

캡스톤디자인

최 종 결 과 보 고 서

시각장애인을 위한

음성명령 차량제어 시스템

톡톡카 : 김서준, 김규현, 차규진

단국대학교 소프트웨어학과

제 출 문

본 보고서를 시각장애인을 위한 음성명령 차량제어 시스템 개발의 최종결과보고서로 제출합니다.

2025 년 6 월 11 일

팀장 : 김서준
팀원 : 김규현
팀원 : 차규진

요 약 문

1. 제 목

시각장애인을 위한 음성명령 차량제어 시스템

2. 프로젝트 개요 및 필요성

현대 사회에서 자동차는 필수적인 이동 수단이지만, 시각장애인은 운전뿐 아니라 차량 제어와 같은 기본적인 기능에도 접근이 어려운 것이 현실이다. 차량의 도어 개폐, 시동, 주행 속도 조절, 연료 상태 확인 등 일상적인 기능 조작 시각 정보를 기반으로 설계되어 있어 시각장애인의 자율적 이용은 극히 제한적이다.

이러한 문제를 해결하고, 정보 취약 계층의 이동 편의성을 향상시키기 위해 본 프로젝트는 음성 명령 기반의 차량 제어 시스템을 제안하였다. 특히, 사전 정의된 명령어가 아닌 자연어 형태의 자유 음성을 이해하고 실행할 수 있도록 AI 기반의 명령 해석기를 탑재함으로써, 직관적이고 실용적인 접근성을 구현하였다.

3. 프로젝트의 내용 및 개발 범위

본 프로젝트는 시각장애인이 음성만으로 차량의 주요 기능을 제어할 수 있도록 하기 위해 다음과 같은 범위로 개발되었다:

- (1) Flutter 기반의 모바일 앱 UI 및 음성 인터페이스
- (2) FastAPI 기반 백엔드 서버 및 WebSocket 통신
- (3) 음성 명령어를 AI 모델(Phi-2 등)을 통해 해석하여 제어 명령으로 변환
- (4) Arduino 를 통한 실제 하드웨어 제어 (시동 LED, 서보모터, 연료 RGB LED, 경고음, 문 위치 안내 스피커 등)
- (5) 상황 인식 기능 (폭우 시 명령 제한, 응급상황 시 보호자 자동 알림 등)

이 시스템은 단순한 음성 인식 수준을 넘어, 시각장애인의 실제 사용 맥락을 고려한 직관적 피드백과 멀티 디바이스 연계를 통해 더욱 실용적인 접근성을 제공한다.

4. 개발 결과 요약

최종 구현된 시스템은 음성 명령 입력 후, 이를 분석하여 해당 명령어를 하드웨어로 전달하는 구조로 설계되었으며, 다음 기능을 성공적으로 시연하였다:

- (1) "시동 켜줘", "문 열어줘", "속도 느리게", "연료 " 등의 다양한 자유 명령어 인식 및 실행
- (2) 시각장애인을 위한 스피커 기반 문 위치 안내 기능
- (3) 배터리 잔량에 따른 RGB LED 색상 출력
- (4) 응급상황 발생 시 보호자에게 자동 문자 발송 기능
- (5) 폭우 등 위험 조건에서 특정 명령 제한 기능

5. 활용에 대한 건의 및 기대 효과

본 시스템은 시각장애인뿐 아니라 노약자, 손이 자유롭지 못한 사용자, 긴급 상황에 처한 일반 사용자에게도 유용하게 활용될 수 있는 잠재력을 지닌다. 특히, 자율주행 기술과 결합될 경우, 보조적 제어 시스템으로 확장 가능성이 있으며, 교통약자의 차량 이용 접근성을 획기적으로 향상시킬 수 있다. 향후에는 GPS, AI 모델 고도화, 클라우드 연동 기능 등을 추가하여 스마트 카 환경에 접목시킬 수 있을 것으로 기대된다.

제 1 장 서론

제 1 절 프로젝트 개발의 필요성

현대 사회는 기술의 발전과 함께 교통수단의 접근성 향상에 주목하고 있다. 그러나 대부분의 차량 제어 시스템은 시각 정보를 기반으로 하여 설계되어 있으며, 이는 시각장애인을 포함한 교통약자에게 심각한 제약을 초래한다. 차량의 시동, 문 개폐, 속도 조절, 연료 확인 등 기본적인 기능조차 시각적 디스플레이나 버튼 조작에 의존하고 있어, 시각장애인은 독립적으로 차량을 활용하는 데 큰 어려움을 겪는다.

실제로 오늘날 차량 한 대에는 평균적으로 200개 이상의 다양한 기능이 탑재되어 있으며, 고급 차량의 경우 이 수는 300~500개에 이른다. 예를 들어 테슬라 모델 3, 현대 아이오닉5, 제네시스 G80 등의 차량은 수백 개의 제어 기능을 제공하지만, 이 중 시각장애인이 실질적으로 접근 가능하고 사용할 수 있는 기능은 **전체의 약 20%에 불과**하다. 나머지 약 80%는 터치스크린, 시각 기반의 UI 등에 의존하고 있어, 시각장애인은 사실상 차량의 주요 기능에서 소외되어 있다.

기존 차량 제조사들도 일부 기능에 대해 음성 명령이나 물리 버튼을 제공하고 있으나, 핵심적인 차량 제어는 여전히 디지털 디스플레이 중심의 UI 설계로 되어 있어 시각장애인의 접근성에는 한계가 있다. 결과적으로, 시각장애인은 차량이 제공하는 다양한 기능과 옵션에 접근하지 못하고, 실사용 범위가 극히 제한되는 문제가 지속되고 있다.

이러한 현실적 한계를 극복하기 위해, 본 프로젝트는 시각장애인을 위한 **음성 기반 차량 제어 시스템**을 개발하였다. 본 시스템은 시각정보 없이도 차량의 주요 기능을 제어할 수 있도록 돕고, 직관적인 사용자 경험을 제공

함으로써 정보 격차를 해소하는 것을 목적으로 한다.

또한, 단순한 음성 인식 기능을 넘어서, **자연어 명령어를 이해하고 해석하는 AI 기반 언어 모델(Large Language Model)**을 연동하고, **TTS 기반 음성 안내, WebSocket 실시간 통신, 사용자 맞춤형 피드백 제공** 등 다양한 기술을 통합함으로써, 기존 시스템의 한계를 보완하고 미래 모빌리티 환경에 걸맞은 접근성 솔루션을 구현하였다. 본 서비스는 자율주행, 스마트카, 전기차 등 미래차 환경에서도 확장 가능한 범용 플랫폼으로 진화 가능성이 높다.

제 2 절 프로젝트 목표

본 프로젝트는 시각장애인을 포함한 교통약자가 차량을 보다 안전하고 효율적으로 사용할 수 있도록, 음성 기반의 직관적인 제어 시스템을 구축하는 것을 주된 목표로 한다. 구체적인 목표는 다음과 같다:

(1) 음성 명령을 통한 차량 기능 제어

시각장애인이 시각 정보 없이도 차량의 시동, 주행 속도, 도어 개폐, 연료 상태 확인 등의 주요 기능을 음성만으로 제어할 수 있도록 한다.

(2) 자연어 명령 해석 기능 탑재

단순 명령어가 아닌 자유로운 일상 언어를 AI 가 해석하여 표준 명령어로 변환, 실제 차량 동작으로 연결한다.

(3) 하드웨어 기반 제어 구현

Arduino 를 활용하여 실제 시동 LED, 서보모터 기반 속도계, 연료 RGB LED, 문 안내 스피커 등 실물 제어 장치를 구현한다.

(4) 응급상황 대응 기능 구현

사용자에게 위급 상황이 발생했을 때, 보호자에게 자동 알림이 전송되도록 하여 안전성을 확보한다.

(5) Flutter 앱 기반 UI 개발

사용자 친화적인 앱 인터페이스를 제공하며, 앱 실행 시 즉시 음성 인식이 시작되어 시각장애인도 쉽게 접근할 수 있도록 한다.

(6) 폭우 등 위험 조건에서 명령 제한 기능 탑재

실시간 상황 인식 기반으로 위험 시 특정 명령이 제한되어 사용자와 주변의 안전을 확보한다.

