



# 주제 선정 : 음량 균형 보정 기기

## <조건부 승인 2차>

조건 1. 실제로 공연예술학과 등의 실습현장에서 사용성 테스트 진행한 결과를 제출

공연예술학과 문의에 대한 답변

실제 테스트 및 사용성 평가

조건 2. 제품을 사용하기 전/후 비교 필수

## <조건부 승인 1차>

조건 1. 기존 음향기기에 연동하여 사용할 수 있는 형태로 제작

하드웨어 구현 수정

오디오 믹싱 콘솔: 연동 테스트 위한 음향 기기

조건 2. 실시간 노이즈 제거

조건 3. 실시간 증폭 처리

## 1. 작품개요

## 2. 주제 선정 과정

### 2.1) 문제인식

관련 사례(tvN '뽕뽕 지구오락실')

자문 Q&A (음향 관련 일을 하고 있는 지인)

### 2.2) 가치

편리한 노이즈 제거 기능

일관된 사운드 레벨

휴대성과 간편한 조절

## 3. 구현 방법

### 3.1) 구현개요

하드웨어 구성:

소프트웨어 구현:

### 3.2) 하드웨어

마이크 입력 장치

ADC(Analog-to-Digital Converter):

MAX98357A 모듈:

<라즈베리 파이와 각 모듈 연결 방법>

### 3.3) 소프트웨어

#### 3.3.1) 노이즈 제거

실시간 노이즈 제거 알고리즘

<노이즈를 제거한 후에 신호를 증폭하는 것이 좋은가?>

#### 3.3.2) 사운드 레벨 동기화

<RMS 레벨>

[FLOW CHART]

## 4. 유사제품 비교

1. Behringer의 X32 디지털 믹서:

2. Samsung Sound Assistant:

## 5. 팀원 간 업무 내용

## 6. 작품 제작 추진 계획 및 일정표

## 7. 지원 경비 사용 계획

아래 보고서는 기존과 동일합니다.

<조건부 승인 2차>에 대한 상세계획서를 추가하였습니다.

## <조건부 승인 2차>

### 조건 1. 실제로 공연예술학과 등의 실습현장에서 사용성 테스트 진행한 결과를 제출

#### 공연예술학과 문의에 대한 답변

공연예술학과에서 사용하는 전문적인 장비에 실험적인 장비를 연결하는 것은 저희 쪽 장비에 고장을 유발할 수 있는 위험한 실험이기 때문에 어려울 것 같습니다.

- 공연예술학과에서 사용하는 믹서 콘솔은 고가의 장비에 해당하기에 검증되지 않은 '음량 균형 보정 기기'를 적용하는 것은 어려워보입니다.
- 이에 디바이스마트에서 구매 가능한 믹서 콘솔을 활용하여 실제 사용되는 **음향 기기와의 연결성**을 테스트 해보고자 합니다.

#### 실제 테스트 및 사용성 평가

- 공연예술학과 교수님 평가

💡 현재, 공연예술학과 교수님들께 메일로 사용성 평가 가능 여부에 문의를 드린 상태입니다.  
아직 답변이 오지 않아 보고서에 첨부하지 못하였습니다.

만약, 공연예술학과 교수님들께서 평가가 어려울 것 같다고 하신다면, 자문을 구한 공연예술학과 음향감독님을 통해 사용성 평가를 진행할 계획입니다.

### 조건 2. 제품을 사용하기 전/후 비교 필수

#### 실험절차

1. 두 참여자가 동시에 마이크 앞에 위치하고 동일한 문장을 읽습니다.
2. 마이크를 통해 두 사람의 음성이 동시에 입력됩니다.
3. 연동성 테스트를 위해 사용하는 오디오 믹서의 기능을 사용하여 음성 파일을 저장합니다.
4. 녹음된 음성 파일을 통해 제품 사용 전/후를 비교합니다.



#### 18. MP3 컨트롤 버튼

▶ 짧게 누르면 정지/재생.

길게 누르면 녹화 상태로 전환합니다.

메인 채널의 출력 녹음 신호는 USB에 저장됩니다.

## 제품 사용 전/후 비교 방법

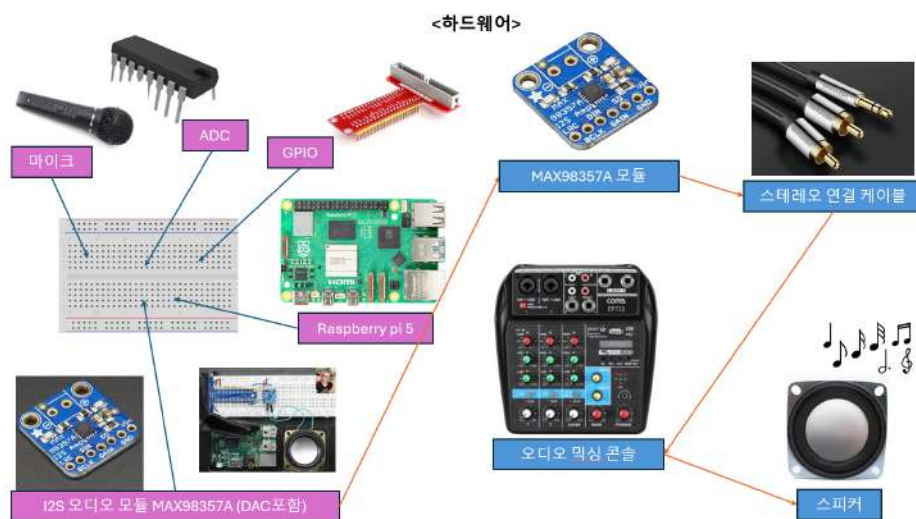
1. 제품을 적용하기 전/후의 비교를 위해 파이썬 오디오 처리 라이브러리인 librosa를 사용합니다.
2. 사용자의 음성 파일을 각각 파형으로 보여줍니다.
3. 제품 적용 후, 균형을 이룬 음성의 파형이 출력되는 것을 확인할 수 있습니다.

## <조건부 승인 1차>

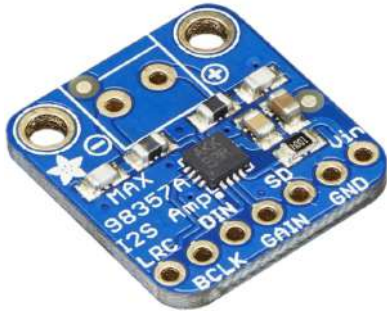
### 조건 1. 기존 음향기기에 연동하여 사용할 수 있는 형태로 제작

#### 하드웨어 구현 수정

오디오 믹서와 연동하는 부분을 추가할 계획입니다. 기존에 사용하는 오디오 증폭 모듈인 MAX98357A 내에 DAC가 포함되어 있어, 이를 오디오 믹서에 연결하여 출력할 수 있습니다.



- MAX98357 모듈  
: 오디오 신호를 스피커로 전달하기 위해 설계된 디지털 오디오 앰프 모듈



MAX98357 모듈



스테레오 연결 케이블

- MAX98357A 모듈은 일반적으로 출력을 위한 단일 스피커 연결을 가지고 있습니다. 따라서 이 모듈을 통해 오디오 믹서의 입력으로 직접 연결이 가능합니다.
  1. 라인 레벨로 연결하기
 

MAX98357A 모듈의 출력에 RCA, 1/4인치 TRS, 또는 3.5mm 헤드폰 잭과 같은 라인 레벨 출력을 추가하여, 이를 오디오 믹서의 라인 레벨 입력으로 직접 연결할 수 있습니다. 이렇게 하면 모듈의 출력이 오디오 믹서로 전달됩니다.
  2. 라즈베리파이의 DAC를 사용하여 연결하기
 

라즈베리파이의 DAC를 사용하여 디지털 오디오 신호를 아날로그로 변환한 후, 이를 MAX98357A 모듈의 입력으로 연결할 수 있습니다. 따라서 라즈베리파이의 DAC를 사용하여 오디오를 생성하고, 이를 MAX98357A 모듈에 연결하여 출력할 수 있습니다.
  3. 오디오 인터페이스를 사용하여 연결하기
 

라즈베리파이에 오디오 인터페이스 모듈을 추가하여 MAX98357A 모듈의 입력에 연결할 수 있습니다. 이를 통해 라즈베리파이의 오디오 신호를 MAX98357A 모듈로 전달할 수 있습니다.

