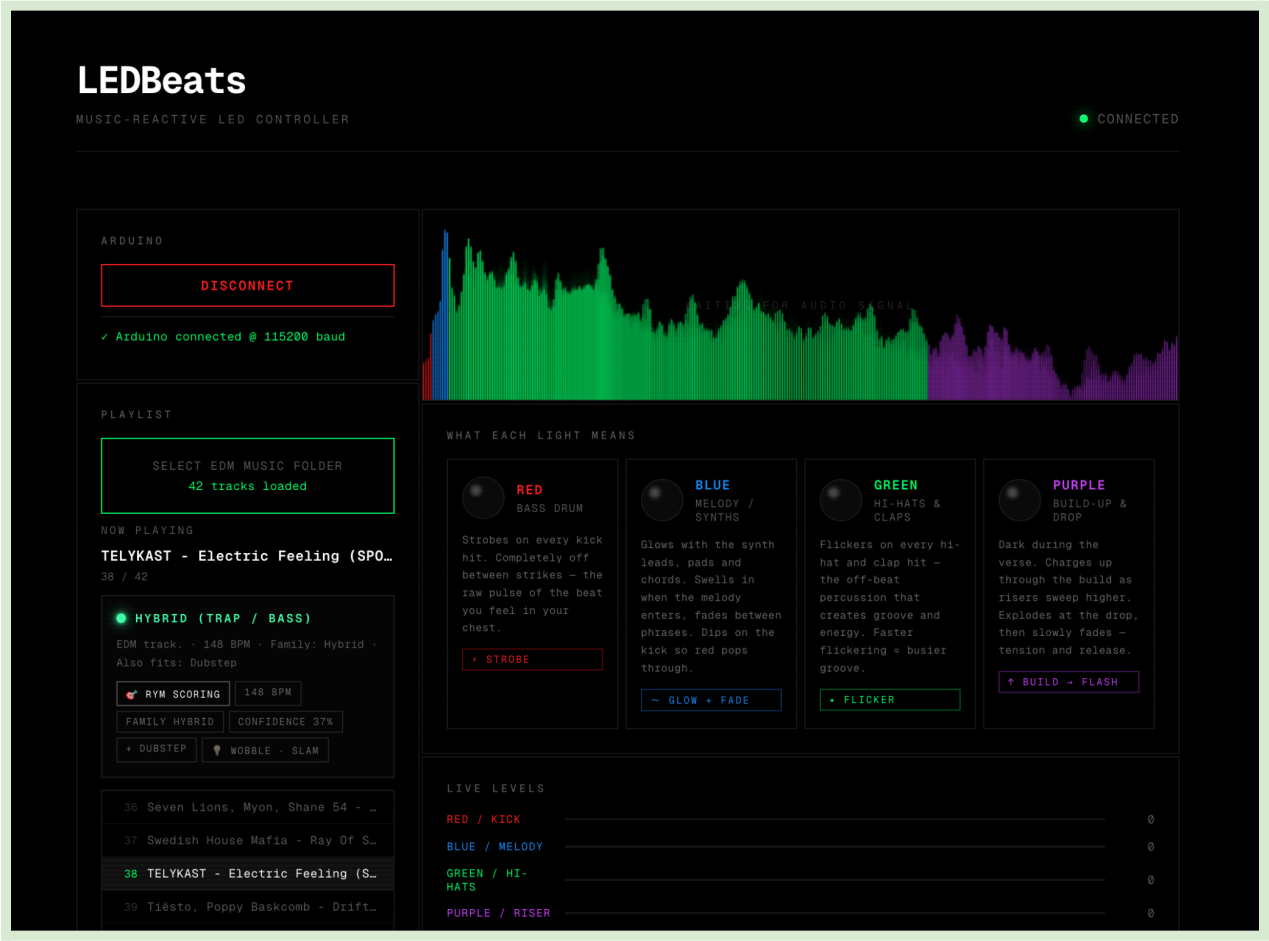
LEDBeats

By Joanna Le

March 13, 2026



Showcase / Description of Finished Piece

LEDBeats Video.MOV

This project built off my previous project for “Sense of Self,” where I visualized all the music events I’ve gone to for the last 3 years in an interactive purple planet system. I’ve been listening to Electronic Dance Music (EDM) ever since I was a kid, but now that I’ve been going to more events, I’ve been growing increasingly curious about how people are able to differentiate all the nuanced subgenres. There’s so many elements that go into music production; for example, dubstep is distinguished by their syncopated half-time rhythms, and heavy sub-bass with a 140 BPM tempo, while techno follows a repetitive 4/4 "four-on-the-floor" beat and typically ranges from 120-150 BPM. I only knew a couple generic genres until recently as friends and family introduced me to more artists and songs. Also, the LED aspect of this project reminded me of the lasers and lights I would see at events so I wanted to replicate that on a minor scale. Ultimately, the goal of this project was to create another visual and programmed way for me to understand and recognize the subgenres within EDM using songs that I’m currently fixated on!

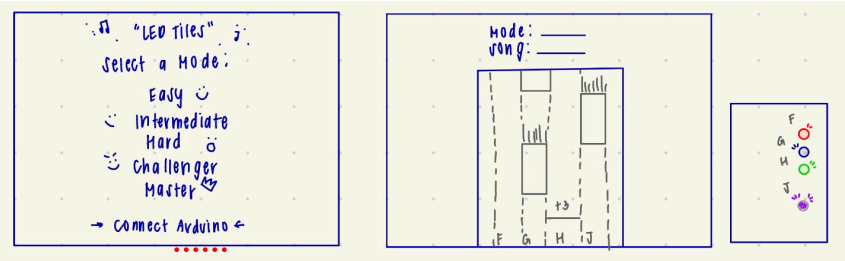
Ideation/Design Process

Ideas:

- Vietnamese lady in a boat
- Koi fish (yin yang)
- Lotus flower in a pond
- Laser harp
