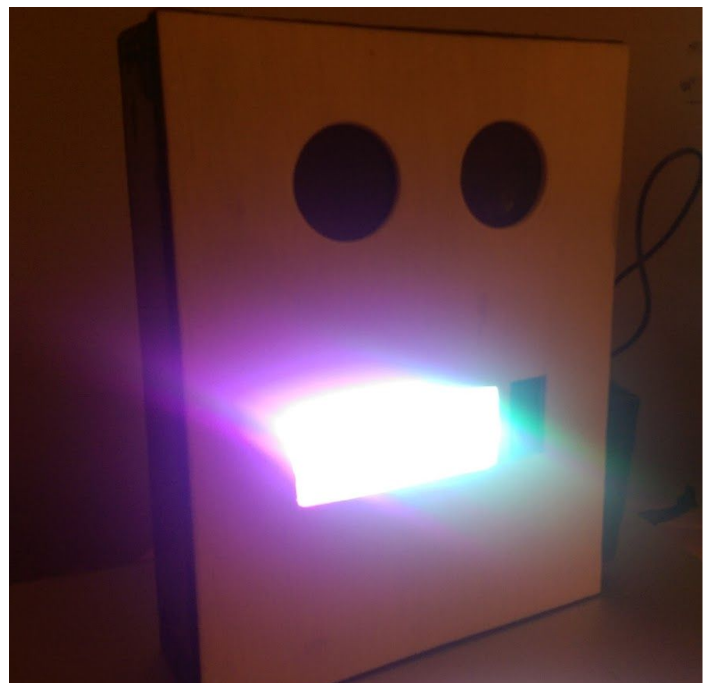
Portable Wireless Speakers with LED Capabilities



EE 403W Final Design Report

Team JAK EE chan

Members: Jeremy Doll, Ahmad Mostafa, Keil Toso

Executive Summary

Many types of wireless speakers based off of the use of Bluetooth connectivity are available today. However, few provide lighting, and those that do, do not offer user-control over the lighting experience. Based off of the usefulness and practicality of Bluetooth speakers, team JAK-EE chan saw an opportunity to capitalize on existing technologies to implement a user-controllable wireless speaker set with lights.

The solution is a set of portable, wireless speakers with LED strips that light to different audio frequency ranges. The colors the LED strips emit can be selected by the user using two push buttons (one for each LED strip). There are two LED strips: one associated with bass frequencies and the other for treble frequencies. To have the system be easily transportable by one person, the entire speaker system, including the LEDs, is enclosed in a small wooden chassis. The resultant system measures 11” x 8.5” x 2.875” and weighs less than one pound. Furthermore, the LED’s take on seven distinct colors, with 8 single LED’s in each strip. The entire system can play music, with the LED’s responding, for over six hours before requiring the battery to be re-charged. Additionally, with no physical obstructions, the system was found to have a range of 45 feet between the system and the Bluetooth-capable device.

Figure 1, below, is a block diagram representing the different components visible to the user, and all the hardware hidden inside the casing.

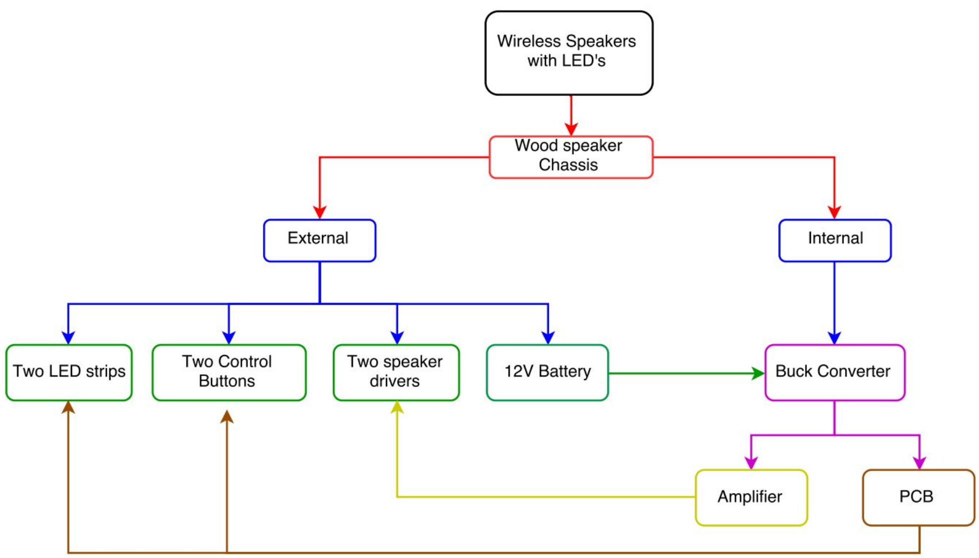


Figure 1: Speaker System Block Diagram

All of the goals of the project were met or exceeded, except for the implementation of a third LED strip. The original design contained another LED strip that would respond to midrange frequencies. However, due to a soldering mistake and time constraints, team JAK-EE chan decided to complete the project with only two LED stirps. The original cost estimate projected a total cost of $51,887.17, while the actual cost was only $50,772.54.