⇒ 위의 세 가지 방법 중 저희는 첫 번째 방법으로, **라인 레벨**로 연결하여 오디오 믹서와 연동하고자 합니다.
- MAX98357A 모듈은 일반적으로 스피커를 연결하는 데 사용되는 터미널 블록을 통해 출력을 제공합니다. 3.5mm TRS to RCA 케이블은 3.5mm 헤드폰 잭(TRS)과 RCA 입력 잭을 연결하는 데 사용됩니다. 따라서 MAX98357A 모듈의 출력을 RCA 입력으로 연결하려면, 케이블을 사용하여 다음과 같이 연결하면 됩니다.
  1. MAX98357A 모듈의 출력단 접근하기:
    - MAX98357A 모듈에는 스피커 연결을 위한 터미널 블록이 장착되어 있으며, 이 터미널 블록은 "+"와 "-"로 표시된 두 개의 연결점을 포함합니다. "+"는 긍정적인 신호를, "-"는 부정적인 신호를 나타냅니다. 이 두 터미널에 스피커 케이블을 연결하여 오디오 신호를 스피커로 전달할 수 있습니다.
  2. 3.5mm TRS to RCA 케이블 사용하기:
    - 3.5mm TRS to RCA 케이블을 사용하여 MAX98357A 모듈의 3.5mm 헤드폰 잭에 연결합니다. 3.5mm TRS 케이블의 헤드폰 잭을 MAX98357A 모듈의 출력 잭에 연결하고, RCA 플러그를 오디오 믹서의 RCA 입력 잭에 연결합니다.
  3. 연결 확인하기:
    - 모든 연결이 올바르게 이루어졌는지 확인하는 것이 중요합니다. 이는 오디오 시스템에서 소리의 질과 안정성에 직접적인 영향을 미칩니다. "+"와 "-" 터미널이 올바르게 연결되었는지, RCA 케이블이 오디오 믹서나 앰프의 RCA 입력에 올바르게 삽입되었는지 확인해야 합니다.

## 오디오 믹싱 콘솔: 연동 테스트 위한 음향 기기



#### 1. 모노 입력 단자

MIC / LINE: 캐논 단자를 지원하는 일반 마이크 또는 콘덴서 마이크(팬텀파워 자체지원)를 연결할 수 있습니다.

#### 2. 스테레오 입력 단자

전자 키보드 또는 오디오 장치를 연결할 때 사용하는 단자입니다. 6.3 모노 단자 및 RCA 단자 타입 두가지를 지원합니다.

#### 3. 팬텀 파워 전원 스위치(콘덴서 마이크)

#### 4. 팬텀 전원 표시 LED

#### 5. [GAIN] 휠

입력 신호의 GAIN 값을 조절할 때 사용합니다.

#### 6. [High] 휠

채널의 고음 영역 톤을 조절할 때 사용합니다.

#### 7. [Low] 휠

채널의 저음 베이스 톤을 조절할 때 사용합니다.

#### 8. [EFX] 버튼

음장 효과 조절 버튼입니다.

#### 9. 채널 볼륨 휠

볼륨 크기를 조절할 때 사용합니다.

#### 10. [ST/USB] 버튼

#### 12. [Phone] 휠

헤드폰 단자의 볼륨을 조절할 때 사용합니다.

#### 13. [Repeat] 휠

에코 반복 및 빈도를 조절할 때 사용합니다.

#### 14. [Deley] 휠

에코 반복 간격을 설정할 때 사용합니다.

#### 15. 헤드폰 단자

#### 16. 전원 표시등

#### 17. 신호 출력 상태 표시 디스플레이

#### 18. MP3 컨트롤 버튼

▶|| 짧게 누르면 정지/재생

길게 누르면 녹화 상태로 전환합니다.

메인 채널의 출력 녹음 신호는 USB에 저장됩니다.

◀◀ / — 짧게 눌러 이전 곡, 길게 눌러 볼륨을 낮춥니다.

◀◀ / + 짧게 눌러 다음 곡, 길게 눌러 볼륨을 높입니다.

MODE/BT : 짧게 눌러 LINE/BT/PC를 전환합니다. 길게 누르면 반복 플레이 모드에 진입합니다.

#### 19. MP3 디스플레이

MP3 작동 조절 및 상태를 표시합니다.

스테레오 입력과 USB 메모리 입력 신호를 전환할 때 사용합니다.

## 20. MP3 USB 포트

## 21. 스테레오 출력 인터페이스

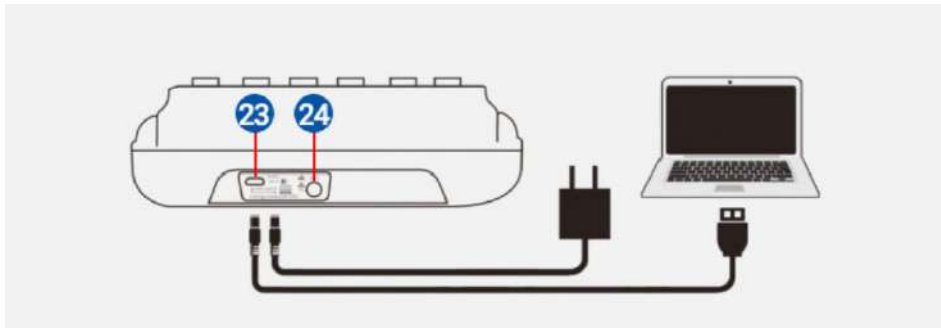
### 11. [Main] 휠

메인 볼륨을 조절합니다.

최적의 값을 설정해주시고.

## 22. 녹음 단자

스테레오 입/출력단자와 녹음 장치를 연결합니다.



## 23. USB 전원 출력 인터페이스

외부 5V 전원 공급 장치 또는 컴퓨터의 USB 단자를 연결합니다. 컴퓨터와 연결될 경우 장치는 외장 사운드 카드로 인식됩니다.

## 24. MP3 / PC 연결 전환 스위치

컴퓨터의 전원을 통해 공급됩니다. 스위치 버튼을 눌러 MP3 USB / PC 연결을 전환합니다.

## 조건 2. 실시간 노이즈 제거

GitHub - SaneBow/PiDTLN: Apply machine learning model DTLN for noise suppression and acoustic echo cancellation on Raspberry Pi  
Apply machine learning model DTLN for noise suppression and acoustic echo cancellation on Raspberry Pi - SaneBow/PiDTLN

<https://github.com/SaneBow/PiDTLN>

라즈베리파이3를 이용하여 실시간으로 입력되는 소리에 대해 노이즈를 제거하여 출력하는 모델인 'PiDTLN'을 분석해보면, 노이즈를 제거하는 과정에서 오디오 입력은 블록 단위로 샘플링되어 처리됩니다. 이 모델의 노이즈 제거 과정에서의 주요 파라미터 중 하나는 오디오 처리를 위한 블록 길이(block\_len)와 블록 이동(block\_shift)입니다. 이 값들은 오디오 데이터가 어떻게 샘플링되어 처리되는지 결정합니다.

