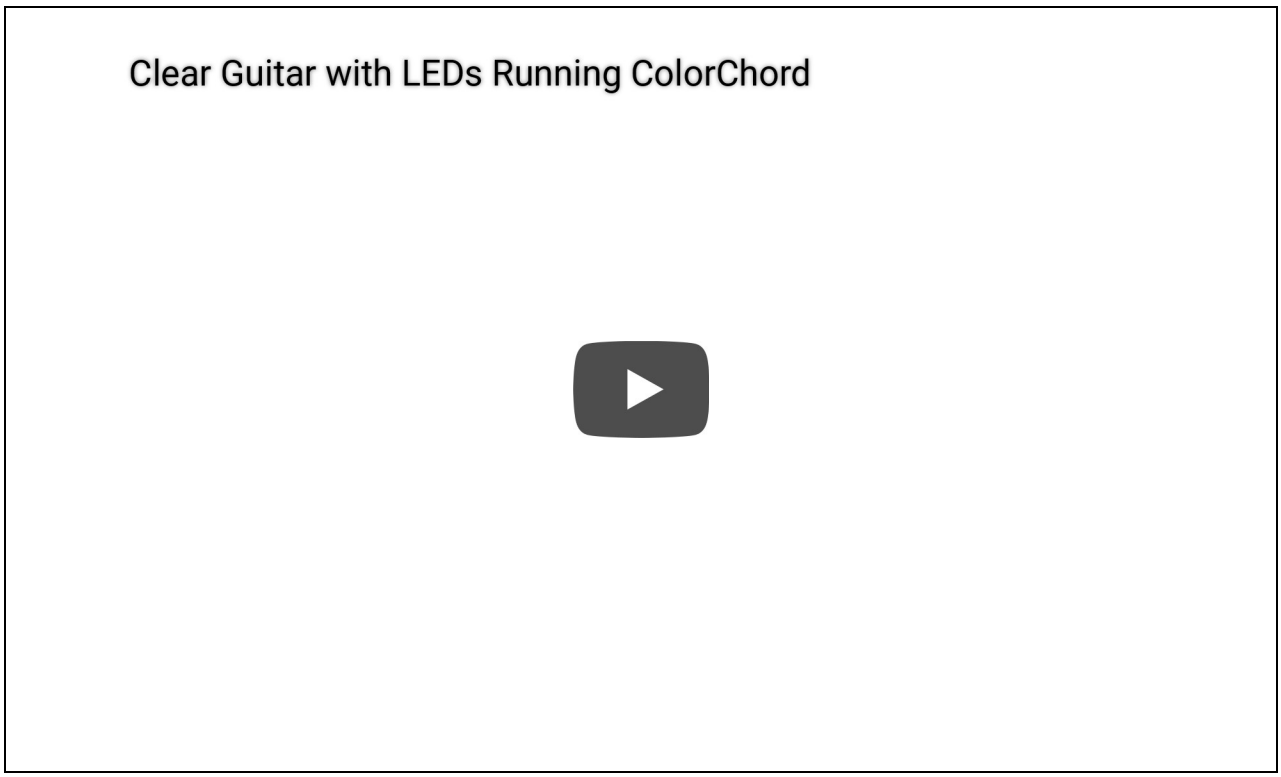
THURSDAY, NOVEMBER 18, 2010

ColorChord Sound Lighting

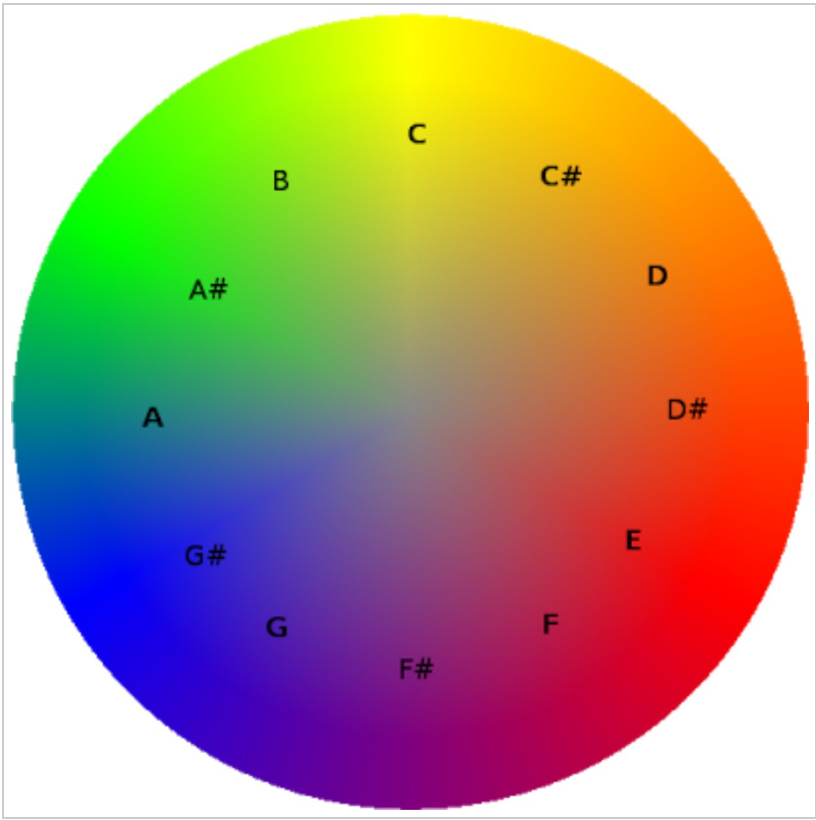
About a year and a half ago, I wanted to put some LEDs in a clear guitar. This has been done before, but I wanted to give it a try and see if I could come up with something really cool. I bought a clear guitar, put USB controlled LEDs into it and plugged it up to my computer before I found out: No one has any tone-based algorithms to convert sound into color! What did I do? I invented ColorChord (TM).



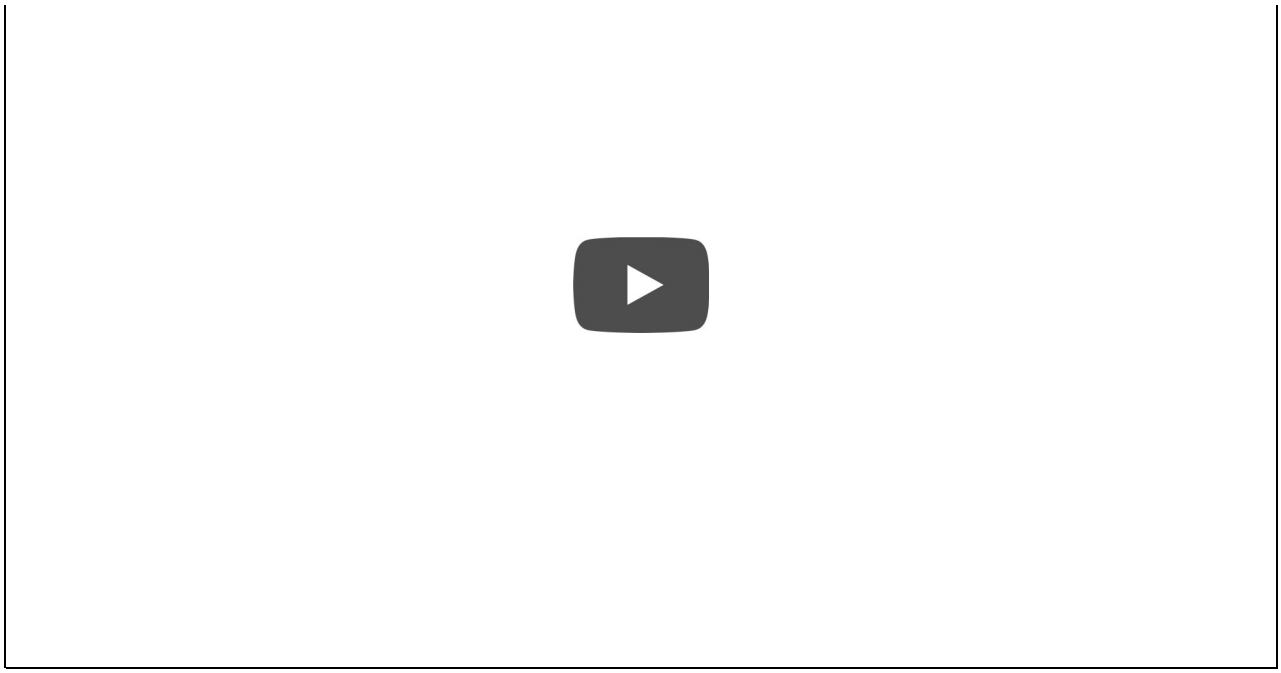
Many people have attempted to assign colors to sound in the past. Many of them were either synesthetic, had perfect pitch (or worse) both. Why is this worse? Because most people aren't synesthetic and don't have perfect pitch. Many of the systems that these people produced involved unusual color associations to sounds, or revolved around a given note always applying to a given color regardless of the key of the music it was used in. Many others had color systems that were unable to transition smoothly while increasing in pitch up a musical scale.