Problem Statement

With the wide availability of devices such as smartphones, people turn to streaming services or storage on multimedia devices for music listening. Dedicated technologies, such as CD's holding music, are no longer the norm. As a result, audio playback devices must have wireless connectivity with digital devices. Advances in Bluetooth capabilities enable the ability to seamlessly and quickly connect to playback devices wirelessly.

Many types of wireless speakers, based off of the use of Bluetooth, are available today. However, few provide lighting, and those that do, do not offer user-control over the lighting experience. Based off of the usefulness and practicality of Bluetooth speakers, team JAK-EE chan saw an opportunity to first implement a user-controllable wireless speaker set with lights. The team drew inspiration from various speaker designs currently on the market.

Such speakers offer a more engaging musical experience. Team JAK-EE chan decided to create a system that not only offers user-input, but which has lights that respond directly to the music. The light response would be such that there would be a clear correlation to certain tones or beats and the flashing of the lights.

Proposed Solution

The solution is a set of portable, wireless speakers with LED strips that light with different audio frequency ranges. For this solution, an LED strip is defined as a set of closely spaced (less than one inch apart) LED's in a horizontal line. The colors the LED strips emit can be selected by the users. There are three LED strips: one associated with bass, middle-range, and treble frequencies.

To achieve the above-described system, the following design goals were set as part of a requirement specifications:

- The system shall output audio selected from Bluetooth-capable devices to two speaker drivers.
- The system shall illuminate three LED strips consisting of at most 10 LED's each.
- The LED's can alternate between at least three different colors.
- Speakers can be carried and moved by one person.

In addition to the design goals, the performance parameters found in Table 1, below, were set as requirements.

Attribute	**Specification**
Battery	Rechargeable, polymer Lithium-Ion: 1500mAh, 9.9V
Run-time	2 hours minimum before battery needs to be re-charged

Size	No larger than 36in x 36in x 36in
Weight	No more than 10 lbs
LED Supply Voltage	5V
Minimum Speaker Voltage	5V

Table 1: Performance Parameters

Implementation Details

Hardware:

The speaker system is divided into two main components: the sound system output and the LED output. Both components, however, receive the same analog signal through a Bluetooth module. To connect the speaker system to any Bluetooth capable device (such as smartphones, tablets, laptops, etc.) the Amebay Bluetooth Audio Receiver module was implemented. The module has already been pre-configured by the manufacturer of the product to connect to any Bluetooth device. Through a USB connection, the module only requires 5V power (no data connections) to run. Once a wireless connection has been established, the analog signals can then travel one-way from the Bluetooth device to the speaker system. From there, the analog signal is sent to the two subsystems of the device via an auxiliary cord, which can be seen in Figure 1 below.

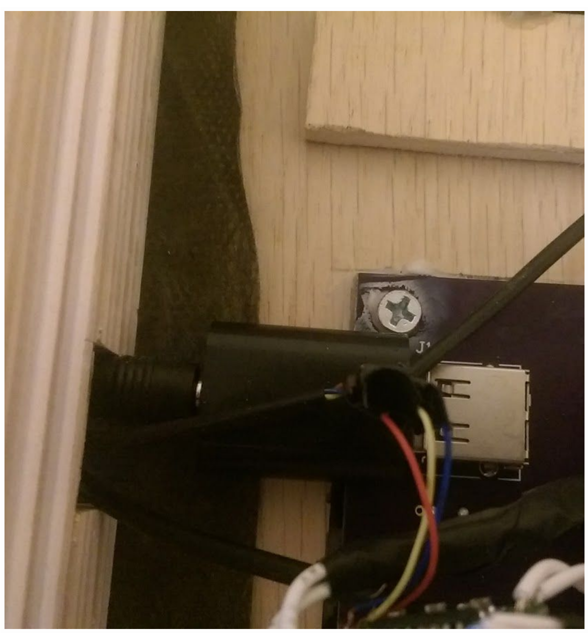


Figure 1 : Bluetooth module with AUX cord output

The first subsystem is the sound output. To configure this, the auxiliary cord coming out of the Bluetooth module has been stripped to reveal three wires (ground, left output, and right output.) Each of the three wires is then soldered to a small audio amplifier, specifically the Uxcell D Class Amplifier Board shown in Figure 2. The amplifier board has three input ports where the ground, left and right outputs can be connected. In addition, there are two other input ports used to power the amplifier chip with 5V (positive and negative terminal.) Once the input is received to the amplifier, it is then output from the amplifier as well using four more ports on the chip. Two are for the left speaker output, and two for the right speaker output (positive and negative terminals for each.) One end of a copper wire is then soldered to an output port, while the other is then soldered to one of the positive or negative terminals of the 50mm 3W Speaker Drivers. A flow diagram of the sound output subsystem can be seen in Figure 3.

