ELEC 327 Final Project Report

By: John Reko, Peter Humphreys, and Sean Hamilton

Concept: Our idea for this project was to create a design utilizing the ADC module on an MSP430 to create an LED system that responded to the audio being inputted. Inspired by the ELEC 244 final project, we wanted our LEDs to respond differently to different frequency ranges of the audio that the user was playing. Our ultimate goal was to create a light up speaker that users could plug a 3.5mm audio cable into, and watch as the LEDs put on a light show in response to the volume levels of the different instruments being played in the song.

Design: For our input, we used a 3.5mm through hole aux connector. This is the standard input for computers and phones, so it allows the user to easily connect their device to the module. The inputted audio signal led to two different places. First, it led to a potentiometer which was then connected to an audio amplifier before being outputted through a speaker. The potentiometer was included so the user could adjust the volume. Additionally, the inputted audio signal was passed through 3 different bandpass filters that were constructed from ADA4098-2 op-amps, resistors, and capacitors, as shown in Fig. 1.

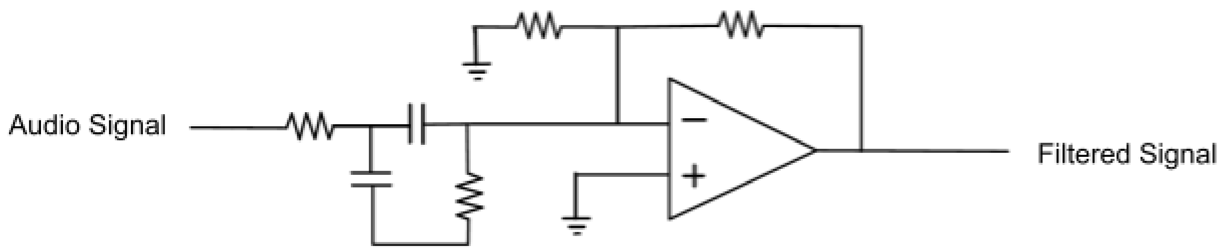

Fig. 1 Bandpass Filter circuit

The output of each of these filters led to a different input of the MSP430G2553. We chose to use the G2553 version for its easier implementation of ADC samples. Connected to the MSP430, we had these three audio signals connected to p1.0, p1.1, and p1.2. In addition to the ground, voltage, test, and reset connections, we also had a button connected to p2.2 and a data output connected to p2.5. Pin 2.2 was used to connect to a button which the user could use to toggle the lighting mode. The data output from the MSP430 led to an LED connector, meant for a ws2812b LED strip, which is where the light would be shown.

For the layout of the board, a few more ICs were needed to complete the circuit. Firstly, audio amplifiers were needed to operate the speaker, so we found some low power amplifiers that ran off a 5V power supply. This led to us needing a boost converter to get 3.3V to 5V. The audio amplifiers in question were dual input amps that were connected to an appropriate feedback network to achieve the desired gain. One amplifier was used to attach to the speaker while another amplifier was used to connect to the frequency spectrometer that was made from passive elements and smaller op amps. The reason two audio amplifiers were used was because one of the amplifiers had a useful breakout board that allowed us to attach a speaker more easily. For the frequency spectrometer, we used smaller op amps that use a 3.3v power supply. These allow us to directly put the output into the MSP430 and use the ADC. A block diagram of the hardware of our design and a picture of the board can be seen below (Fig. 2). The board was laid out so that some resistors and a battery pack were on the back. This was done to make the board more compact as we had a lot of resistors and capacitors. To attach the LEDs and speaker amplifier, we had pins on the side of the board that are used to interface with external components.