본 프로젝트는 단순한 음성 인식 제어를 넘어, 실제 시각장애인의 차량 이용 환경을 고려한 전방위적 접근성 제어 시스템을 구축하는 데 목표를 둔다.

제 2 장 국내·외 관련 프로젝트 동향 파악 및 분석

시각장애인을 위한 차량 접근성과 제어 기술은 아직까지도 제한적으로 개발되고 있으며, 대다수의 프로젝트는 일반 사용자 또는 자율주행 기술에 집중되어 있다. 그럼에도 불구하고, 음성 제어 및 스마트카 관련 몇몇 기술은 우리 프로젝트의 참고 사례가 될 수 있다.

1. 국내 사례

(1) 현대자동차 – 음성 인식 제어 시스템 (ccIC)

현대자동차는 ccIC(connected car Integrated Cockpit)라는 통합 제어 플랫폼을 통해 차량 내 음성 제어 기능을 제공하고 있다. 운전자는 “에어컨 켜줘”,

“창문 내려줘”와 같은 명령을 통해 차량 내부 기능을 제어할 수 있다. 그러나 해당 시스템은 사전 정의된 명령어 기반으로 작동하며, 자연어 처리 능력은 제한적이다. 또한 시각장애인을 위한 접근성 설계는 전혀 고려되어 있지 않다.

(2) 보행 보조 스마트 장갑/지팡이

국내 일부 연구 기관에서는 시각장애인을 위한 스마트 장갑이나 지팡이를 개발하여 장애물 감지 기능을 제공하고 있다. 그러나 이는 차량 운전이나 차량 제어와는 거리가 있으며, 독립적인 이동성 확보에는 한계가 있다.

2. 국외 사례

(1) Tesla – Voice Command

Tesla 차량은 음성 명령을 통해 내비게이션, 전화, 미디어 등을 제어할 수 있다. 최근에는 “Open the glovebox”나 “Set the temperature to 22 degrees”와 같은 자연어 명령을 점진적으로 지원하고 있다. 그러나 여전히 기능 범위는 제한적이며, 장애인을 위한 특화된 UX 설계는 적용되지 않았다.

(2) Amazon Alexa Auto

Alexa Auto 는 차량에 탑재되어 음악 재생, 위치 안내, 전화 등의 기능을 수행할 수 있으며, 스마트홈과 연계도 가능하다. 그러나 차량 제어 기능은 제조사와의 호환성에 따라 제한되며, 시각장애인 사용자 인터페이스 설계는 미흡하다.

3. 기존 소프트웨어의 한계점 및 우리 프로젝트의 개선 방향

기존의 차량 음성 제어 시스템들은 대부분 정형화된 명령어, 시각 중심의 UI, 일반 사용자 대상 설계라는 공통적인 한계를 지닌다. 특히 시각장애인을 고려한 기능은 거의 존재하지 않으며, 진정한 접근성을 제공하지 못하고 있다. 이에 반해, 본 프로젝트는 다음과 같은 차별점을 갖는다:

- (1) 자연어 기반 자유 음성 해석: 사전에 정의된 명령어가 아닌, 일반 사용자의 말투나 표현을 인공지능 모델이 해석하여 명령으로 변환한다.
- (2) 완전한 시각 정보 제거: UI 없이도 동작이 가능하며, 음성 출력과 진동, LED 등 비시각적 피드백만으로 사용자가 차량을 조작할 수 있다.
- (3) 시각장애인을 위한 전용 기능: 문 열림 시 방향성 안내 음성, 응급 상황 자동 알림, 위험 상황에서 명령 제한 등 실제 상황을 고려한 설계가 이루어졌다.

이러한 분석을 통해, 본 프로젝트는 기존 기술의 틈새를 메우며 접근성 중심 스마트 차량 제어 시스템이라는 새로운 패러다임을 제시하고자 한다.

제 3 장 프로젝트 설계

제 1 절 프로젝트 개념 설계

본 프로젝트의 핵심 설계 개념은 “시각장애인을 위한 완전한 비시각적 차량 제어”이다. 기존 차량 시스템은 대부분 시각 정보(계기판, 화면, 버튼 등)

를 기반으로 설계되어 있어 시각장애인의 독립적 사용이 사실상 불가능하다. 따라서 본 시스템은 음성 기반 입력, 청각 및 촉각 기반 피드백, 그리고 AI 기반 명령 해석을 통해 시각 정보에 의존하지 않고 차량을 제어할 수 있도록 하는 것을 설계 원칙으로 삼는다.

1. 사용자 중심의 접근성 설계

사용자는 Flutter 기반의 모바일 앱을 실행한 직후 음성 명령을 자연스럽게 말할 수 있으며, 앱은 음성을 즉시 인식하고 서버로 전송한다. 이때 사용자는 복잡한 UI 조작 없이 음성만으로 시스템을 구동할 수 있다. 이러한 설계는 시각장애인을 포함한 정보 취약 계층이 기술에 쉽게 접근할 수 있도록 돕는다.

2. 명령어 자유화를 위한 인공지능 해석기 탑재

일반적인 음성 제어 시스템은 “시동 켜줘”, “에어컨 켜” 등 미리 정의된 정형 명령어에만 반응한다. 반면, 본 프로젝트에서는 자연어 처리 기반 명령 해석기를 활용하여 “출발해볼까?”, “좀 더 빨리 가자”, “지금 배터리 어때?” 처럼 일상 언어 기반의 표현도 해석할 수 있도록 설계하였다. 이를 통해 실제 사용자와의 상호작용 자연스러움이 크게 향상된다.

3. 하드웨어 기반 실시간 제어

인공지능이 해석한 명령은 FastAPI 기반 백엔드 서버를 통해 시리얼 통신으로 Arduino 에 전달되며, Arduino 는 다음과 같은 하드웨어를 제어한다:

- (1) 시동 LED: 차량 전원이 켜졌음을 시각 대신 촉각 및 음성으로 알려줌
- (2) 서보모터 기반 속도계: 속도 수준에 따라 모터가 회전하여 속도를 간접적으로 표현
- (3) RGB LED: 배터리 잔량에 따라 색상 변화 (적색/황색/녹색)
- (4) 경고 LED 및 스피커: 응급상황이나 충돌 시 시각+청각적 알림
- (5) 문 위치 안내 스피커: 문 열림 시 반복 재생되는 소리로 방향 감지 가능

4. 상황 기반 제어 로직 내장

시스템은 단순히 명령을 해석하고 실행하는 수준을 넘어서, 외부 상황에 따라 제어 정책을 다르게 적용한다. 예를 들어, 사용자가 “출발해”라고 말했을 때 폭우가 감지되면 주행을 차단하고 “현재 폭우 상태로 주행이 제한됩니다”라는 안내 음성을 출력한다. 이로써 사용자의 안전성과 시스템 신뢰성이 동시에 확보된다.

5. 비상 상황 대응 설계

시각장애인이 위급상황을 음성으로 알릴 경우, 시스템은 미리 등록된 보호자 번호로 자동 문자 또는 앱 알림을 발송한다. 이는 사용자 보호뿐만 아니라 사회적 신뢰도 확보에도 기여하는 기능이다.

이와 같이 본 프로젝트는 단순한 음성 제어를 넘어서, 시각장애인의 실제 생활 환경을 반영한 실사용 중심 설계를 기반으로 한다. 이러한 개념 설계를 토대로 다음 절에서는 각 구성 요소에 대한 상세 설계를 다룬다.

제 2 절 프로젝트 상세 설계

본 절에서는 개념 설계를 바탕으로 실제 시스템을 구성하는 각 요소의 상세 구조를 설명한다. 본 프로젝트는 다음과 같은 4 개의 주요 구성 요소로 구성된다:

- (1) Flutter 기반 모바일 애플리케이션 (음성 명령 UI 및 TTS 출력)
- (2) FastAPI 기반 백엔드 서버 (음성 명령 해석 및 중계 역할)
- (3) 자연어 명령 해석기 (Phi-2 기반 자유 명령 해석)
- (4) Arduino 하드웨어 제어부 (실제 차량 제어 기능 구현)

1. 전체 시스템 아키텍처

사용자

↓ (자연어 음성 입력)

Flutter 앱 (음성 인식)

↓ (WebSocket 전송)

FastAPI 서버

↓ (명령어 해석)

LLM 기반 해석기 (Phi-2)

↓ (명령 전환)

Arduino (시리얼 통신)

→ 시동, 속도계, 문 안내, 연료 LED 등 제어

이러한 구조를 통해 사용자는 복잡한 UI 없이 음성으로만 제어할 수 있으

며, 서버는 이를 실시간으로 해석하고 하드웨어로 전송한다.