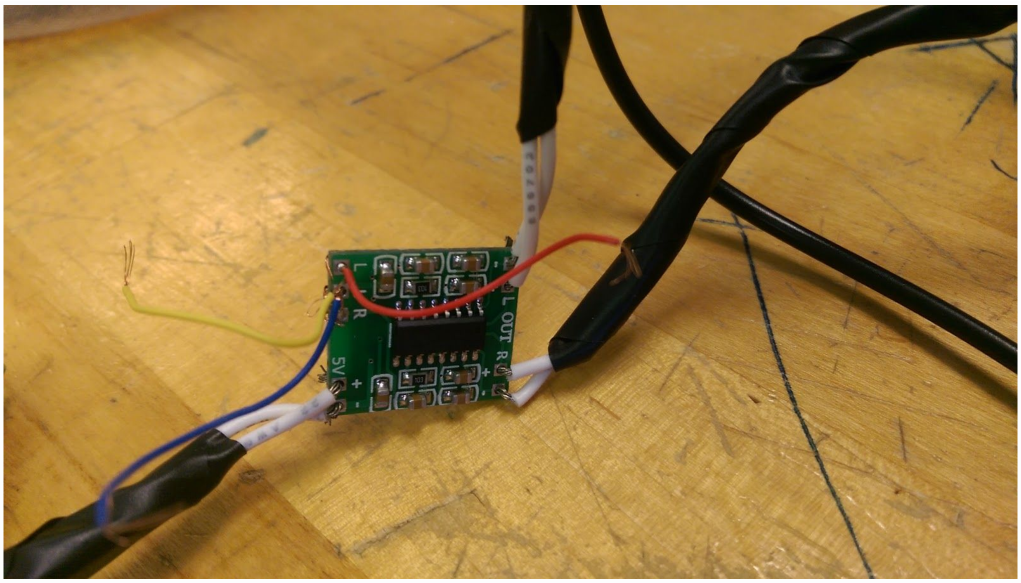


Figure 2: Uxcell D Class Amplifier Board

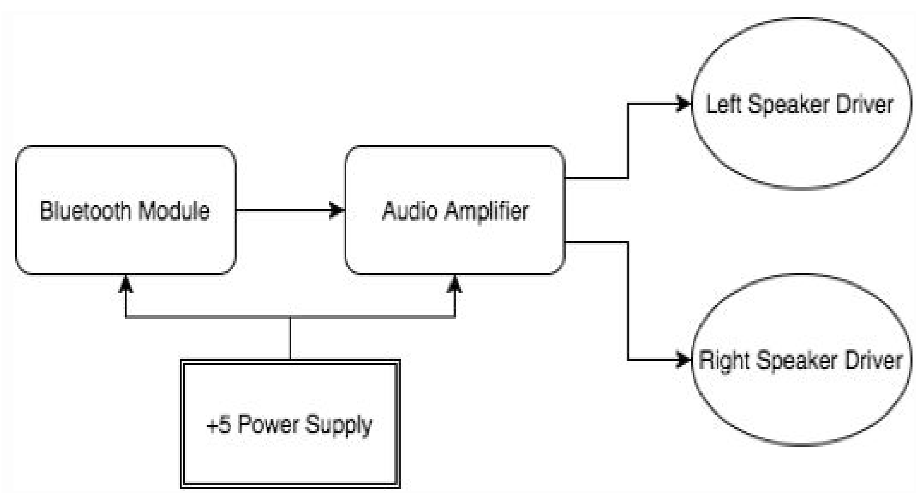


Figure 3: Flow diagram of the sound output system

The second subsystem is the LED output. The three LED strips are designed to activate with the three different frequency ranges, and change color depending on three binary control buttons. All the LED manipulations must be programmed through software. To do this, the Teensy 3.2 microcontroller with the Audio shield adaptor add on is used, as seen in Figure 4. Once again, to connect the Bluetooth module with the Teensy microcontroller, 3 copper wires are soldered to each of the three wires of the stripped auxiliary cord (ground, left output, and right output.) On the Teensy, there is a 'line in' port, where each of the three wires can be connected. This port is used to receive the analog signal and send it to a PC where it can be manipulated through software. The three binary control buttons are also connected to the Teensy via pins 16, 17, and 20. After the microcontroller receives the analog signal and the binary values from the push buttons, it then outputs the commands to the set of the APA102 LED strips. Each component in the LED output subsystem is powered with 5V. A flow diagram of the sound output subsystem can be seen in Figure 5.

