Electrical Overview

Year: 2017 **Semester: Spring** **Team: 12** **Project: Guitutar**
Creation Date: January 29, 2017 **Last Modified:** May 2, 2017
Author: Austin Peterson **Email: peter174@purdue.edu**

Assignment Evaluation:

Item	Score (0-5)	Weight	Points	Notes
Assignment-Specific Items				
Electrical Overview	4.5	x3	13.5	Too many specifics
Electrical Considerations	4.5	x3	13.5	See comments
Interface Considerations	5	x3	15	
System Block Diagram	4	x3	12	Not a block diagram
Writing-Specific Items				
Spelling and Grammar	4	x2	8	Some grammar and spelling mistakes
Formatting and Citations	3.5	x1	3.5	Use in-text citations
Figures and Graphs	5	x2	10	
Technical Writing Style	5	x3	15	
Total Score	90.5/100			

5: Excellent 4: Good 3: Acceptable 2: Poor 1: Very Poor 0: Not attempted

General Comments:

Generally, this assignment came out great. I would redo the figure as a block diagram, not as a pre-/pseudo-schematic. The detail is great here, but I would rearrange some of the info as it does not always seem to be in the right sections (e.g. power values in Interface Considerations?)

1.0 Electrical Overview

Guitutar will utilize a 32-bit microcontroller as the brains of the device and an external device will act as the interfacing tool. The user would send a command through the app that will send the translated song to the microcontroller. This will be done through a wireless Bluetooth transceiver using UART. The song will be translated into bytes as a result of an algorithm that will convert tab songs into code. This code is then sent to the microcontroller. The microcontroller will then send power and illuminate the LEDs based on the bytes from the algorithm. Since a chord would use no more than nine notes, the maximum number of LEDs that will be on will be nine. The LEDs are configured in a multiplexed configuration where all the positive leads are connected by column (string) and the negative leads are connected by row (fret). A shift register will be implemented that would help the LEDs iterate through the combinations. This configuration will allow the microcontroller to determine what note is pressed based on what string is sending power to which fret.

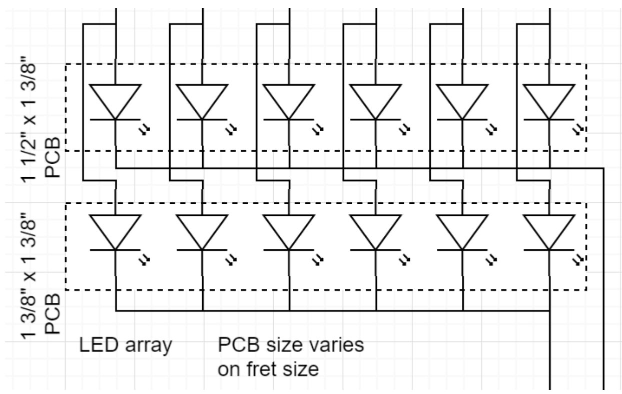

Figure 1: Multiplexed LED array

The microcontroller will then wait for the user to press the correct note(s). This is done by configuring the strings and frets as a switch matrix. Each string of the guitar will be powered with 3.3 volts from the microcontroller and the circuit will be completed by physical contact between the string and the fret. As a safety mechanism, the microcontroller will clock the power to the strings at a frequency greater than 360 Hz to prevent an accidental short across the strings. We will implement an amplifier and filter so that the microcontroller will only recognize strums from the pickup as the accepted vibration and disregard any bumps from the body or strings. This would either be done with a high pass filter or by having the microcontroller disregard any waveforms above a certain limit. The microcontroller will then interpret the strum waveform from the amplifier as a strum input. Once the micro receives input from the switch matrix and the pickup, the micro will then match the input with the tab. If it is correct, then the micro will proceed with the next note/chord. All of the components will be powered through a 3.7 volt battery that will be ramped up to 5.2 volts through a third party USB charging circuit and then regulated to 3.3 volts by a linear voltage regulator to avoid overvoltage on the microcontroller. The device will use USB, but this peripheral is only for charging the device battery.