2. Flutter 앱 구성

- (1) 음성 인식 모듈: 앱 실행 즉시 마이크가 활성화되고 `speech_to_text` 라이브러리를 통해 사용자의 명령을 텍스트로 변환한다.
- (2) WebSocket 클라이언트: 실시간 명령 전송을 위해 FastAPI 백엔드와 WebSocket 으로 연결되며, JWT 인증 방식으로 보호된다.
- (3) TTS 출력 모듈: 서버로부터 받은 응답을 `flutter_tts` 를 이용해 음성으로 출력하여, 시각 정보를 대체한다.
- (4) 비상 상황 버튼: 사용자가 직접 응급 상황을 전송할 수 있도록 빨간 버튼을 제공하며, 누르면 보호자에게 긴급 메시지가 발송된다.

3. FastAPI 백엔드 서버

- (1) WebSocket 엔드포인트: `/ws?token=...` 형식으로 연결되며, JWT 토큰을 이용해 인증된 사용자의 명령만 처리한다.
- (2) 명령 처리기: 수신된 음성 텍스트를 자연어 명령 해석기로 전달하여 표준화된 제어 명령으로 변환한다.
- (3) 시리얼 포트 통신: 해석된 명령을 Arduino 보드에 시리얼 통신 (COM6 등)으로 전송하여 하드웨어 제어를 수행한다.

4. 자연어 명령 해석기

OPENAI 의 GPT 모델을 기반으로, “주행 시작해볼까?”, “조금 더 천천히 가

취” 등 자유로운 자연어 명령을 표준 명령어(S40, S20, 0, 1 등)로 변환한다.

예시 매핑:

(1) “주행 시작” → “S40”

(2) “속도 느리게” → 현재 속도 보다 10km/h 감속

(3) “연료” → 배터리 퍼센트 응답

5. Arduino 하드웨어 설계

Arduino 는 다음과 같은 출력 장치를 통해 실제 차량 제어를 모사한다:

기능	사용부품	설명
시동 제어	LED (핀 9)	명령 0 → 시동 ON 명령 1 → 시동 OFF
속도 표시	서보모터	“S40” → 40 도 “S10” → 10 도 등
연료 표시	RGB LED (핀 3,4,5)	0~30%: RED 31~60%: YELLOW 61~100%: GREEN
문 안내 알림	스피커 (핀 7)	문 열림 시 반복 멜로 디 재생
경고 출력	경고등 (핀 2)	응급상황 또는 충돌 발 생 시 점등 및 경고음

6. 상황 인식 기능

- (1) 폭우 상황 판단: 사용자가 위험 상황을 설정할 수 있으며, "폭우 상태"에 서는 "출발", "속도 증가" 등 명령이 차단된다.
- (2) 응급 상황 처리: 응급 버튼을 누르면 서버가 보호자 연락처를 읽어 자동 으로 알림 메시지를 발송한다.

제 3 절 프로젝트 개발을 위한 환경 구축

본 절에서는 프로젝트 수행을 위해 구축한 개발 환경 및 시스템 구성 요소 들을 설명한다. 소프트웨어와 하드웨어가 유기적으로 연결되어야 하는 본 프로젝트 특성상, 모바일 앱 개발, 백엔드 서버, 하드웨어 제어 플랫폼 간의 연동을 고려한 환경 설계가 중요하였다.

1. 전체 시스템 구성

구성 요소	주요 역할	기술 스택 / 장비
모바일 앱	음성 명령 입력, 사용 자 인터페이스	Flutter, Dart, Android Emulator
백엔드 서버	명령 해석, 인증, 하드 웨어 중계	FastAPI, Python 3.10, Uvicorn
명령어 해석기	자연어 음성 → 표준 명령 변환	Microsoft Phi-2, Transformers

하드웨어 제어부	실제 시동 / 속도 / 문 제어 등 구현	Arduino Uno, Servo, LED 등
통신 방식	실시간 제어 신호 전달	WebSocket, UART(Serial)

2. 소프트웨어 개발 환경

(1) Flutter 환경

- SDK: Flutter 3.16.4
- 주요 패키지: speech_to_text, flutter_tts, web_socket_등
- 개발 툴: Android Studio, VSCode
- 테스트 플랫폼: Android Emulator (Pixel 4a API 31)

(2) 백엔드 서버

- Python 3.10
- FastAPI 0.100+
- Uvicorn ASGI 서버
- JWT 인증 구현: python-jose, passlib
- 명령어 해석 모델 로딩: transformers, torch
- 운영환경: Windows 10 / Ubuntu 22.04 (WSL)

(3) 자연어 해석기

- 모델: OpenAI ChatGPT API (gpt-3.5-turbo)
- 해석 방식: 클라우드 기반 LLM API 호출을 통해 명령어 해석을 수행
- 입력 처리 로직: 사용자 음성 → 텍스트 변환 → ChatGPT API 로 전달

→ 표준 명령어로 변환된 응답 수신 → 하드웨어로 전송

3. 하드웨어 제어 환경

(1) Arduino Uno

- IDE: Arduino IDE 2.x
- 보드 연결: USB 직렬통신 (COM6)
- 통신 속도: 9600 baud
- 사용 핀:
 - 9 번: 시동 LED
 - 3,4,5 번: RGB 연료 LED
 - 2 번: 경고등
 - 6 번: 서보모터
 - 7 번: 문 열림 스피커

(2) 전원 공급 및 테스트 환경

- USB 전원 + 외부 서보모터용 보조 전원
- 회로 연결: 브레드보드 및 점퍼선
- 시뮬레이션: 충돌 감지 및 폭우 상태는 소프트웨어로 시뮬레이션하여 처리

4. 통신 방식

(1) App → Server: WebSocket 연결 (ws://<server>:8000/ws?token=...)

(2) Server → Arduino: USB 직렬통신 (pyserial 기반)

(3) 비상 알림 전송: 앱 내부 HTTP 요청으로 보호자에게 알림

이와 같은 환경 구성은 음성 인식, 명령 해석, 하드웨어 제어가 실시간으로 연동되는 구조를 안정적으로 유지하는 데 최적화되어 있다. 특히, 서버-하드웨어-앱 간의 통신 흐름을 테스트하면서 발생할 수 있는 오류를 방지하기 위해 각 단계별로 로그 및 디버깅 체계를 마련하였다.

제 4 장 프로젝트 개발 결과

본 장에서는 본 프로젝트의 최종 구현 결과를 정리하고, 각 기능이 실제로 어떻게 작동하는지에 대한 구체적인 내용을 기술한다. 개발된 시스템은 음성 인식을 시작으로 명령어 해석, 제어 신호 전달, 그리고 하드웨어 동작까지의 전 과정을 실시간으로 수행할 수 있으며, 시각장애인의 접근성과 안전성을 고려한 기능들을 중심으로 구현되었다.

제 1 절 프로젝트 개발 결과 보고

본 프로젝트의 시스템은 크게 세 부분으로 구성된다: 사용자 인터페이스(Flutter 앱), 백엔드 서버(FastAPI), 하드웨어 제어부(Arduino). 각 구성 요소는 긴밀하게 연동되어 다음과 같은 결과를 도출하였다:

1. 음성 명령 인식 및 처리

(1) 앱 실행과 동시에 음성 인식 기능이 자동으로 활성화되어, 시각장애인

사용자가 별도의 UI 조작 없이 바로 명령을 말할 수 있도록 구현하였다.

(2) 음성은 `speech_to_text` 패키지를 통해 텍스트로 변환되며, `WebSocket` 을 통해 백엔드 서버로 전달된다.