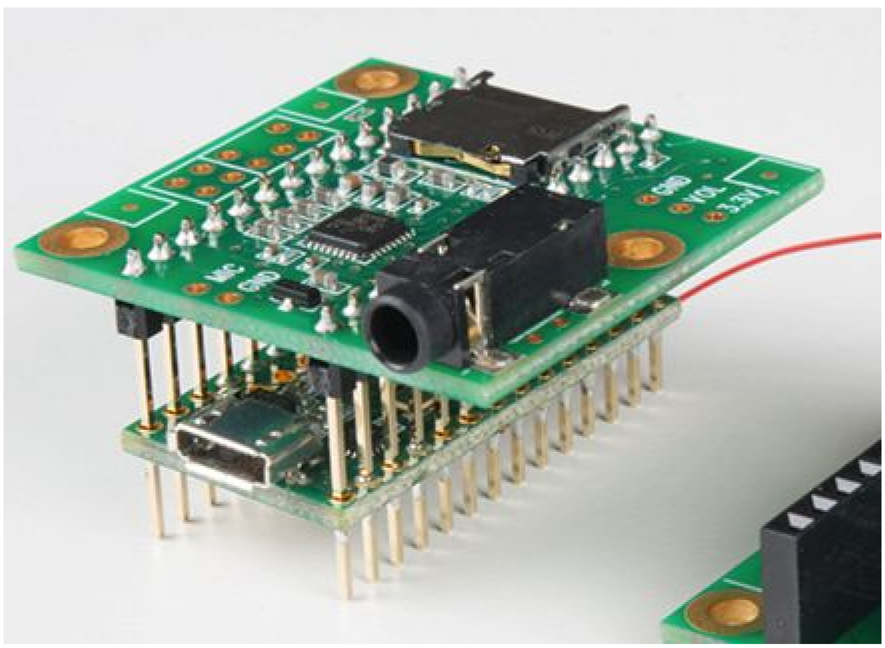


Figure 4: Teensy 3.2 Microcontroller with Audio Shield add on

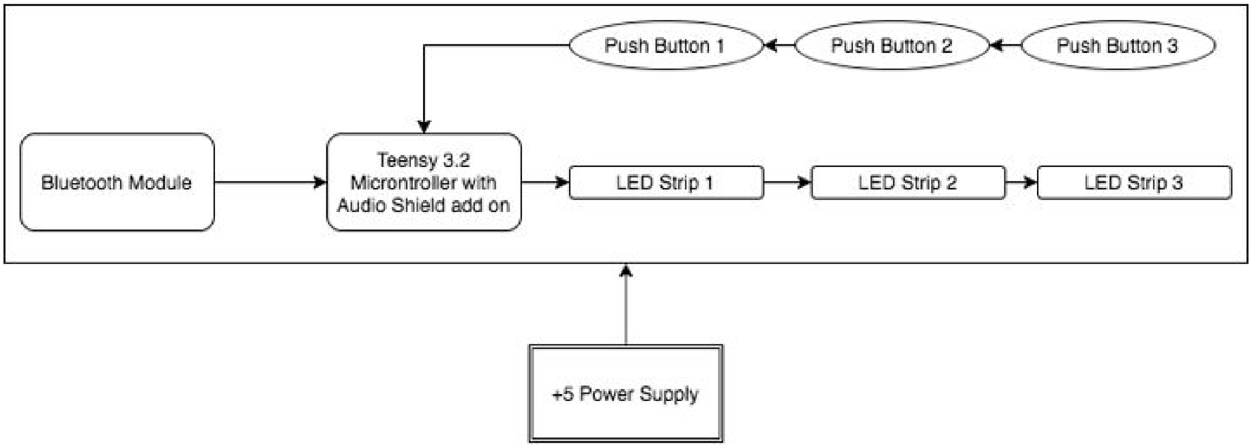


Figure 5: Flow Diagram of LED Output Subsystem

Every single component in this device requires 5 volts. Having all LED's enabled simultaneously would draw about 2 amperes from rough calculations. However, when testing our device with a power supply, the current never exceeded 0.5 amperes. Still airing on the side of caution, team JAK-EE chan selected a 12V, 3800 mAh battery, specifically, the HitLights Rechargeable 12V Battery This battery has two external features: an on/off switch and an output/charging port. The on/off switch is conveniently used to turn on and off the whole speaker system. The port is used to charge the battery and also output 12V. This means, that when using the device, one can either charge the battery or use the speakers independently. Both functions cannot be done simultaneously. To regulate the 12V power output to 5V, the SMAKN DC/DC Converter (buck converter) is used. This can then reduce the voltage without reducing the maximum current draw of the battery. The buck converter is then connected to the PCB which contains all the components of the speaker system. A flow diagram of the power subsystem is shown in figure 6.

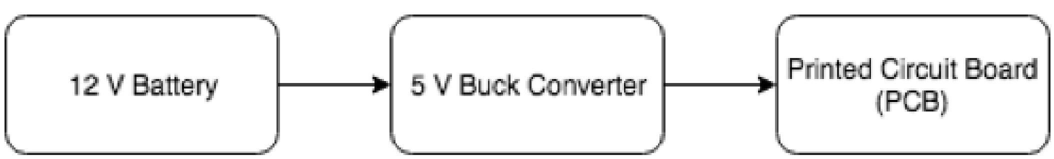


Figure 6: Power Design

Printed Circuit Board (PCB):

The circuit board provides connections between the Teensy microcontroller, LED's, and buttons, while also providing power ports for external components, and a USB connector for the Bluetooth module. As seen in a portion of the full schematic below in Figure 7, the pins selected as the clock and data pins connect to the first LED, whose data out and clock out pins connect to the data in and clock in pins of the next LED. This process continues, so that the LED's are "daisy-chained" together.

