

# GLMv4.exe Decompiler

## 개발료해진행보고서

2023. 6. 21

郑光明

## GLMv4.exe Decompiler

No	수정날자	수정내용	비고
1	2023.4.20	개발방향확정	100%
2	2023.5.1 ~ 6.10	개발과정진도(1 단계)	100%
3	2023.6.11 ~ 6.21	개발과정진도(2 단계)	65%
4	2023.7.30	개발과정진도(3 단계)	
5	2023.8.30	개발과정진도(4 단계)	

# GLMv4.exe Decompiler

## 目 录

目 录 .....	3
1. GLMV4.exe 소개 .....	7
-GML 장치들사이 통신 .....	11
2. IDA 를 리용한 GLMV4.exe 분석 .....	13
3. 프로그램 Decompiler 과정에 필요한 지식과 도구들 .....	14
4. JUCE 프레임워크를 리용한 코드분석진행 .....	15
5. 참고문헌 .....	21

## GLMv4.exe Decompiler

프로그램을 개발하는것보다 다른사람이 개발한 프로그램으로부터 원천코드를 얻는과정은 몇배로 품이드는 작업이다. 분석하려는 실행파일은 VC++언어로 작성된 상위응용프로그램(64 비트용)이다. 개발도구는 Visual Studio 를 리용하였으며 JUCE 프레임워크를 리용하였다.

JUCE(trial 판 7.0.5)는 C++언어 기반의 Audio 관련기능이 특화된 프레임워크이다. GUI, 네트워크, 자료구조 등의 기본적인 기능으로부터 오디오플러그인 포맷 ,오디오파일입출력에 이르기까지 Audio 관련된 기능을 총 망라해서 제공하고 있다는것이 특징이다.

JUCE 는 DAW 에서 사용되는 플러그인들을 만들기 위한 프레임워크이다. JUCE 는 실제 개발 할때에는 유료판본을 계정에 등록하고 받아야 한다.

<https://juce.com/stories/>

### -DAW

일반적으로 음악가들은 음악을 만들때 DAW(Digital Audio Workspace)를 거의 필수적으로 사용한다. 사진 보정같은 작업을 할 때는 포토샵과 같은 이미지 편집 툴을 사용하듯이 음악 작업은 음악 편집 툴인 DAW를 사용한다. 이러한 DAW에는 Logic, Cubase, StudioOne, Ableton Live 등등 여러 소프트웨어가 존재한다.

### -Plug-in

DAW 는 음악 편집 툴이지만 음악가들은 DAW 소프트웨어 하나만을 사용하여 곡을 만들지 않는다. 비유를 하자면 DAW 는 음악 작업을 위한 기본적인 시설을 갖춘 음악 녹음실과 같다. 그러나 좀 더 완성도 있는 곡을 만들기 위해서는 이에 필요한 더 좋은 피아노, 기타와 같은 악기를 별도로 구성해야 하므로 DAW 역시 악기 소프트웨어나 여러 효과를 줄수 있는 이펙터 등의 소프트웨어를 불러와 추가로 사용할 수 있다. 이렇게 DAW 에서 불러와 사용할 수 있는 소프트웨어 툴 들을 플러그인(Plug-in) 이라고 한다.

그런데 이러한 Plug-in 들을 만드는 회사는 한 두개가 아니다. 다양한 회사에서 만들어진 Plug-in 들을 다양한 DAW 에서 구동시키려면 서로 간의 약속(=표준)이 필요하며 이러한 표준 들에는 VST, AU 등이 있다.

VST(Steinberg's Virtual Studio Technology)는 전자 음악 편집 소프트웨어 및 레코딩 시스템과 신시사이저 및 이펙트를 이어주는 플러그인 형식의 표준 규격이다.

오디오유닛(Audio Unit, AU)은 OS X 의 코어 오디오에서 구현되는, 애플 컴퓨터에서 개발한 오디오 플러그인 기술 및 규격이다.

## GLMv4.exe Decompiler

JUCE 는 크로스플랫폼이기 때문에 똑같은 코드를 가지고 맥, 리눅스, 윈도우 용 프로그램을 빌드할 수 있다.

프로그램의 주요기능은 음성자료처리교정과 관련된 폭넓은 기능을 가진 관리프로그램이다.

아래의 그림은 프로그램의 설치된 폴더안의 내용을 보여준다.

Name	Date modified	Type	Size
ConsoleApplication1	4/18/2023 11:27 AM	File folder	
concrct140.dll	7/12/2021 5:00 PM	Application exten...	309 KB
GLM4Manual.pdf	5/31/2022 9:05 AM	WPS PDF 文档	3,824 KB
GLMv4.exe	10/28/2022 10:10 AM	Application	15,404 KB
GLMv4.exe.c	4/18/2023 11:20 AM	C Source	28,973 KB
GLMv4.exe.i64	4/18/2023 6:31 PM	I64 File	262,918 KB
hidapi.dll	11/19/2020 11:29 AM	Application exten...	18 KB
msvcp140.dll	7/12/2021 5:00 PM	Application exten...	553 KB
msvcp140_1.dll	7/12/2021 5:00 PM	Application exten...	24 KB
msvcp140_2.dll	7/12/2021 5:00 PM	Application exten...	182 KB
unins000.dat	4/17/2023 2:43 PM	DAT File	6 KB
unins000.exe	4/17/2023 2:43 PM	Application	2,521 KB
unins000.msg	4/17/2023 2:43 PM	MSG File	23 KB
vccorlib140.dll	7/12/2021 5:00 PM	Application exten...	327 KB
vcruntime140.dll	7/12/2021 5:00 PM	Application exten...	95 KB
vcruntime140_1.dll	7/12/2021 5:00 PM	Application exten...	37 KB

그림 1. 프로그램의 설치폴더내용

exe 파일을 Decompile 하기 위해서는 IDA 7.5 판본을 리용하고 있다.

## 시스템 구성요소

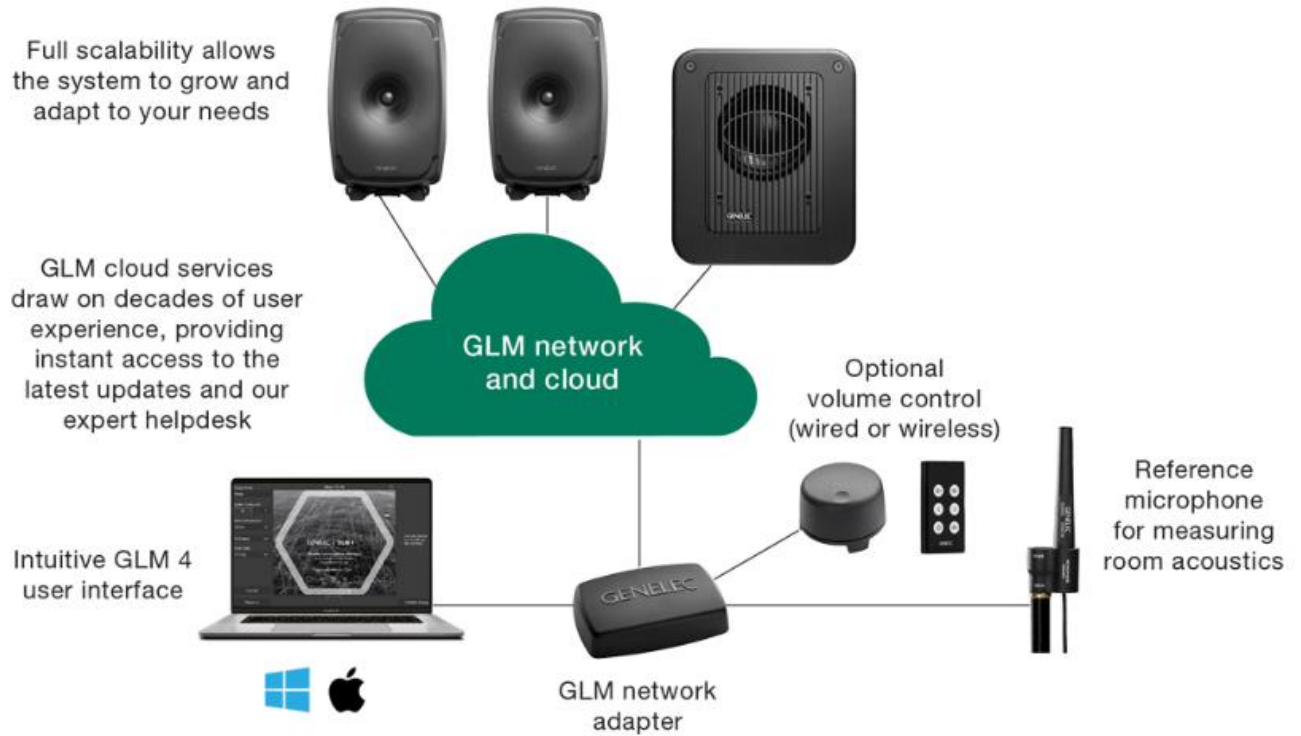


그림 3. 체계구성

## 1. GLMV4.exe 소개

프로그램의 동작과정에 대해서는 GLMV4 Manual(98 페이지)을 참고하기 바란다.