코드 내에서 블록 길이(block\_len)와 블록 이동(block\_shift)은 다음과 같이 설정됩니다:



**block\_len\_ms = 32**

→ 이 값은 처리할 오디오 블록의 길이를 밀리초 단위로 지정합니다. 즉, 한 번에 32ms 길이의 오디오가 처리됩니다.



**block\_shift\_ms = 8**

→ 이 값은 오디오 블록 사이의 이동(오버랩)을 밀리초 단위로 지정합니다. 즉, 처리 과정에서 8ms마다 새로운 오디오 블록이 시작됩니다.

이러한 값들은  $fs\_target = 16000$  (타겟 샘플링 레이트가 16kHz임을 나타냄)과 함께 사용되어 실제 샘플 수를 계산합니다.

- **block\_len** =  $\text{int}(\text{np.round}(fs\_target * (block\_len\_ms / 1000)))$
- **block\_shift** =  $\text{int}(\text{np.round}(fs\_target * (block\_shift\_ms / 1000)))$

여기서 **block\_len**과 **block\_shift**는 샘플 단위로 계산됩니다. 예를 들어, 16kHz의 샘플링 레이트에서 32ms의 블록 길이는  $16 * 32 = 512$  샘플을 의미하며, 8ms의 블록 이동은  $16 * 8 = 128$  샘플을 의미합니다. 따라서 이 스크립트는 512 샘플 길이의 오디오 블록을 처리하고, 각 처리 단계에서 128 샘플만큼 오디오 블록을 이동시키며 노이즈를 제거합니다. 이는 고정된 크기의 오디오 블록을 연속적으로 처리하면서, 각 블록 사이에 일정량의 오디오가 오버랩되도록 합니다.

Model	ns.py	aec.py	aec_mp.py
128	N/A	4.43 ms	2.52 ms
256	N/A	6.36 ms	3.76 ms
512	2.56 ms	12.35 ms	8.48 ms

모델 별 실시간성

1. **ns (Noise Suppression)**: "ns"는 잡음 억제(Noise Suppression)를 나타냅니다. 이는 주로 입력 오디오에서의 잡음을 감소시키는 알고리즘이며, 주요 목표는 클리어한 음성을 보다 잘 감지하고 전달하는 것입니다. 잡음 억제는 주로 통화나 음성 인식과 같은 응용 프로그램에서 사용됩니다.
2. **aec (Acoustic Echo Cancellation)**: "aec"는 음향 에코 캔슬레이션(Acoustic Echo Cancellation)을 의미합니다. 이는 주로 통화나 화상 통화와 같은 상황에서 발생하는 음향 에코를 제거하는 알고리즘입니다. 예를 들어, 상대방의 음성이 마이크를 들어가서 스피커로 재생되는 경우 발생하는 에코를 제거하는 데 사용됩니다.
3. **aec\_mp (Acoustic Echo Cancellation with Multi-Point)**: "aec\_mp"는 멀티포인트 음향 에코 캔슬레이션(Acoustic Echo Cancellation with Multi-Point)을 의미합니다. 이는 다중 마이크나 다중 스피커를 사용하는 경우에 발생하는 복잡한 음향 에코를 처리하는 알고리즘입니다. 예를 들어, 회의실에서 여러 사람이 다른 위치에 마이크와 스피커를 사용하는 경우에 유용합니다.

위의 표에서 "128", "256", "512"는 DTLN 모델의 블록 크기를 나타냅니다. "블록 크기"는 모델이 처리하는 입력 샘플의 수를 의미합니다. 이 값은 모델이 한 번에 처리하는 샘플의 양을 나타냅니다. 여기서는 각 모델이 128, 256 또는 512개의 샘플을 한 번에 처리한다는 것을 의미합니다.

여기서 "2.56ms" 값은 각 노이즈를 제거하는데 소요된 시간을 나타냅니다. 예를 들어, "aec\_mp.py" 스크립트에서 블록 크기가 512인 경우 처리 시간이 8.48ms라고 나와 있습니다. 이는 해당 모델이 512개의 샘플을 처리하는 데 평균 8.48ms가 걸린다는 것을 의미합니다. 따라서 해당 표는 다른 블록 크기의 DTLN 모델이 처리하는 데 필요한 시간을 보여주며, 각각의 처리 시간이 실시간 처리에 적합한지를 판단하는 데 도움이 됩니다.

일반적으로 실시간 오디오 처리 시스템에서는 입력 신호를 실시간으로 처리해야 하므로, 처리 시간이 입력 신호의 샘플링 주기보다 짧아야 합니다. 이 모델의 경우 샘플링 주기는 8ms(0.008초)이므로, 한번에 처리되는 샘플이 128인 모델의 경우 최대 4.43ms의 시간이 소요되므로 실시간성을 유지할 수 있습니다.

위의 실험은 라즈베리파이3를 이용한 실험으로 실시간으로 입력되는 소리의 노이즈를 제거하여 출력되는 것이 가능함을 보여줍니다. 우리의 시스템은 성능이 더 좋은 라즈베리파이5를 사용하므로 충분히 실시간으로 노이즈 제거가 가능함을 알 수 있습니다.

### 조건 3. 실시간 증폭 처리

일반적으로 사람이 오디오 딜레이를 느끼기 시작하는 임계값은 약 20ms(0.02초) ~ 30ms(0.03초) 정도입니다. 이 임계값 이상의 딜레이가 발생하면 사람들이 오디오의 시간적 불일치를 인식하기 시작할 수 있습니다. 따라서 내부적인 처리로 인한 오디오 딜레이가 20ms를 초과하는 경우에는 사용자가 딜레이를 느낄 수 있을 가능성이 높습니다. 해당 시스템에서 사용되는 알고리즘은 이 범위 내에서 실행되도록 구현하겠습니다.

음성 신호의 경우 8kHz에서 16kHz 사이의 샘플링 주파수를 사용하는 것이 일반적입니다. 높은 샘플링 주파수를 사용할수록 더 정확한 신호를 얻을 수 있지만 해당 시스템 구현에서 사용할 고감도 유선 다이내믹 마이크 경우 주파수가 60Hz에서 최대 14kHz이므로 8~16kHz 사이의 값인 14kHz 샘플링 주파수를 사용할 예정입니다.

## 1. 작품개요

이 작품은 노이즈 제거 기술을 적용하고 사운드 레벨 동기화 기술을 구현하여 입력되는 오디오 신호의 음량을 조정하는 장치를 개발하는 것을 목표로 합니다.

#### • 노이즈 제거 기술

- 증폭된 소리에 노이즈가 섞일 수 있으므로 노이즈를 제거하는 기능이 필요합니다.
- 이를 위해 디지털 신호 처리 기술을 사용하여 노이즈를 감지하고 제거할 수 있는 알고리즘이 필요합니다. 주로 사용되는 기술로는 주파수 영역에서 노이즈를 제거하는 FFT 기반의 알고리즘인 'noisereduce'가 있습니다.

#### • 사운드 레벨 동기화 기술

- 주로 사용되는 알고리즘으로는 RMS (Root Mean Square) 분석을 기반으로 한 볼륨 조절 알고리즘이 있습니다.
- 이 알고리즘은 각 소스에서 입력된 오디오의 RMS 값을 측정하여 신호의 강도를 평가하고, 이를 기반으로 볼륨 조절을 수행하여 일관된 음량을 유지합니다.
- 각각의 다른 소리 신호의 볼륨을 조정하여 사용자에게 일관되고 편안한 오디오 청취 경험을 제공하도록 합니다.

## 2. 주제 선정 과정

### 2.1) 문제인식

- 공연예술학과 친구와 식사를 하던 중...

무대를 구성할 때, 사람마다 성량이 다르기 때문에 각각의 **성량에 따라 사운드 레벨을 직접 조절**해야 한다고 한다.