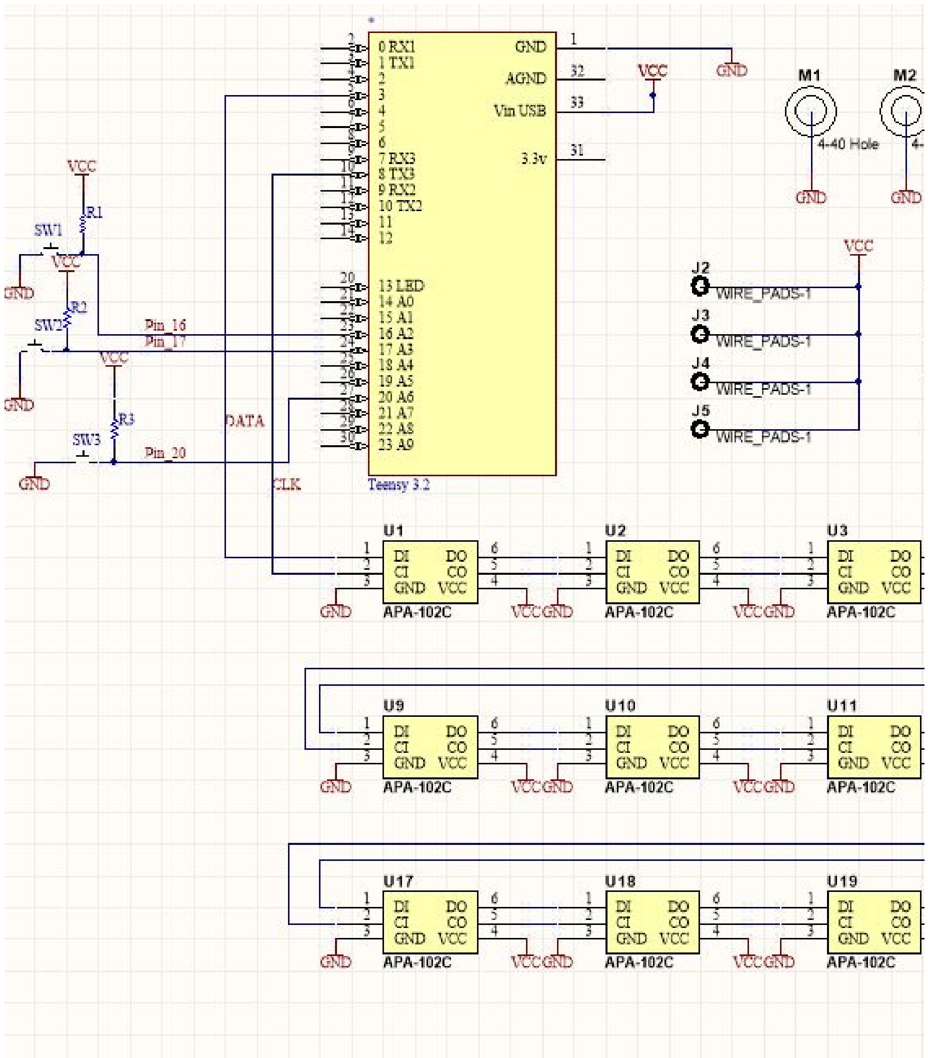


Figure 7: Partial schematic of the circuit on the PCB (USB connector and other LED's not shown)

The circuit controlling the buttons works by the Teensy analog input pins reading the voltages they receive. Without a button being pressed, the button input is "floating" by not closing the line to ground, and so the 5V across the receiver is read by the Teensy. When the button is pressed, the 5V does not go across the resistor, as there is a direct connection to ground.

The USB connector requires to only be powered with 5 volts so that the Bluetooth module can operate, while the data connections can be left floating. A view of the actual PCB with all components attached can be seen in Figure 8. The USB connector and Teensy are not visible as they are on the back side of the board.

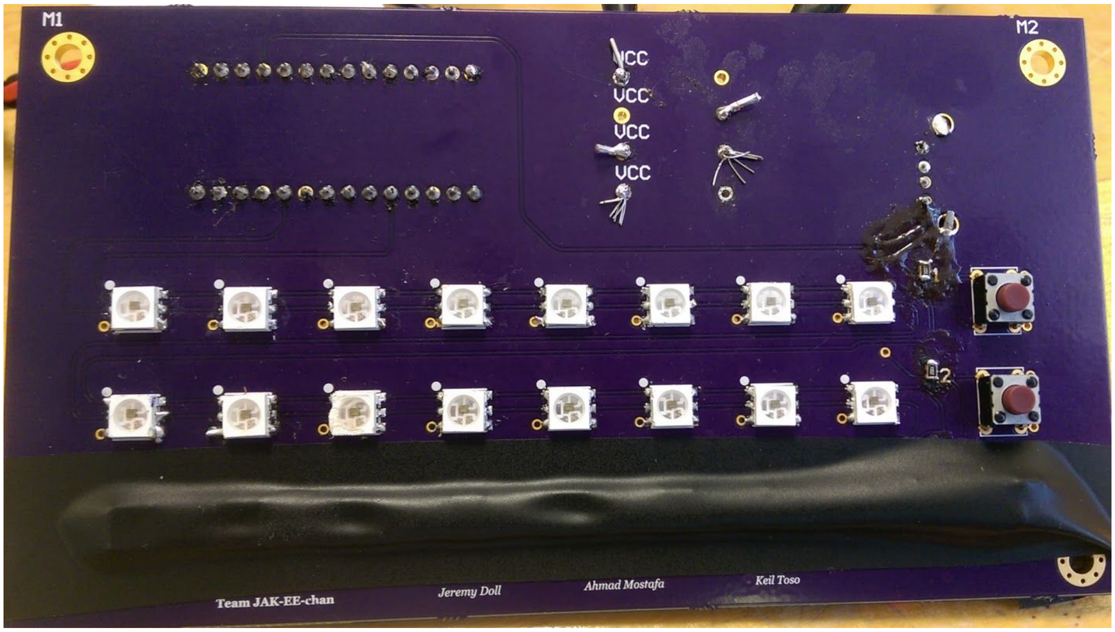


Figure 8: Finished PCB before being inserted into casing

Software:

The software for the system is a single C source file. This is written in Arduino IDE 1.6.7 with the Teensyduino 1.28 add-on. This add-on enables access to the FastLED library, as well as the Teensy audio shield functions, and the Teensy FFT functions.