※ 모든 교정알고리즘은 클라우드서버에서 실행된다. 따라서 클라우드계정을 창조해야 한다.

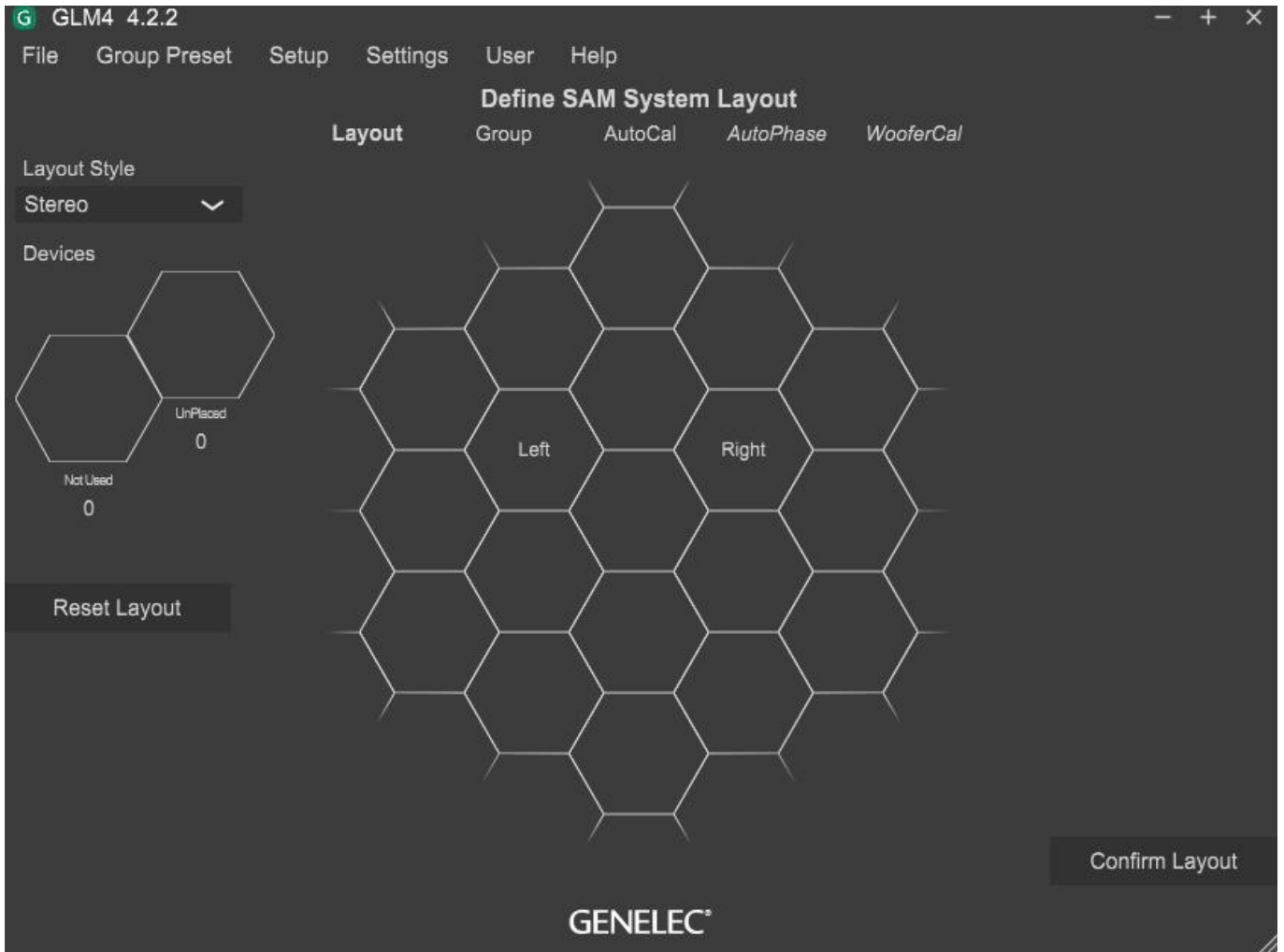


그림 3. 프로그램기본화면-1

## GLMv4.exe Decompiler

GLM MIDI Settings

☐ Enable GLM MIDI interface

MIDI Input Device

MIDI Output Device

MIDI Channel

1 - +

MIDI Command Log

GLM Function	CC	Type
<input checked="" type="checkbox"/> Volume	20 - +	Value (0-127)
<input checked="" type="checkbox"/> Volume+	21 - +	Momentary
<input checked="" type="checkbox"/> Volume-	22 - +	Momentary
<input checked="" type="checkbox"/> Mute	23 - +	Momentary
<input checked="" type="checkbox"/> Dim	24 - +	Momentary
<input checked="" type="checkbox"/> Preset Level1	25 - +	Momentary
<input checked="" type="checkbox"/> Preset Level2	26 - +	Momentary
<input checked="" type="checkbox"/> BM Bypass	27 - +	Momentary

Volume Step Size

0.5 - +

GLM Function	CC	Type
<input checked="" type="checkbox"/> Group1	31 - +	Momentary
<input checked="" type="checkbox"/> Group2	32 - +	Momentary
<input checked="" type="checkbox"/> Group3	33 - +	Momentary
<input checked="" type="checkbox"/> Group4	34 - +	Momentary
<input checked="" type="checkbox"/> Group5	35 - +	Momentary
<input checked="" type="checkbox"/> Group6	36 - +	Momentary
<input checked="" type="checkbox"/> Group7	37 - +	Momentary
<input checked="" type="checkbox"/> Group8	38 - +	Momentary
<input checked="" type="checkbox"/> Group9	39 - +	Momentary
<input checked="" type="checkbox"/> Group10	40 - +	Momentary
<input checked="" type="checkbox"/> Groupx	30 - +	Value (1-10)