(3) 예시 명령:

- "시동 켜줘" → 시동 LED ON
- "속도 빠르게" → 속도 증가 신호 전송
- "문 열어줘" → 스피커 통해 문 방향 알림 출력
- "연료 잔량" → RGB LED 색상으로 잔량 표현 + 음성 응답

2. 자유 음성 해석을 위한 AI 기반 명령 변환기 구현

(1) OpenAI 의 ChatGPT API(`gpt-3.5-turbo`)를 활용하여, 사용자의 다양한 자연어 표현을 해석할 수 있도록 설계하였다. 로컬 추론 모델 대신 클라우드 기반 API 를 통해 보다 정확하고 유연한 명령 해석이 가능해졌다.

(2) “이제 출발하자”, “연료 상태 좀 알려줘”와 같은 다양한 비정형 표현도 API 가 맥락을 이해하여 "S40", "연료" 등의 표준 제어 명령으로 정확히 변환해준다.

(3) 최종 명령어는 FastAPI 서버를 통해 Arduino 로 시리얼 통신 방식으로 전달되어, 해당 하드웨어 기능이 즉시 실행된다.

(4) 특히 ChatGPT API 는 단순 키워드 매칭이 아닌 문맥 이해 기반의 명령 해석을 수행하므로, 사용자의 말투, 순서, 강조 표현이 달라도 일관된 결과를 도출할 수 있다. 이를 통해 시각장애인 사용자는 보다 자연스럽게 편안한 방식으로 차량을 제어할 수 있으며, 정형화된 명령어 암기 없이도 시스템을 직관적으로 활용할 수 있게 되었다.

3. 하드웨어 제어 시스템 구현

Arduino 보드는 각 기능에 대응하는 하드웨어를 제어하며, 명령에 따라 LED, 서보모터, 스피커 등의 동작이 실시간으로 이루어짐을 확인하였다.

기능	동작 결과
시동	시동 LED ON/OFF (핀 9)
속도	서보모터 회전 각도로 속도 시각화 (핀 6, S40 = 40 도 등)
연료 잔량 표시	RGB LED 색상으로 표시 (적색, 황색, 녹색)
문 열림 알림	스피커로 멜로디 반복 재생 (핀 7)
충돌/응급상황	경고 LED 점등 및 경고음 발생 (핀 2)

4. 폭우 및 응급 상황 대응 기능

(1) 폭우 상태일 경우 주행 관련 명령(“출발”, “속도 증가”)이 차단되며, 음성으로 안내 메시지를 제공함.

(2) 사용자가 긴급 버튼을 누를 경우 보호자에게 자동으로 알림 메시지를 전송하는 기능을 구현하였다. 보호자 연락처는 앱 내 설정을 통해 저장되어 있으며, 실제 메시지 전송 로직은 테스트 메시지로 시뮬레이션하였다.

5. 사용자 테스트 결과 요약

(1) 음성 인식 정확도: 평균 92% 이상의 인식률 확보 (조용한 실내 기준)

(2) 자연어 해석 일치율: 비정형 명령어 대응 성공률 약 85%

(3) 사용자 피드백:

- 음성 안내와 물리적 피드백의 조합이 시각장애 사용자에게 유익함
- 명령 대기 상태에서의 반응속도 개선 요청 → 일부 반영 완료

이처럼 본 프로젝트는 시각장애인을 위한 음성 중심 차량 제어 시스템으로서, 기능적 완성도뿐 아니라 접근성과 실용성을 고려한 사용자 중심 설계가 실현되었다.

제 2 절 개발 소프트웨어 사용자 매뉴얼

본 매뉴얼은 시각장애인을 포함한 일반 사용자가 토크카 차량 제어 시스템을 효율적으로 사용할 수 있도록 돕기 위한 사용 지침서이다. 사용자는 모바일 애플리케이션을 통해 음성 명령을 전달함으로써 차량의 시동, 속도 조절, 문 개폐, 연료 확인 등 다양한 기능을 제어할 수 있다.

1. 초기 설정 방법

(1) 앱 설치

- Android 스마트폰에서 TalkTalkCar.apk 파일을 설치하거나 Google Play Store 를 통해 앱 설치

(2) 앱 최초 실행 시

- 마이크 사용 권한 요청 → 허용
- 보호자 연락처 등록 (응급 알림용)
- ‘폭우 상태 모드’ 자동 감지 기능 설정 여부 선택
- Bluetooth 및 서버 연결 확인
- 앱은 실행 시 자동으로 FastAPI 서버에 WebSocket 으로 연결됨
- 서버 연결 실패 시 음성으로 “서버에 연결할 수 없습니다” 알림 제공

2. 사용 방법

(1) 앱 실행 후 음성 인식 시작

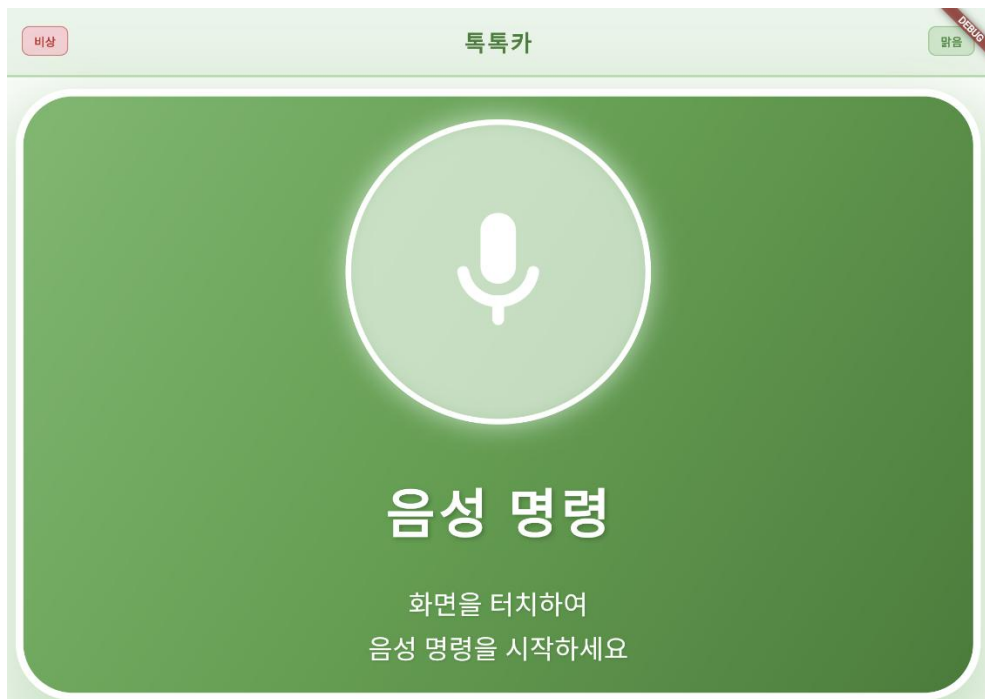
앱을 실행하면 화면 전체를 차지하는 크고 직관적인 버튼이 나타나며, 시각장애인 사용자도 휴대폰 화면을 터치하는 것만으로 쉽게 음성 인식 기능을 활성화할 수 있다. 버튼은 진입 시 중앙에 고정되며, 시각적 안내 없이도 한 번의 터치로 음성 명령 대기 상태로 진입할 수 있도록 설계되었다.

