

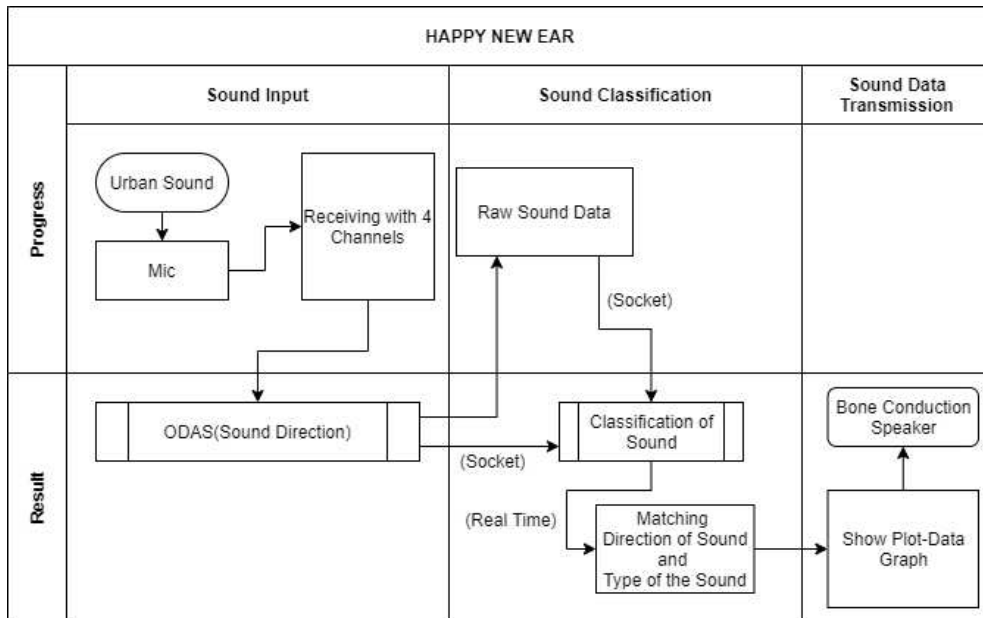
제19회 임베디드SW경진대회 개발완료보고서

[자유공모]

□ 개발 요약

팀 명	HAPPY NEW EAR
	
작품명	Happy New Ear
작품설명 (요약)	청각 장애인을 위한 소리 위험 감지 임베디드 디바이스
소스코드	https://github.com/HyunJemml/2021ESWContest_free_1155
시연동영상	https://www.youtube.com/watch?v=RFun6zqWqaE

□ 개발 개요



○ 개발 작품 개요

- 이 작품은 청각 장애인이 일상생활 속에서 겪을 수 있는 위험을 정확하고 빠르게 알아낼 수 있도록 한다. 마이크에서 소리를 인식하면 프로그램에서 어떤 종류의 소리인지 분류하고, 그 소리가 들리는 방향을 그래프 위에 점으로 나타낸다. 이를 쉽게 확인하기 위해 8가지 소리를 위험군 소리로 선별했다. 선별된 소리는 위험 정도에 따라 세 단계로 분류되고 각 단계에 해당하는 진동 횟수를 내보내어 골전도 스피커로 위험 단계 정보를 전달한다.

본 시스템은 크게 3가지 영역으로 나누어져 있다.

1. 소리 수신 영역: 마이크를 사용해 소리를 받아들여 ODAS(Open embeddeD Audition System)로 보내고 디스플레이에 소리 방향을 나타낸다.
2. 소리 분석 영역: ODAS에서 socket 통신을 통해 받아온 소리 데이터를 tensorflow lite를 이용해 어떤 종류의 소리인지 분석한다. 그 후, 소리 방향 데이터와 소리 분류 결과 데이터를 각각 4개의 채널에 할당해 짝을 짓는다.
3. 소리 데이터 전달: 소리 분석 영역에서 실시간으로 받은 결과를 그래프 위에 나타낸다. 알고리즘을 통해 특정 신호를 받으면 골전도 스피커 및 화면상의 경고 이미지를 통해 사용자에게 위험을 알려준다.

○ 개발 목표

■ 소리 수신 영역

- 4-mic-array를 통한 소리 데이터 수신
- 4-mic-array와 USB 포트를 연결하여 라즈베리 파이로 소리 정보를 전달한다. 전달된 소리 정보는 ODAS를 통해 4개의 채널로 할당된다.
- 4개의 채널로 할당된 소리는 그래프 위의 점으로 나타나 소리의 방향을 가시적으로 확인할 수 있다. 이는 4개의 마이크를 개별적으로 연결하여 사용하는 것에 비해 정확한 방향 확인이 가능하다.

■ 소리 분석 영역

■ 오픈 API 사용

- 머신러닝 API인 TensorFlow는 딥러닝 된 모델을 이용해 결과를 받아온다. 그중 발생 빈도가 높은 소리를 선별해 정보를 특정 구간으로 좁힘으로써 소리 데이터 정보를 보다 정확하게 도출할 수 있게 되었다.