2.0 Electrical Considerations

Selecting the right battery is important so we have all peripherals and LEDs powered and a satisfactory battery life. The peripherals we need to be powered are the LEDs, the Bluetooth transceiver, amplifier, and the microcontroller. The Bluetooth transceiver operates at 3.3V and has a maximum of 14.87 mA of current draw when operation in Data, Slave mode. The amplifier will run at 3.3 volts and will amplify the guitar pickup with a gain of 30. This equates to using a R2 of 30 kΩ and R1 of 1 kΩ. Each LED string needs a current limiting resistor to prevent damage to the LED [7]. A 1kΩ resistor was selected to get to the operating current of the LED. The selected LED has a forward voltage of 2.65 volts and an operating current of 2 mA [5]. The absolute tolerances are 30 mA of forward current and 3 volts of forward voltage. A bar chord is a chord that requires all six notes of a fret to be pressed with additional notes below that bar. This means that the most LEDs that will be on will be nine LEDs. Based on this assumption, the largest voltage drop will be 5.3 volts or two LEDs on in series. Power calculations are as follows:

$$\boldsymbol{Voltage\ drop: 5.2V - 2.65V = 2.55V}$$
$$\boldsymbol{Current\ through\ 1\ LED: 2.55V\ /\ 1000\Omega\ =\ 2.55mA}$$
$$\boldsymbol{Current\ through\ LED\ string: 2\ *\ 2.55mA\ =\ 5.1mA}$$
$$\boldsymbol{Total\ Power: 5.2V\ *\ 5.1mA\ =\ 25.5mW}$$
$$\boldsymbol{Power\ per\ LED\ string: 5.2V\ *\ 2.55mA\ =\ 13.26mW}$$
$$\boldsymbol{Total\ LED\ Power: 2.65V\ *\ 5.1mA\ =\ 13.515mW}$$
$$\boldsymbol{Total\ Resistor\ Power: 2.55V\ *\ 5.1mA\ =\ 13mW}$$
$$\boldsymbol{Efficiency: 13.515mW\ /\ 25.5mW\ *\ 100\%\ =\ 53\%}$$

To power everything, a battery of 3.7 volts and 6600 mAh was selected. This battery has plenty of power to power every peripheral and have a good battery life. The PowerBoost 1000 charging circuit was selected since it will ramp up our battery to 5.2 volts if needed for the LEDs and other peripherals. This equates to 34.32 W of available power. A clocking frequency above the 60Hz that is recognizable to the human eye will be needed to blink the LEDs so that no two strings are powered at the same time. “Blinking” the LEDs in this way will prevent any possible shorts that could occur from powering two strings at once. The chosen PIC32 microcontroller has more than enough speed with its 50MHz operating frequency. Its maximum and minimum voltages are 2.3V to 3.3V. Total power drawn is about 0.4525 W considering 9 LEDs on (2.65V * 10mA * 9), the micro (50mA * 3.3V), and Bluetooth (3.3V * 14.87mA).

3.0 Interface Considerations

The interfaces that will be required are USB for charging the system’s battery and UART through Bluetooth for data transferring and interfacing. The USB device chosen is the AdaFruit PowerBoost 1000 which will take an input from our 3.7V battery and step that up to the 5.2V needed. The PowerBoost will be able to charge the 600 mWh battery at a 1000mA rate. The Bluetooth/UART interface has a voltage supply of 3.3-4.2V with a current of 37mA while receiving and 43 mA while transmitting. The maximum power output is 1.5dBm, or 0.0014W. The data rate for the UART system is not listed for this specific device, but after looking at 3 other similar Bluetooth devices, we can assume that the data rate should be around 2Mbps.

4.0 Sources Cited:

[1] Microchip.com (2017, January 27). PIC32MX570F512L - 32-bit PIC Microcontrollers [Online]. Available: http://www.microchip.com/wwwproducts/en/PIC32MX570F512L

[2] Adafruit (2017, January 27). PowerBoost 1000 Charger - Rechargeable 5V Lipo USB Boost @ 1A - 1000C [Online]. Available: https://www.adafruit.com/products/2465

[3] Adafruit (2017, January 27). Lithium Ion Battery Pack - 3.7V 6600mAh [Online]. Available: https://www.adafruit.com/product/353

[4] SparkFun (2017, January 27). Piezo Element [Online]. Available: https://www.sparkfun.com/products/10293

[5] Digikey (2017, January 26). Kingbright APT3216LVBC/D [Online]. Available: http://www.digikey.com/product-detail/en/kingbright/APT3216LVBC-D/754-1943-1-ND/5177473

[6] DigiKey (2017, January 27). Microchip Technology BM78SPP05NC2-0002AA [Online]. Available: http://www.digikey.com/product-detail/en/microchip-technology/BM78SPP05NC2-0002AA/BM78SPP05NC2-0002AA-ND/6098092

[7] K.E. Clothier (2017, February 3). Power Consumption of a 4x4 LED Matrix [Online]. Available: http://electronics.stackexchange.com/questions/71092/power-consumption-of-a-4x4-led-matrix

Appendix 1: System Block Diagram