driving 값을 정상적으로 수신하는지 확인	서버 응답: {"is_drunk_driving": true}	앱이 JSON 파싱 성공 후 알림 트리거
TC-SA-02	음주 아님 응답 처리	앱이 false 값에 대해 정상적으로 무반응 처리하는지 검증	서버 응답: {"is_drunk_driving": false}	앱은 로그 출력無, 알림 없음
TC-SA-03	응답 필드 누락	서비스 응답에 필드가 빠졌을 때 앱이 예의 없이 동작 유지하는지 확인	응답: {"status": "success"}	앱에서 예상 오류 발생, 예외 로그 출력 후 멤 계속 실행
TC-SA-04	응답 구조 오류	JSON 구조가 잘못됐을 경우 앱이 예외 처리하는지 확인	응답: {"is_drunk_driving: true } (문법 오류)	JSON 파싱 실패 → catch 블록에서 오류 처리
TC-SA-05	네트워크 끊김	Wi-Fi 해제 후 서버 호출 시 앱 반응 확인	서버 접속 실패	앱에서 ConnectException 또는 TimeoutException 처리, UI 명령 없음
TC-SA-06	서버 지연 응답	응답까지 5초 이상 걸리는 경우 앱의 처리 확인	5초 후 응답 반환	앱은 응답 대기 후 정상 파싱, 타임아웃 없으면 정상 알림
TC-SA-07	서버 500 오류 발생	서비스에서 내부 에러 발생 시 앱의 예외 처리 여부 확인	서버 응답: HTTP 500	앱에서 예외 로그 출력, 알림 없이 종료 또는 재시도 가능
TC-SA-08	주기적 호출 테스트	일정 주기로 데이터를 호출하여 정상 작동 여부 확인	5초마다 서버 요청 → 정상 응답	앱은 모든 응답을 처리, is_drunk_driving=true 시 알림 반복 발생
TC-SA-09	앱 재실행 후 호출	앱을 깨끗이 켰을 때 첫 호출이 정상적으로 작동하는지 확인	앱 재시작 후 자동 요청	앱이 정상적으로 서버 호출 및 응답 파싱 수행
TC-SA-10	잘못된 URL	API 주소가 잘못됐을 경우 앱 동작 확인	잘못된 URL: http://wrong-url.com/data	앱에서 HTTP 예의 발생, catch 블록에서 처리됨
TC-Sensor to Board				
전체조건				
- 리셋 버튼 누르면 초기화 상태 진입				
- 센서(스피커/모터 연결 완료 및 전원 공급됨)				
- MCU (STM32) 와 센서, 스피커가 SPI/UART 등으로 정상 통신 중				
TC-SB-01	초기화 트리거	리셋 버튼 입력 시 시스템이 초기화 흐름을 시작하는지 검증	리셋 버튼 눌림	모터 동작 시작, 스피커 1번 트랙 재생
TC-SB-02	알콜 감지 동작	알콜 센서가 기준치를 초과할 때 다음 단계로 전이되는지 확인	알콜 농도 > 기준치	스피커 2번 트랙 재생, 상태값 업데이트
TC-SB-03	파이썬 기초화 미다	파이썬 세션 초기화 시 시스템이 견뎌 오류 없이 핑크 핑크	파이썬 초기화 미다	파이썬 도자기 어오 기초 시대 오지