Cancel

Confirm

그림 4. 프로그램기본화면-2



## GLMv4.exe Decompiler

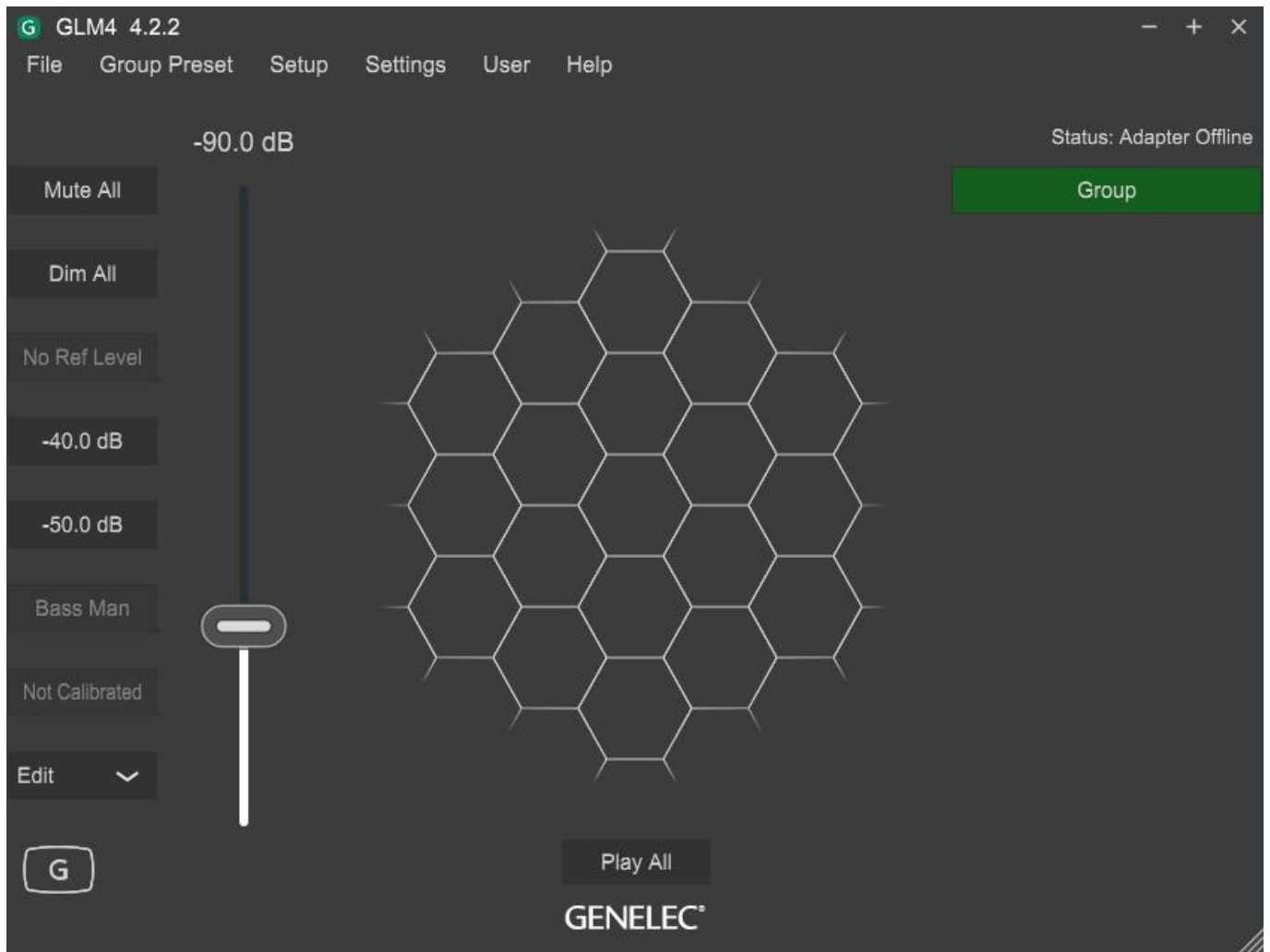


그림 5. 프로그램기본화면-3

Genelec 는 핀란드의 스피커 제조사. 1978 년 Ilpo Martikainen 에 의해 설립된 세계적인 모니터링 스피커 제조사로, 사명 GENELEC 는 천재라는 뜻의 Genius 와 전자기기이라는 뜻의 Electronics 의 합성어이다.

줄여서 GLM. 제네렉에서 2006 년 처음 선보인 스피커 보정 프로그램이다. SAM(Smart Active Monitoring)을 지원하는 제네렉의 스마트 스피커 라인업에서 지원된다. 스피커에서 발생하는 소리는 스피커가 배치된 공간과 상호작용하여 왜곡되기 마련인데, 이를 자체 알고리즘을 통해 정교하게 보정한다. 특히 공간의 영향을 매우 크게 받는 중저음 대역도 위상왜곡없이 보정해주고, 스피커를 세 대 이상 사용하는 멀티채널 구성도 간편하게 보정할 수 있어서 수동으로 보정해주는 것보다 훨씬 편리하다.

## GLMv4.exe Decompiler

출시된 이래로 꾸준히 이어져오는 지원 및 개발로 업계에서 신뢰성을 인정받고 있다. 제네렉 제품의 특징점 중 하나.

2006 NAMM TEC 에서 제네렉의 GLM 지원 스피커가 우승한 바 있다.

### 네트워킹 준비

GLM 소프트웨어를 스마트 액티브 모니터와 통합하려면 GLM 네트워크 어댑터 , 8300A 보정 마이크 및 마이크 홀더, 1.8m USB 케이블 및 빠른 연결 가이드. 네트워크 어댑터는 제공된 USB 케이블을 통해 Mac 또는 PC 에 연결하고 5 미터 CAT 케이블(각 모니터와 별도로 제공됨)을 통해 모니터에 연결한다.

GLM 4 는 '클래식' Genelec 아날로그 모니터를 제어할 수 없다. 표준 활성화 모델을 사용하는 경우 실내에 배치한 후 중립 주파수 응답을 위해 후면 패널 DIP 스위치를 조정해야 한다. 자세한 내용은 모든 사용 설명서에서 확인할 수 있다.

### GLM 4 의 보정 기능을 사용하여

- 실내 주파수 응답, 거리 지연 및 청취 레벨을 자동으로 조정한다.
- 자동 설정을 자유롭게 조정, 개인화 및 저장하십시오.
- 모든 유형의 스테레오, 다중 채널 또는 3D 몰입형 오디오 형식 에 맞게 시스템을 만들고 확장한다 .
- 무제한의 마이크 위치에서 샘플 룸 응답.
- 모든 생산 환경에서 일관된 성능과 정확한 모니터링의 이점을 누리십시오.

### GLM 4 의 모니터 제어 기능을 사용하여

- 동일한 DAW 출력을 사용하여 여러 모니터링 시스템 간에 전환한다.
- 솔로, 음소거 및 표준 보정 청취 레벨을 활성화한다.
- 저음 관리 또는 전체 시스템 지연 오프셋을 사용하십시오.
- 개별 모니터 시간 오프셋을 포함하여 청취 위치 간에 전환한다.
- 80 개 이상의 Genelec 모니터 및 서브우퍼를 관리하고 제어한다.

### GLM 4 의 GRADE 보고서 기능을 사용하여

- ITU-R BS.1116 권장 사항을 포함하여 모범 사례에 따라 룸 및 모니터링 시스템 성능에 대한 완전한 분석을 생성한다.
- 주파수 응답 정확도, 청취 위치에서의 이른 소리와 늦은 소리, 비행 시간 정확도 및 초기 반사 구조를 포함하여 심층적인 실내 및 시스템 측정을 참조하십시오.
- 어쿠스틱 처리, 시스템 포지셔닝 및 저음 관리 개선을 위한 의사 결정을 안내한다.

## GLMv4.exe Decompiler

- GRADE의 접근 가능하고 직관적인 음향 조안을 기반으로 청취 설정을 간단하게 개선한다.

15년 이상 동안 GLM 소프트웨어는 Smart Active Monitors와 함께 작업하여 방의 원치 않는 음향 영향을 최소화하고 어디에서나 믹스가 훌륭하게 들리도록 도와준다. 다른 새로운 기능 중에서 GLM 4.2는 강력한 새로운 GRADE 보고서를 도입하여 GLM의 범위를 초정밀 시스템 보정을 넘어 심층적인 실내 및 시스템 분석으로 확장한다.



### 번역 중 손실?

자신의 믹스가 다른 시스템으로 잘 변환되지 않아 좌절한 적이 있다면 모니터와 실내가 상호 작용하여 일관되게 들리는 훌륭한 믹스를 만드는 데 필요한 중립적이고 부드러운 청취 경험과는 거리가 먼 주파수 응답을 생성하기 때문입니다. 재생 위치에 관계없이 균형을 이룹니다.

우아한 사용자 인터페이스와 Mac 및 Windows와 호환되는 유용한 기능 호스트를 통해 GLM 4는 타의 추종을 불허하는 진실된 사운드 레퍼런스를 생성하여 짧은 시간에 더 나은 믹스를 만들 수 있도록 도와줍니다.

GLM 4 소프트웨어는 Genelec 스마트 액티브 모니터 및 서브우퍼의 성장하는 제품군에 장착된 고급 내부 DSP 회로와 긴밀하게 통합됩니다. GLM 4의 레퍼런스 마이크를 사용하면 음향 환경을 분석할 수 있으며, 그 후 GLM 4.2의 AutoCal 2 기능은 레벨, 거리 지연, 서브우퍼 크로스오버 위상 및 룸 응답 이퀄라이제이션에 대해 각 모니터를 신속하게 최적화하며, 옵션을 통해 마이크를 더 미세하게 조정할 수 있습니다. 맛보는 시스템.

GLM 4.2는 거대한 물입형 모니터링 시스템, 소형 시스템 및 울트라 니어필드 시스템을 모두 보정합니다. 새로운 Calibration Group ID 및 Subwoofer Group ID 기능은 대형 그룹으로 모니터의 보정을 더욱 개선하고 여러 서브우퍼가 매우 유연한 배열로 함께 작동할 수 있도록 합니다. 따라서 이미 우수한 방에서 청취 경험을 향상시키려는 경우나 더 까다롭고 생생한 사운드 공간에서 작업하는 경우 GLM 4.2는 스테레오에서 지금까지 가장 진보되고 사용자 친화적인 시스템 관리 및 확장성을 제공하는 친구입니다. 물입.

GRADE 보고서의 도입으로 실내 개선 작업을 직접 수행하거나 전문 음향 엔지니어에게 세부 측정 전달하려는 사용자는 이제 GLM 4를 사용하여 실내 및 모니터링 시스템 성능에 대한 완전한 분석을 생성할 수 있습니다. 양문 제안으로 2023년 7월 1일까지 무료로 제공되는 이 보고서는 모든 음향 문제에 대한 구체적인 관찰과 조언을 제공하여 실내 음향 처리를 미세 조정하고 모니터와 청취자 위치를 조정하며 저음 관리를 최적화하는 데 도움이 됩니다. ITU-R BS.1116 권장 사항을 준수하는지 확인하십시오.