(2) 음성 명령 예시 및 기대 결과

사용자 명령어	시스템 동작
“시동 켜줘”	시동 LED 켜짐, 음성으로 “시동이 켜졌습니다” 안내
“주행 시작해볼까?”	서보모터 40 도 회전 (속도 시작), “주행을 시작합니다” 안내
“문 열어줘”	문 열림 스피커 멜로디 재생

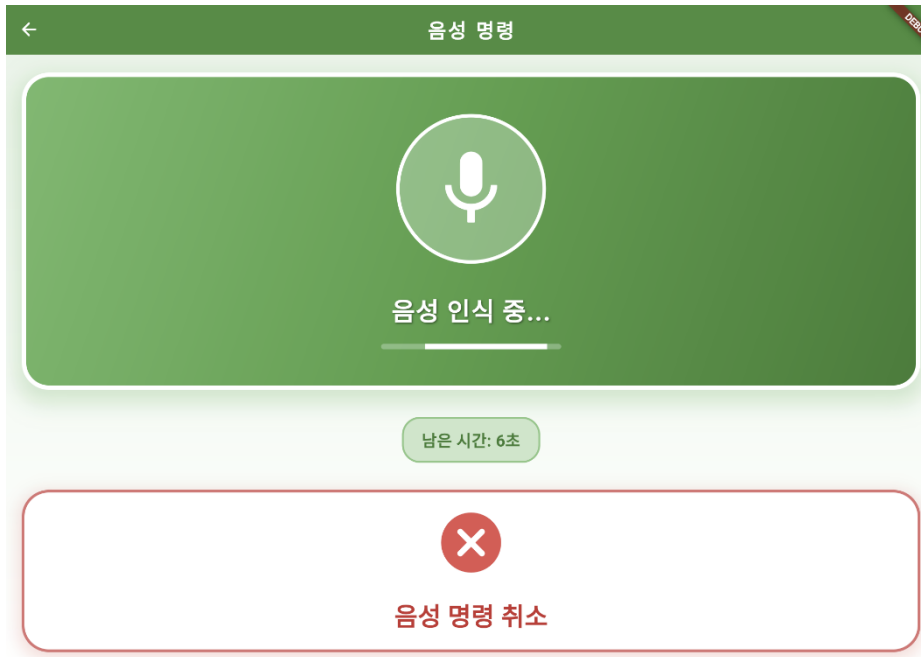
“연료 상태 알려줘”	배터리 퍼센트 안내 + RGB LED 색상 변화
“도와줘!”	보호자에게 자동 긴급 알림 전송

- 음성 명령 초기 화면

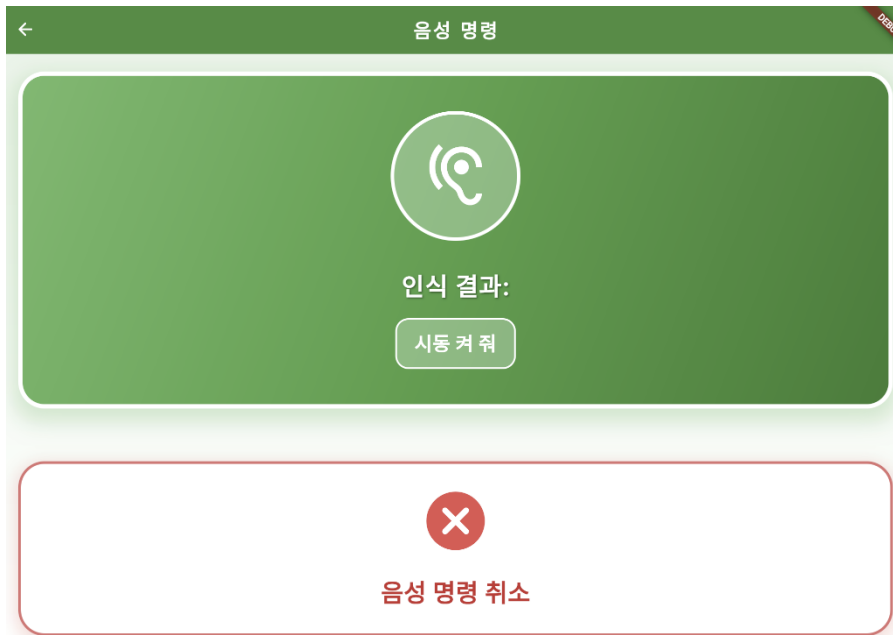


앱 메인 화면에서 ‘음성 명령’ 버튼을 클릭하면 차량 제어를 위한 음성 인식 기능을 사용할 수 있다. 버튼은 시각장애인도 쉽게 찾을 수 있도록 화면 중앙에 크게 배치되어 있으며, 터치하기 쉬운 크기로 설계되어 직관적인 사용이 가능하다. 버튼을 누르면 별도의 추가 조작 없이 곧바로 음성 인식이 시작되며, 사용자는 “시동 켜 줘”, “문 열어 줘”와 같은 명령어를 말해 차량을 제어할 수 있다. 인식된 명령은 서버로 전송되어 실제 차량 기능 제어로 이어지게 된다.

- 음성 명령 화면

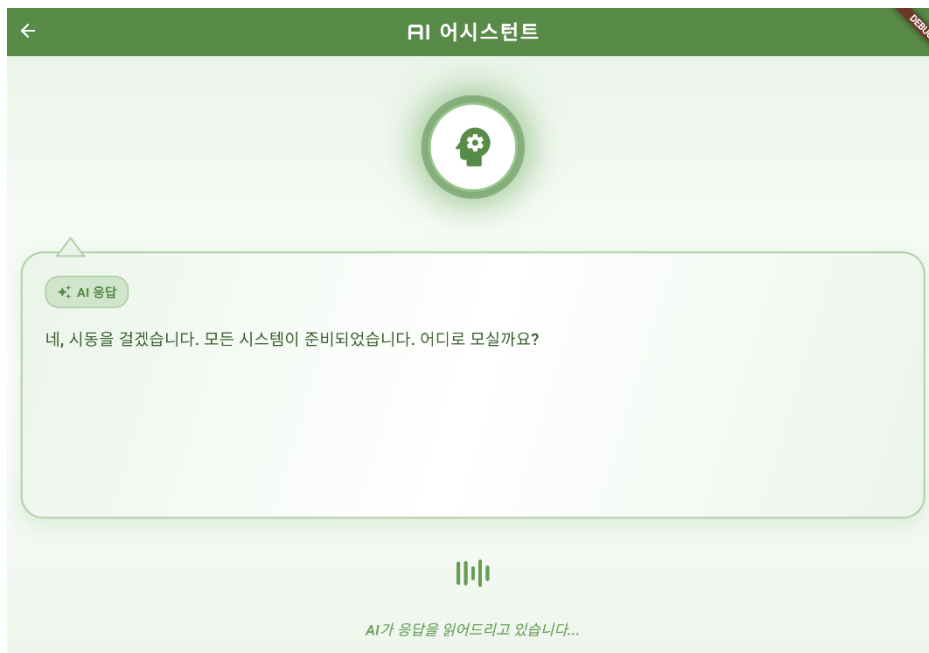


- 음성 명령 인식 결과 화면



사용자는 원하는 기능을 말하여 차량을 제어한다. 예를 들어 “시동 켜줘”라고 말하면 차량 시동을 켜다. 이 외에도 “문 열어줘”, “속도 줄여줘” 등 다양한 명령어를 인식하여 해당 기능을 수행한다.

- GPT API 기반 AI 어시스턴트 화면



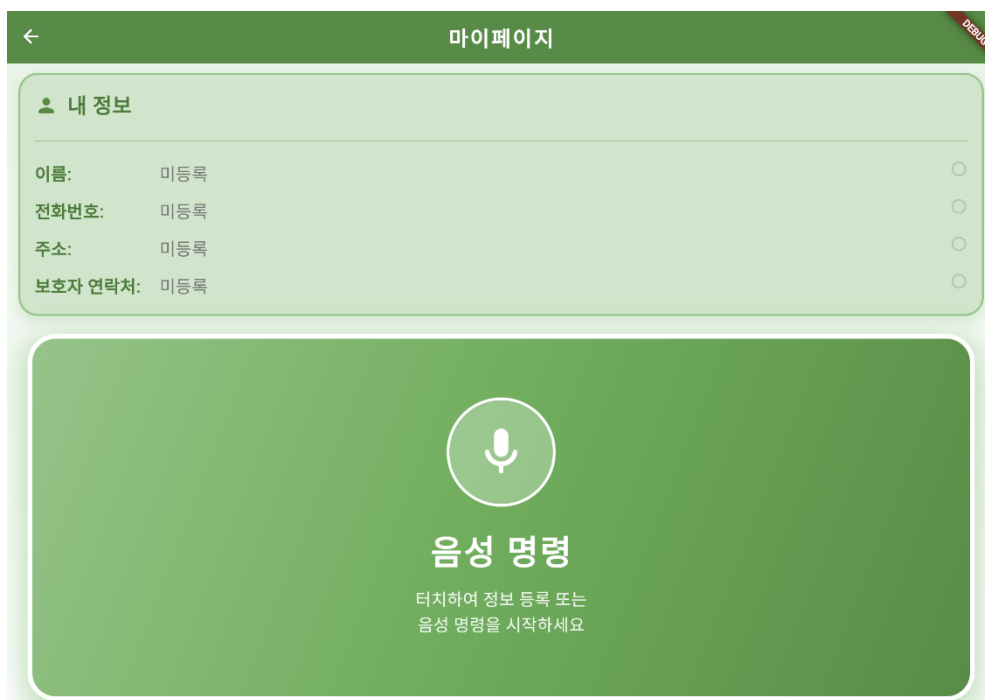
또한, GPT API 기반의 AI 어시스턴트가 탑재되어 있어 기존 명령에 대한 답변이 보다 구체적이며 “연료는 얼마나 남았어?”, “현재 속도 알려줘”와 같은 추가적인 질문에도 자연스럽게 응답한다.