■ 소리 데이터 전달

■ 화면과 골전도 스피커를 이용한 전달

- 특정 과정을 거쳐 분석된 소리 정보와 방향을 라즈베리 파이 LCD 상의 그래프로 확인할 수 있다. 화면 외의 보조 수단으로 골전도 스피커를 사용해 소리 정보의 위험 정도를 3단계로 나누어 진동 횟수를 달리해 전달했다. 이렇게 진동을 줌으로써 사용자가 위험 정도를 알 수 있음과 동시에 디스플레이를 확인하도록 유도했다.

○ 개발 작품의 필요성

- 청각 장애인은 청력 손실과 상실로 인해 의사소통의 장애를 가진 자로 모든 판단을 거의 시각에 의존한다. 이 때문에 비장애인보다 수용할 수 있는 정보량이 제한적이고, 정보 접근성이 비교적 낮다. 따라서 재난이나 위험 상황 발생 시 비장애인에 비해 시간상 현저히 늦게 정보를 획득하거나 아예 얻지 못하는 경우가 존재한다. 이와 같은 상황에 놓여 있는 청각 장애인은 일상생활에서 불편함을 느끼고 있으며 이를 넘어 안전상의 문제를 겪고 있는 것이 사실이다.

실제로 환경미화원으로 일하던 청각 장애인이 뒤에서 돌진한 차량에 부딪혀 사망한 사건이 발생했다. 다른 사례로 청각장애인 60대 부부는 가스 온수기에 연결된 호스에서 가스가 누출되었지만 이 사실을 몰라 대처하지 못해 액화석유가스가 폭발해 화상을 입은 사고가 발생했었다. 이는 우리나라에만 국한된 문제가 아니다. 해외 사례를 살펴보면 9.11 테러와 카트리나 허리케인이 발생한 당시에 청각 장애인은 정보 접근성이 낮고 재난에 대비하는 것에 있어 취약한 집단임이 증명되었다.

이처럼 많은 사고가 있었기 때문에 청각 장애인의 불편을 줄여줄 수 있는 장비의 필요성이 대두되었다. 이러한 상황을 개선하기 위해 청각 장애인에게 청각적 요소로 이루어진 정보를 시각적, 혹은 촉각적 요소로 변환해 주변 상황을 인지할 수 있게 하여 일상생활에서 발생하는 위험을 줄여줄 수 있는 장비의 개발이 필수적이다.

□ 개발 환경 설명

○ Hardware 구성

Raspberry Pi 4
Raspberry Pi Touch Display (800*480)
4-mic-array

보조배터리
3D 출력물
골전도 스피커 (HUMBIRD)

○ Hardware 기능 (제어 방법 등 서술)

■ 4-mic-array

사용자를 기준으로 사방에서 발생하는 소리를 인식하고 사용자의 불편을 최소화하면서 사람의 귀와 가장 근접한 위치에서 소리를 인식할 수 있도록 mic-array를 어깨에 부착했다. 사용자의 위치를 원점으로 두고, x축과 y축을 이용해 주변 공간을 4분할하여 전후좌우를 살필 수 있도록 했다. 사람마다 몸통 둘레와 키가 다른 것을 고려해 mic-array를 부착할 수 있는 버튼의 위치를 5cm 간격으로 박음질하여 4-mic-array가 개개인마다 적합하게 위치할 수 있도록 구성했다. 마이크가 4배열로 내장되어 있기 때문에 4개의 채널로 소리의 근원지를 구분할 수 있다. 또한 자연에서 발생한 현상을 마이크로 인식해서 MCU에 표현하기 위해 아날로그-디지털 컨버터(ADC)로 데이터를 받아들이고, micro-5pin을 통해 라즈베리 파이와 통신할 수 있도록 했다.

■ 라즈베리 파이 터치 디스플레이(Raspberry Pi Touch Display)

머신러닝 된 모델에 데이터를 입력하면 결과가 출력된다. 그 결과를 카테고리에 mapping하여 디스플레이에 표현했다. 5인치 디스플레이를 통해 47가지의 소리 종류와 소리 근원지 방향을 시각정보로 나타내 실시간 정보를 바로 확인할 수 있도록 했다. Raspberry Pi와 디스플레이를 핀으로 연결해 데이터 통신을 하고, 한번에 휴대할 수 있게 만들었다. 디스플레이를 직접 손으로 들어 확인할 때 Raspberry Pi 뒷면 납땜 부분이 사용자에게 불편을 끼칠 수 있다고 생각해 3D 모델링으로 규격에 맞는 맞춤 케이스를 제작했다. 또한 휴대성을 위해 10000mAh 보조배터리를 전원으로 하였고, 슬링백 안에 라즈베리 파이, 디스플레이, 배터리를 모두 내장할 수 있도록 설계했다.

