

# Goal

Manipulate the audio data stream in realtime using analogue input from hardware as variables in code and original audio tracks.

Done!

Hello World from the board's Serial out

For a solid week, I simply could NOT get the board to be recognized on my computer. I even thought that the board could be defective, so I bought another one and that one was not recognized either. So I looked at the board and saw it had a CP2102 chip, which I found out was the chip that let the ESP32 communicate via USB. So even though MacOS usually doesn't need drivers, I got drivers from Silicon Labs (the manufacturer of the CP2102), and it was STILL not being recognized. So in an act of desperation, I bought a micro USB to USB C cable for the board, and finally it was recognized.

Done!

Play audio from the board to the speakers

This has been a complete nightmare. There is ZERO documentation of this board online except for the nearly useless User Guide which assumes that we're only trying to build an Alexa-like device. There is also no example code anywhere from people's projects using this board. It's as if I'm the first person in the English-speaking world to use the ESP32-LyraTD-Syna. The user guide suggested that I use Espressif's ESP-ADF library.

Setting up ESP-ADF and its prerequisite ESP-IDF felt like navigating a labyrinth of the most user-unfriendly UI from the 90s, but even when I somehow managed to pull it off, I found that Espressif didn't provide a board configuration for the LyraTD-Syna. There's one for the Lyra-mini, two for the LyraT, two for the LyraTD-MSC, but nothing for the LyraTD-Syna.

I looked at the documentation for the other boards and they got lengthy user-friendly guides, but the LyraTD-Syna only gets a useless PDF.

It's as if this board was the ugly stepchild in their product lineup.

Not only that, I found out that there were literally zero drivers for interfacing with the CX22721 Codec chip, so I would be going in completely blind. In my desperation, I

turned to AI (Which I HATE using for code) to see if it could generate something slightly usable, but it was incredibly low-level C code that not only I did not understand, it didn't work!!

Figuring out how to use this stupid board was turning into a Sisyphian effort as it dawned on me that I would need to write my own drivers for using the audio codec chip in order to work with it, which would be a whole project on its own.

I was starting to believe that this project was impossible b/c I just spent three weeks just trying to get a test sound to play from the world's least documented devboard. I did some research and heard of a different ESP audio-based devboard with an incredibly powerful set of libraries made by a Phil Schatzmann who specializes in making Arduino/ESP32 based audio projects.

So I forsook the LyraTD-Synn and bought the AI Thinker Audio-Kit, which uses the ESP32-A1S.

Using Schatzmann's code libraries, I was able to get audio playing from the board within an hour of opening the packaging that the board came in. Finally.

Now on to the next thing

Have audio effects be applied upon user input (via one of the boards buttons at first)

WIP

Connect the force sensor to the board and be able to read its values in code

Use F.S.'s analog input to apply audio fx to audio

Create a couple audio tracks that vary in mood

Design and print an enclosure for the whole thing

Connect the heart rate sensor and be able to calculate average bpm in code

Create and implement an algorithm depending on the user's heart rate and how hard they press the force sensor to determine their approximate mood and which track to play

Create schematic for a PCB to connect all the components together