After setting up the appropriate functions, a loop runs continuously, executing the code that allows the LED's to respond to the music and change color via button presses. The general architecture of the software can be seen in Figure 9 below.

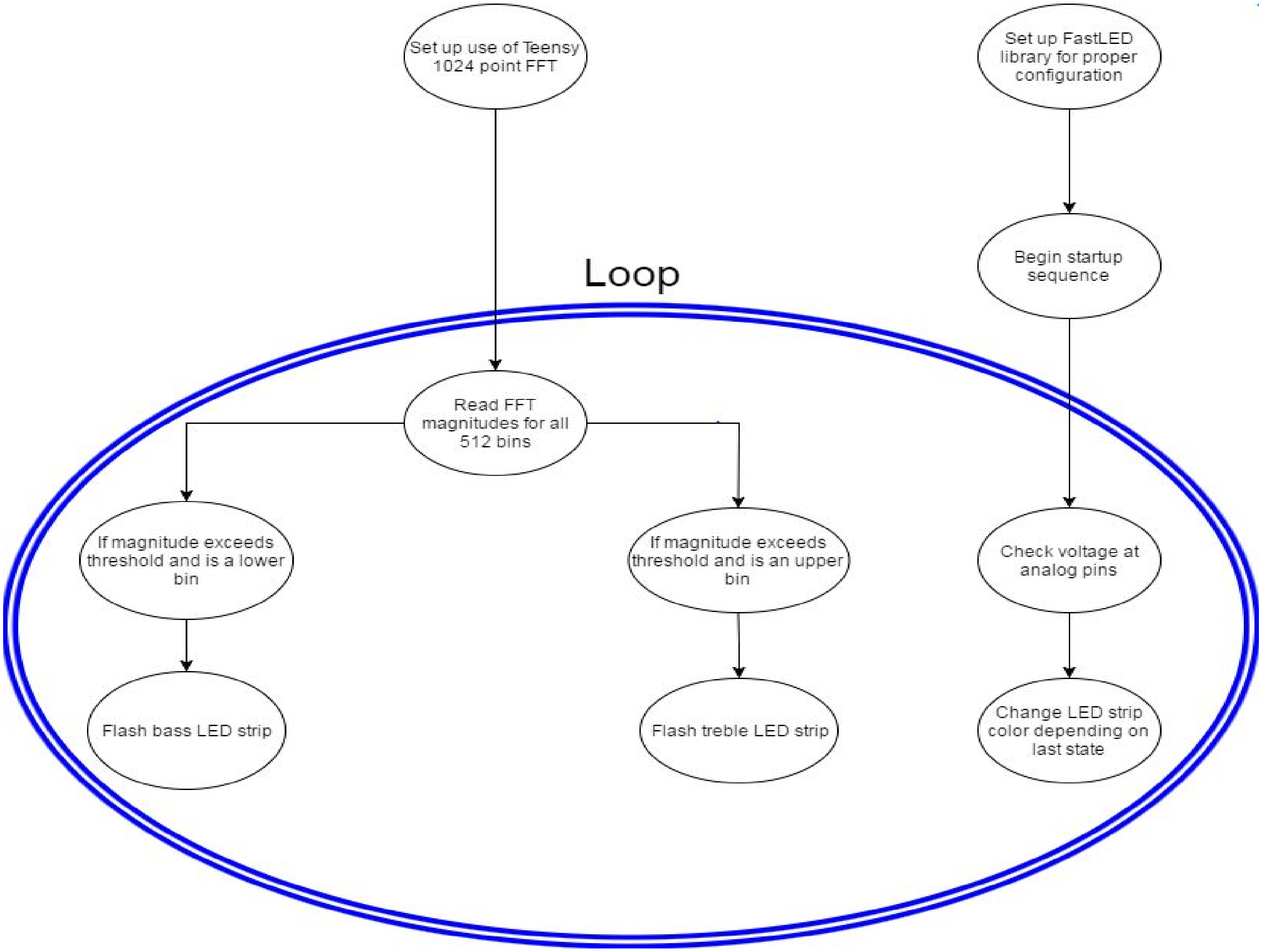


Figure 9: Block diagram of the software architecture.

The initialization for the LED's requires including 'FastLED.h' and then defining which pin from the Teensy will serve as the clock, and which will send data. These functions were carried out on pins 8 and 3, respectively. Also, the number of LED's being utilized is selected. Reading audio from the audio in pins on the Teensy is defined by being in I2S. The 1024 point FFT must be defined, and the audio shield enabled. This code can be designed by a block diagram that takes the I2S audio input and performs a 1024 point FFT in the Audio System Design Tool by Teensy. This tool is available for free from Teensy's website, and allows one to export the required setup code for the desired functions drawn in the block diagram.