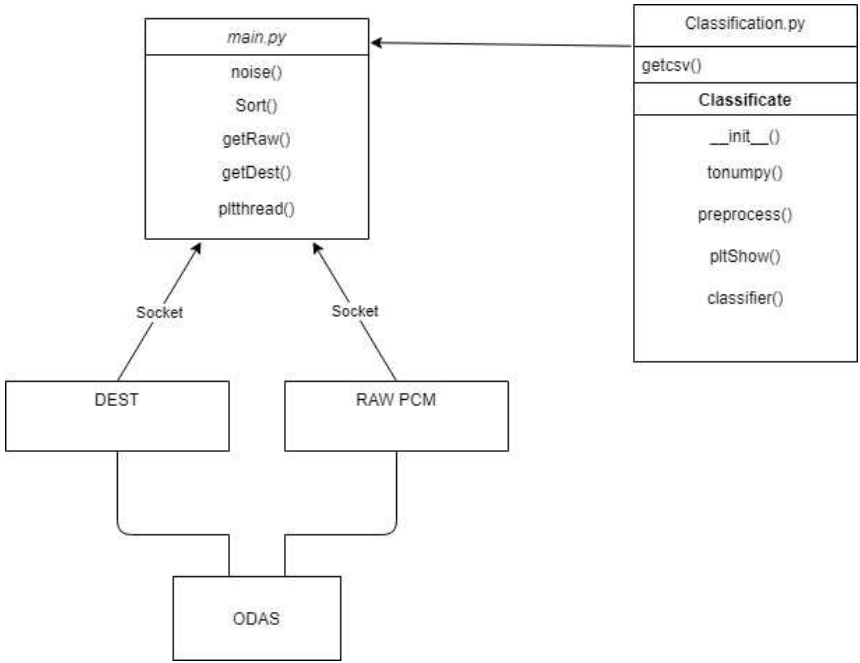
■ 골전도 스피커

라즈베리 파이에서 골전도 스피커로 블루투스 통신을 통해 데이터를 전송한다. 라즈베리 파이와 연결된 디스플레이를 항상 들고 다니기에 불편할 수 있기 때문에 부가적으로 골전도 스피커를 통해 위험을 인식할 수 있도록 했다. 두개골을 진동시켜 내이가 곧바로 소리를 느끼게 해주는 골전도 방식을 채택하여 청각 장애인들에게 도움이 되도록 했다. 또한 사용자가 원하는 대로 주머니 등에 골전도 스피커를 자유롭게 위치시켜 촉각으로도 위험 상황을 인지할 수도 있게 했다.

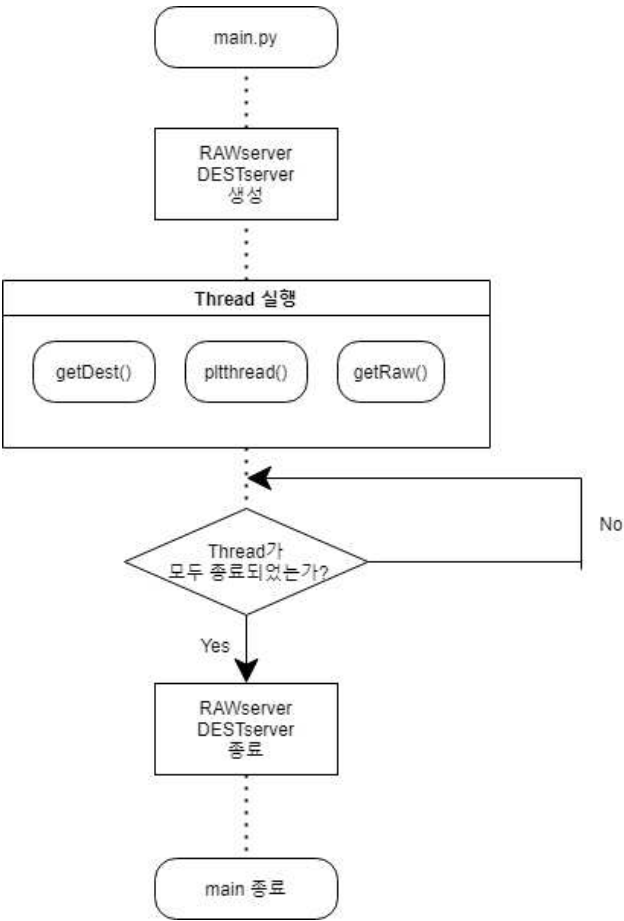
○ Software 구성

- HappyNewEar의 SW는 크게 ODAS(Open embeddeD Audition System)와 main 부분으로 구성되어 있다. ODAS는 소리의 방향을 구분하고, beamforming을 통해 소리를 각각의 채널로 분리한다. main에서는 Socket 통신을 통해 ODAS로부터 받은 위치 데이터, RAW PCM 데이터를 Classification 모듈을 활용해 가공, 분류하고 결과를 출력한다.