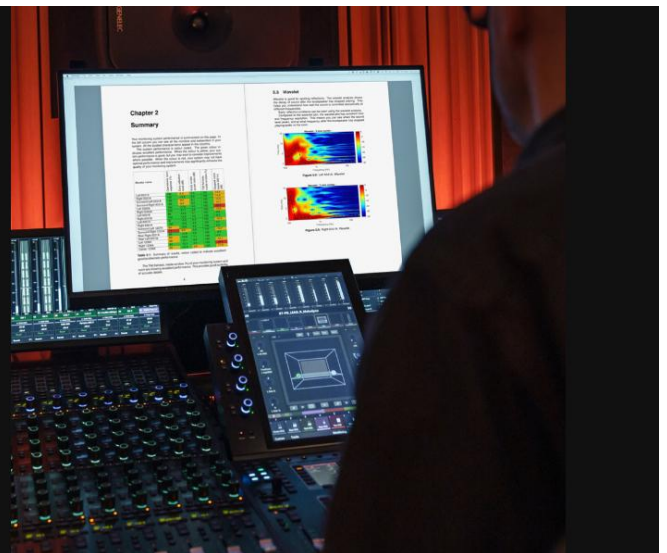


그림 6. GLM 프로그램의 실제리용과정

## -GML 장치들사이 통신

초보적인 분석에 의하더라도 프로그램에서는 USB(HID)와 망통신으로 자료를 주고받는

## GLMv4.exe Decompiler

부분이 있으며 Cloud Server 를 리용하여 조종할수 있게 되었다.

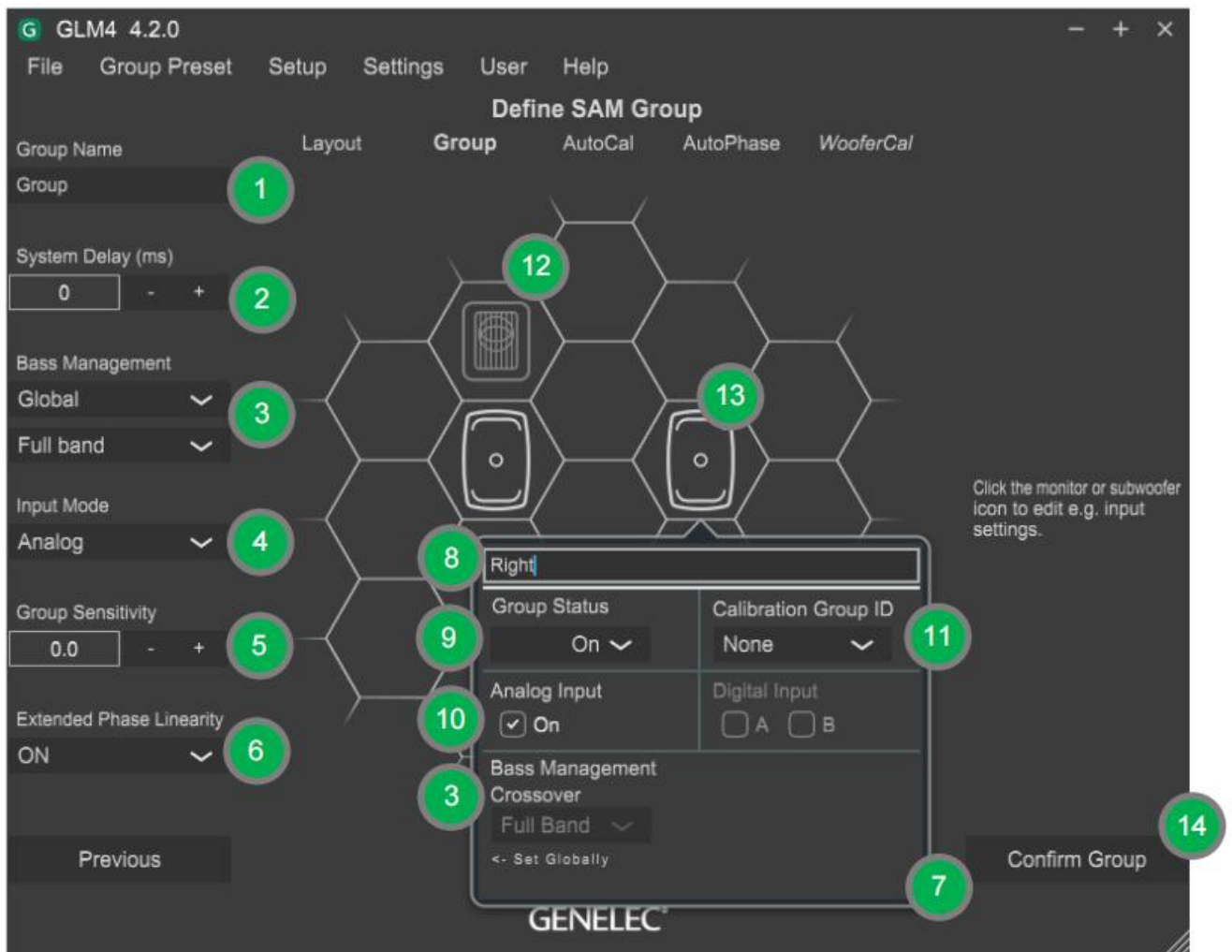
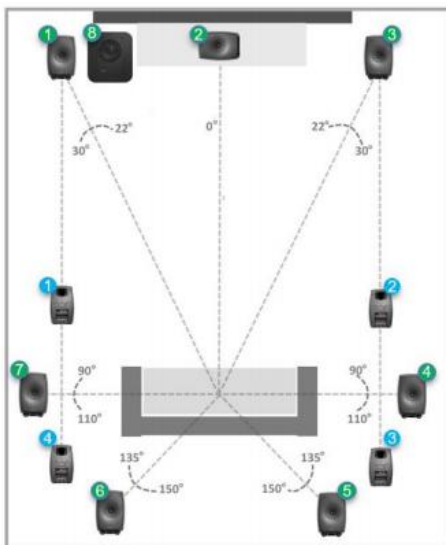


그림 7. GLM 프로그램 그룹설정단계



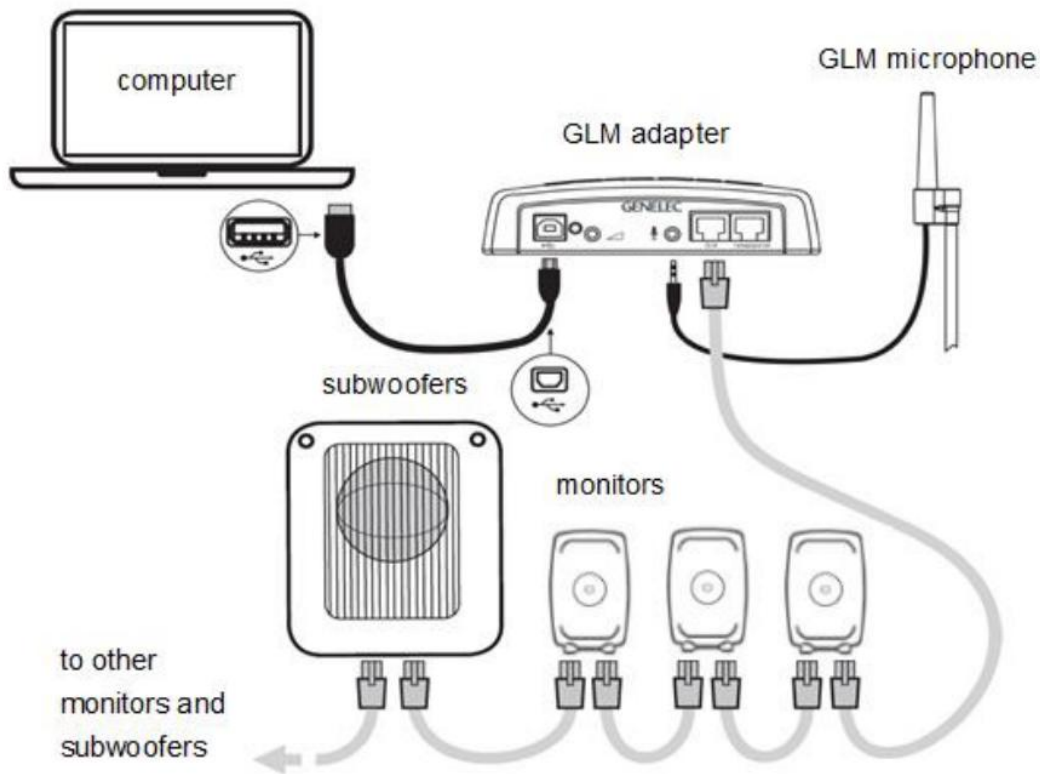


Figure 12. Connections of GLM management network, measurement microphone, and computer

그림 8. GLM 프로그램의 네트워크 환경구성

GLM AutoCal 은 Genelec 클라우드에서 실행되는 강력한 모니터링 시스템 보정 알고리즘이다.

## 2. IDA 를 리용한 GLMV4.exe 분석

분석도구로서는 IDA7.5 도구를 기본으로 리용하도록 한다.