현재 학과에서 사용하는 사운드 레벨 조정 방법은 **교수님이 무대 중앙에서** 배우들의 사운드 레벨을 듣고 “몇 번 배우 소리가 너무 크다”와 같은 피드백으로, **감에 의해 조절**하고 있다고 한다.

이로 인한 문제는, 다수의 배우의 사운드 레벨을 균일하게 맞춰야 하기 때문에 이를 조절하는 데에 **오랜 시간**이 필요하는 것이다. 또한 교수님의 **청력에 따라** 판단되기에 **정확도가 떨어진다**는 문제가 있다.



## 관련 사례(tvN '뽕뽕 지구오락실')

- tvN '뽕뽕 지구오락실' 시리즈를 맡았던 김선우 오디오 감독이 가수 이영지의 목소리 크기를 언급했습니다.
- 2023년 12월 6일 유튜브 '채널십오아' 채널에 나영석 PD와 김대주 작가, 김선우 오디오 감독이 함께한 라이브 영상의 편집본이 게재되었습니다.
- 나영석 PD가 이영지의 목소리 크기에 대한 질문을 하자 김선우 오디오 감독은 "목소리가 진짜 크다"라고 언급하였습니다.

네 분 중에서 상대적으로 유진 씨가 목소리가 좀 작다. 네 분 중에서는. 그래서 **유진 씨의 목소리에 맞게 음량을 좀 키워 놓으면 옆에서 얘기하는 영지씨 목소리가 엄청 크게 들어간다.** 그래서 현장이 넓어서 공간이 많으면 4개의 마이크를 운용하는데, **좁은 공간에서는 영지 씨 마이크를 아예 꺼버린다.**

'지락실' 오디오 감독, "이영지 목소리 진짜 커...마이크 아예 꺼버리기도" : 네이트 연예

한눈에 보는 오늘 : 연예가 화제 - 뉴스 : [톱스타뉴스 정은영]tvN '뽕뽕 지구오락실' 시리즈를 맡았던 김선우 오디오 감독이 가수 이영지의 목소리 크기를 언급했다. 지난 6일 유튜브 '채널십오아' 채널에는 나영석 PD와 김대주 작가, 김선우 오디오 감독이 함께한 라이브 영상의 편집본이 게재됐

<https://news.nate.com/view/20231208n08774>



## 자문 Q&A (음향 관련 일을 하고 있는 지인)

- 공연예술학과 음향감독

**Q1.** 보컬의 성량에 따른 음량 차이는 어떻게 조절하나요? 차이가 있을 시 감으로 조절하는 것인지, 기계로 밸런스를 측정해서 조절하는 것인지도 궁금합니다.

**A1.** 성량에 따른 데시벨이 나오는 기계도 있고, 단순히 두 귀로 듣고 차이가 있으면 콘솔을 사용해서 수동으로 조절합니다.

**Q2.** 현재 음향 조절은 어떻게 하고 계시나요?

**A2.** 저희는 X32라는 콘솔로 음량을 조절하고 있습니다. 이는 자동화가 아닌, 사람이 직접 조작해야 합니다.

**Q3.** 무대에서 발생하는 다수의 배우들 간 성량 차이는 어떻게 해결하나요?

**A3.** 제가 알기로는 마이크 내부에 음량을 조절하는 기능이 탑재되어 있는 것도 있다고 들었는데, 여러 배우의 음량을 맞추는 건 직접 해야 하는 부분입니다.

**Q4.** 저희의 구상 방향성이 필요한 기술인지 여쭙보고 싶습니다.

예를 들어, 10명의 사람이 마이크를 쓰고 있을 경우에 10명 각각의 음향을 조절하는 것이 아니라 음향 감독이 음량을 조절하기 전에 시스템 내부적으로 일정한 볼륨으로 설정해주는 것입니다.

즉, 사람 A의 소리가 사람 B의 소리보다 작을 경우 내부적으로 사람 A의 소리를 조금 크게 하고, 사람 B의 소리를 조금 작게 해서 모든 채널의 음량을 동일하게 맞추어 각각의 음량을 따로 조절하지 않아도 되도록 하는 것입니다.

**A4.** 인간이 듣고 판단하는 것이 아닌 기계가 정확한 데시벨을 측정하여 자동적으로 볼륨을 조절해주는 것은 너무 좋은 것 같습니다.

## 2.2) 가치

### 편리한 노이즈 제거 기능

증폭된 소리에 노이즈가 섞이는 문제를 해결하여 사용자들이 더욱 깨끗하고 정확한 오디오 청취를 할 수 있도록 하여 오디오 경험을 향상 시킵니다.

### 일관된 사운드 레벨

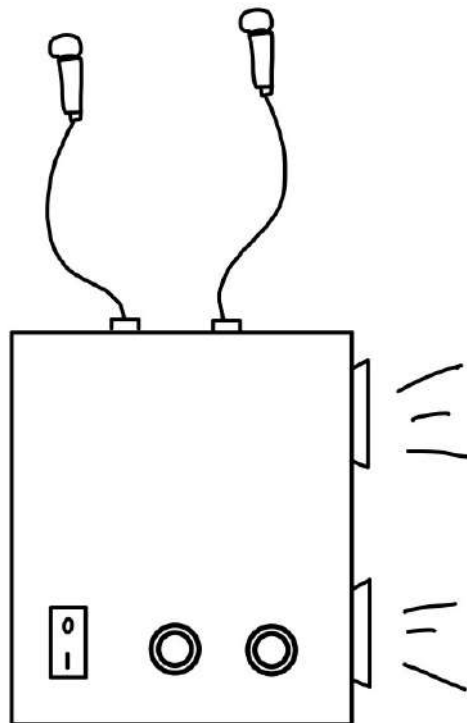
다양한 소스에서 입력되는 오디오 신호의 볼륨을 조정하여 일관된 사운드 레벨을 유지합니다. 이를 통해 사용자는 다른 소스 간의 음량 차이로 인한 불편함 없이 일관된 음향을 들을 수 있습니다.

### 휴대성과 간편한 조절

휴대용으로 제작된 콘솔은 사용자가 어디에서든지 손쉽게 사용할 수 있습니다. 또한 사용자 친화적인 인터페이스를 통해 노이즈 제거 및 사운드 레벨 조절을 간편하게 수행할 수 있습니다. 이는 이동 중이거나 다양한 장소에서의 오디오 작업을 수행하는 사용자에게 큰 편의를 제공합니다.

## 3. 구현 방법

### 3.1) 구현개요



#### 1. 마이크 입력 및 노이즈 제거:

- 마이크로부터 소리의 입력을 받고 ADC를 통해 디지털 소리로 변환 후 노이즈를 제거합니다.

#### 2. 신호 증폭 알고리즘 적용:

- 증폭할 크기를 제어할 파라미터를 설정합니다. 이때 파라미터 값은 사용자 인터페이스를 통해 조정할 수 있습니다. 각각의 마이크로 들어온 소리 신호를 ADC 모듈을 이용해 디지털 신호로 변환 후 각각 증폭합니다. 이때 증폭 정도는 두 소리 신호 크기가 동일해지도록 자동으로 각각 조절됩니다.

#### 3. 앰프 모듈 제어:

- 파라미터 값을 기반으로 실제 하드웨어인 앰프 모듈을 제어합니다.

#### 4. 앰프 모듈로 출력:

- 앰프 모듈을 통해 증폭된 오디오 신호를 출력 장치로 전달합니다. 이때 DAC 모듈이 내장되어 있는 앰프 모듈을 사용하여 별도 장치 없이 스피커로 출력할 수 있습니다.

