



2025 장애인 분야 해커톤 대회  
「장애 플러스 기술」 개발제안서  
- 장애인을 위한 APP 개발 공모전 -

팀 명	SignGlove				
APP명	SignTalk (사인톡)				
분야 [대주제]	■ 분야 1 (디지털 포용)		□ 분야 2 (자립생활 지원)		
중주제 [택1]	■ ① 디지털 기기/기술 활용 보조	□ ② 디지털 콘텐츠 제작 지원	□ ① 일상생활 돌봄 지원	□ ② 재활 및 치료 지원	
	□ ③ 정보 접근성 향상	□ ④ 기타 자유주제	□ ③ 교육 및 훈련	□ ④ 기타 자유주제	
사용가능 장애유형 [중복선택 가능]	□ 지체장애	□ 뇌병변장애	□ 시각장애	■ 청각장애	■ 언어장애
	□ 지적장애	□ 자폐성 장애	□ 정신장애	□ 신장장애	□ 심장장애
	□ 호흡기장애	□ 간장애	□ 안면장애	□ 장루요루 장애	□ 뇌전증 장애
활용기술	■ AI(인공지능) □ 빅데이터 □ 가상현실(VR) □ 증강현실(AR) □ 음성 또는 동작인식 □ 기타(        )				
1. 기획의도 및 필요성					

## 청각장애인 소통 현황과 기술적 필요성

- 대상 인구: 국내 청각장애인 약 25.5만명(언어장애 포함 27.2만명), 65세 이상 고령층 비중 73.2%로 상승
- 소통 인프라 부족: 전국 수어통역센터 192개소, 공인 통역사 1,134명으로 수요 대비 절대 부족
- 실생활 소통 공백: 의료·공항·관공서 등에서 수어통역 부재로 인한 의사소통 장애 빈발 (한겨레 2014.11 보도)
- 디지털 격차: 고령 청각장애인의 스마트폰/앱 접근성 제약으로 기존 텍스트 기반 솔루션 한계

## 기존 의사소통 도구의 구조적 한계

- 비장애인 중심의 준비 부담: 현재 청각장애인용 의사소통 도구들은 대부분 비장애인의 적극적인 참여나 준비를 필요로 합니다[1]:
  - 수어 통역 서비스: 비장애인이 통역사를 호출하고 비용을 부담해야 함
  - 수어 학습 프로그램: 비장애인이 수어를 배우는 시간과 노력 투자 필요
  - 음성-문자 변환 앱: 비장애인이 앱을 다운로드하고 사용법을 익혀야 함
  - 전문 통신 장비: 비장애인 측에서 장비 설치 및 운영 준비 필요
- 소통 주도권의 불균형: 기존 도구들은 청각장애인이 소통을 주도적으로 시작하기 어려운 구조로, 비장애인의 준비 상태에 따라 의사소통 가능 여부가 결정되는 한계가 있습니다.

## 우리 팀의 차별화된 접근 방식

- 장애 당사자 중심의 자립적 소통 도구: SignGlove는 청각장애인과 언어장애인이 비장애인의 추가적인 준비나 참여 없이도 주도적으로 의사소통을 시작할 수 있는 전용 솔루션입니다.

## 개발 목표

- 한국어 수어(KSL) 특화 웨어러블 디바이스와 AI 서버 시스템으로 실시간 수어 인식 및 음성/텍스트 변환을 통한 양방향 소통 지원:
- 플렉스 센서 5개 + LSM6DS3 IMU 센서 융합으로 손동작 정밀 측정
  - FastAPI 기반 서버로 실시간 데이터 수집·전처리·추론·TTS 파이프라인 구축
  - KSL 34개 클래스(자음 14개, 모음 10개, 숫자 10개) 체계적 분류 및 학습
  - 비장애인의 사전 준비 없이도 즉시 사용 가능한 독립적 소통 도구
  - 청각장애인이 주도하는 능동적 의사소통 환경 구축

## 2. 국내·외 유사 개발물 사례 비교 및 차이점

### 국내외 수어 번역 기술 한계점

#### 해외 제품들의 제약사항:

- SignAll(헝가리): ASL 기반, 인식률 70% 수준, 복합 문장 번역 한계
- Signily(미국): 컴퓨터 비전 기반으로 조명/배경 환경 의존성 높음 (인식률 40-80% 편차)
- Google MediaPipe: 손동작 추적에 특화되나 실제 수어 번역 기능 부족