After this portion, a setup() function runs. Contained within this, is a two second delay before proceeding to allow the system to restart itself. The previously defined LED data is implemented into FastLED's functionality with the function 'FastLED.addLeds< >. Then the audio shield's inputs are defined, audio memory is allocated, and a window function is selected for the FFT. The windowing function prevents spectral leakage so that specific frequencies are centered around a few frequency bins rather than being more spread out. A 1024 point Hanning window was selected, but this setting could be tampered with depending on the responsiveness the user is looking for.

The start-up sequence then begins. Within a for-loop, individual LED's are controlled by leds[k].setRGB(). The 'k' represents the integer representing the LED to be accessed (0 to 23 in the order they are daisy-chained), and within the parentheses are three numbers, separated by commas, representing the red, green, and blue values of the output color, respectively. These numbers can be any integer from 0 to 255. The command FastLED.show() is then required to send the data to the LED's. There are three for-loops that each access one LED from each row on a single iteration, separated by a delay. These delays decrease in time from one loop to the next to create an effect that the LED's are "moving" slowly along the rows and then faster and faster. After this, the LED's are all turned off by accessing each one in a final for-loop.

The setup() function then ends. After this, a series of integer variables are defined and initialized to zero. Then the loop() function runs, which operates continuously until the Teensy loses power.

The loop begins by reading the analog voltage values at the pins connected to the buttons. Without the button being pressed, 5V (1023 from 10-bit analog to digital conversion) is read, and when the button is pressed, 0V (1 from 10-bit analog to digital conversion) is read. These values are stored into variables. Next a series of if-statements in an if-else if-etc sequence is utilized to check those voltage values. The variable 'count1' has been initialized to zero. As seen in Figure 10 below, the red, green, and blue values are decided to produce a green LED output if the analog voltage value dropped low enough (below 501) and if 'count1' is equal to 0.

```
// Row 1
if ((button1 <= 500) && (count1 == 0))
{
  // Green
  b1 = 0;
  g1 = 255;
  r1 = 0;
  count1 = 1;
  delay(200);
}
```

Figure 10: Setting of red, green, and blue values after a button press.

Also, 'count1' is then set to 1, so that the next time the analog voltage drops below 501, the first if-statement will be ignored, and the program will run into the else if-statement seen below in Figure 11.

```
else if ((button1 <= 500) && (count1 == 1))
{
    // Blue
    b1 = 255;
    g1 = 0;
    r1 = 0;
    count1 = 2;
    delay(200);
}
```

Figure 11: Setting of red, green, and blue values after the second button press.

This allows a new set of red, green, and blue values to be decided for the color blue, This design pattern is continued so that users can cycle through green, blue, red, cyan, purple, yellow, and white colors, or turn the LED's off. Note that this design pattern means the LED's in a row will not light until the button for that row has already been pressed. This design pattern is utilized for the other two LED rows.

An if-statement checks if FFT data is available, ensuring no analysis is completed or LED's are illuminated unnecessarily. Next, a for-loop runs 512 times. This represents the 512 bins containing frequency information. With a 44.1 kHz sampling rate, the highest possible frequency is 44.1 kHz / 2 = 22.05 kHz. 22.05 kHz / 512 bins yields about 43 Hz as the distance between bins. So bin 0 represents 0 Hz, bin 1 represents 43 Hz, and so on. Each bin contains a magnitude value between 0 and 1 representing the intensity of that frequency value. These values are read, and they must exceed a threshold to light any LED's. Through experimentation, a threshold of 0.05 for bass frequencies was selected. As can be seen in Figure 12 below, if the threshold is exceeded for bins less than 12, representing frequencies from 0 to about 500 Hz, the LED's for the top row are set to the red, green, and blue values defined from the button design pattern.

```
for (i=0; i<512; i++)
{
  n = myFFT.read(i);
  if (n >= 0.03)
  {
    Serial.print(n);
    Serial.print(" ");

    // Bass frequencies
    if ((n >= 0.05) && (i < 12))
    {
      for(int j=0; j<8; j++)
      {
        leds[j].setRGB(b1,g1,r1);
      }
      FastLED.show();
      delay(10);
      for(int j=0; j<8; j++)
      {
        leds[j].setRGB(0,0,0);
      }
      FastLED.show();
    }
```

Figure 12: Reading of FFT data and controlling the first LED row for bass frequencies

After a short delay, these same LED's are turned off to create a flashing effect. This design sequence is repeated for the higher frequency bins, to control the other two LED strips.

<u>*Casing:*</u>

The casing for this project was designed around the internal hardware and the portability requirement. Due to this, the first issue dealt with was the weight. To make sure the casing would be light enough to carry around and play on the go, plywood planks were the material of choice for the face and the more sturdy 2x4 option was used for the sides.

The second issue determining the casing was sound. Because we had delicate speakers that had to produce audible sound, the design had to take into account acoustics and protection. To produce a clean sound the speakers were fitted up against holes cut into the face and secured by wooden rigs to cancel out any vibrations. To protect the speaker drivers, a layer of landscaping fabric was spread across mentioned holes. This fabric was both porous to allow sound through but also very tough and hard to penetrate, protecting the drivers from damage.