무엇보다도 참고로 할만한 원천코드가 있는 프로젝트를 창조하는것이 중요하다.

프로젝트원천코드를 얻었다고 하여 완전히 프로그램으로 실행시킬수 있는것이 아니며 이것을 실행시킬수 있는 기틀 즉 골격프로젝트가 작성되어야 한다.

IDA 를 리용한 분석에서 Main 함수를 찾고 그 구조를 파악한 상태에서 다운받은 JUCE 프레임워크를 리용하여 새로운 GUI 프로젝트를 만들고 Window 창문을 가진 응용프로그램을 창조하여 실행파일(exe)를 생성한 다음 IDA 로 import 하여 분석해본다.

다시말하여 분석하려는 exe 파일을 IDA 로 열어보았을때와 JUCE 로 만든 exe 파일을 IDA 로 열어보았을때의 차이점을 찾아야 한다. 이렇게 볼수 있는 근거는 목적파일(GLM4.2.exe)을 IDA 로 열어보았을때 `juce::xxxx` 형식의 문자열이 많기때문이다.

다음으로 C#코드분석과 달리 C++코드분석에서는 포인터(\*)가 상당히 중요한 자리를 차지하는데 기억구역에 대한 주소관리를 전부 포인터에 의해 진행하므로 함수포인터찾기가 난문제



## GLMv4.exe Decompiler

로 나서고 있다. 이것을 해결하자면 IDA 상에서 debug를 부단히 해보며 원리와 방법을 찾아야 한다.

다음으로 중요한것은 프로그램코드를 분석해보면 OpenGL(2 차원 및 3 차원그래픽스표준 API) 들을 리용하였는데 이것을 Decompile 하는데서 문제로 제기되는것들이 있다. OpenGL 서고에 대한 표상이 있어야 한다.

다음으로 통신패킷분석모듈을 정확히 찾는것이다.(USB 통신과 소켓통신)

참고로 교정이란 어떤 재려고하는값과 지시값사이 관계를 고려하여 즉 최대값,최소값을 고려하여 측정값을 얻을수 있는 관계를 교정이라고 할수 있다.

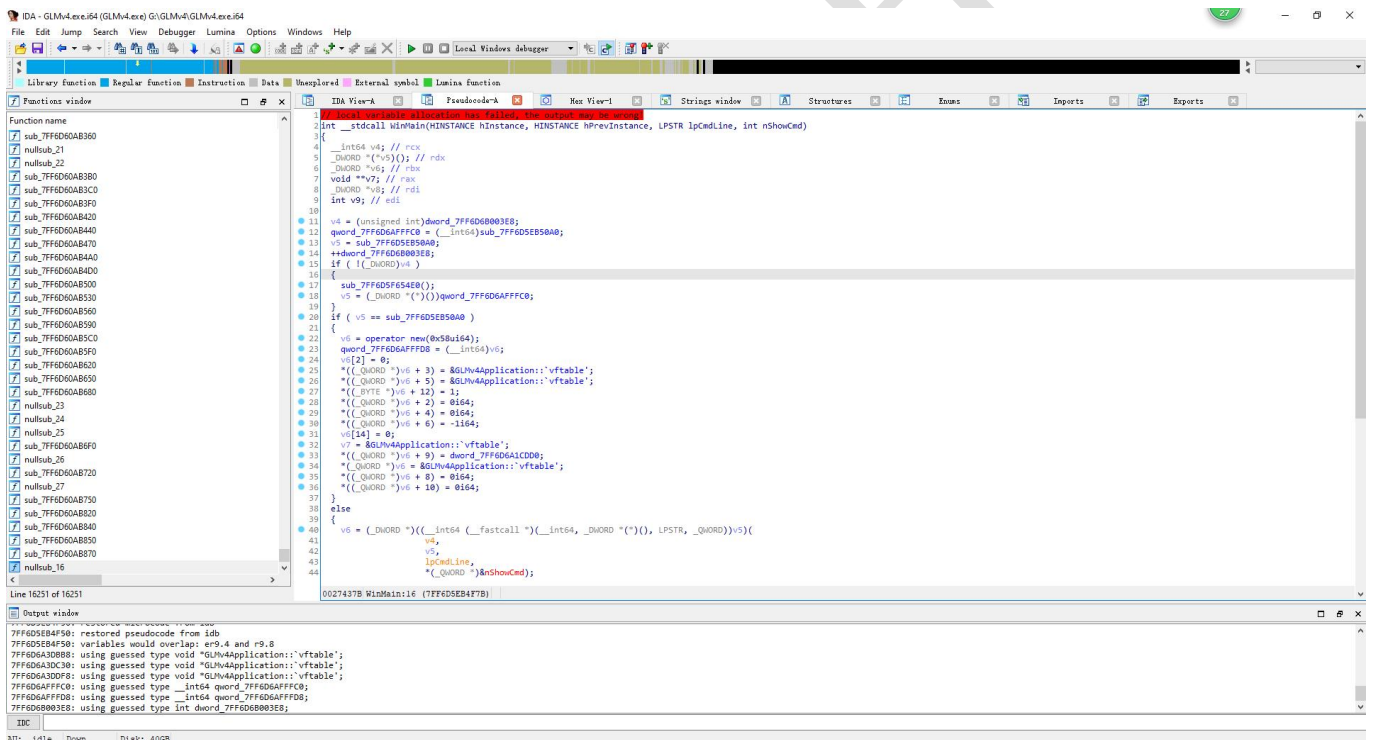


그림 9. GLM 프로그램의 IDA 분석과정

### 3. 프로그램 Decompiler 과정에 필요한 지식과 도구들

- C++언어에 정통하고 있어야 한다.
- Intel CPU에 대한 Assembly 언어에 대한 지식이 있어야 한다.
- JUCE Library 관련지식이 있어야 한다.
- 음성처리에 대한 깊은 지식이 있어야 한다.
- 실험을 정확히 진행하기 위해서는 주변실험환경장치가 필수적으로 필요하다. 왜냐하면 통

## GLMv4.exe Decompiler

신과케트분석과 프로그램의 동작과정에 고찰해야 할 변수 혹은 자료를 알아야 하기 때문이다.

프로그램을 장치와의 런결하에 실행시키면서 분석하는것과 장치를 런결시키지 않고 분석하는것과는 엄연히 큰 차이가 존재한다.

### 4. JUCE 프레임워크를 리용한 코드분석진행

현재 JUCE Library(판본 7.0.5)를 리용하여 GLM4.2 프로그램을 해석하기 위해서 아래와 같은 4 가지 단계를 설정하고 작업을 진행하려고 한다.

>1 단계:표준함수확정(5.1~6.5)

>2 단계:표준함수기능정의(6.6~6.30)

>3 단계:사용자함수정의찾기(7.1~7.30)

>4 단계:론리적기능찾기(8.1~8.30)

그러기 위해서 JUCE Library 를 리용하여 골격프로젝트(AudioSample.sln)를 만들고 이것을 Release 방식에서 빌드하여 exe 파일을 만든다음 그것을 IDA(7.0.5)를 리용하여 분석하여 하나의 C 파일(NewProject.exe\_total.C)로 만든다.

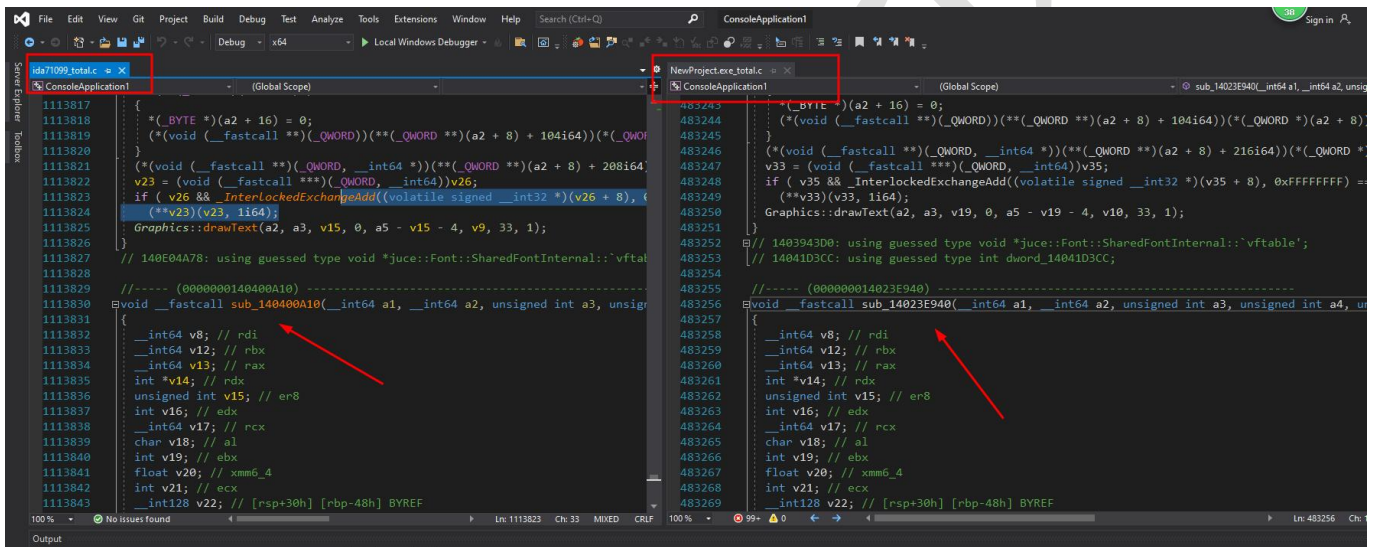


그림 10. GLM 프로그램과 표준 JUCE 의 IDA 를 통해 얻은 C 파일비교

이때 IDA 상에서 나타나는 모든 함수명들은 앞붙이 Sub\_가 붙은 함수들로 이루어졌다는것을 알수 있다.