The concept of ColorChord here is that the primary color (most times Yellow) is applied to the note that the piece's key is. I.e. If you're playing many guitar songs, you're playing in the key of E. Therefore, the E note will be Yellow. If you're playing something in C major, the C note will become Yellow.

After much experimentation, some friends and I found that the chromatic color wheel applies surprisingly well to the chromatic scale. I have noticed that in some situations sliding the scale slightly seems to work better, but for the most part this seems generally applicable and is what is used in all of the videos.



There are some neat properties that immediately jump out of this. Chords make some semblance of artistic color sense. In this picture, it is a wheel turned to C major. A C Chord would produce [Yellow, Bluish purple, and Red] (in descending brightness) You'll notice that fifths produce nearly complementary? colors. This works out very well. After trying some other combinations this is the layout that I just couldn't do any better than. This is the color



Technically, this process is very complicated, and if people want the source code, I have no problem publishing it. But here's a rough overview. Just as a word of warning, everything has to happen with very low latency. This system takes sound in via a microphone and transforms it into colors in real time. More than 1/30th or 1/20th of a second and it starts to get annoying.

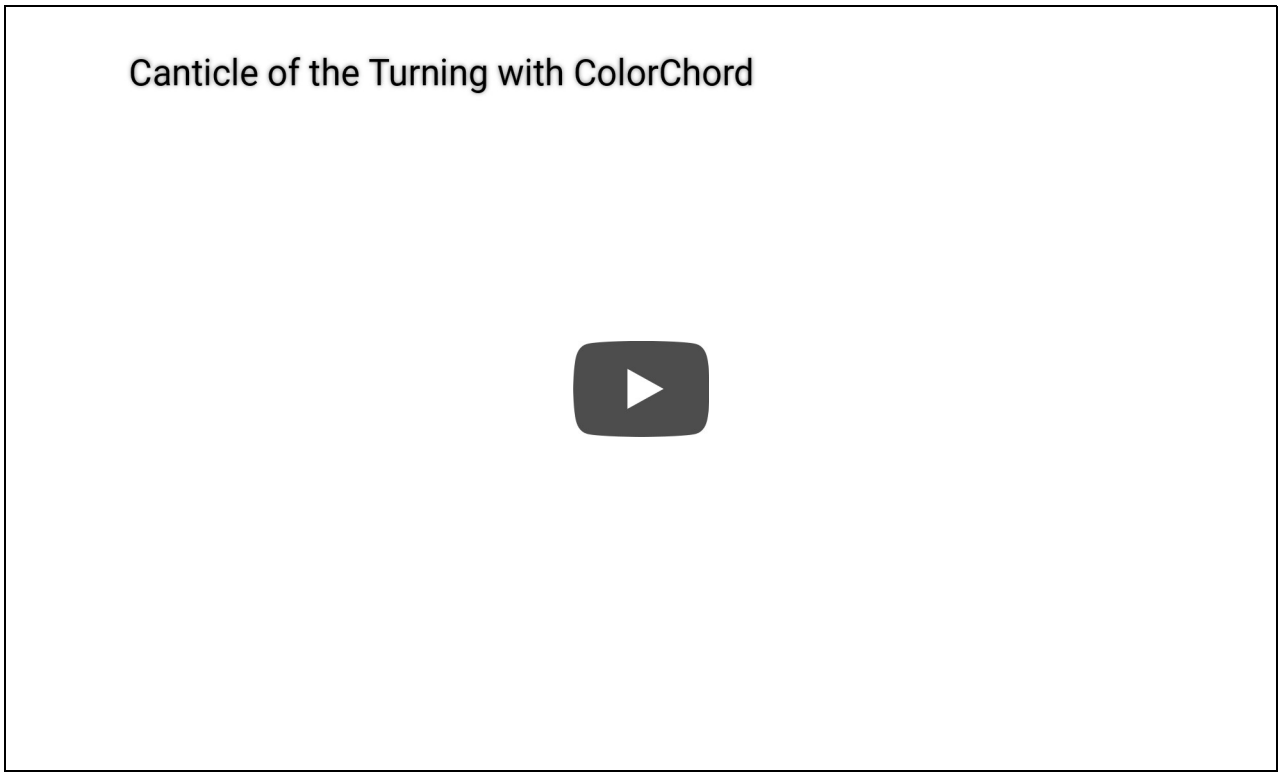
Step 1: Record the sound (512 to 1024 samples at a time) must be constantly streaming, minimise any buffering.