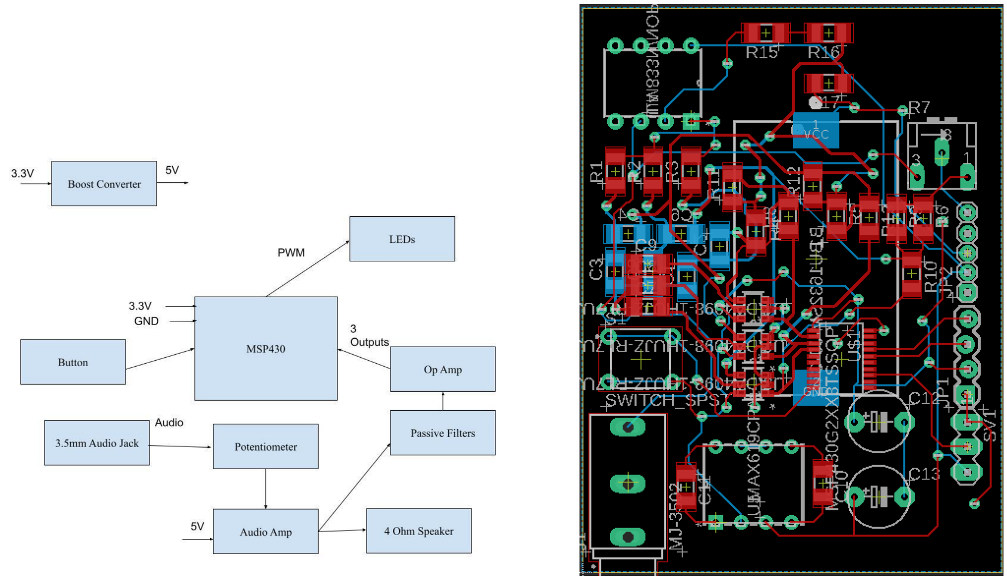

Fig. 2 Board diagram and board design

To build the board, we tested components with a solderless breadboard initially and then applied them to the PCB. The potentiometer was placed on the backside of the PCB as it was too close to one of the pins on the edge. There were a lot of surface mount parts and in the lab we were not able to find all the correct resistor and capacitor values for our band pass filters. We opted to use different values but try to get as close as possible on our filter characteristics given the components in the lab. Additionally, two 0.22uF capacitors were needed for the boost converter, so we had to use a through hole capacitor and solder the leads to the surface mount pad. Given errors with the LM833P audio amplifier, we had to change to a LM386 audio amplifier that we attached to an auxiliary board. This design change made the project less elegant, but gave much better results for audio amplification. Another auxiliary board was used to connect a string of LEDs to the MSP430. The LEDs were put on another board and were interfaced with a jumper wire attached to the pin on the main board (Fig. 3). We initially sought to use an LED strip that was programmable with an Arduino library and modify the library for the MSP430. However, this was not possible to implement with the MSP430, so we substituted standard LEDs into the auxiliary board and used PWM to alter their brightness. After soldering the board, we were able to see that the analog frequency network was functional and gave varied inputs based on frequency into the MSP430. This test was done using a NI virtual bench to set VCC = 3.3V and GND = 0V.