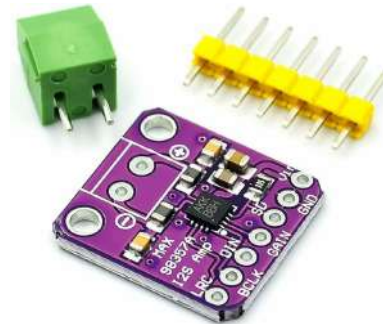
#### 하드웨어 구성:



- **마이크** : 아날로그 소리 신호를 입력 받습니다.



- **ADC** : 마이크를 통해 입력 받은 아날로그 신호를 노이즈 제거 및 앰프에 전달하기 위해 디지털 신호로 변환합니다.



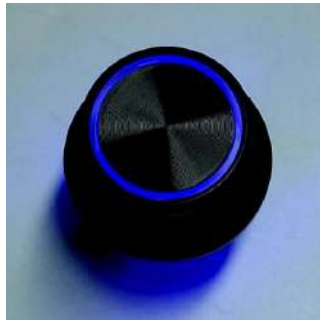
- **DAC가 내장된 앰프 모듈** : 소리 신호 증폭을 제어하고 증폭시킨 디지털 신호를 아날로그 신호로 변화합니다.



- **스피커** : 아날로그 신호로 변환된 소리를 스피커를 통해 출력합니다.



- **전원 스위치 및 충전모듈** : 휴대가 가능하도록 충전 모듈을 달아 따로 전원이 없어도 작동이 가능하도록 합니다. 전원 스위치를 추가하여 배터리가 빨리 방전되는 것을 예방합니다.



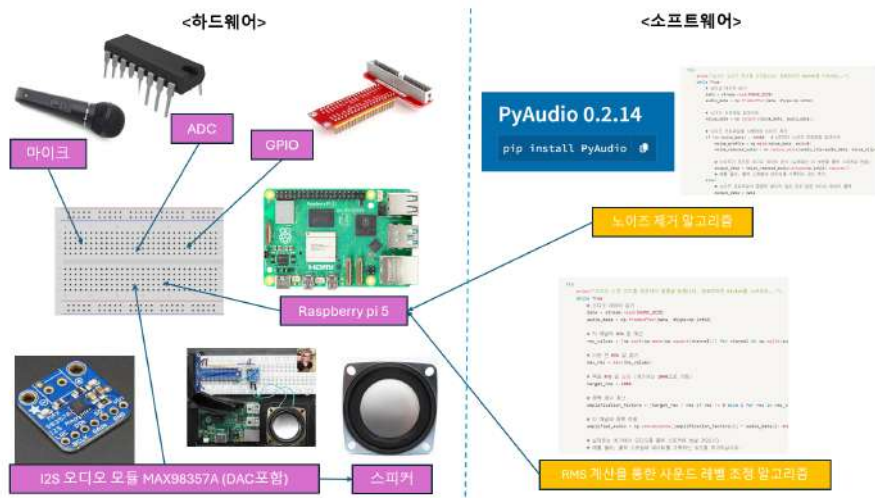
- **다이얼** : 사용자 인터페이스로 다이얼을 사용하여 사용자 지정 노이즈 강도를 지정합니다.

#### 소프트웨어 구현:

- **노이즈 파라미터 값 조정**: 사용자 인터페이스를 통해 전달 받은 사용자가 지정한 노이즈 강도를 통해 파라미터 값을 조절합니다. 해당 파라미터를 가지고 노이즈를 제거합니다. 노이즈를 제거한 각각의 디지털 신호를 각각의 앰프 모듈로 전달합니다.
- **앰프 모듈 제어**: 하드웨어 모듈을 제어하여 각각의 신호를 결과적으로 동일한 크기의 소리로 출력될 수 있게끔 증폭을 수행하는 알고리즘을 작성합니다.

### 3.2) 하드웨어

<구상도>



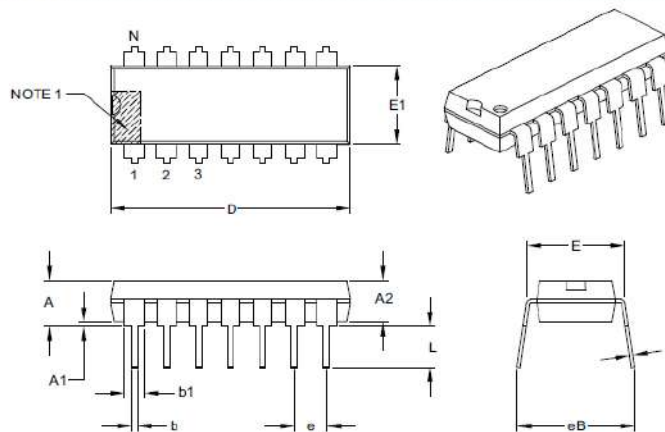
## 마이크 입력 장치

마이크로폰 또는 다른 오디오 입력 장치를 사용하여 소리를 감지하고 아날로그 신호를 생성합니다.

## ADC(Analog-to-Digital Converter):

라즈베리 파이에는 내장된 ADC가 없으므로 외부 ADC 모듈이 필요합니다. MCP3008과 같은 SPI(Serial Peripheral Interface)를 통해 통신하는 ADC 모듈을 선택할 수 있습니다.

### Dimensions



Units		INCHES		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N		14	
Pitch	e		.100 BSC	
Top to Seating Plane	A	—	—	.210
Molded Package Thickness	A2	.115	.130	.195
Base to Seating Plane	A1	.015	—	—
Shoulder to Shoulder Width	E	.290	.310	.325
Molded Package Width	E1	.240	.250	.280
Overall Length	D	.735	.750	.775
Tip to Seating Plane	L	.115	.130	.150
Lead Thickness	c	.008	.010	.015
Upper Lead Width	b1	.045	.060	.070
Lower Lead Width	b	.014	.018	.022
Overall Row Spacing	eB	—	—	.430

#### Notes:

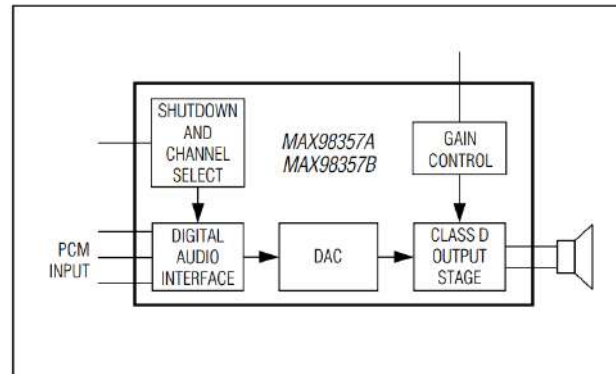
- Pin 1 visual index feature may vary, but must be located with the hatched area.
- § Significant Characteristic.
- Dimensions D and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed .010" per side.
- Dimensioning and tolerancing per ASME Y14.5M.

BSC: Basic Dimension. Theoretically exact value shown without tolerances.

## MAX98357A 모듈:

라즈베리 파이의 디지털 GPIO 핀을 사용하여 제어할 수 있습니다. 이 모듈은 DAC와 앰프를 통합하고 있으며, 디지털 입력 신호를 오디오 출력 신호로 증폭합니다.