때문에 GLM4.2.exe 파일을 IDA(7.0.5)를 리용하여 Open 한다음 전체함수들을 하나의 C 파일(ida71099\_total.c)로 넘기고 NewProject.exe\_total.C 파일과 비교하면서 서로 같은 내용을 가진 함수들을 표식해두는 작업을 진행하는것이 1 단계과정이다.

다음 이러한 표식된 두개의 C 파일을 리용하여 2 단계에서는 원천코드(JUCE Library)에 대응하는 함수명을 찾아서 IDA 상에서 함수이름을 표식해두는 작업을 진행하였다.

## GLMv4.exe Decompiler

즉 IDA 상에서 'vftable'에 의해 전체클래스정보를 검색하고 여기서 const juce::클래스명::'vftable'의 형식으로 표준클래스에 대응하는 함수들을 찾아서 두개의 IDA(juce와 GLM)상에서 대응하는 함수들(Sub가 붙은)을 원천코드에 대응한 함수명으로 바꾸어 놓는 작업을 진행하였다.

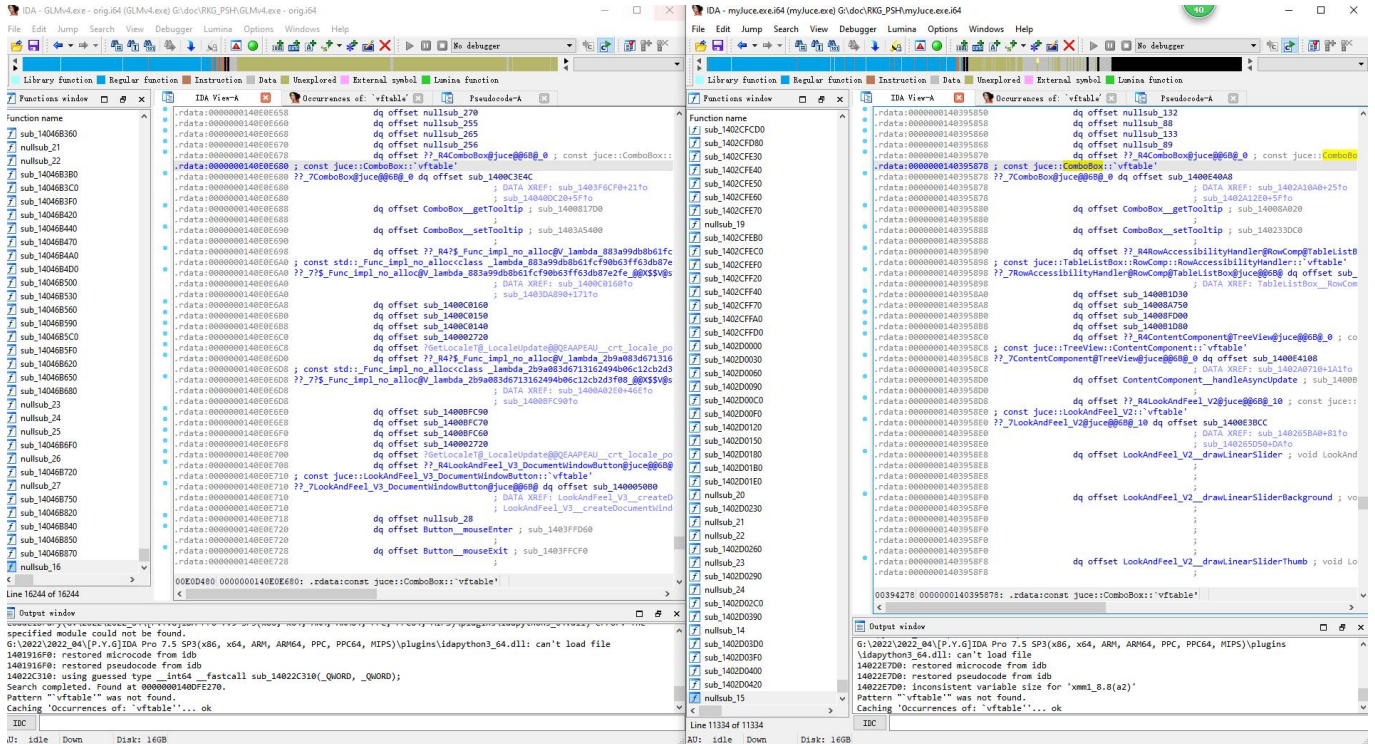


그림 11. GLM 프로그램과 표준 JUCE의 IDA 상에서 함수이름변경

이 단계는 필수적으로 필요하다. 이유는 목적파일(GLM4.2)이 바로 JUCE 프레임워크를 기반으로 구축되었기 때문이다. 따라서 적어도 표준함수들만이라도 선별하여 코드해석의 품을 줄이기 위해서이다.

3 단계에서는 표준클래스들을 기초클래스로 하여 파생된 사용자정의 클래스들을 정확히 찾아서술하는 단계이다.

※○▲△(코드분석에서 한가지 찾은 방법)

코드를 넘기면서 중요한것은 SVN을 설치하고 협동을 잘하는것이다. 하나의 프로젝트(AudioSample)에 개별적인 성원들이 만든 클래스파일들을 추가하는 방식으로 작업을 진행하며 사용자정의 클래스파일만 SVN 서버에 upload 하고 기본프로젝트는 각자의 컴퓨터상에 배치한다(용량상관계와 공통부분이므로). IDA 상에서 찾은 함수명들을 기록한 Excel 파일도 매일 upload 한다.