테스팅

시스템 테스트

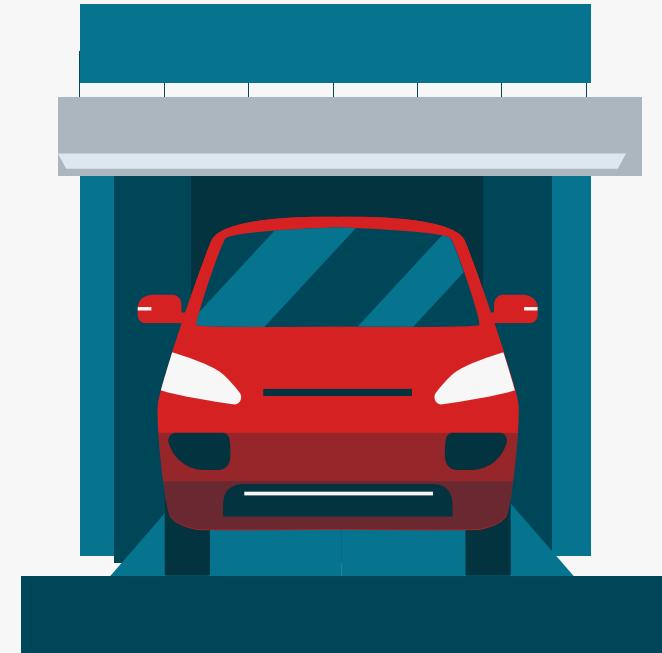


테스트 ID	테스트 명칭	테스트 목적	입력	예상 결과
TC-ST-01	전체 시스템 정상 작동 확인	시스템이 정상적으로 작동하는지 확인	모든 센서 및 모듈 연결 후 시스템 시작	모든 센서가 정상 작동하고, 시스템이 초기화됨
TC-ST-02	자이로 센서 및 알콜 감지 연동	자이로 센서와 알콜 감지 연동 확인	자이로 센서 값 100 이상, 알콜 센서 값 150 입력	알콜 감지 상태가 "Alcohol detection"으로 변경되고, 자이로 카운트 증가
TC-ST-03	모터 스피드 설정 확인	모터 스피드 상태가 정상적으로 설정되는지 확인	PWM 500 입력	모터 스피드 상태가 "Driving"으로 설정됨
TC-ST-04	모터 스피드 경고 출력 확인	모터 스피드가 200일 때 경고 메시지 확인	모터 스피드(PWM) 200 설정	스피커에서 "10초 후 차량이 정지됩니다." 출력
TC-ST-05	자이로 카운트 증가 확인	자이로 센서 값에 따라 카운트를 증가시킴	자이로 센서 값 100 이상 입력	카운트가 1 증가, JSON 응답에 카운트 포함
TC-ST-06	20초 경과 후 상태 확인	20초 후 카운트에 따라 상태 확인	자이로 센서 값 100을 20초 동안 유지	200 OK 응답, JSON 데이터에 is_drunk_driving: true
TC-ST-07	알콜 감지 후 상태 확인	알콜 감지 후 상태 변화 확인	알콜 감지 상태에서 값 감소	상태가 "Normal"으로 변경됨
TC-ST-08	모터 작동 확인	모터가 정상적으로 작동하는지 확인	알콜 감지 상태에서 모터 작동 요청	모터가 정상적으로 작동 시작
TC-ST-09	시스템 장애 처리 확인	시스템 장애 시 적절한 오류 처리 확인	자이로 센서 탈거 후 요청	오류 메시지 반환, 시스템 안전 모드 활성화
TC-ST-10	ESP32로 True 신호 전송 확인	자이로 센서 조건 만족 시 ESP32로 True 신호 전송 확인	자이로 센서가 100 이상으로 20초 동안 10회 흔들림	ESP32가 True 신호를 수신하고, 스피커에서 "즉시 차량을 멈추십시오." 출력
TC-ST-11	알콜 감지 음성 출력 확인	알콜 감지 시 음성 출력 확인	알콜 센서에 알콜 솔을 대고 100 이상 측정	스피커에서 "음주 상태가 감지되었습니다. 차량이 제한될 수 있습니다." 출력
TC-ST-12	자이로 흔들기 알림 확인	자이로 센서 흔들기로 인한 경고 메시지 확인	자이로 센서 흔들기 (100 이상 카운트 10회)	조심하세요. 주위에 음주운전 차량이 있습니다. 메시지가 출력됨
TC-ST-13	모터 스피드 감속 확인	모터 스피드가 100 미만으로 감소했을 때 확인	자이로 센서가 100 이상이고 20초 동안 10회 흔들림	모터 속도가 100씩 감소하고, 스피커에서 "즉시 차량을 멈추십시오." 출력
TC-ST-14	상태 확인 메서드 검증	상태 확인 메서드의 정상 동작 검증	GET 요청 (상태 확인)	현재 상태 반환
TC-ST-15	시스템 전체 통합 검증	모든 시스템 모듈이 통합되어 작동하는지 확인	모든 센서 및 모듈 정상 입력	모든 시스템 모듈이 정상적으로 작동함



07

업무 분장 / 협업





업무 분장



장술우
센서 통합 및 코드
설계
알코올, 모터 센서



윤용성
웹 서버 및 **DB** 구축
보드 통신



오동훈
보드 통신
안드로이드 앱 개발



이지호
스피커 센서
요구사항 분석



민경서
보드 통신
문서 관리



협업

Notion, Github



The screenshot shows a Notion workspace with a To Do list. The list items are:

- 준비물 - 알콜 감지 센서 구입
- 준비물 - 선 구매
- 개발 환경 설정 - 파이썬 웹서버 구축
- 개발 환경 설정 - 라즈베리 기초 OS 세팅
- 개발 환경 설정 - ESP32 vscode 세팅
- 요구사항 분석서
- 시스템 아키텍처 설계서
- 인터페이스 설계서
- 모듈 설계 - 모터
- 모듈 설계 - 알콜 센서
- 모듈 설계 - 스피커
- ESP & Django 서버 통신
- 모듈 통합
- UART - ESP32 통신
- 단위테스트 - 테스트 레포트
- 통합&시스템테스트 - 테스트 레포트
- 발표자료 PPT
- 시연영상
- 한줄평

Each item has a status column (완료 or 진행 중) and a responsible person column (장승원, 민민경서, 윤윤성).



AlcoholWatcher

README.md

AlcoholWatcher

주요 기능

- 실시간 혈중알코올농도 측정: 센서를 통해 사용자의 혈중알코올농도를 감지
- 차량 시동 제어: 알코올 수치가 기준치를 초과하면 차량 시동 차단
- 운전자 데이터 기록: 운전자의 음주 여부 및 운전 기록 저장
- 모바일 앱 연동: 경고 알림 및 데이터 확인 기능 제공

기술 스택

- 하드웨어: 알코올 센서 (MQ-3 등)
- 임베디드 시스템: Raspberry Pi / Arduino
- 백엔드: Python (Flask, FastAPI)
- 프론트엔드: React Native (모바일 앱)
- 데이터베이스: PostgreSQL

설치 및 실행 방법

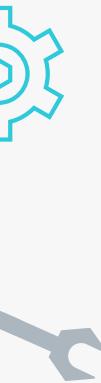
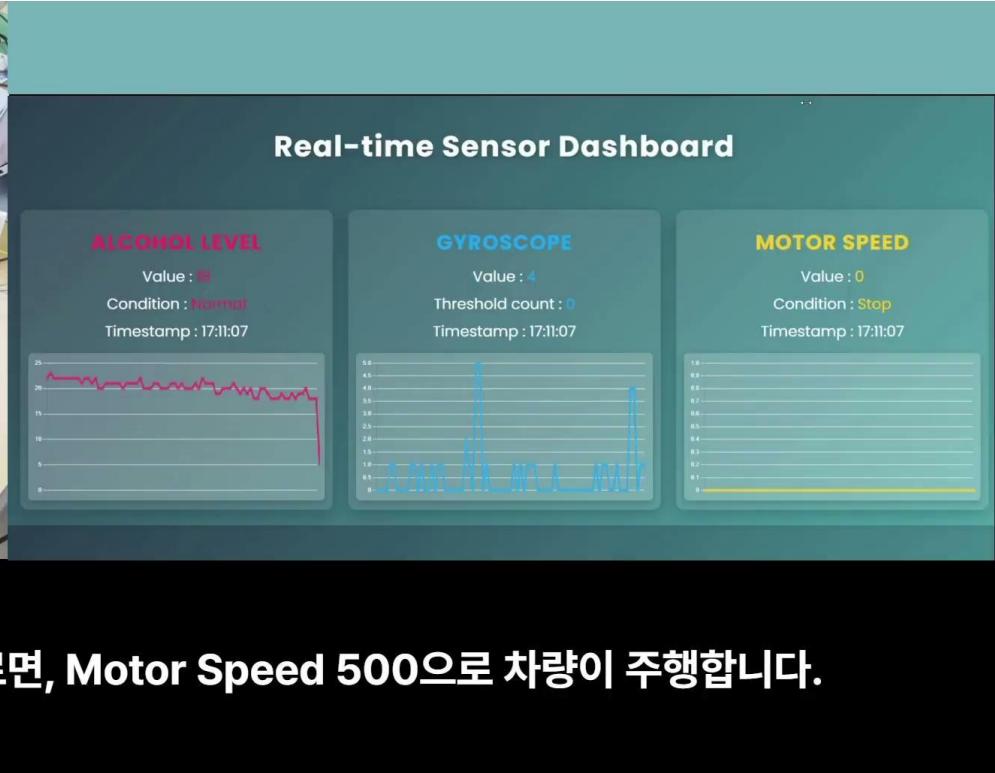
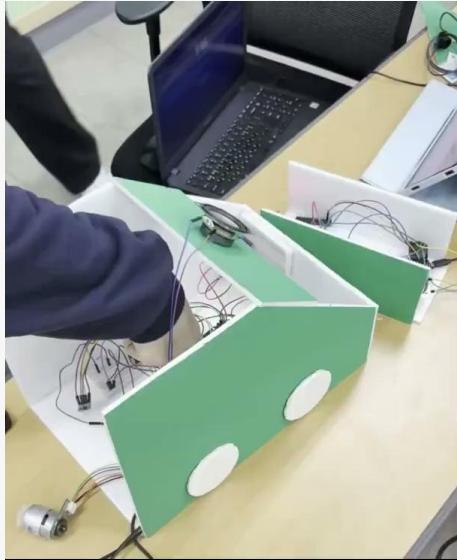
- 프로젝트 클론