Step 2: Copy the sound to a buffer on the GPU. Because I'm using GLSL, this is just a 1D texture.
Step 3: Perform a windowed DFT, with the beginning of the sound niblet and ends tapering off to silence. This DFT cannot be implemented as an FFT. It needs to be in chromatic space. For example, on some of these demos, every octave may have 24-512 individual bins. The output is "folded." This means a C in one octave will contribute to the same bucket as a C in another octave. Several octaves may fall into the same bucket.
Step 4: Copy back to the CPU, and filter against previous samples. More filtering produces a less jittery result, but also adds latency in the response of the lights.
Step 5: Find all the peaks. These are the pitches within the octave that are loudest in the sample.
Step 6: Select the highest few peaks.
Step 7: Assign your lights to your peaks. Try to keep some inter-frame coherence here. Also, assign your lights based on the brightness of your peaks. Very high peaks may take over several lights (or all) while quieter peaks may only take on a few. If you're playing a chord, you should get roughly even distribution among all three peaks).
Step 8: Assign your peaks (or lights) the colors according to the wheel.
Step 9: Modify the brightness of the lights based on the strength of the peak. Don't worry here - percussion will simply raise the noise floor of the output because of the DFT we performed earlier, it will not show up.
Step 10: Output to the lights.

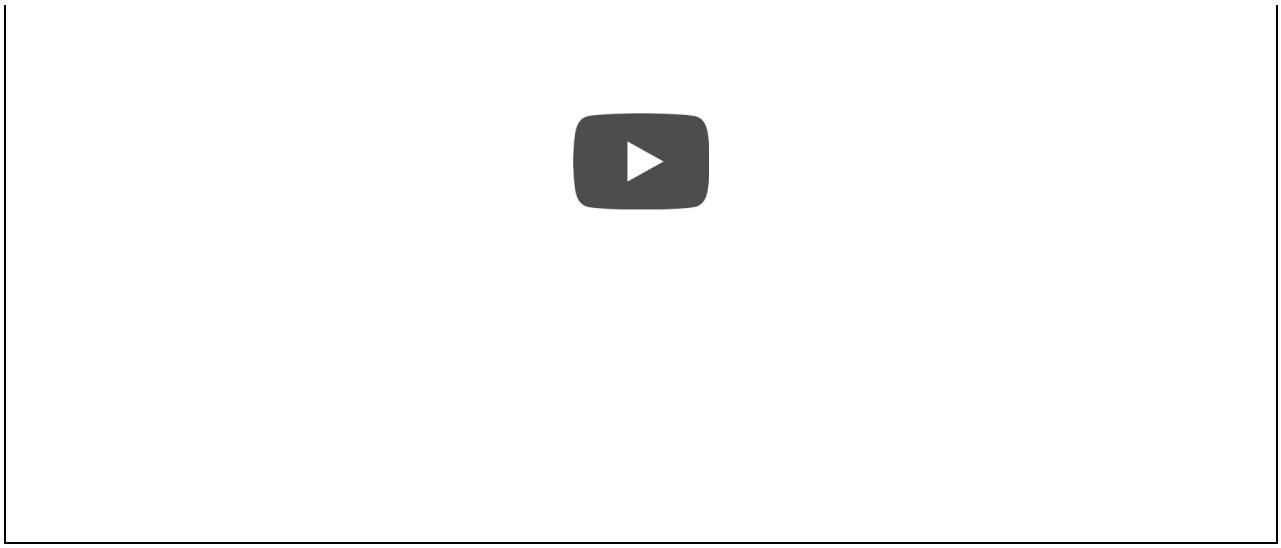
networks, but that just didn't cut it. Oh well.