## GLMv4.exe Decompiler

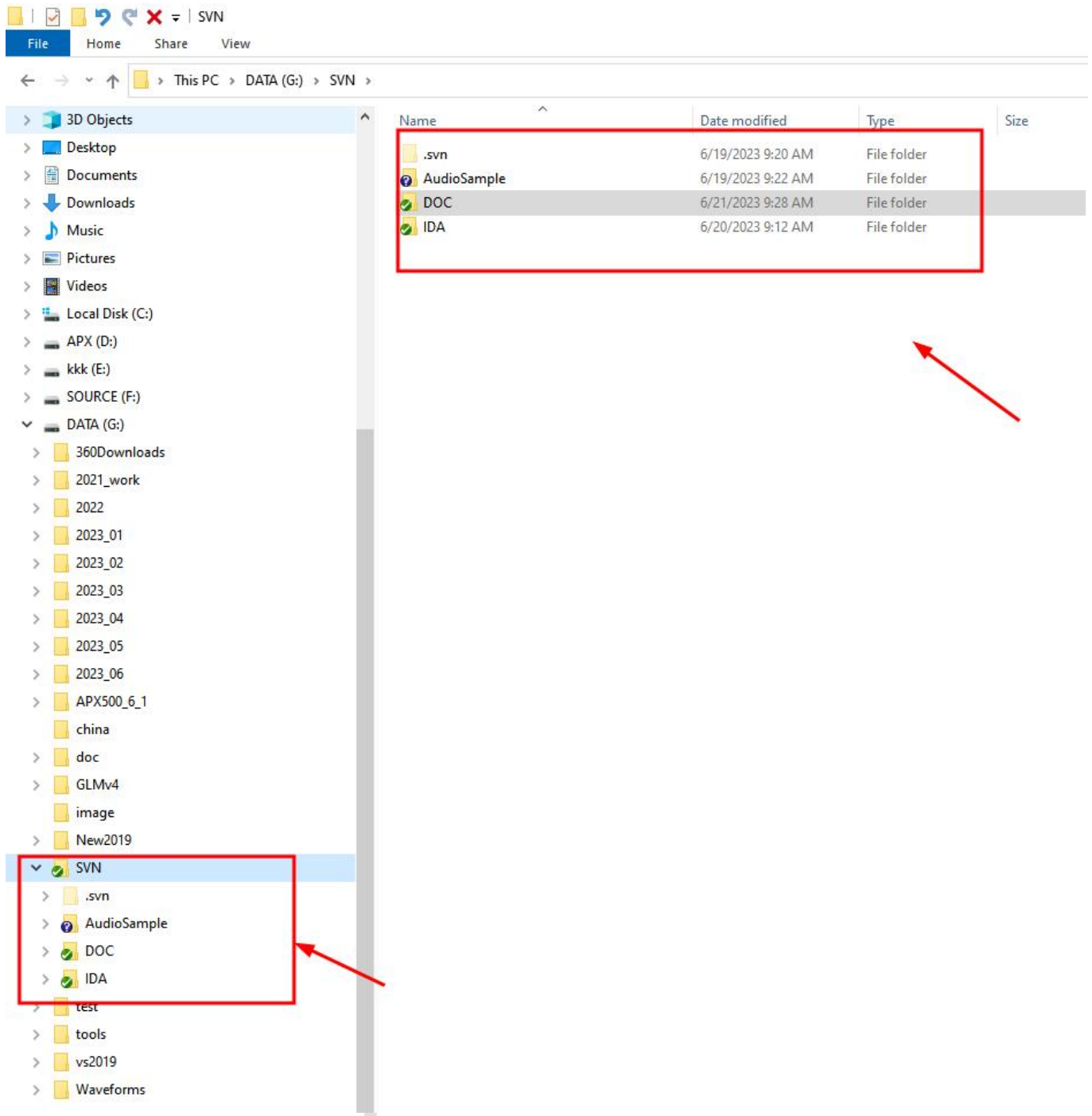


그림 12. SVN 설치환경에서 작업과정

Excel 파일 종합은 박성혁동무가 맡아서 진행하며 SVN 서버관리와 전체프로젝트관리는 본인이 한다. 클래스명을 추가할때 몇가지 준칙을 토의 하였다.

○ IDA 상에서 Sub 로 된 함수명들을 원천코드에 옮길때 반드시 Override 표식을 해야 되는가를 판단한다.

○ 어떤 클래스로부터 파생되었가를 표시하며 불명확할때에는 뒤에 other 클래스 개수를 붙인다. public 와 private 속성인자를 반드시 기입해주어야 한다. 오류를 없애기 위해 코멘트를

## GLMv4.exe Decompiler

넣어주어 IDA 에서 복사한 코드의 오류를 없애야 한다. 왜냐하면 앞으로 빌드를 하면서 프로젝트를 Debug 해야 하므로 현재는 아래의 그림에서와 같이 IDA 코드들을 Comment 해주기로 토론.

○ Excel 표를 정확히 기입해야 한다.

○ 한가지 JUCE 원천코드분석에서 중요한것은 클래스들의 파생관계를 어떻게 IDA 상의 코드분석과 일치시키겠는가 하는것이다. 방법은 다음과 같다.

먼저 IDA 상에서 vtable 에 해당하는 클래스명을 찾고 실레(`; const juce::Button::`vtable'`)로 검색하면 아래의 그림과 같이 검색되어 나온다.2 개 Button 에 해당하는 Button.h 파일을 JUCE 원천프로젝트에서 찾고 그것의 기초클래스의 정의파일을 찾는식으로 virtual 과 override 관계를 차례차례 고찰해 나갈수 있다. 이때 Button 클래스는 바로 두개의 클래스(Component 와 SettableTooltipClient)로부터 파생되었기때문에 IDA 상에서 두개의 vtable 이 생성되어있었던것이다.

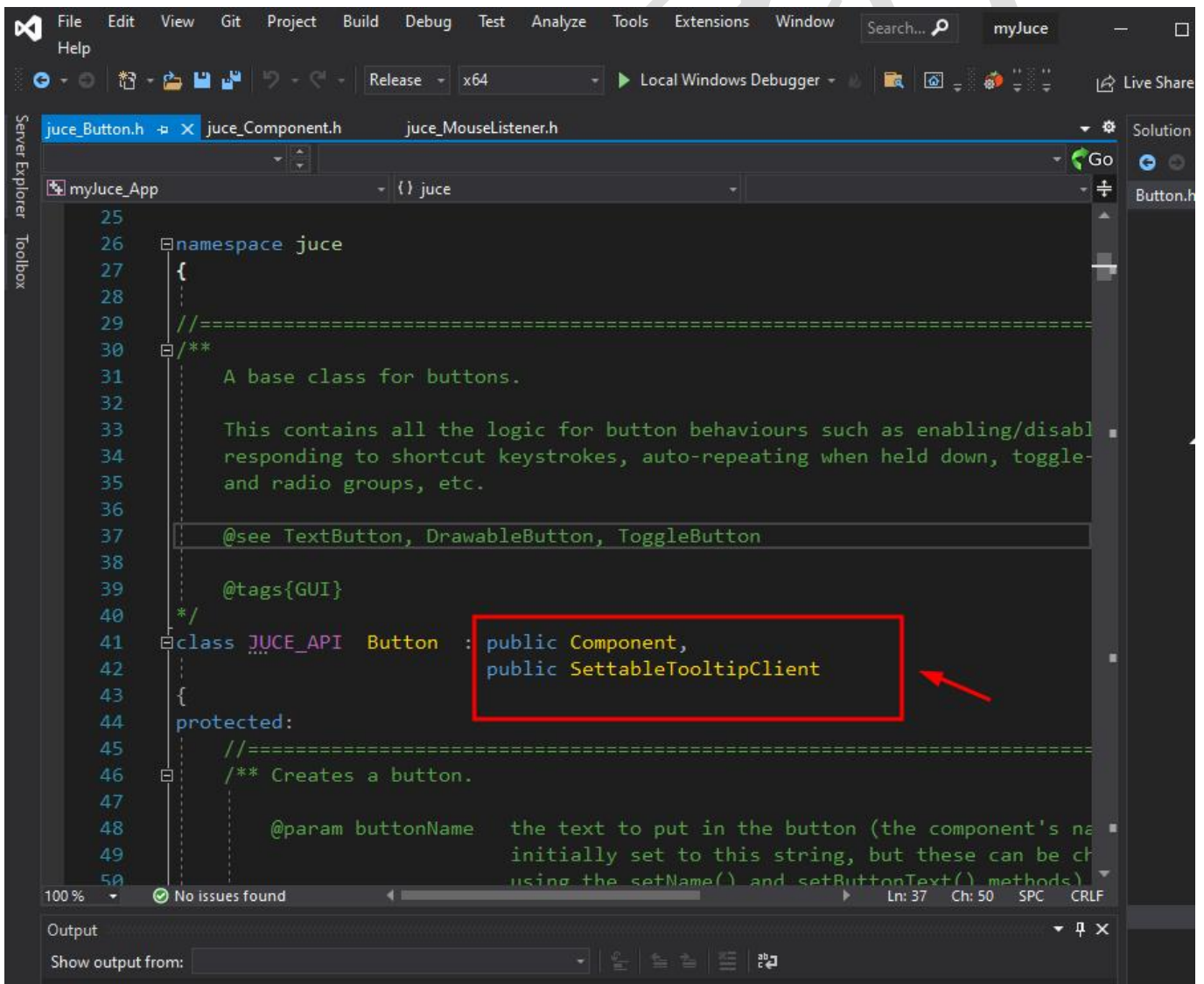


그림 13. Button 클래스의 기초클래스 2 개(Component,SettableTooltipClient)

결론적으로 IDA 상에 나타난 vftable 은 바로 자기가 파생한 클래스의 개수와 반드시 일치한다.

vftable 에서 순서는 반드시 가장 기초로 되는 클래스의 virtual 함수들의 순서가 우선이고 그다음 파생된 클래스들의 순서로 virtual 함수순서가 놓인다. 이 원리를 잘 알면 함수이름을 확정짓는데서 애매하거나 모순되는 점을 해결할수 있다.

그리고 IDA 상에서 nullsub\_수자형식의 함수들은 함수본체가 없는 부분이며 \_purecall 형식은 virtual 형식의 함수명에서 =0 으로 정의된 함수들이며 파생클래스에서 override 가 되지 않은 함수들이다.

또 중요한것은 클래스의 Template 형식으로 이것은 Template 에 어떤 클래스를 넣는가에 따라 클래스형이 달라지므로 성원함수들도 정적이지 못하며 동적으로 변한다. 다시말하여 Template 에 정의된 성원함수는 고정적이지만 그것을 리용하는 클래스들에서는 그 고정된 함수들을 자기의 성원함수처럼 리용하므로 IDA 상에서 같은 성원함수이지만 이름을 서로 다를 sub 를 가진 함수로 될수 있다는것이다. 즉 함수내용은 같고 함수명만 차이난다.

○ 함수명은 같고 파라미터가 차이날때에는 <\_함수명 >식으로 \_를 붙이도록 하였다.