○ Software 설계도 (흐름도 및 클래스 다이어그램 등 / 개발언어에 따라 선택)



<함수 개요>



<프로그램 전체 흐름도>

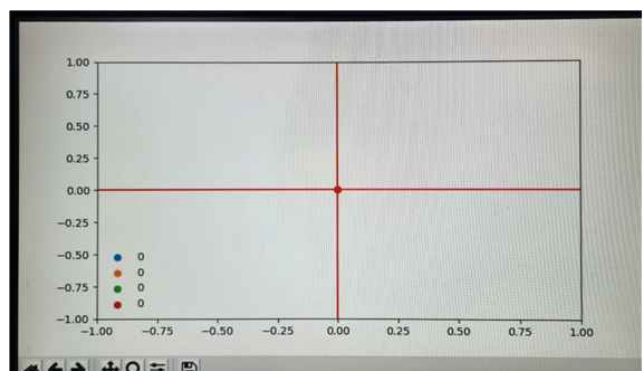
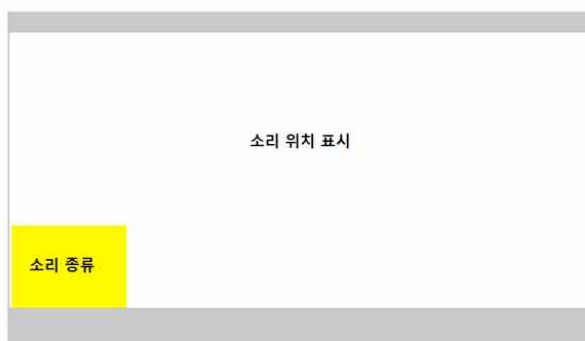
- main.py()

1. ODAS와 통신하는 RAWserver(port: 9001), DESTserver(port: 9000) 서버를 생성하고, Client를 Listen 한다.
2. 함수들의 실행을 기록하는 log.txt 파일 스트림을 생성한다.
3. getRaw(), getDest(), pltthread() 함수를 실행하는 Thread를 생성한 후 실행한다.
4. 터미널 상에서 ODAS를 실행한다.
5. ODAS와 Server가 소켓으로 연결되어 통신 및 데이터 처리와 출력을 시작한다.
6. Thread가 모두 종료되면 서버를 닫고 프로그램이 종료된다.

○ Software 기능 (필요시 알고리즘 설명 포함)

- 4 mic array를 통해 받아들인 소리를 분석한다. ODAS를 이용해 음원 방향을 구분하고, beamforming을 사용해 소리를 각각의 채널로 분리한다.
- 텐서플로와 Open API를 사용했다. 기존 API의 521가지의 소리를 발생 빈도 혹은 중요성을 기준으로 47개의 소리로 정리했다. 그중 위험군으로 지정된 8가지 소리가 인식되면 특정 화면이 그래프 배경화면으로 나타남과 동시에 골전도 스피커로부터 소리 및 진동 신호를 받도록 해 착용자에게 경고한다.
- 음원 방향 데이터와 소리 분류 데이터를 각각 4개의 채널에 할당해 어떤 지점에서 어떤 소리가 들리는지를 그래프 위에 표현했다. matplotlib Real-time scatter plot을 사용해 지속적으로 변화하는 도심 환경 속에서도 끊김 없이 사용할 수 있도록 했다.
- 기존에는 ODAS로 소리의 위치를 받고, PyAudio로 소리를 분류하려고 하였으나, 마이크가 두 환경에 대해 동시에 작동되지 않는 한계가 있었다. 이를 해결하기 위해 ODAS에서 socket으로 실시간 데이터를 수신하도록 했다.
- ODAS는 방향 정보를 3차원 구 모양으로 제공한다. 이는 직관적인 방향 파악이 어렵다는 단점이 있어 3차원 좌표를 xy평면에 사영함으로써 평면 좌표로 나타냈다.

○ 프로그램 사용법 (Interface)



- 4-mic-array를 연결한 후 전원을 켜다. 그 후 HappyNewEar 폴더 안의 HappyNewEar.sh 파일을 더블클릭하여 실행한다.

○ 개발환경 (언어, Tool, 사용시스템 등)