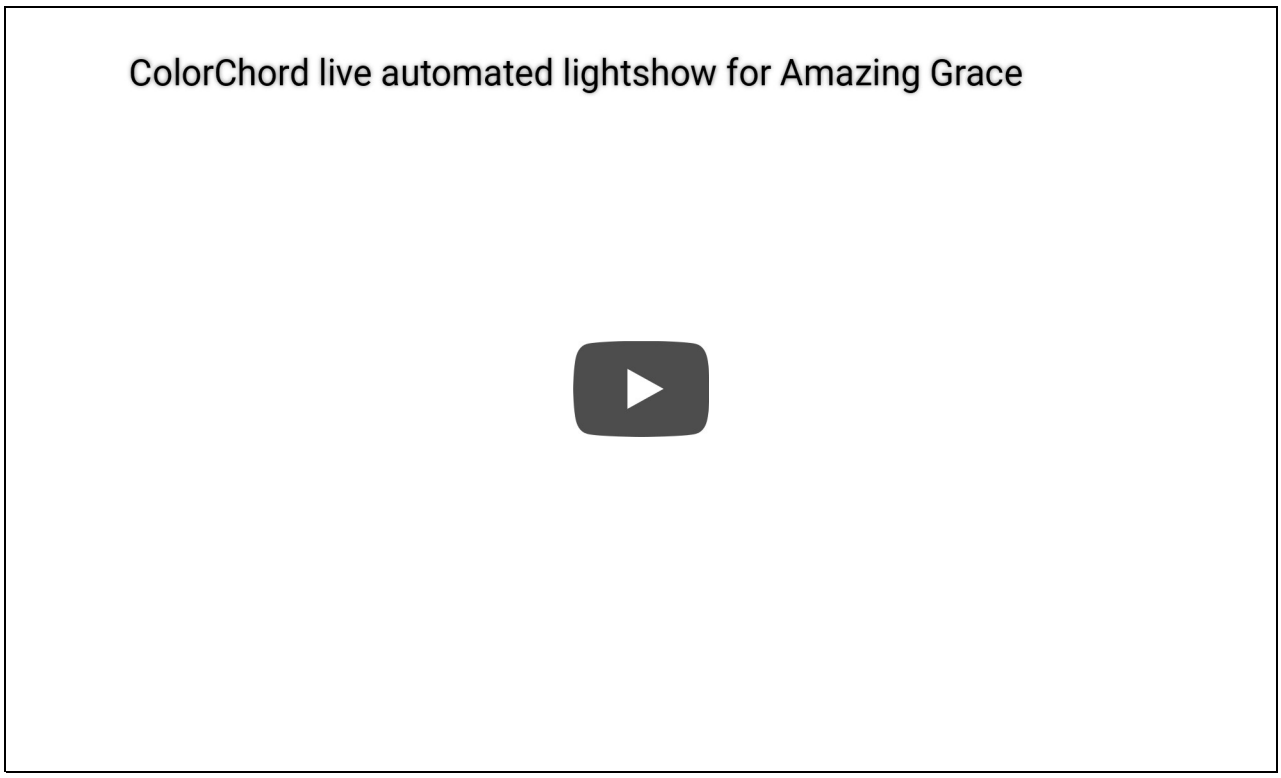
If you want to use this technology commercially or just for for a project, feel free to contact me.

Note that all of this material is patent pending.

Another few videos:







POSTED BY CHARLES AT 8:40 PM
LABELS: AUTOMATED, CHORDS, CHROMATIC, COLOR, COLORCHORD, DMX, GUITAR, LIGHTING, LIGHTS, LIGHTSHOW, LIVE, MIDI, MUSIC, SOUND, SYNESTHESIA

20 COMMENTS:

Charles said...

