Harmonizing Sound and Light: MIDI-Enabled LED Visualization System

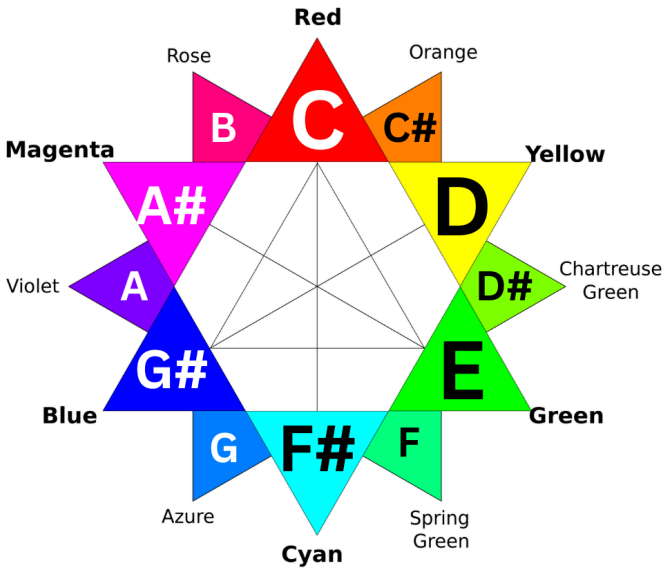
Welcome to the “read me” document for my COS981 independent work project. Here, you will find some more specific information about the code used to create the music visualization system in the form of answers to specific and general questions.

How do the colors of the LEDs correspond to the pitch of the musical notes?

In music, the chromatic scale consists of 12 notes with 12 distinct pitches. Knowing this, I divided the RBG color palette into 12 distinct sections, each of which can be mapped to a specific musical pitch. For instance, the musical note “D” maps to (255,255,0), a color that most would refer to as “yellow.” As the chromatic scale repeats in either direction, so does the color palette.

X (MIDI Pitch Value)	X mod 12	Musical note	RGB color scheme	Color name
60	0	C	(255,0,0)	“Red”
61	1	C♯ / D ♭	(255,128,0)	“Orange”
62	2	D	(255,255,0)	“Yellow”
63	3	D♯ / E ♭	(128,255,0	“Chartreuse”
64	4	E	(0,255,0)	“Green”
65	5	F	(0,255,128)	“Spring”
66	6	F♯ / G ♭	(0,255,255)	“Cyan”
67	7	G	(0,128,255)	“Azure”
68	8	G♯ / A ♭	(0,0,255)	“Blue”
69	9	A	(128,0,255)	“Violet”
70	10	A♯ / B ♭	(255,0,255)	“Magenta”
71	11	B	(255,0,128)	“Rose”

This is a chart displaying the translations between MIDI values, musical notes, and colors used by MIDIVis. The chart starts at MIDI pitch 60 (C3, middle C) and ends at 71 (B4), assigning each of the 12 notes to a specific RGB color.



This is a color wheel that shows how musical notes are mapped to colors in the music visualization system. Notes a semitone apart have neighboring colors; notes a whole tone apart have one color between them.

Why does this system use the Hairless MIDI to Serial Bridge and the loopMIDI virtual port?

To make a long story short, getting an Arduino to *read* MIDI information was much easier than getting it to *write* MIDI information. I tried many implementations where the Arduino would send all of the data by itself for auditory output without the need for any other software, but there was no method that really accomplished what I set out to do. Perhaps in a more advanced implementation of the system, I can find a way for the system to operate using only the Arduino software.

Does this music visualization system work with any MIDI controller?

At this stage, no. The system is only designed to read input from MIDI buttons. As such, controllers with knobs, sliders, or other input devices will not register with the system in the same way. Most controllers that use digital piano keys will work with full functionality.

However, I know for a fact that there does exist support in the Arduino MIDI library for other controllers. If and when I continue to work on this system, one of the first things I want to do is

develop functions that work with these other input methods. Perhaps a knob could be used to change the brightness of the LEDs, or a slider could alter the color palette entirely. The creative possibilities are (somewhat) endless now that I know how the system generally operates.

Does this music visualization system work with any digital audio workstation?

Yes! I used Ableton Live because I was very familiar with it, and it was easy for me to incorporate the virtual MIDI ports into it for data transmission. But the code itself isn't tied to Ableton in any way; any other DAW that you like more (Logic Pro, GarageBand, etc) should also work, provided that you have the loopMIDI virtual port and the Hairless MIDI to Serial Bridge set up in the same way that I did.