raspberry pi4 (Linux)	라즈베리 파이에서 PI OS를 다루려면 리눅스(Linux) 운영체제를 사용해야 한다. 리눅스 자체에 파이썬, 스크래치 등과 같은 도구들을 포함하고 있어 개발 목적에 맞는 tool들을 sudo 권한 부여를 통해 쉽게 구동시킬 수 있다는 장점이 있다.
ODAS (Open embeddeD Audition System)	Sound Localization을 통해 음원을 추적 및 분리하고 필터링을 수행하기 위한 전용 라이브러리이다. 음원을 정확하게 찾고 추적하려면 마이크 위치를 정확하게 알아야 한다. C언어를 기반으로 완전한 시스템이 구축되어 있고, 마이크 어레이의 신호를 제대로 처리하도록 추가 매개 변수를 지정해야 한다. 우리가 사용하는 소리 유형의 정보에 따라 구성 파일을 수정해서 사용했다. 각 데이터 유형에 대한 형식과 인터페이스 유형을 변경해 소스 폴더를 구성했다.
python3.7.3	파이썬은 고급 언어이며 인터프리터(interpreter) 언어이다. C나 Java와 같은 경우는 코드 작성 후 컴파일러(compiler)에 의해 변환 후 실행하지만, 파이썬은 코드 작성 후 바로 실행하여 결과를 볼 수 있는 장점이 있다. 인터프리터를 제공하기 때문에 프로그램이 끝나야만 결과를 볼 수 있는 다른 언어와는 달리, 도중에 값을 확인할 수 있어 유용하게 임베디드 시스템 개발을 진행할 수 있다. 이러한 장점이 있어 python을 사용해 ODAS에서 소리의 위치와 종류의 결괏값을 도출하고, tensorflow로 데이터를 수신해 소리의 종류를 구분했다. 또한 모델의 결괏값을 mapping 해 matplotlib에 pyplot을 이용해 표현했다.
tensorflow lite	텐서플로의 경량화 버전으로 다양한 데이터 흐름 프로그래밍을 위한 오픈소스 소프트웨어 라이브러리이다. 학습된 TensorFlow 모델을 통해 소리의 종류를 분류하는 기계 학습 응용프로그램에 활용하기 위해 사용했다. 효율적인 On-Device 추론을 수행할 수 있는 런타임으로 텐서플로 라이트는 케라스 모델을 지원함과 동시에, 임베디드 장치에 모델을 배포할 수 있다. 높은 버전의 파이썬에는 TensorFlow 지원이 안되고, 32비트 라즈베리 파이 OS에서 TensorFlow 사용이 불가능해, TensorFlow Lite와 그 모델을 사용했다.
Thonny python IDE	Thonny는 초보자를 위해 설계된 Python 용 통합 개발 환경이다. 따라서 간단한 전체 코드를 단계별로 실행하기 위해 사용했고, 단계별 표현 평가, 콜 스택의 세부적인 시각화, 참조 및 heap의 개념을 설명하는 모드를 지원한다.

JSON Socket 통신	<p>JSON(JavaScript Object Notation)이란 데이터 오브젝트를 전달하기 위해 인간이 읽을 수 있는 텍스트를 사용하는 개방형 표준 포맷이다. python으로 작성된 client와 server 사이에 JSON으로 데이터를 주고받는 프로그램을 활용해서 데이터 통신 출입구 객체를 생성해 연결을 요청하는 프로그램을 작성했다. 자료의 종류에 큰 제한이 없으며, 특히 컴퓨터 프로그램의 변수값을 표현하는 데 적합하다. JSON 데이터와 Python 데이터는 1:1로 mapping 해 사용하기 쉽다. 실시간으로 위치 데이터를 수신하고 가공하도록 별도의 Thread에서 실행한다.</p>
-------------------	--

□ 개발 프로그램 설명

○ 파일 구성

HappyNewEar

- └ img - 특정 소리의 이미지 저장
 - └ Alarm.png
 - └ Crushing.png
 - └ Emergency vehicle.png
 - └ Explosion.png
 - └ Fire alarm.png
 - └ Shout.png
 - └ Siren.png
 - └ Vehicle horn.png
- └ odas - ODAS의 소스 폴더
 - └ bin
 - └ odaslive
 - └ odas.cfg - 4 mic array에 맞게 Tuning된 설정 파일
- └ _pycache_ - Classification 모듈의 컴파일 캐시
 - └ Classification.cpython-3.7.pyc
- └ Classification.py - 소리 분류를 위한 모듈
- └ HappyNewEar.sh - 프로그램의 실행 파일
- └ log.txt - 프로그램의 실행 로그
- └ main.py - 실행
- └ yamnet.tflite - 사전에 학습된 소리 분류 모델
- └ yamnet_class_map.csv - 모델 결괏값의 매핑 파일

○ 함수별 기능

■ main.py

- noise(rate): 분류된 소리의 위험도(rate)에 따라 스피커로 알림 소리를 출력하는 함수
100Hz, 0.5초간 지속되는 소리 알림을 rate 회 발생시킨다.
- Sort(data): RAW PCM 데이터를 채널별로 분류하고, 이를 학습된 TensorFlow 모델을
통해 분류하는 함수
Classification 모듈의 tonumpy() 와 preprocess() 함수를 사용하여 RAW 데이터를
numpy 배열로 가공하고, 이를 4개의 채널로 분리한 후 classifier() 함수로 결과를
도출한다.
- getRaw(): ODAS의 Socket을 통해 RAW PCM 데이터를 수신하는 함수
(*주요 함수의 흐름도에서 후술)
- getDest(): ODAS와 Socket을 통해 소리의 위치 데이터를 수신하는 함수
(*주요 함수의 흐름도에서 후술)
- plttheard(): 변수에 저장된 소리의 위치, 종류 데이터를 사분면에 표시하는 함수