#### 서비스의 한계:

- 국립국어원 한국수어사전: 일방향 검색만 가능, 실시간 번역 부재
- 손말이음 앱: 단어 수준 번역에 머물러 문장 단위 소통 어려움

### SignTalk의 기술적 우위

- 환경 독립적 인식: 플렉스 센서 + IMU 센서 융합으로 조명/배경 영향 최소화[2]
- KSL 특화 설계: 한국어 수어 34개 클래스 체계적 정의 및 문법 구조 반영[3]
- 센서 퓨전 알고리즘: Madgwick Filter 적용으로 방향 모호함 해결 및 드리프트 보정
- 실시간 처리 파이프라인: 수집→전처리→추론→TTS까지 104Hz 고속 처리

## 3. 서비스 기능 및 활용기술 설명

## 하드웨어 구성

### 센서 시스템:

- **Arduino Nano 33 IoT**: WiFi 내장 메인 컨트롤러
- **LSM6DS3 IMU 센서**: 6축 가속도계/자이로스코프, 104Hz 고정 출력
- **플렉스 센서 5개**: 손가락별 굽힘 각도 측정 (A0-A6 핀 할당)
- **통신 방식**: WiFi TCP 1차, UART 115200bps 백업

### 데이터 스펙:

LSM6DS3 센서 사양 (TECHNICAL\_CHALLENGES.md 기준)

- Accelerometer:  $\pm 4g$  범위,  $\pm 0.122mg$  해상도
- Gyroscope:  $\pm 2000dps$  범위,  $\pm 70mdps$  해상도
- 샘플링: 104Hz 고정, 아두이노 20-50ms 간격 목표

## 서버소프트웨어 아키텍처

### FastAPI 기반 실시간 데이터 파이프라인:

#### 데이터 수집 API (server/main.py):

- POST /collect\_data: 센서 데이터 실시간 수집 및 검증
- GET /health: 서버 상태 점검
- GET /statistics: 수집된 데이터 통계 분석

#### 전처리 엔진 (server/preprocessing.py):

- Butterworth 저역통과 필터: 센서 노이즈 제거
- 정규화 및 평활화: 개인별 센서 범위 보정
- 슬라이딩 윈도우 기반 시계열 처리

#### KSL 클래스 관리 (server/ksl\_classes.py):

- 34개 한국어 수어 클래스 정의 및 검증
- 자음 14개, 모음 10개, 숫자 10개 체계적 분류
- 카테고리별 통계 및 검증 상태 API

#### 추론 엔진 (server/inference\_engine.py):

- 실시간 파이프라인 + 5-window 안정성 체크
- 현재: 테스트용 Mock 모델 (실제 모델 연동 준비 완료)

#### TTS 엔진 (server/tts\_engine.py):

- 한국어 음성 합성 (pyttsx3/GTTS)
- KSL 기호→발화 텍스트 매핑 후 음성 변환

### 배포/운영 시스템:

- Ubuntu 자동 배포 스크립트
- Docker 컨테이너화
- 헬스체크/로그/백업 시스템
- 교수님 서버 프록시 (환경변수 제어)

## 4. 상용화 및 구체화 전략

## CNN+LSTM 하이브리드 모델

**모델 구조** (external/KLP-SignGlove/models/deep\_learning.py 기준):

```
# 입력: (batch_size, sequence_length=20, features=8) # flex1-5, pitch, roll, yaw
# Conv1D layers: 64→128→256 채널, 특징 추출
# LSTM: hidden_dim=128, 시계열 패턴 학습
# Attention: 중요 시점 가중치 부여
# Output: num_classes=5 (현재 ㄱ, ㄴ, ㄷ, ㄹ, ㅁ)
```

**모델 특징:**

- **1D CNN 레이어:** 센서 데이터의 공간적 특징 추출, BatchNorm으로 안정성 확보
- **LSTM:** 시계열 패턴 학습, dropout으로 과적합 방지
- **어텐션 메커니즘:** 중요한 시점에 가중치 부여로 인식 정확도 향상
- **이중 출력:** 분류 결과 + 특징 벡터로 확장성 확보