- **Connect to song → lights up based on bpm, bass, or other musical notes**
- Guitar hero / piano tiles inspired
    - Upload music file → turns into a four Piano tiles like website game and when you hit a tile → light pops up based on music
    - Keys → F G H J
- Play tetris → certain colored tiles light up depending on block dropping

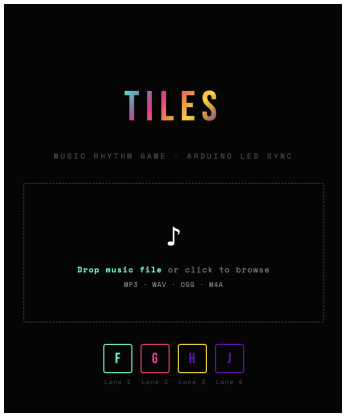
List of ideas I brainstormed

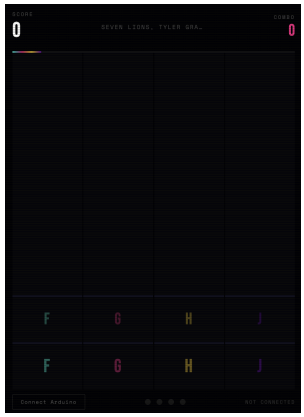
Initial Ideas + Failures:
I had two ideas: drop a YouTube link of a song into the website and based on the song's BPM and other musical elements, the LEDs on the Arduino board would light up.



Sketch of my first idea

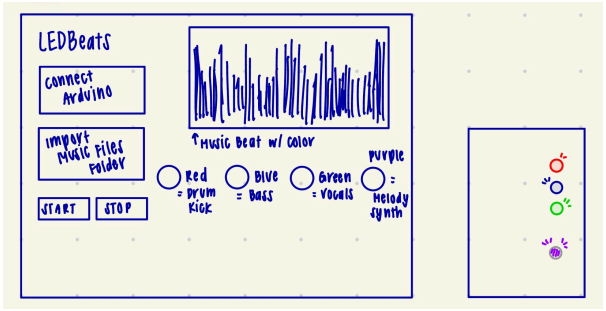
However, inserting links and using YouTube APIs took a lot of time and storage, and there were a lot of inaccuracies (e.g. the song that loaded onto the website was different from the link). A second idea came from my childhood fondness and memories for Guitar Hero and Piano Tiles; the website would allow the user to play the a Piano Tiles like game, pressing the keys "F", "G", "H", "J" (inspired by the online game https://magictiles.org/) and depending on the key pressed, it would light up one of the four LEDs on the Arduino board. But to be honest, I was still pondering my first idea and had more to say regarding the first idea. I think I gave up on my first idea too early, so I decided to go back and try to iterate some more.