### Simplified Block Diagram



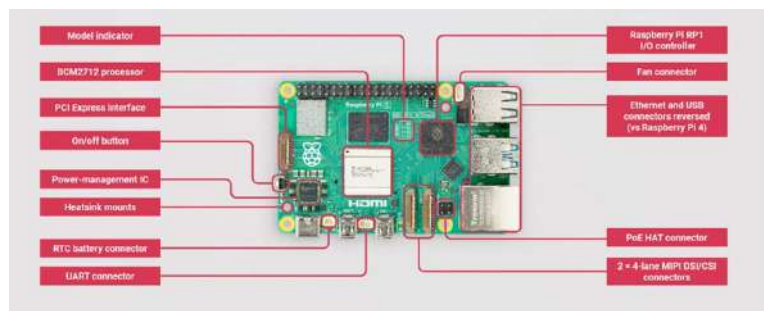
## <MAX98357A>



### Pin Description

PIN	WLP	TOFN	NAME	FUNCTION
A1	4		SD_MODE	Shutdown and Channel Select. Pull SD_MODE low to place the device in shutdown. In IFS or LJ mode, SD_MODE selects the data channel (Table 5). In TDM mode, SD_MODE and GAIN_SLOT are both used for channel selection (Table 7).
A2	7, 8		VDD	Power Supply Input
A3	9		OUTP	Positive Speaker Amplifier Output
B1	1		DIN	Digital Input Signal
B2	2		GAIN_SLOT	Gain and Channel Selection. In IFS and LJ mode determines amplifier output gain (Table 8) in TDM mode, used for channel selection with SD_MODE (Table 7). In TDM mode, gain is fixed at 12dB.
B3	10		OUTN	Negative Speaker Amplifier Output
C1	16		BCLK	Bit Clock Input
C2	3, 11, 15		GND	Ground
C3	14		LRCLK	Frame Clock. Left/right clock for IFS and LJ mode. Sync clock for TDM mode.
—	5, 6, 12, 13		N.C.	No Connection
—	—		EP	Exposed Pad. The exposed pad is not internally connected. Connect the exposed pad to a solid ground plane for thermal dissipation.

## <라즈베리 파이와 각 모듈 연결 방법>



- 마이크를 ADC 모듈의 입력에 연결합니다.
- ADC 모듈을 라즈베리 파이의 SPI 인터페이스에 연결합니다. 이때 SPI 핀은 라즈베리 파이의 GPIO에 연결됩니다.
- MAX98357A 모듈을 라즈베리 파이의 GPIO 핀에 연결합니다. 이때 MAX98357A 모듈의 DIN 핀은 라즈베리 파이의 MOSI(Master Out Slave In) 핀에 연결될 수 있습니다.
- MAX98357A 모듈의 출력을 스피커에 연결합니다.

## 3.3) 소프트웨어

### 3.3.1) 노이즈 제거

#### 실시간 노이즈 제거 알고리즘

1. 파이썬과 PyAudio를 사용하여 실시간으로 마이크로부터 오디오를 읽고, 노이즈를 제거하는 알고리즘을 작성합니다.
2. PyAudio를 사용하여 실시간으로 오디오를 읽어들이고, NoiseReduce 라이브러리를 사용하여 노이즈를 제거합니다. 코드를 실행하면 마이크로부터 들어오는 오디오가 노이즈 제거되어 출력됩니다.
3. 사용자 인터페이스를 통해 파라미터를 조정하여 노이즈를 얼마나 제거할 것인지 사용자가 필요에 맞게 조정할 수 있습니다.
4. 노이즈를 제거한 각 신호를 애플 모듈로 전달합니다.

- **NumPy 라이브러리:** 다차원 배열과 행렬을 처리하기 위한 파이썬 라이브러리입니다. 이 코드에서는 오디오 데이터를 다룰 때 NumPy 배열을 사용합니다.
- **PyAudio 라이브러리:** 오디오 스트림을 처리하기 위한 라이브러리입니다. 이를 통해 오디오 입력을 받아들이고 오디오 출력을 생성할 수 있습니다.

#### Example: Callback Mode Audio I/O

```
"""PyAudio Example: Play a wave file (callback version)."""

import wave
import time
import sys

import pyaudio

if len(sys.argv) < 2:
    print('! Plays a wave file. Usage: {sys.argv[0]} filename.wav')
    sys.exit(-1)

with wave.open(sys.argv[1], 'rb') as wf:
    # Define callback for playback (1)
    def callback(in_data, frame_count, time_info, status):
        data = wf.readframes(frame_count)
        # If len(data) is less than requested frame_count, PyAudio automatically
        # assumes the stream is finished, and the stream stops.
        return (data, pyaudio.paContinue)

    # Instantiate PyAudio and initialize PortAudio system resources (2)
    p = pyaudio.PyAudio()

    # Open stream using callback (3)
    stream = p.open(format=p.get_format_from_width(wf.getsampwidth()),
                    channels=wf.getnchannels(),
                    rate=wf.getframerate(),
                    output=True,
                    stream_callback=callback)

    # Wait for stream to finish (4)
    while stream.is_active():
        time.sleep(0.1)

    # Close the stream (5)
    stream.close()

    # Release PortAudio system resources (6)
    p.terminate()
```

참고: PyAudio Documentation

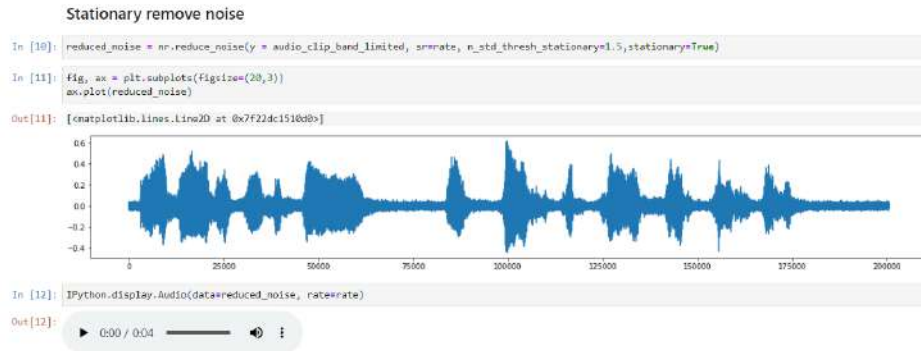
#### PyAudio

Cross-platform audio I/O with PortAudio

 <https://pypi.org/project/PyAudio/>



- **noisereducer 라이브러리**: 주어진 오디오 클립에서 노이즈를 감소시키는 기능을 제공하는 라이브러리입니다. 이를 사용하여 노이즈가 포함된 오디오 입력에서 노이즈를 제거합니다.

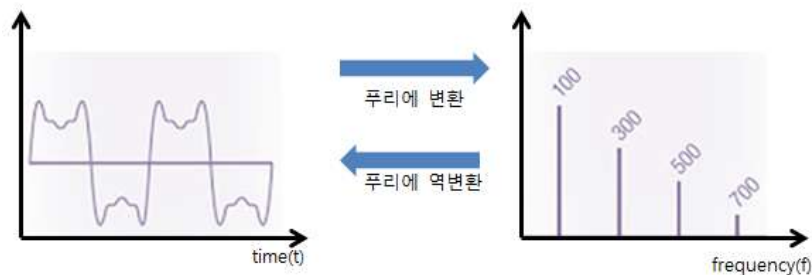


참고: noisereducer Documentation

noisereducer 라이브러리는 주로 **푸리에 변환**을 활용하여 주파수 도메인에서의 노이즈를 감소시키는 기능을 제공합니다.

1. **푸리에 변환(Fourier Transform)**: noisereducer는 입력 오디오 신호를 주파수 도메인으로 변환하기 위해 푸리에 변환을 사용합니다. 이를 통해 주파수 영역에서 신호와 노이즈를 분리할 수 있습니다.

$$Y(f) = \int_{-\infty}^{\infty} y(t)e^{-i2\pi ft} dt \quad : \text{푸리에 변환}$$



→ 푸리에 변환을 이용해서, 어떤 신호가 시간에 대한 함수(시간별로 그 파형의 크기가 주어지는)로 주어 졌을 때, 그 신호를 주파수에 대한 함수(각 주파수별 크기를 나타내는)로 표현할 수 있습니다.