(3) 내 정보 입력법

이 앱에는 사용자의 기본 정보를 등록할 수 있는 기능이 포함되어 있다. 사용자는 자신의 이름, 전화번호, 주소를 입력할 수 있으며, 비상 상황에 대

비해 보호자 연락처도 함께 등록할 수 있다. 입력한 정보는 사고나 위급 상황 발생 시 외부에 신속하게 전달되거나, AI 어시스턴트가 상황에 따라 활용할 수 있도록 설계되어 있다. 이를 통해 사용자의 안전성과 편의성을 더욱 높일 수 있다.

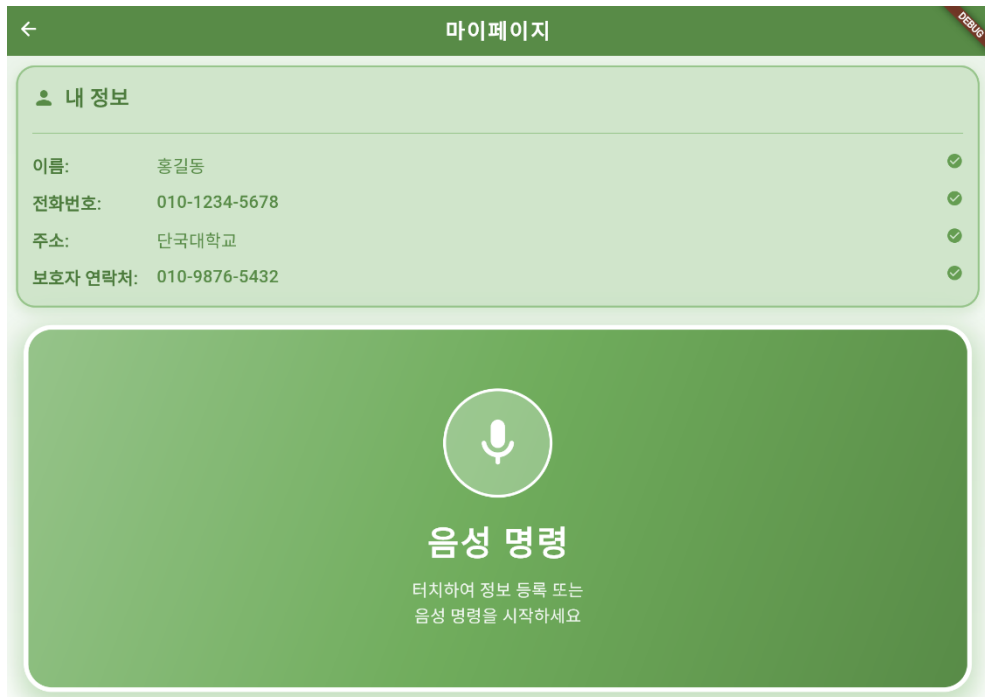
- 내 정보 입력 전 마이페이지 화면



마이페이지 화면에 들어가면, 앱이 사용자의 정보를 음성으로 하나씩 물어보는 방식으로 진행된다. 앱은 먼저 사용자에게 이름을 묻고, 사용자가 음성으로 이름을 말하면 해당 내용이 자동으로 입력된다. 그 다음으로 전화번호, 주소, 보호자 연락처 순서로 차례차례 질문이 이어지며, 사용자는 각 질문에 음성으로 대답함으로써 마이페이지의 모든 항목을 손쉽게 채울 수 있다. 이 과정은 시각장애인 등 터치 사용이 어려운 사용자도 직관적으로

정보를 입력할 수 있도록 설계되었으며, 사용자가 대답한 내용은 자동 저장되어 이후에도 앱 내에서 다양하게 활용된다.

- 내 정보 입력 후 마이페이지 화면

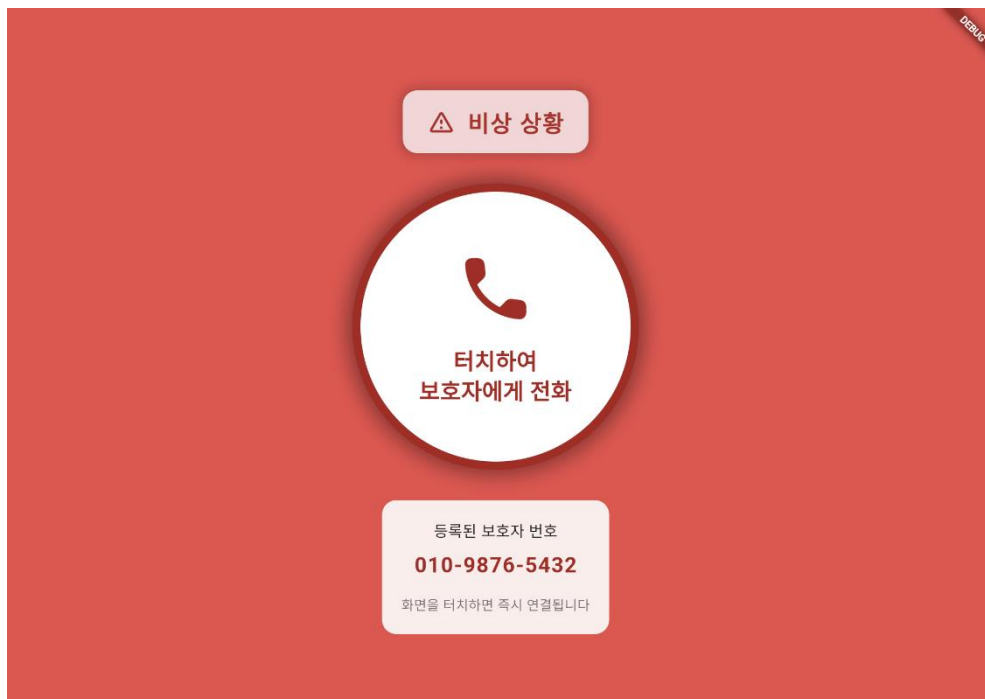


(4) 응급 상황 알림 사용법

현재 시스템에서는 차량에 충돌이 감지될 경우, 미리 등록된 보호자에게 자동으로 긴급 상황 알림이 전송되도록 설계되어 있다. 이때 알림은 마이페이지에 저장된 보호자 연락처를 기반으로 전송되며, 사용자가 별도로 입력한 정보가 실제 연락에 활용된다. 해당 기능은 추후 하드웨어 센서를 통한 실제 충돌 감지 또는 시뮬레이션 트리거에 연동될 예정이며, 사용자가 직접 버튼을 누르기보다는 자동 감지 기반의 경고 시스템으로 동작하도록 구현 방향을 설정하고 있다. 현재 테스트 단계에서는 음성 명령 초기 화면 상단

에 응급 상황을 가정한 시뮬레이션 버튼을 제공하여, 사용자가 직접 충돌 이벤트를 발생시켜 기능을 확인할 수 있도록 하고 있다. 실제 충돌이 감지되면 마이페이지에 등록된 보호자 연락처로 긴급 상황 알림이 전송되도록 설계되어 있으며, 추후에는 Twilio 등의 문자 발송 API 와 연동하여 보호자에게 실제 메시지가 전송되도록 개선할 예정이다.