Sure. I am actually working on a few variants. One was done more recently for live music. I am trying to figure out a way of doing the work on an embedded system so it doesn't take a whole laptop, so we'll see where that goes. Right now, as it says, it needs a GPU to run the colors.

JANUARY 18, 2011 AT 10:35 AM

Austin Wayne Duncan said...

hey, I'm in a band that performs at local venues and is about to release a cd. I was wondering if it would be possible to get this software from you, and if it would work on a mac? thanks.
Austin

APRIL 6, 2011 AT 3:25 PM

Charles said...

@Austin - No, it does not currently work on a mac. Also, it is very specialized to work with whatever system it's being used in at the moment. If you have/are a programmer, you should be able to get it to work on a mac and bend it to your needs.

The raw (pretty gnarly) code is at https://cnlohr.net/pubsvn/projects/voicevis/

You can try checking it out and building it on a mac. It currently on accepts the guitar I have and the DMX controller I have (For lighting).

MAY 15, 2011 AT 11:56 PM

Juris 3D said...

Amazing, i am a bit speechless. I am not into guitars, but i am big fan of making any kinds of color organ, light effects electronic projects since 80-ies. I wonder if there is any chance to get technical details about this project. Its just my hobby, my "soft spot". Thanks in advance!

JULY 25, 2011 AT 12:19 AM

Charles said...

@Juris - This is totally just a hobby, you are more than welcome to rifle through the code I posted and I might be able to provide help here and there.

AUGUST 1, 2011 AT 10:19 PM

David Labay said...

I have been looking into doing this for some time. I teach this relationship to my guitar students as a way of

Charles said...

I'm terribly sorry it too so long, but I am up for helping with just about anything. I recommend contacting me via youtube, since I am much, much more responsive there.

FEBRUARY 5, 2013 AT 9:59 PM

Unknown said...

Hi, where can I get the code to try this? looks awesome, and what program did you write the code in?

MAY 21, 2013 AT 5:17 PM

Anonymous said...

No offense.
When playing out of C chord and on your color wheel: why take blue for A if A is at lower frequency than C. Surely the lowest frequency tone should be Red, which is the lowest frequency color. And the highest frequency tone must be violet. Why?
Simply to let the lights play by light frequencies corresponding to the music frequencies. Check out my project to try and make music visible for once and for all and for the deaf. Wanna collaborate?
https://sites.google.com/site/worldwishweb/music-color-chords

JANUARY 7, 2014 AT 5:50 AM

Anonymous said...

No offense.
When playing out of C chord and on your color wheel: why take blue for A if A is at lower frequency than C. Surely the lowest frequency tone should be Red, which is the lowest frequency color. And the highest frequency tone must be violet. Why?
Simply to let the lights play by light frequencies corresponding to the music frequencies. Check out my project to try and make music visible for once and for all and for the deaf. Wanna collaborate?
https://sites.google.com/site/worldwishweb/music-color-chords

JANUARY 7, 2014 AT 5:52 AM

Charles said...

Interestingly enough, I am juuust about done ColorChord 2. Much more versatile and a lot faster. No need to run on the GPU. https://github.com/cnlohr/colorchord

Check out the video of it in action: https://www.youtube.com/watch?v=UI4eqOP2AU0

About color selection: Be careful that you don't have chromesthesia. That'll throw a lot of things off with your opinions of what color is what. My color wheel is based on informally asking a lot of people and I can

JANUARY 6, 2015 AT 9:23 PM

Andrew Coulson said...

Hi Charles, awesome work! I've been playing around with using FFT (on a teensy) to generate RGB color levels in sync with music and this is much better. A couple of questions: Do you believe the ESP code would port easily to the ESP32? Does running colorchord preclude the use of wifi? (thinking that wifi stack is probably excluded from the build and that even if it's not, colorchord might be so processor intensive that wifi stack couldn't be serviced) And, finally, roughly what would be involved in hooking in a different output method?

NOVEMBER 29, 2017 AT 3:24 PM

Unknown said...

I too would love to use this code to create a SolFa education helper to see the pitches as the kids sing.

The color mapping of pitch class can be used to create a helix of 12 LEDs per turn, like some folks did at USC in a video, so that choice was exactly what is needed.