2. **노이즈 프로파일 추출(Noise Profile Extraction)**: 푸리에 변환을 통해 얻은 주파수 도메인의 신호를 분석하여 노이즈 프로파일을 추출합니다. 이는 노이즈가 포함된 신호의 주파수 성분을 식별하는 데 사용됩니다.
3. **노이즈 제거(Noise Reduction)**: 추출된 노이즈 프로파일을 사용하여 주파수 도메인에서 노이즈를 감소시킵니다. 이를 통해 노이즈가 제거된 신호를 얻을 수 있습니다.
4. **신호 재구성(Signal Reconstruction)**: 주파수 도메인에서 노이즈가 제거된 신호를 시간 도메인으로 다시 변환하여 최종적인 결과를 얻습니다. 이를 통해 오디오 신호의 품질이 향상됩니다.

noisereducer  
Noise reduction using Spectral Gating in Python

<https://pypi.org/project/noisereducer/>



<https://github.com/timsainb/noisereducer>



### <노이즈를 제거한 후에 신호를 증폭하는 것이 좋은가?>

1. **노이즈 증폭 방지:** 노이즈가 이미 증폭된 경우, 노이즈는 원래보다 더 크게 증폭될 수 있습니다. 이는 신호 대비 노이즈 비율이 더 높아지고, 노이즈 제거 알고리즘의 성능을 저하시킬 수 있습니다.
2. **신호 왜곡 방지:** 입력 신호를 노이즈 제거하기 전에 증폭하면, 신호와 노이즈가 모두 증폭됩니다. 이는 노이즈 제거 알고리즘이 신호를 왜곡시킬 가능성이 있습니다. 노이즈를 제거한 후에 증폭하면, 신호만 증폭되고 노이즈는 제거됩니다.

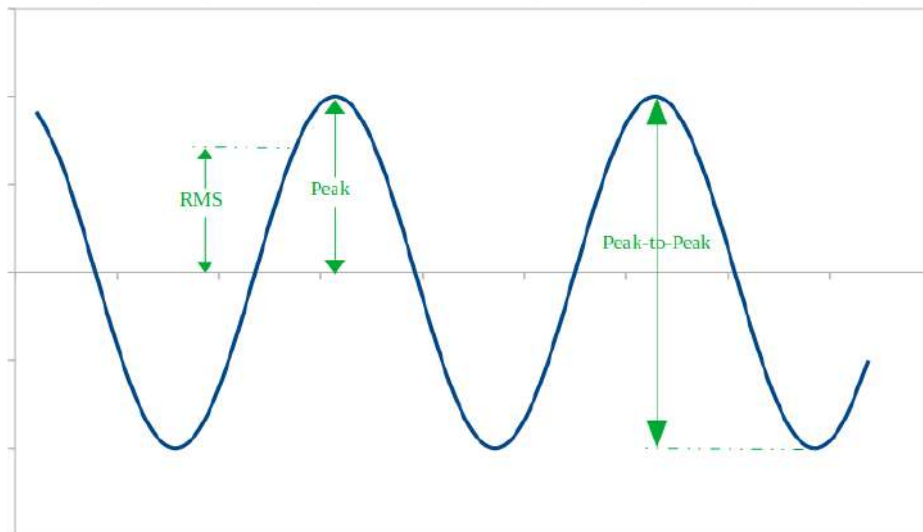
### 3.3.2) 사운드 레벨 동기화

각각의 서로 다른 크기의 신호를 일정한 볼륨 수준으로 맞추기 위해서는 각 신호의 크기를 측정하고 앰프를 통해 적절한 증폭을 수행해야 합니다.

아래는 이를 구현하기 위한 간단한 절차입니다.

- 약 300ms(일반적으로 RMS는 300ms 동안의 오디오 트랙 평균 음량을 측정) 시간 동안 입력된 각신호의 Root Mean Square (RMS) 값을 계산합니다. 초반 신호가 입력되는 시간동안 계산된 각 신호의 RMS값을 기반으로 평균 크기의 신호를 계산합니다. 이 평균 값을 기준으로 각 신호의 증폭도를 계산하여 적용합니다. (RMS를 계산하는 시간과 고정 기준을 정확하게 정하기 위해서는 직접 코드의 파라미터 변환을 통해 실험하여 가장 정확도가 높은 시간 범위를 찾도록 합니다.)
- 두 신호의 크기가 모두 작은 경우 증폭도를 평균값에 맞추어 증폭시킬 경우 소리의 전반적인 소리가 너무 작아지는 문제가 발생할 수 있습니다. 이를 막기 위해 증폭도를 조절하는 파라미터의 최저 값을 설정합니다.

### <RMS 레벨>



Root Mean Square(RMS) 레벨은 음량의 평균적인 값을 보여주는 지표입니다. 음량의 평균적인 값을 보여주기 때문에, 두 노래 간의 실질적인 음압 차이를 가늠할때 사용하기 편리합니다.

1. 오디오 신호의 RMS 값은 다음과 같이 계산됩니다.

$$RMS = \sqrt{\frac{1}{N} \sum_{i=1}^N x_i^2}$$

2. 신호 크기 비교 및 적절한 증폭 계산:

- 각 신호의 RMS 값을 비교하여 보정할 증폭 계수를 계산합니다.

- 예를 들어, 두 개의 마이크로부터 들어오는 신호의 RMS 값이  $RMS_1$ 과  $RMS_2$ 일 때, 중간값인  $RMS_{avg}$ 를 계산합니다.

$$RMS_{avg} = \frac{RMS_1 + RMS_2}{2}$$

- 각 신호의 증폭 계수를 계산합니다. 보정된 RMS 값이 중간값인  $RMS_{avg}$ 와 같아지도록 계수를 조절합니다.
- pydub 라이브러리:** 오디오 파일의 형식 변환, 볼륨 조절, 오디오 분할 및 결합, 오디오 파일에 대한 페이드 인/아웃 효과 적용 같은 다양한 오디오 처리 작업을 간단하게 수행할 수 있습니다.
  - audio.rms** 메서드는 오디오 파일의 RMS 레벨을 계산하여 반환합니다. 반환값은 해당 오디오의 RMS 레벨을 나타내는 정수입니다. 큰 값일수록 오디오 신호의 강도가 높다는 의미입니다. 이 값을 출력하면 해당 오디오 파일의 RMS 레벨을 확인할 수 있습니다.

```
from pydub import AudioSegment

# 오디오 파일 로드
audio= AudioSegment.from_file("audio.wav")

# RMS 레벨 계산
rms= audio.rms

# 결과 출력
print("RMS 레벨:", rms)
```