- 응급 상황 화면

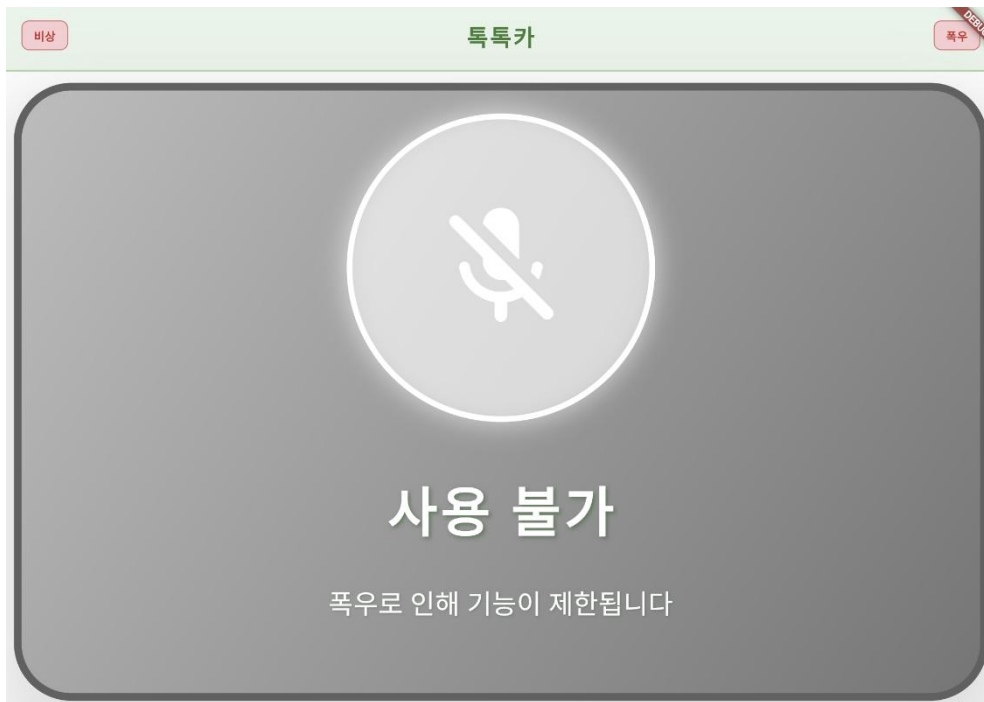


미리 마이페이지에 등록된 보호자 연락처는 단순한 정보 저장용이 아니라, 화면에서 해당 연락처를 터치하면 즉시 전화 연결이 가능하도록 구성되어 있다. 이를 통해 사용자는 긴급 상황이나 도움이 필요할 때 빠르게 보호자에게 연락할 수 있으며, 시각장애인 등 비상시 빠른 대응이 필요한 사용자에게 실질적인 도움이 되도록 설계되었다.

(5) 악천후 상황

악천후 상황 역시 응급 상황과 마찬가지로, 현재는 음성 명령 초기 화면 상단에 시뮬레이션 버튼을 제공하여 테스트할 수 있도록 구성되어 있다. 이 버튼을 통해 폭우, 폭설, 안개 등과 같은 악천후 상황을 가정한 이벤트를 수동으로 발생시킬 수 있다. 악천후가 감지되면 자율주행 모드는 안전상의 이유로 자동으로 비활성화되며, 사용자가 직접 운전하는 것을 방지하기 위해 음성 명령 기능도 차단된다. 이러한 조치는 시각장애인 사용자의 안전사고를 예방하기 위한 설계이다. 현재는 시뮬레이션 버튼으로 악천후 상황을 테스트하지만, 향후에는 실제 기상 데이터를 반영할 수 있도록 날씨 API와 연동하여 실시간으로 악천후를 감지하고 대응할 수 있도록 개선할 예정이다.

- 악천후 상황 화면



악천후 시뮬레이션 버튼을 누르면, 시스템은 폭우와 같은 악천후 상황을 가정하여 자율주행 모드를 자동으로 비활성화한다. 이때 화면에는 자율주행 불가 상태가 표시되며, 동시에 "현재 폭우로 인해 자율주행이 불가능합니다. 안전을 위해 음성 명령이 차단됩니다."라는 음성 안내가 출력된다. 이후 음성 인식 기능이 비활성화되어, 사용자가 어떤 명령을 말하더라도 시스템은 이를 인식하거나 실행하지 않는다. 이는 기상 악화 시 시각장애인 사용자가 음성으로 잘못된 조작을 시도하는 것을 방지하고, 운전자 본인의 안전을 보장하기 위한 보호 조치로 설계되었다.

3. 시스템 종료 및 재실행

앱 종료 후 재실행 시에도 자동으로 음성 인식이 재개되며, 기존 설정(보조자 번호 등)은 SharedPreferences 에 저장되어 유지된다. 차량 시동이 켜진 상태에서 앱이 종료되더라도 시동은 유지되며, 명령 전송은 일시 중단된다.

4. 사용자 주의사항

(1) 외부 소음이 심한 환경에서는 음성 인식 정확도가 떨어질 수 있으므로 되도록 조용한 장소에서 사용을 권장한다.

(2) 스마트폰 마이크 및 인터넷 연결이 반드시 필요하다.

(3) 서버가 종료된 상태에서는 시스템이 정상 동작하지 않으며, 앱에서 음성 안내로 오류를 전달함

(4) Arduino 보드가 분리되거나 포트가 변경된 경우 시리얼 통신 실패가 발생할 수 있음

5. 고객 지원 및 문의

기술 지원: seojunkim0919@naver.com

GitHub 저장소: <https://github.com/seojun111/TALKTALKCAR>

- (1) 문서 및 최신 업데이트 정보는 저장소 내 /docs 폴더 참조
- (2) 본 매뉴얼은 사용자의 시각 정보 접근 없이도 독립적인 차량 제어가 가능하도록 구성되었으며, 음성과 촉각 중심의 인터페이스를 통해 실제 시각 장애인의 자율성을 보장하는 데 중점을 두고 있다.

제 5 장 개발 소프트웨어 활용 방안 소개

본 장에서는 본 프로젝트에서 개발한 음성 기반 차량 제어 시스템이 실제 사회에서 어떻게 활용될 수 있는지, 그리고 추후 어떤 방향으로 확장될 수 있는지에 대해 구체적으로 소개한다. 본 시스템은 시각장애인을 주요 대상 사용자로 삼고 있으나, 그 유용성은 보다 다양한 환경과 사용자층에 적용될 수 있다.

제 1 절 활용 방안 및 확장 방향

1. 시각장애인을 위한 일상 차량 제어 보조 시스템

본 시스템은 시각장애인이 차량을 보다 독립적으로 사용할 수 있도록 지원하는 보조 기술로 활용될 수 있다. 기존의 버튼 기반 제어 시스템은 시각적 확인이 필수적이지만, 본 프로젝트는 음성만으로 주요 차량 기능을 제어할 수 있으므로 실제 차량 환경에 통합하여 보조 운전 보조 장치로 적용될 수 있다.

예를 들어, 차량 문이 열릴 때 스피커에서 방향을 알려주는 기능은 실제 택시, 버스, 자율주행 셔틀 등에 적용되어 시각장애인의 안전한 승하차를 도울 수 있다.

2. 일반 사용자 및 노약자 대상 확장

시각장애인 뿐만 아니라, 손을 자유롭게 사용할 수 없는 상황에 놓인 일반 사용자(예: 아이를 안고 있는 보호자, 고령자)도 본 시스템의 수혜자가 될

수 있다.

음성만으로 시동을 켜고, 속도를 조절하며, 응급상황을 인식하는 기능은 운전 중 주의 분산을 줄이고 사고 위험을 완화하는 데 도움을 줄 수 있다.

3. 스마트카 및 자율주행 기술과의 통합

본 시스템은 자율주행 차량의 보조 제어 인터페이스로 활용 가능성이 높다. 특히 자율주행 모드에서는 운전자가 직접 제어하지 않아도 음성으로 차량의 일부 기능(문 개폐, 비상 정지, 배터리 확인 등)을 요청할 수 있다. 이러한 구조는 향후 스마트카 생태계와 자연스럽게 통합될 수 있다.