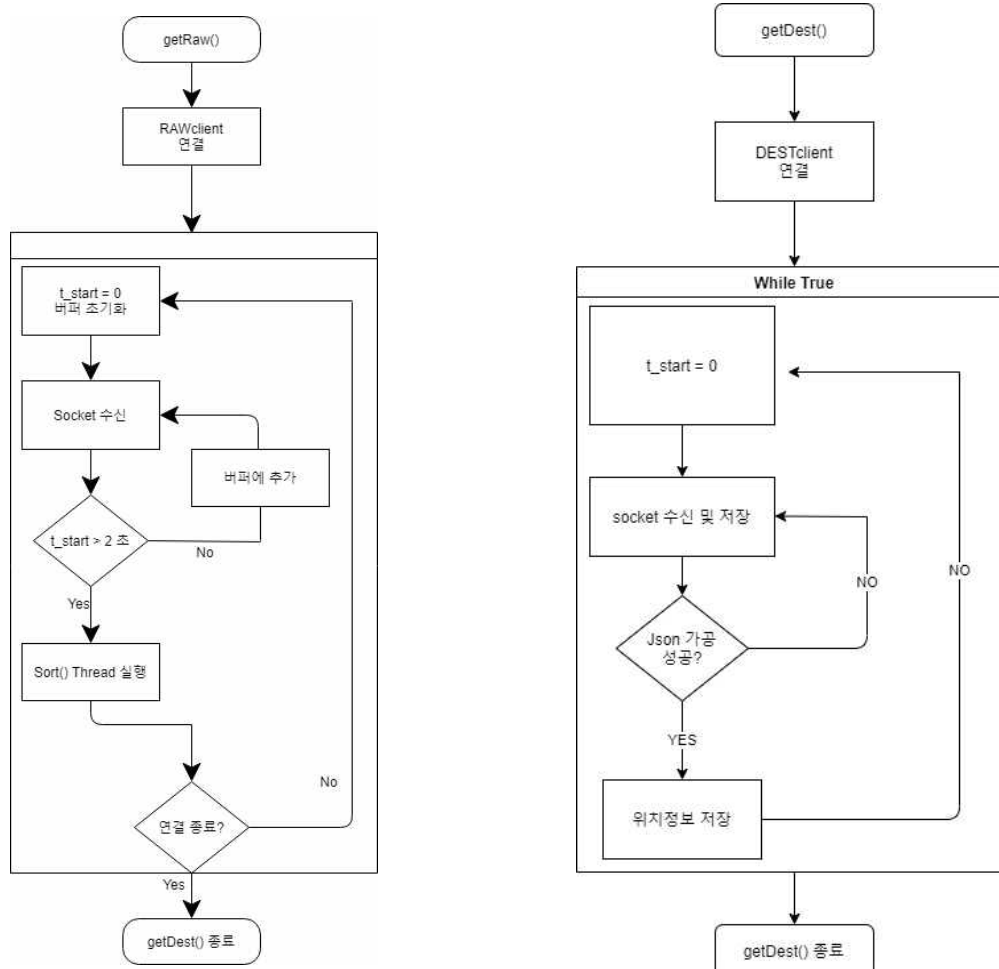
■ Classification.py

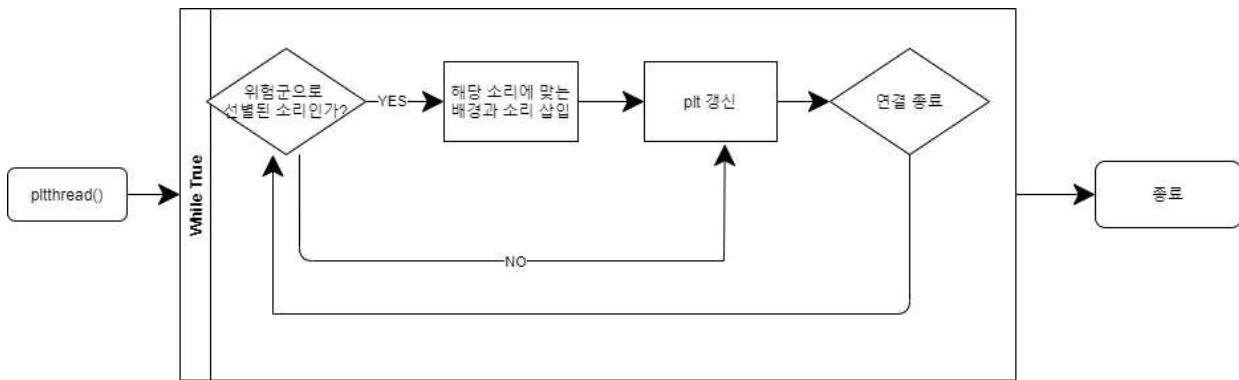
- getcsv(path): 결괏값이 매핑된 (path).csv 파일을 불러와 리스트로 저장되는 함수

■ Classificate(클래스)

- _init_(): 클래스를 객체화할 때 실행되는 함수
Tensorflow Lite 모델을 불러오고, getcsv()를 통해 결과를 매핑할 리스트를 받는다.
- tonumpy(buff): ODAS로부터 수신한 RAW PCM 버퍼의 데이터형을 변환하는 함수
numpy 모듈의 frombuffer() 함수를 활용해 float32형의 데이터로 변환하고, 중간중간 잘린 데이터를 이어붙인다.
- preprocess(data): 데이터들의 크기를 조정하는 함수
데이터의 크기를 TensorFlow Lite 모델에서 요구하는 [-1,1] 사이의 범위로 조정한다.
- classifier(data): 처리된 데이터를 모델에 넣어 결괏값을 받는 함수
Tensorflow Lite 모델에 data를 넣어 그 결괏값(숫자)을 얻고, 이를 getcsv()로 얻은 리스트에 대입해 영문으로 된 결과를 반환한다.
- pltShow(data): 소리 데이터를 확인하기 위한 함수
matplotlib 모듈의 pyplot을 사용하여 data의 파형을 그린다.