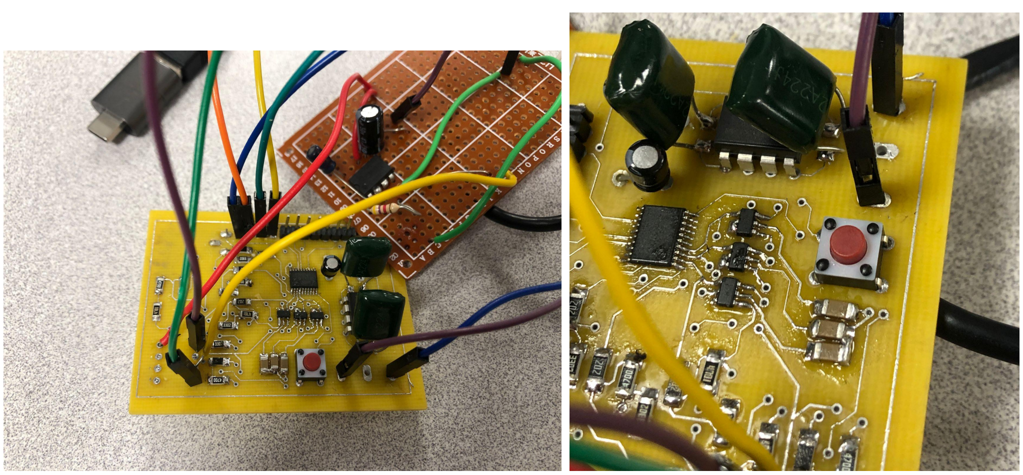

Fig. 3: Auxiliary board connected to whole board (left), MSP430, frequency spectrometer, button, and boost converter attached to the board (right)

Code: The code for this project ended up being relatively simple due to the fact that most of our complexity was within the actual hardware implementation and PCB design. Our main subject of interest was our ADC implementation. This was due to the fact that this was the main method by which we would control the LEDs that would be affected by the audio being input to the system. To this extent we used the ADC capabilities of the MSP430 on the three available pins that we could. The idea behind this was that each would correspond to different frequencies, and would thus enable us to define the noise levels coming from different instruments. Initially this was to be used to control the brightness and colors of the LED strip we had through SPI, however, we ran into issues with this as it did not respond to SPI and instead had a custom communications system. Due to this fact we instead decided to focus on PWM and change the brightness of our LEDs depending on the signal. This was done by changing the duty cycle of our PWM depending on the ADC value that was read out from one of our pins. Along with this our actual implementation was dependent on the button that was on our board as this either enabled or disabled the LEDs. We decided to add this in case someone wanted to use the speaker to listen to music or to maybe go to sleep to white noise without having the lights to match it that could keep them awake.

Errors: Unfortunately we ran into many errors during our construction of our project and implementation of our hardware. One of the first problems we encountered was with our 3.5 mm input, where we found that although we were able to propagate the signal to its outputs, once we tried to run it through a wired to our board it would instead turn into a single harmonic. Along with this we had more component issues with our op amps and LED strip that we tried. With one of our op amps despite our calculations being correct and checking with simulation, the feedback circuit we created did not produce the gain that we expected and this led to us having to change up our design to accommodate. The LED strip on the other hand simply just did not communicate with the MSP430 as it used its own custom communication system and so we were unable to control it using PWM as we had initially planned. After further research we realized that SPI would be better to control the strip, but unfortunately it was too late to change our board and we did not have an SPI output connected to the LED input. Despite an excellent soldering job that was checked and rechecked again to make sure there was no shorting of any kind, we still experience issues with our board. One of these issues was something we encountered when we would connect our board to the ground pin on the MSP430. If we left our board connected to ground on the NI virtual bench, we could check inputs and they would read as intended, showing what appeared to be music coming from the audio input. However, when connecting our board to ground, most inputs would instead turn into nearly pure sinusoidal signals and we were unable to get the proper readings from our ADC due to this fact. Upon connecting to the launchpad no inputs would register correctly and would instead just appear like ambient noise. Along with this our ground plane would start reading a sinusoidal signal with an amplitude equal to what our audio input should be. In our investigation as to what was causing this we tried measuring the dc values coming from the launchpad itself and we found that from the 3.3 V pin we read an output of 0 V, and from the ground pin we read an output of -3.3 V. Despite trying multiple different launchpads we encountered this every single time with the same effect on our ground plane. We feel that there may have been a short that we could not find, but we did not have enough time to fix this issue. In the future, we would try to use less surface mount components if we are hand soldering the board as it made it very difficult to actually attach all the components. We would also try to leverage the power of the MSP430 more and connect SPI LEDs to the board to create a more dynamic visual display.

Overall, this project featured a lot of analog signal processing that made our PCB have a lot of components. This made the manufacturing process difficult and cost us time that we could have used for testing the software. A large scale solderless prototype of the PCB before could have helped us solve some connection issues. We were able to create a frequency spectrometer that operated on the board and wrote a program to access the ADC values, but we were unable to finish building the design as we had errors that would have required us to start over.