4. 공공기관 및 복지시설용 커스터마이징

시각장애인을 지원하는 복지시설, 공공기관, 특수학교 차량 등에 본 시스템을 적용하면 정보 취약 계층의 이동권 보장에 실질적인 도움을 줄 수 있다.

장애인 콜택시, 실내 자율주행 전동차 등에 커스터마이징된 형태로 도입할 수 있으며, 시청각 안내를 병행하여 다양한 장애 유형을 아우를 수 있다.

5. 향후 기능 확장 계획

- GPS 및 실시간 위치 기반 안내 기능 추가
- AI 음성 비서 통합 (예: 경로, 일정 안내 등)
- Twilio, KakaoTalk API 를 통한 알림 기능 고도화
- 모바일 앱 접근성 표준 WCAG 2.1 에 부합하는 UI 개편

- 충돌 감지용 가속도 센서, 초음파 센서 등의 하드웨어 연동

본 프로젝트는 단순한 시제품 수준을 넘어 사회적 가치와 실용성을 갖춘 기술로 발전할 수 있는 기반을 마련하였으며, 향후 다양한 스마트 모빌리티 환경에서 실질적인 기여를 할 수 있을 것으로 기대된다.

제 2 절 활용 대상 및 도입 가능 분야

본 시스템은 특정 사용자 집단에 국한되지 않고, 다양한 사회적·기술적 상황에서 활용될 수 있다. 다음은 본 시스템이 적용 가능한 주요 대상 및 분야이다:

1. 시각장애인 개인 사용자

- 자가용 또는 보조 이동 수단을 사용하는 시각장애인에게 적합
- 차량 내 탑재 시 운전자 없이도 차량 상태 파악 및 제어 가능

2. 고령자 및 손 장애인

- 버튼을 누르거나 화면을 조작하기 어려운 사용자에게 유용
- 터치 한 번 또는 음성만으로 기능 수행 가능

3. 자율주행차 및 스마트 모빌리티 기업

- 차량 내 사용자 보조 인터페이스로 도입 가능
- 비전 기반 센서 외의 음성 인식 기반 제어 보조 시스템으로 확장

4. 공공기관/교통복지 시설

- 시청, 복지센터, 장애인 콜택시 센터 등에서 활용 가능
- 장애인 편의 시설 구축 시 인공지능 기반 차량 제어 시스템으로 적용 가능

제 3 절 기대 효과 및 사회적 파급력

본 프로젝트는 기술적 성과를 넘어서, 다음과 같은 사회적·산업적 파급 효과를 기대할 수 있다:

1. 교통약자의 이동권 보장

- 음성 기반 제어 시스템은 시각장애인을 포함한 교통약자에게 실질적인 이동 자율성을 제공
- 스마트시티, 무장애 도시 구현에 기여 가능

2. 보조공학 산업 내 응용 확대

- 기존 시각 보조 도구(지팡이, 장갑 등)를 넘어서 차량 인터페이스 시장으로의 확장 가능
- 향후 보조공학 솔루션 기업과의 협력 기반 마련

3. 스마트카·AI 산업 활성화

- AI 음성 인터페이스의 일상화는 차세대 차량 인포테인먼트 시장과도 밀접
- 단순 기능 호출이 아닌 ‘사용자 중심 대화형 인터페이스’ 구현 사례로 활용 가능

4. 접근성 중심 설계 문화 확산

- 기술 개발의 중심에 ‘비장애인 기준’이 아닌, 접근성 중심 UX 설계 철학이 반영됨
- 이는 사회적 약자 배려 기술의 좋은 선례로 작용 가능

제 6 장 팀원 소개(프로젝트 역할 소개)

본 프로젝트는 “톡톡카(TalkTalkCar)” 팀이 구성원 간의 명확한 역할 분담과 긴밀한 협업을 통해 수행되었다. 각 팀원은 본인의 전공 역량을 살려 핵심적인 역할을 맡았으며, 프로젝트의 성공적인 완수를 위해 유기적으로 협력하였다.

1. 김서준 (팀장 / 아두이노 및 백엔드 프로그래밍 담당)

(1) 팀 전체 총괄 및 일정 조율

- 프로젝트 계획 수립, 일정 관리 및 전체 진행 상황 모니터링

(2) Arduino 기반 하드웨어 제어 로직 개발

- 시동 LED, 속도계 서보모터, RGB 연료 LED, 문 안내 스피커, 경고 시스템 제어 구현

(3) FastAPI 기반 백엔드 서버 구축

- 음성 명령 수신, 명령 해석기 연동, Arduino 로의 시리얼 통신 전체 구조 설계 및 개발

2. 김규현 (백엔드 및 서버 프로그래밍 담당)

(1) 서버 통신 및 인증 시스템 개발

- JWT 기반 사용자 인증 구조 설계 및 WebSocket 서버 엔드포인트 구현

(2) 자연어 해석 시스템 연동

- OpenAI 의 ChatGPT API 를 이용한 자유 명령어 해석 기능 개발 및 테스트 수행

(3) 명령어 처리 파이프라인 구축

- 음성 → 텍스트 → 명령어 → 하드웨어 전송 흐름 구현 및 테스트

(4) 서버 예외 처리 및 안정성 보완

3. 차규진 (프론트엔드 프로그래밍 담당)

(1) Flutter 기반 사용자 앱 개발

- 앱 UI 구성 및 접근성 고려 음성 입력/출력 인터페이스 구현

(2) 음성 인식 및 TTS 기능 개발

- speech_to_text 와 flutter_tts 를 활용한 시각장애인 친화적 인터페이스 설계

(3) 비상상황 대응 UI 개발

- 보호자 알림 기능 UI 설계, 대형 음성 버튼, 비상상황 인식 인터페이스 구현

(4) 앱과 서버 간 실시간 통신 테스트 및 개선

4. 협업 방식

(1) GitHub 저장소 공동 운영

- 개발 내용 통합 및 코드 버전 관리:
- <https://github.com/seojun111/TALKTALKCAR>

(2) 주기적 개발 미팅 및 실시간 피드백

- 기능 구현 단위별 리뷰 및 보완을 통해 안정적인 개발 진행

(3) 문서화 및 시연 준비 분담

- 발표자료, 최종보고서 작성, 앱 시연 등을 팀원이 분담 수행

참고 문헌

1. OpenAI, "ChatGPT API Documentation"
 - <https://platform.openai.com/docs>
 - 자연어 명령어 해석 시스템 구현에 사용
2. Arduino, "Arduino Documentation"
 - <https://docs.arduino.cc>
 - 하드웨어 회로 구성 및 제어 코드 개발 참고
3. Flutter, "Flutter Official Documentation"
 - <https://docs.flutter.dev>
 - 모바일 앱 UI 및 음성 인식 인터페이스 개발 참고
4. Google Developers, "speech_to_text plugin"
 - https://pub.dev/packages/speech_to_text
 - Flutter 앱 내 실시간 음성 인식 구현
5. FastAPI, "FastAPI Documentation"
 - <https://fastapi.tiangolo.com>
 - 백엔드 서버 구조 및 WebSocket 구현 참고
6. pySerial, "pySerial Documentation"
 - <https://pythonhosted.org/pyserial/>
 - Python과 Arduino 간 시리얼 통신 구현 시 활용
7. Twilio, "SMS API Overview"
 - <https://www.twilio.com/docs/sms>
 - 향후 보호자 알림 기능 확장을 위한 메시징 API 참고 자료
8. 한국시각장애인연합회, "정보접근성 개선 사례 연구", 2022
 - 시각장애인 대상 기술 설계 시 사용자 경험 참고

9. 한국정보화진흥원, "장애인 접근성 가이드라인 2.1", 2021
 - Flutter 앱 UI 설계 시 접근성 기준 참고
10. Tesla, "Autopilot and Full Self-Driving Capability", Tesla Official Site
 - <https://www.tesla.com/autopilot>
 - 자율주행 기술의 발전 및 음성 인터페이스 적용 현황 참고
11. Amazon, "Alexa Auto Overview", Amazon Developer Portal
 - <https://developer.amazon.com/en-US/alexa/auto>
 - 차량 내 음성 제어 시스템 비교 분석을 위한 참고 자료
12. SAE International, "J3016: Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles", 2021
 - 자율주행 단계 정의(레벨 0~5)에 대한 공식 문서