○ 주요 함수의 흐름도





- getRaw()

1. RAWserver에 ODAS가 연결되면 RAWclient로 객체화한다.
2. 시작 시간을 t_start 변수에 저장한다.
3. ODAS로부터 RAW PCM 데이터를 수신한다.
- 4-1. t_start로부터 2초가 지나지 않았으면 raw 변수에 데이터를 append 한다.
- 4-2. t_start로부터 2초가 지나면 Sort 함수를 실행하는 Thread를 생성하고, t_start를 초기화한다.
5. ODAS로부터 10회 이상 응답이 없으면 Thread를 종료한다.

- getDest()

1. DESTserver에 ODAS가 연결되면 DESTclient로 객체화한다.
2. ODAS로부터 데이터를 받아온다.
3. 받아온 데이터를 json으로 로드한다.
4. json에서 위치 정보를 pos에 저장한다.

○ 기술적 차별성

■ 3차원 상의 음원 정보를 2차원으로 나타냈다.

- ODAS는 4-mic-array로 소리를 받아 소리 정보를 반지름이 2.5m인 구 표면에 점으로 표현한다. 이는 위치를 3차원 상으로 나타내는 것이기 때문에 더 정확한 위치를 알 수 있다는 장점은 있지만 한눈에 방향을 파악하는데 어려움이 있다. 따라서 이를 해결하기 위해 3차원 상의 모든 점을 xy 평면으로 사영해 나타냈다.

■ 인터넷 없이 즉시 사용 가능하다.

- 오프라인은 온라인의 반대 개념으로, 컴퓨터나 스마트폰, 카메라와 같이 인터넷이나 인트라넷, 엑스트라넷에 접속할 수 있지만 접속을 하지 않은 상태를 뜻한다. 네트워크로 연결되어 있지 않은 상태에서 사용이 가능해 네트워크 연결 문제로 인한 렉이나 프레임 드랍 버그 같은 문제가 발생할 일이 없다. 이러한 오프라인의 특징을 활용하여 지속적이고 중단 없는 서비스가 가능한 컴퓨팅 환경을 구축해 터치 한 번으로 딜레이 없이 구동할 수 있도록 했다. 결과적으로 청각 장애인이 웨어러블 인터페이스를 쉽게 조작할 수 있다.

□ 개발 중 발생한 장애요인과 해결방안

- 소리가 발생한 위치까지의 거리를 구하는데 어려움을 겪었다.
 - 음원까지의 거리를 구하기 위해서 DTW를 이용하여 시간 순서상에 여러 개의 연속적인 데이터를 비교함으로써 두 종류의 데이터가 얼마나 유사한지 판단해 그 두 소리의 시차로 음원까지의 거리를 구하고자 했다.
 - 이때 쌍곡선 항법, 혹은 삼각측량법을 이용하여 거리를 구하려고 했다. 하지만 라즈베리 파이에 이를 구현하려 했을 때 딜레이가 길어지는 등 여러 문제가 발생했다. 따라서 기존에 사용하던 MAX9814 대신에 4-mic-array를 사용하여 방향과 위치를 나타낼 수 있게 했다.
- ODAS와 파이썬의 PyAudio 모듈에서 같이 마이크를 쓰지 못하는 문제가 발생했다.
 - 소리를 받아와서 어떤 소리인지 분류하기 위해 ODAS로 위치를 받은 후 PyAudio를 통해 소리를 분류하려고 시도했다. 하지만 ODAS와 PyAudio에 동시에 마이크를 사용할 수 없는 문제가 발생했다.
 - 이를 해결하기 위해 PyAudio를 사용하지 않고, 소켓을 통해 ODAS에서 바로 소리 방향과 소리 데이터를 받아서 사용했다.
- 몇몇 프로그램이 개발 환경과 호환되지 않는 문제가 발생했다.
 - 프로젝트 초반, 32비트 라즈베리 파이 OS에 Anaconda, Tensorflow, PyAudio 등의 모듈을 활용하여 구현하고자 했다. 그러나 Anaconda가 라즈베리 파이의 CPU를 지원하지 않고, Tensorflow가 32비트 환경에서 실행되지 않는 문제가 발생했다.
 - 32비트 -> 텐서플로가 실행되지 않음.
 - 64비트 -> Anaconda가 실행되지 않고, 파이썬의 버전이 너무 높아 Tensorflow가 깔리지 않음.
 - 이를 해결하기 위해 32비트 라즈베리 파이 OS에 TensorFlow Lite와 그 모델을 사용했다.
- 프로그램 구조상 여러 가지 함수가 동시에 실행되어야 하는 문제가 발생했다.
 - 프로그램의 구조상 Socket을 통해 데이터를 수신하는 함수가 계속해서 실행되어야 하는데, 지금까지 학습한 방법으로는 이를 구현할 수 없었다.
 - Python의 Threading 모듈을 이용해 함수들을 별도의 thread에서 실행시킴으로써 기존 계획대로 구현할 수 있었다.
- 소켓 통신을 하며 수신 버퍼가 밀리는 문제가 발생했다.
 - ODAS와 main 소켓 통신에서 main 데이터 처리 함수 구조상의 문제로 인해 수신되는 데이터 버퍼가 점점 밀리는 병목현상이 발생했다.
 - 데이터의 병목이 일어나는 함수를 다른 thread로 분리함으로써 병목현상을 해결할 수 있었다.