## 센서퓨전 알고리즘

**방향 모호함 문제** (SENSOR\_FUSION\_GUIDE.md, TECHNICAL\_CHALLENGES.md):

현재 이슈: X: -0.031g, Y: -0.762g, Z: 0.588g → "방향 모호함" □

해결방안: Madgwick Filter로 Roll/Pitch/Yaw 각도 + 신뢰도 기반 판단

## Madgwick Filter 구현 계획:

```
void updateSensorFusion() {
    IMU.readAcceleration(ax, ay, az);
    IMU.readGyroscope(gx, gy, gz);
```

```
    filter.updateIMU(gx, gy, gz, ax, ay, az); // 104Hz
```

```
    roll = filter.getRoll();
    pitch = filter.getPitch();
    yaw = filter.getYaw();
```

```
    direction = getOrientationWithConfidence(roll, pitch, yaw);
```

## 신뢰도 기반 방향 판단:

Roll/Pitch/Yaw 각도별 임계값 설정 (60도, 45도)

신뢰도 점수 (0-1) 계산으로 모호함 해결

드리프트 보정 및 누적 오차 제거

## 5. 리스크 및 해결방안

## 기술적 리스크

### 하드웨어 제작 문제:

- 이슈: 플렉스 센서 납땜 연결 불안정 (TECHNICAL\_CHALLENGES.md)
- 영향도: 중간 (8월 일정 지연 가능)
- 대응: flex→pin header→resistor→cable 연결구조, 브레드보드 프로토타입 병행

### 데이터 모호함:

- 이슈: IMU 센서 방향 판단 실패 시점 발생
- 영향도: 높음 (인식 정확도 직접 영향)
- 대응: Madgwick Filter 센서퓨전으로 신뢰도 기반 판단, 드리프트 보정

### 성능 불확실성:

- 이슈: 실제 KSL 데이터 부족으로 모델 성능 예측 어려움
- 영향도: 중간 (9월 마일스톤 영향)
- 대응: ASL→KSL 전이학습, 데이터 증강 기법 적용

## 개발 일정 리스크

### 하드웨어 개발 지연:

- 위험 시나리오: 납땜 문제로 8월 하드웨어 목표 미달성
- 완화 전략: 소프트웨어 개발 우선 진행, 외부 하드웨어 전문업체 협력 검토

### 데이터 수집 지연:

위험 시나리오: 충분한 품질의 KSL 데이터 확보 실패

완화 전략: 다양한 연령대/지역 사용자 모집, 외부 수어 전문가 협력 확대

## 사용자 수용성 리스크

### 고령층 기술 적응:

- 이슈: 65세 이상 73.2% 사용자의 웨어러블 기기 적응 어려움
- 대응: 단순한 인터페이스 설계, 착용 편의성 최우선, 사용법 교육 프로그램