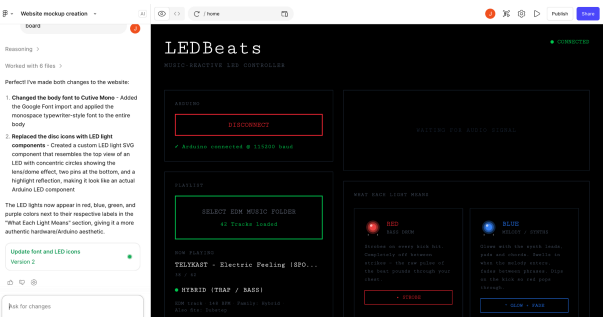




Inspirations:
I was on a call with a friend asking about any upcoming events they were planning on going to. They were sharing some new EDM songs they discovered and I kept asking them “What genre is this under? Do you know the subgenre?”. Then a brilliant idea came to mind: why not answer it myself?! I’ve always struggled to recognize the subgenres myself, so I thought that combined with my new motivation to explore new subgenres within EDM and attend as many concerts and raves as my wallet would allow, I could build something that would help me learn more about how people or computers are able to identify the nuanced subgenres and visualize them.

Ideation + Making Progress:
I used Figma Make to mockup what I wanted the design to look like and then used Claude AI to prompt-engineer and flesh it out with viable buttons. The prompt I used to initialize the website included the design template, goal of the website, what each of the colors means and a display, a place for the user to connect the arduino and import music files, and the music visualization chart. I initially wanted to have it be able to link to my Spotify and I could just drag and drop songs, but getting the API took too much time (once again) and I thought it would be easier if I just downloaded songs I am currently listening to locally so it felt more personal to me.





Sketch + mockup of my second and implemented idea

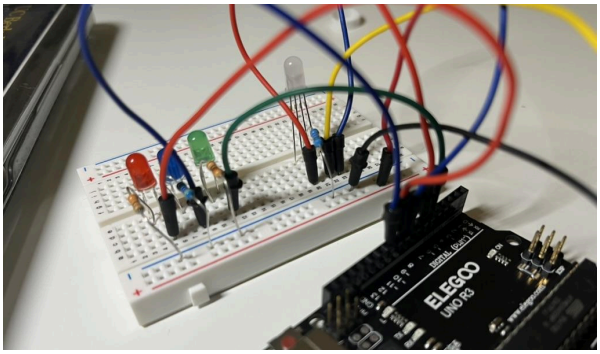
How it works:

** I used someone's EDM Subgenre Classifier code I found on Github as a base and then made my own changes **

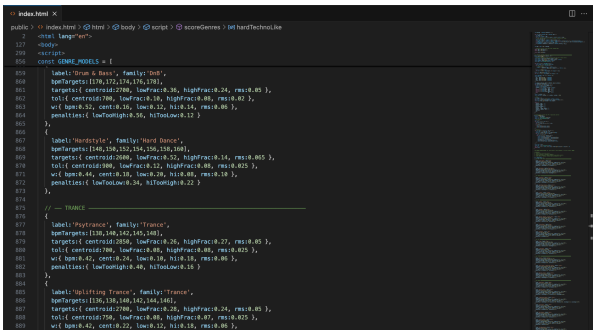
1. When a track loads, the app decodes the audio file offline into raw PCM samples, runs two independent analyzers, then scores every genre model against the results.
2. Tempo/BPM - detects recurring drum hits and measures the time between them. Most EDM subgenres live in distinct BPM ranges, so this immediately narrows down candidates.
3. Bass Weight - how much of the song's energy lives below 300Hz which is the kick body and bassline.
4. Brightness (Spectral Centroid) - ex. higher frequencies include bright leads, acid synths, hi-hat density while low centroid means the sound is dark and heavy.
5. Loudness - how compressed and loud the master is; the deciding factor between genres that are otherwise spectrally similar.