The ESP32 seems to be perfect for this kind of project and I too was trying to see what I could do with the FFT and FastLED libraries now that FastLED supports the rmt so the ESP32 can do WiFi and bluetooth while controlling LEDs and using other inputs to make the LEDs interactive.

Not sure if this is a dead project but it seems you have done a lot of work to get a long way to a tool that could be used by music educators all over this globe. Tech people can do so much to change music education if they wanted to help music teachers. It's sad that with all this tech there are virtually no open source tools for music education. There is nothing like SolFa for kids but even among adult musicians in the US the majority cannot pick up a score and sing it.

MARCH 9, 2018 AT 2:16 PM

Andrew Coulson said...

@Steve Anken, although I didn't get a reply from Charles (he seems like a very busy guy :), your post popped up in my email and got me looking at this again. That led me to the discovery of this YouTube video: https://www.youtube.com/watch?v=qE3zEM8qM00, which includes links to an IEEE Spectrum article he wrote, as well as the github repo for the ColorChord code. I'd looked at the code briefly before, but it's all a bit lower-level than I'm used to (I haven't dove any deeper than using Arduino for ESP projects) and I couldn't deduce much about what is really there for the ESP.

The IEEE article was super-informative, though. It's clear that his code can not only run on the ESP8266 (without even over-clocking!), but drive a string of WS2812's (so you probably don't need the FastLED lib if

up in VirtualBox or something.

Unfortunately, another thing that is pretty clear is that there would be a fair amount of work involved in building it for the ESP32. Significantly, he uses the NONOS (No RTOS) SDK, for which no equivalent exists on the ESP32. (Although I DID find a thread indicating you could potentially run a non-preempted process on a thread with the non-WiFi-servicing core dedicated to it, which seems ideal for this application! way beyond me though.) Stuff he did like building custom ADC sampling and repurposing I2S for driving the WS2812's would also need to be re-done (although the latter may just be a port of his excellent esp8266ws2812i2s library, which I've used in the past). So unless Charles get's back into it for some new project or someone else pretty skilled comes along and 'adopts' the idea, I'm not thinking there an easy path to ESP32. Still, an overclocked 8266 should be plenty!

MARCH 10, 2018 AT 9:04 AM

Andrew Coulson said...

@Steve Anken, you might also want to check out the Teensy 32 (https://www.pjrc.com/store/teensy32.html) and, especially apropos to your applications, the Teensy audio library (https://www.pjrc.com/teensy/td_libs_Audio.html). While it doesn't do the chord "bucketing" it does have FFT libraries that are very adequate for analysis of tonal frequency levels. You can drive WS2812's directly from it using this library (https://github.com/PaulStoffregen/WS2812Serial). If you need WiFi, you can always pair it with an ESP8266 connected via serial lines or whatever. I did essentially that in one of my projects: https://austinlightguy.wordpress.com/2016/03/15/lightsuit-v2-0-part-2/

MARCH 10, 2018 AT 9:29 AM

Charles said...

Somehow, I don't get comments any more on this, please visit the github colorchord page. I strongly recommend just using the 8266, since it has so much horsepower it's got a fair bit left over for other stuff afterward. Regardless, discussion should only happen on the github page.

JULY 19, 2018 AT 7:29 PM

Brother One said...

I'd love to whip up the embedded version of this, trouble is, never have done anything embedded. Any links/guidance for the neophyte? I tend to be a Maker-of-things, can solder, etc., but this is a whole new realm for me. Looking to build the device as shown in your sax video.
Thanks in advance if you can guide me in the right direction!

JANUARY 10, 2019 AT 9:54 AM

Andrew Coulson said...

@Brother One, as Charles mentions a couple of messages up, the place to start is https://github.com

JANUARY 10, 2019 AT 10:08 AM

Post a Comment

Newer Post Home Older Post

Subscribe to: Post Comments (Atom)

BLOG ARCHIVE

► 2014 (2)

► 2012 (1)

► 2011 (2)

▼ 2010 (2)

▼ November (1)

ColorChord Sound Lighting

► June (1)

► 2008 (1)

ABOUT ME

CHARLES

VIEW MY COMPLETE PROFILE