The LED strips were another feature that needed to be displayed but came with a couple complications. Due to the APA102's very bright display and the desire for a more diffused light, a layer had to be made to cover the LED's to dim and diffuse them. To do this, four layers of plastic wrap were inserted in intervals between the LED's and the face. This gave the diffusing/dimming effect needed.

Finally the power circuit was the last design requirement. Due to the battery charging and powering through the same input, the battery was left on the outside of the case to allow easy access to the input in order to switch modes on the battery. To keep the battery connected to the case a mesh holder was attached to the side of the case in order to secure and maintain the battery's connection.

Manpower:

To maximize efficiency and to assure the completion of the project within the allotted deadline of one semester, each individual in the group was given a set of tasks to accomplish individually. Even though each member had specific tasks, the team as a whole made sure to give assistance where needed and to give updates on completions of tasks. The tasks were split accordingly:

Ahmad Mostafa:

- Assemble list of required parts and order the parts.
- Design power, audio conditioning, and voltage step up/down circuitry.
- Realize PCB layout and submit to prospective company.

Jeremy Doll:

- Interface between Bluetooth-capable device and Bluetooth module.
- Implement frequency analysis to trigger the LED strips.
- Program the user selection of LED colors.

Keil Toso:

- Program communication between Bluetooth module and microcontroller.
- Solder components to completed PCB board.
- Design and build speaker casing.

Parts List and Pricing:

All of the materials used, as well as their associated costs, are listed in Table 2 below.

Category	**Description**	**Quantity**	**Unit Cost ($)**	**Total Cost ($)**
Speaker Chassis				
	Wood	3 ft^2	0.42 /ft^2	1.26
	Screws	10	1.00	10.00

	Screw Driver	1	5.00	5.00
	Panasonic EVQ-PAG05R (Tactical Buttons)	3	0.27	0.81
	Super Glue	1	5.97	5.97
	Spray Paint	1	3.87	3.87
Electronics				
	Teensy 3.2 Microcontroller	1	19.80	19.80
	Teensy Audio Adaptor Board	1	14.25	14.25
	Uxcell D Class Amplifier Board	1	5.25	5.25
	Amebay Bluetooth Audio Receiver	1	9.20	9.20
	50mm 3W Speaker Driver	2	4.87	9.74
	OshPark Printed Circuit Board (PCB)	15.76 in^2	5.00 /in^2	78.80
	APA102C LED’s	30	9.27 (per 10)*	27.81
	HitLights Rechargeable 12V Battery	1	24.99	24.99
	Wurth 8492121 USB Connector	1	8.50	8.50
	SMAKN DC/DC Converter	1	6.99	6.99
Testing Equipment	Power Supply	1	400.00	400.00
	Oscilloscope	1	2,000.00	2,000.00

	Multimeter	1	100.00	100.00
Miscellaneous	Panasonic ERJ-6ENF1002V (Resistors)	3	0.10	0.30
	Shipping & Handling	N/A	N/A	40.00
	Labor (20 hours/week)	300 hours	80.00	24,000.00
Engineering Work	Overhead (20 hours/week)	300 hours	80.00	24,000.00
Total Cost ($)				**$ 50,772.54**

*LED's must be purchased in packs of 10

Table 2: List of all required parts and their associated prices.

Conclusion/Results

All of the goals of the project, as discussed in the problem statement as design goals and specifications, were met or exceeded except for the implementation of three LED strips. A soldering mistake towards the end of the allotted time presented difficulties in using the third row. The button on the third row would need to be replaced, so with the deadline approaching, and the need to construct the casing, the team decided it would be best to only utilize 2 LED strips instead. Our original cost estimate projected a total cost of $51,887.17, while the actual cost was only $50,772.54.

The resultant system measures 11" x 8.5" x 2.875" and weighs less than one pound, far exceeding the less than 10 pounds and smaller than 36 cubic inches specifications. Furthermore, the LED's take on seven distinct colors, with two strips of 8 LED's each. The top strip responds to frequencies below 500 Hz, while the bottom strip responds to frequencies above 1000 Hz. The entire system can play music, with the LED's responding, for over six hours before requiring the battery to be re-charged. Additionally, with no physical obstructions, the system was found to have a range of 45 feet between the system and the Bluetooth-capable device.

With more time, the third LED strip would be utilized by simply replacing the button for the bottom row of LED's. The software could also be tinkered or optimized by implementing interrupt service routines for setting the LED colors so as to avoid using small delays within loops. A different approach could be taken, involving the application of different filters and measuring the responses over a range of frequencies, since the measuring of 512 individual frequency bins is not necessary for the current application. Conversely, an FFT could still be used, but the LED response could be modified to respond differently to individual frequency components.

More of the design, including the buck converter, and audio amplifier could have been implemented on the PCB. Less “air-wiring” would have consolidated the design and enabled a simpler and more compact case design. On the plus side, the current design still allows the chance to implement a potentiometer to control the speaker volume. The Bluetooth-capable device must send data near maximum volume for the LED’s to respond, meaning there is currently little variation possible in the volume from the speakers. Finally, the ability to measure the remaining battery charge, and warn the user when the battery is almost devoid of charge, would be a useful addition.

Appendices

The following resources were utilized or referenced to create this project:

https://www.arduino.cc/en/Main/Software

http://circuitmaker.com/

http://fastled.io/

https://www.pjrc.com/teensy/