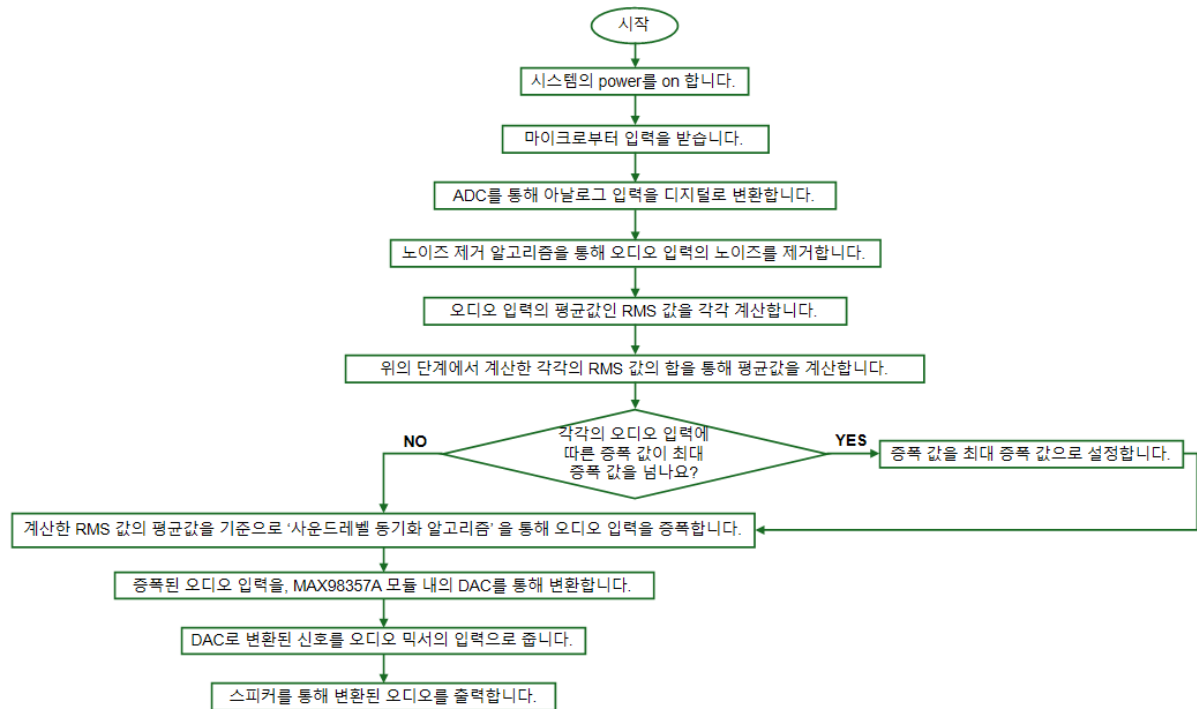
pydub  
Manipulate audio with an simple and easy high level interface

 <https://pypi.org/project/pydub/>



<https://github.com/jiaaro/pydub>

## [FLOW CHART]



## 4. 유사제품 비교

### 1. Behringer의 X32 디지털 믹서:



- 유사성:

- Behringer의 X32 디지털 믹서는 프로 품질의 음향 조절에 사용됩니다. 다양한 입력 소스를 조절하고 믹싱하여 최종 출력을 제어할 수 있습니다.
- **차별성:**
  - Behringer의 X32는 주로 전체 오디오 믹싱과 관련된 기능을 제공하는 반면, 목소리 크기의 균일화와 같은 개별적인 음향 처리에 대한 기능은 강조하지 않습니다.
  - 사용자가 수동으로 입력 소스의 볼륨을 조절하고 사운드 이펙트를 조절하는 것이 가능하나, 목소리 크기에 따라 자동으로 조절되지는 않습니다.

## 2. Samsung Sound Assistant:



- **유사성:**
  - 삼성 사운드 어시스턴트는 다양한 앱이나 시스템을 사용할 때 발생하는 소리 크기의 불균형을 조절하기 위해 만들어진 앱입니다. 사용자가 각 앱의 출력 소리를 직접 조절하고 메인 스피커를 조절할 수 있는 Samsung Sound Assistant와 유사하게, 각 사용자의 목소리를 균일하게 조절하는 시스템은 사용자에게 높은 편의성을 제공할 수 있습니다.
- **차별성:**
  - 자동으로 각각의 앱 및 시스템의 사운드 레벨을 맞춰주는 기능이 없습니다. 사용자가 직접 각각의 앱에서 출력되는 소리 크기를 설정한 후 메인 스피커를 조절하는 방식입니다.
  - 해당 앱은 모바일 폰에서 사용되는 것으로 공연장이나 인터뷰 현장, 회의 같은 오프라인 환경에서의 아날로그 소리 신호를 입력 받을 수 없습니다.

## 5. 팀원 간 업무 내용

팀원	업무
김지윤(팀장)	<ul style="list-style-type: none"> <li>—하드웨어 조립</li> <li>—하드웨어 동작 테스트</li> </ul>
	<ul style="list-style-type: none"> <li>—라즈베리파이 셋팅</li> <li>—사운드레벨 동기화 알고리즘 구현</li> <li>—사운드레벨 동기화 알고리즘 테스트</li> </ul>

팀원	업무
박수진	<ul style="list-style-type: none"> <li>—하드웨어 조립</li> <li>—하드웨어 동작 테스트</li> <li>—라즈베리파이 셋팅</li> <li>—사운드레벨 동기화 알고리즘 구현</li> <li>—사운드레벨 동기화 알고리즘 테스트</li> </ul>
정지원	<ul style="list-style-type: none"> <li>—외관 모델링 3D 프린팅</li> <li>—외관 조립</li> <li>—노이즈 제거 알고리즘 구현</li> <li>—노이즈 제거 알고리즘 테스트</li> <li>—노이즈 제거 알고리즘 오류 수정</li> </ul>
이다은	<ul style="list-style-type: none"> <li>—외관 모델링 3D 프린팅</li> <li>—외관 조립</li> <li>—노이즈 제거 알고리즘 구현</li> <li>—노이즈 제거 알고리즘 테스트</li> <li>—노이즈 제거 알고리즘 오류 수정</li> </ul>
공동작업	<ul style="list-style-type: none"> <li>—사운드레벨 동기화 알고리즘 오류 수정</li> <li>—소프트웨어 적용 테스트</li> <li>—소프트웨어 적용 오류 수정</li> </ul>

## 6. 작품 제작 추진 계획 및 일정표

분류	업무내용	3월	4월				5월			
		4	1	2	3	4	1	2	3	4
하드웨어	하드웨어 조립									
	하드웨어 동작 테스트									
	소프트웨어 적용 테스트									
	소프트웨어 적용 오류 수정									
소프트웨어	라즈베리 파이 셋팅									
	노이즈 제거 알고리즘 구현									
	노이즈 제거 알고리즘 테스트									
	노이즈 제거 알고리즘 오류 수정									
	사운드레벨 동기화 알고리즘 구현									
	사운드레벨 동기화 알고리즘 테스트									
	사운드레벨 동기화 알고리즘 오류 수정									
외관 제작	외관 모델링 및 3D프린팅									
	외관 조립									

## 7. 지원 경비 사용 계획

모델명	단가	수량	총액	링크
라즈베리파이5	148,170	1	148,170	제공되는 보드 사용
라즈베리파이 40Pin GPIO 확장 보드 T-Cobbler [SMP0053]	6,000	1	6,000	<a href="https://www.devicemart.co.kr/goods/view?no=1327563">https://www.devicemart.co.kr/goods/view?no=1327563</a>
[Adafruit] Adafruit I2S 3W Class D Amplifier Breakout - MAX98357A [ada-3006]	11,000	2	22,000	<a href="https://www.devicemart.co.kr/goods/view?no=12496231">https://www.devicemart.co.kr/goods/view?no=12496231</a>

모델명	단가	수량	총액	링크
[Coms] 고감도 유선 다이내믹 마이크 / Metal / 3.5mm / 케이블 3M [BB070]	14,000	2	24,000	<a href="https://www.devicemart.co.kr/goods/view?no=12741240">https://www.devicemart.co.kr/goods/view?no=12741240</a>
[Microchip Technology] MCP3008-I/P	4,200	2	8,400	<a href="https://www.devicemart.co.kr/goods/view?no=1322885">https://www.devicemart.co.kr/goods/view?no=1322885</a>
LED 조그다이얼 타입 포텐셔미터/엔코더 노브 30mm 블랙	2,000	2	4,000	<a href="https://www.devicemart.co.kr/goods/view?no=13538750">https://www.devicemart.co.kr/goods/view?no=13538750</a>
[Coms] 4채널 오디오 믹서(블루투스) / 믹싱콘솔 / 유튜브 방송전용 [EP723]	50,100	1	50,100	<a href="https://www.devicemart.co.kr/goods/view?no=12741210">https://www.devicemart.co.kr/goods/view?no=12741210</a>
			262,470	