08

시연영상 / 회고



시연 영상





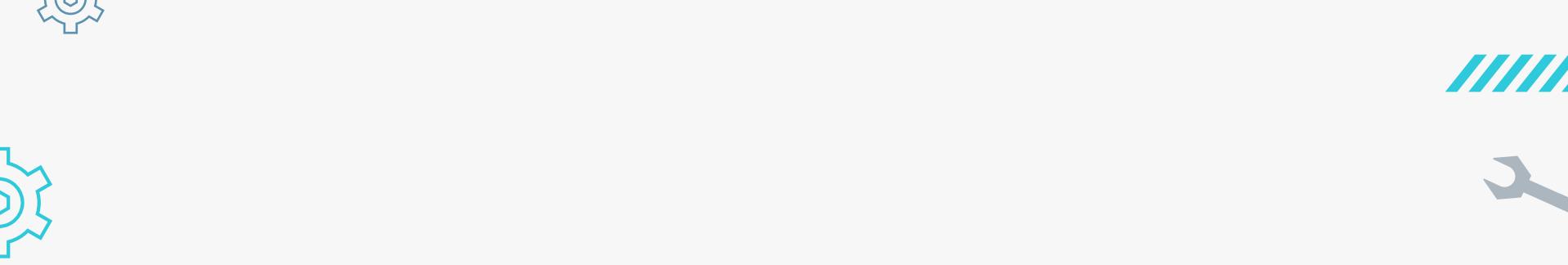
회고

배운 점, 어려웠던 점, 아쉬웠던 점

이지호	임베디드 시스템 설계를 진행하면서 테스트 케이스를 잘 정립하고 디버깅을 철저히 하지 않으면, 문제 발생 시 어느 부분에서 문제가 생긴 것인지 파악하기 어려운 점을 느낍니다. 이를 통해 단위별, 모듈별, 시스템별로 테스트를 진행하고, 산출물을 체계적으로 작성하여 정리해 나가는 과정이 프로젝트 성공에 중요한 부분임을 깨달았습니다.
민경서	STM32 마이크로컨트롤러, ESP32 통신 모듈, 다양한 센서들을 연동하는 과정에서 실제 임베디드 시스템 개발의 복잡성과 그 해결 방법을 배울 수 있었습니다. 테스트 결과 기록을 더 상세히 남기고 문제 해결 과정을 문서화하는 부분에서 개선이 필요함을 느꼈습니다. 향후 프로젝트에서는 테스트 문서의 버전 관리와 결과 공유 프로세스를 더 체계화하고 싶습니다.
오동훈	개별 기능 개발 단계에서 사전에 디버깅을 철저히 해두는 것이 협업 효율에 큰 영향을 미친다는 점을 깨달았습니다.
장승원	설계를 한 번에 완벽하게 한다는 건 정말 어려운 것을 느낍니다. 한단계씩 고쳐나가는 부분에서 힘들었지만 많은 것을 배울 수 있었습니다.
윤용성	이번 프로젝트에서 순차 로직을 구현하는 데 어려움을 겪었습니다. 각 기능이 서로 의존하는 구조에서 로직 흐름을 명확히 설정하는 것이 쉽지 않았습니다. 또한, 새로 고침 시 초기화 로직을 제대로 작성하지 못해 시스템 상태를 유지하는데 문제가 발생했습니다. 이러한 점들을 반성하며, 다음 프로젝트에서는 보다 체계적으로 로직을 설계하고 초기화 로직을 철저히 구현해 나가겠습니다.







Alcohol Watcher

음주 운전 방지 시스템

감사합니
다