□ 개발결과물의 차별성

- 애플리케이션, 화면을 보지 않고 골전도 스피커만으로도 위험 정도 파악이 가능하다.
 - 기존에 연구되었던 소리 감지 IoT 웨어러블 디바이스는 감지된 자동차 경적소리와 사이렌 소리를 애플리케이션을 통해서 방향을 나타내 화면을 봐야만 위험을 감지할 수 있다. 하지만 이번에 개발한 작품은 골전도 스피커를 통해 위험 정도를 알려주어 LCD 화면을 보지 않고도 위험을 인지할 수 있다.
 - 위험 정도를 3단계로 나누어 위험할수록 진동이 울리는 시간을 길게 하여 사용자가 인지할 수 있게 했다. 진동이 울린 다음 화면을 확인해야만 위험요소의 정보를 파악할 수 있던 기존 작품과는 달리 진동만으로도 대략적인 위험 정도를 알 수 있다.

- 분류된 소리를 LCD 화면을 통해 사진으로 알 수 있다.
 - 4-mic array를 통해 소리를 받아오고, ODAS 프로그램을 사용해 이 소리의 종류를 분석한다. 그중에서 위험하다고 판단되는 몇 가지의 소리는 화면에 관련된 사진이 나타나면서 진동이 울리게 된다. 글씨뿐만이 아니라 그림이 함께 나타남으로써 어떤 위험이 발생했는지 쉽고 빠르게 판단할 수 있도록 했다. 이로써 위험 상황 발생 시 더 빠른 대처가 가능해졌다.

- 소리 분류 가짓수를 더 세분화했다.
 - 기존에 있던 작품은 경적 소리와 사이렌 소리만 분류해서 알려주었다. 그러나 이 작품은 위험 소리뿐만 아니라 말소리, 웃음소리, 우는소리, 기침 등 다양한 소리를 포함했다. 위험 소리에는 개 짖는 소리, 사이렌 소리, 폭발음, 드릴 소리 등을 추가해 기존보다 더 세분화하여 나타냈다.
 - 기존 API를 조금 수정해 소리 분류 종류는 줄이되, 보편적인 소리로 묶어 정리했다. 이를 통해 소리 분류를 기존보다 세분화하면서도, 정확도를 향상시킬 수 있었다.
 - 이를 통해 위험성이 높은 소리는 진동을 통해 알 수 있고, 위험성이 높지는 않지만 필요한 소리 또한 알 수 있게 되었다. 이는 청각장애인이 일상생활을 좀 더 편리하게 생활할 수 있게 될 것으로 기대된다.

□ 개발 일정

No	내용	2021年											
		6月			7月			8月			9月		
기획	프로젝트 아이디어 선정												
	자료 수집 및 기능 설계												
개발 및 구현	하드웨어 모델링												
	라즈베리 파이 설정												
	SW 개발 환경 구축												
	음성 인식 및 소리 방향 분석												
	tensorflow 머신러닝												
	실시간 그래프 그리기												
	3D 모델링												
	디스플레이 설정												
	블루투스 통신												
	프로토타입 테스트 및 수정												
기타	시연 영상 제작												
	보고서 작성												

□ 팀 업무 분장

No	구분	성명	참여인원의 업무 분장
1	팀장	이현재	개발 총괄, 알고리즘 설계
			Linux 개발 환경 구축
			소켓 통신 코드 작성
			ODAS 소리 데이터 분류
			RAW PCM 데이터 통신
			소리 정보 분석 및 방향 결과 표시 함수 작성
			보고서 작성
2	팀원	이주형	소리 파형 확인 함수 작성
			소리 분류를 위한 모듈 작성
			소켓 통신 코드 작성
			tenserflow 오디오 빅데이터 정리
			블루투스 모듈 설정
			실시간 그래프 그리기
			보고서 작성
3	팀원	이지수	하드웨어 설계
			라즈베리 파이 디스플레이 통신
			3D 모델링
			프로토타입 테스트
			시연 영상 제작
			보고서 작성
4	팀원	장예진	하드웨어 설계
			마이크 모듈 설정
			소리 데이터 수신
			고위험군 알림 그림 삽입
			3D 프린팅
			보고서 작성
5	팀원	정하연	하드웨어 설계
			소리 데이터 조사 및 수집
			골전도 스피커 알림 설정
			프로토타입 테스트
			시연 2영상 제작
			보고서 작성