Every genre has a target value for each of these features. The app scores each candidate using a bell curve, so genres whose targets match the measured audio score highest and drives the LED behavior for that track.

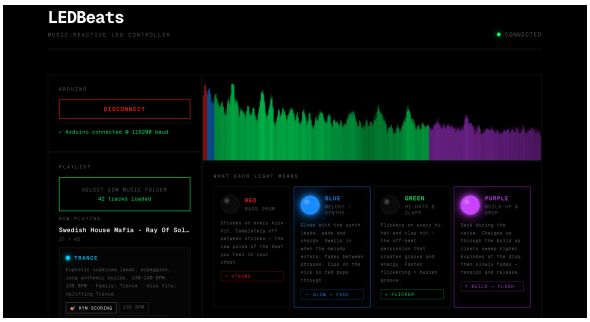
Iterations I made:



I used red, blue, green, yellow, but the color yellow looked too muted to resemble music lights at festivals so I switched it out for the RGB (purple)



The longest part of the project was editing the 'Subgenre Scoring Engine' since slight changes in music elements would make it fall under the correct/incorrect bin



What the colors mean:

Red - The Kick Drum

- Strobes on every kick hit, off between strikes

Blue - Melody & Synths

1. I first used an online, free music subgenre identifier called Last.fm API, so when a track loaded, the app parsed the filename to extract the artist and track title, then queried the Last.fm API for community- tagged genre data. However, it was so inaccurate, I pivoted. 2. I used a live audio sampling, so while a track played, the app collected 8 seconds of FFT data from the Web Audio API and classified it based on BPM and rough frequency ratios using if/else rules. The problem was that short live sampling produced unstable readings, and the gate-based logic meant any track that didn't perfectly fit a condition would fall through to a default label. 3. I upgraded the BPM detection and moved to a scoring model where every genre gets a score based on how closely its expected audio fingerprint matches the measured values, and the highest score wins. I tested individual songs one by one, logging each track's metrics, then manually adjusted each genre's target values to match what I was actually seeing. I repeated this across every subgenre until each one scored highest on its correct label.	● Glow - genres with sustained, flowing melodies ● Stab - genres with punchy, percussive synth hits ● Wobble - Dubstep and Psytrance Green - Hi-Hats & Percussion ● Glow - genres with sustained, flowing melodies ● Stab - genres with punchy, percussive synth hits ● Wobble - Dubstep and Psytrance Purple - Build-Up & Drop ● Riser - House and Trance genres with clear 8 or 32-bar builds ● Always On - Melodic Techno and Psytrance which sustain continuous tension ● Off - Deep House and Trap don’t have a dramatic structure

Conclusion / Reflection

LEDBeats started as a way to make my EDM listening experience more immersive. I wanted my LED setup to respond to the music in a way that felt intentional and replicate the intensity and emotional resonance I have when going to events in person. What I didn't expect was how deep the technical rabbit hole would go. Getting the subgenre detection right pushed me to learn real signal processing concepts I had never touched before, from autocorrelation to spectral feature extraction, and the iterative process of testing individual songs and manually tuning the model taught me more about how EDM is actually constructed; I learned so much about the relationship between tempo, bass weight, and brightness across genres. This project also deepened my interest in music production, which is something I want to explore more in the future. Understanding how subgenres are defined by their frequency profiles and arrangement structure made me think differently about sound design. If I had more time, I would have taken this further: swapping out the LEDs for something more expressive, like a laser harp, where the physical interaction with light becomes part of the musical experience itself.

Links

Code	https://github.com/joannale-berk/LEDBeats → To check out the website, you need to deploy on Vercel, but here is the link to the website to make things easier.
Final video	LEDBeats Video.MOV

Sources

- I used Claude AI & v0 (vibe-coding website) to write the code for my project. I used Figma (Figma Make) to help with design choices and mockups.

EDM Genre Classifier	https://github.com/jrobin2791/EDM-Genre-Classifier → I used this code as a base for identifying the subgenres.
EDM Research	Essential Guide to EDM Song Structure - Hyperbits EDM Subgenre Classification Using Supervised Learning | by David Wismer