### 경제적 접근성:

- 이슈: 하드웨어 제작 비용으로 인한 높은 기기 가격
- 대응: 정부 보조기구 등재 추진, 렌탈/리스 서비스 모델 검토

## 6. 개발 일정

## 8월 핵심 목표

### 주차별 세부 계획:

#### 1. 1주차 (8/1-8/7): 센서 퓨전 알고리즘 구현

- Madgwick Filter 라이브러리 통합
- 신뢰도 기반 방향 판단 로직 개발
- 실시간 104Hz 처리 최적화

#### 2주차 (8/8-8/14): 하드웨어 프로토타입 완성

- 플렉스 센서 납땜 문제 해결 (pin header 방식)
- 글러브 형태 프로토타입 제작
- WiFi/UART 통신 안정성 테스트

#### 3주차 (8/15-8/21): 기본 데이터셋 수집

ㄱ, ㄴ, ㄷ, ㄹ, ㅁ 클래스별 100개+ 샘플 수집

통합 수집기 활용 체계적 데이터 관리

데이터 품질 검증 및 라벨링

#### 4주차 (8/22-8/31): Alpha 버전 통합 테스트

전체 시스템 연동 테스트

성능 벤치마킹 및 최적화

사용자 피드백 수집 준비

## 9월 주요 마일스톤

딥러닝 모델 학습: CNN+LSTM 실제 KSL 데이터로 훈련 시작

실시간 추론 엔진: Mock 모델에서 실제 AI 모델로 전환

정확도 측정 및 개선점 도출

실제 청각장애인 사용자 대상 파일럿 테스트

## 10-11월 최종 검증

성능 최적화: 레이턴시 최소화, 인식률 90% 이상 목표

시스템 안정성: 24시간 연속 운영 테스트, 오류 처리 강화

배포 자동화: CI/CD 파이프라인 완성, 문서화 완료

사용성 검증: 복지관 협력 실사용 테스트, 접근성 개선

## 7. 장애인 특성 및 욕구를 반영한 개발 계획

## 7. 장애인 특성 및 욕구를 반영한 개발 계획

### 당사자 참여형 개발 프로세스

사용자 중심 설계:

- 개발 초기부터 청각장애인 당사자를 자문위원으로 참여
- 매주 정기 사용자 인터뷰를 통한 요구사항 수집
- 실제 수어 사용 환경에서의 테스트 및 피드백 반영

다양성 고려:

- 지역별 수어 방언 및 개인적 표현 습관 반영
- 연령대별 수어 사용 패턴 분석 (특히 65세 이상 고령층)
- 표준 수어뿐만 아니라 일상 자연스러운 표현 포함

### 접근성 우선 설계

감각적 특성 고려:

- 시각적 피드백: 명확한 텍스트 표시 및 상태 인디케이터
- 촉각적 피드백: 진동 알림으로 시스템 상태 전달
- 직관적 조작: 최소한의 버튼으로 핵심 기능 접근

사용 편의성:

- 한 손 사용 제약 고려한 글러브 설계
- 간단한 착용 방식으로 자립적 사용 가능

-

- 이러한 개발 계획을 통해 청각장애인의 실제 생활 패턴과 욕구를 충분히 반영하여, 단순한 기술 도구가 아닌 일상의 동반자 역할을 하는 솔루션을 완성하고자 합니다.

※ 8페이지 이내로 작성

※ 모든 기재 내용에 허위사실이 없어야 하며, 허위사실로 인한 불이익은 참가신청자의 책임입니다.

※ 반드시 pdf 파일로 제출요망

# 참가자 이력사항

<b>팀원 1</b> <b>이민우(팀장)</b>	팀원 1 - 이민우(팀장) (학력) 2022. ~ 2025. 강남대학교 인공지능전공 (기술) Back-end : Python, FastAPI Cloud : Google Cloud, Docker Data : Pandas, 실시간 데이터 처리 (프로젝트) SignGlove 실시간 센서 데이터 수집 시스템 구축 FastAPI 기반 서버 개발 및 데이터 파이프라인 구축 Mobile VLA 관련 프로젝트 진행 및 작성 중
<b>팀원 2</b> <b>양동건</b>	팀원 2 - 양동건(하드웨어 개발) (학력) 2020. ~ 2025. 강남대학교 소프트웨어전공 (기술) <ul style="list-style-type: none"><li>- Embedded : Arduino, C/C++, WiFi/UART 통신</li><li>- Hardware : LSM6DS3 센서, 플렉스 센서</li><li>- Signal Processing : 실시간 신호 처리</li></ul> <b>(프로젝트)</b>
<b>팀원 3</b> <b>하유빈</b>	팀원 3 - YUBEEN(ML 학습 모델) (학력) 2020. ~ 2025. 강남대학교 소프트웨어전공 (기술) <ul style="list-style-type: none"><li>- Machine Learning : TensorFlow, PyTorch</li><li>- Deep Learning : CNN, LSTM, Transformer</li><li>- Data Processing : Python, 시계열 데이터 분석</li></ul> <b>(프로젝트)</b>
<b>팀원 4</b> <b>정재연</b>	팀원 4 - 정재연(데이터 분석) (학력) 2024. ~ 2025. 강남대학교 인공지능전공 (기술) <ul style="list-style-type: none"><li>- Data Science : Python, 통계 분석</li><li>- ML/AI : 머신러닝 모델링, 데이터 전처리</li><li>- Analysis : 센서 데이터 분석, 성능 평가</li></ul> <b>(프로젝트)</b>
<b>팀원 5</b>	