## GLMv4.exe Decompiler

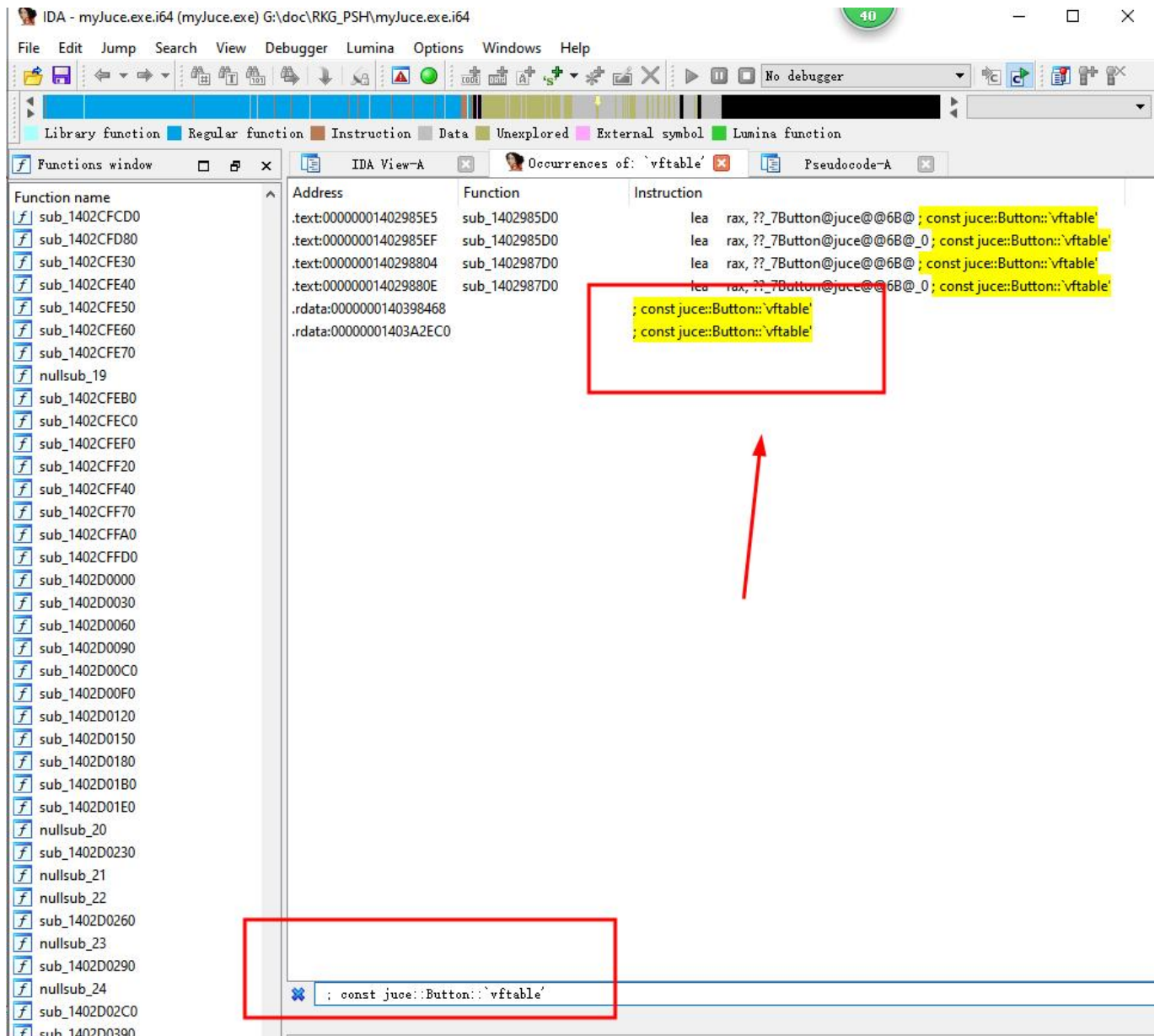


그림 14. IDA 상에서 Button 클래스의 vtable 관계도

## GLMv4.exe Decompiler

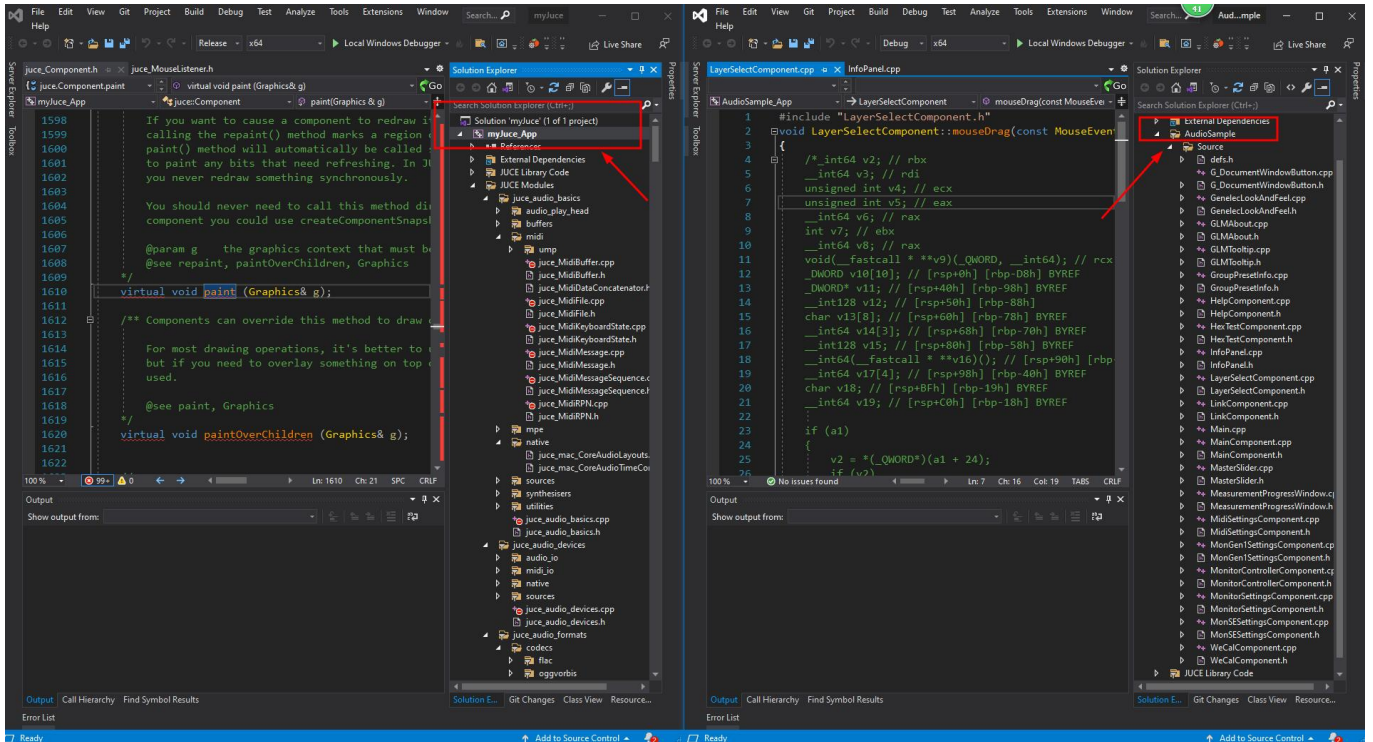


그림 15. AudioSample 프로젝트와 표준 JUCE 프로젝트

이 단계에서는 사용자클래스가 어떤 클래스로부터 파생되었는가를 명확히 하는것이 중요하다. 즉 기초클라들의 개수와 기초클래스명을 정확히 찾는것이 관건이다.

왜냐하면 오브젝트프로그래밍(Object Oriented Programming)에서는 일반 C 프로그램과 달리 모든 객체를 클래스로 묘사하며 클래스들의 파생과 의존관계에 기초하여 모든 처리를 진행하므로 클래스와 구축자들을 정확히 찾아 정의하는것이 중요하다.

물론 원천코드의 매개함수가 정확히 IDA 상에서 하나의 함수로 대응되지 않는다. 이것은 compiler 와 빌드환경에 따라 차이난다.

4 단계에서는 논리적기능을 찾는 단계이다. 실제로 장치를 프로그램에 려동시켜서 IDA 상에서 Debug 를 진행하면서 장치런결과 단기, 교정, 동작흐름에 해당하는 알고리즘을 도출하는 단계이다. 3 단계까지 확정한 클래스와 구조체들을 어떻게 런결시키고 어떤 관계를 가지고 프로그램이 자기의 기능을 수행하는가를 확정짓는 단계라고 볼수 있다.

## 5. 참고문헌

- [1] GLM4 System operating Manual.pdf (Genelec Loudspeaker Manager ,GLM) ,98 페이지,2022.5
- [2] Juce Programming Tutorial.pdf 75 페이지
- [3] Juce Library Source code download
- [4]

