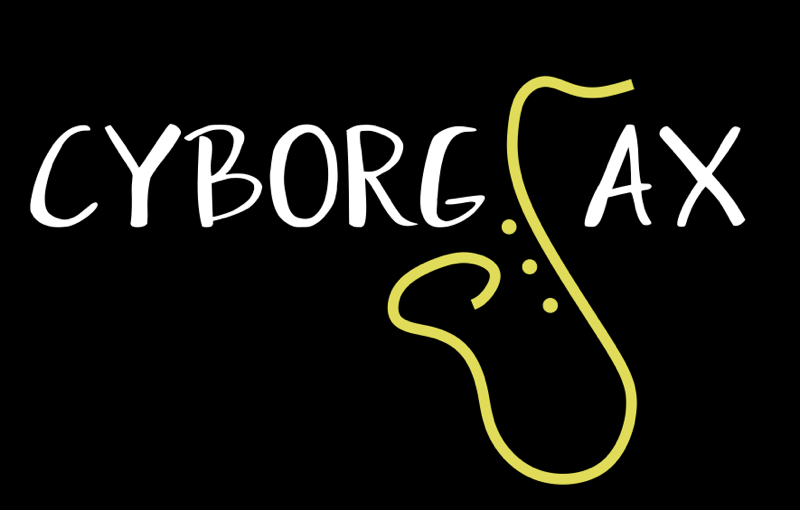
CyborgSax Test Plan for Second Prototype

Team 7: Tariq Ahsan, Raymond Liang, Esiri Madagwa Jr., Qinglang Yu, Andrew Zannetos



Hardware Setup:

The CyborgSax has a complex hardware setup. Our hardware setup consists of many parts: the Teensy Microprocessor, a Teensy audio shield, two 8 X 8 RGB LED Matrices, an encoder knob, an XLR/TRS input, a lavalier microphone, a battery pack, and an outer shell for enclosing the Teensy. The Teensy 3.6 microprocessor is used to control the visual patterns on the LED matrices. A micro-usb cable connects the Teensy to the laptop which will be used to code the Teensy for the LED matrices. Our TRS adaptor for our aux cable allows for our 3.5mm aux cable to connect to the system. Wires are connected from the Teensy/Audio Shield to the led matrix - these wires are +5V, ground - these two power the LEDs - and a digital signal connection to control the pattern and intensity of the LEDs on the matrices. The encoder knob will be connected to 3 digital pins on the teensy and a ground pin. Inside our casing will be the Teensy, the audio shield, and all of the wiring, with the outside of our casing having our encoder knob on the top of our shell and wires connecting to the LED matrices.

What to perform for hardware tests:

When it comes to testing the hardware components, we will connect the Teensy micro-usb port to a 5V battery pack. After it has been connected, the programmed Teensy with the FFT software and real time analysis will run which will create the lighting visualisations on the LED matrix. One member from the team will then talk into the microphone to generate a sound signal to test the real time analysis software.The same will be done with a song and a function generator to get an input into the microphone. Based on the outputs of the FFT, an algorithm will communicate with the Teensy to light up the LED lights according to the FFT values. The aim of the hardware test is to test the functionality of the LED matrix, the encoder, the microphone and the audio shield with the FFT code. Another goal of this hardware testing is to make sure each system (LED matrices, encoder and microphone) works synchronously with each other.

Hardware Measurable Criteria:

1. Be able to make sure that the microphone integration works properly by getting the LED matrix to light up according to the frequency of the noise signal coming in.
2. Be able to check whether the wires are connected properly with solder by making sure the lights on the LED light up.
3. Making sure that the encoder is working properly by making sure the colours of the LED lights can change with the push of a button on the encoder and the amplitude of the lights can change with the turn of a knob.
4. Making sure all the LED lights light up with correspondence to the FFT code.
5. Making sure that the system is powered properly with the use of a 5V battery by making sure that everything is working as planned.

Software Setup:

* We need a functional PC (Windows/Mac) to run the Arduino Software
* We need a micro-USB cable to connect the laptop with the Teensy Microprocessor
* The program will run with the Arduino Software with the Teensy driver

Software Tests:

1. Visual Implementation
   1. LED should light up once the Teensy is loaded with power source (Laptop or Battery Pack) - If powered by Battery pack, must be plugged in once in the middle, and then repowered by the first USB slot (Bottom LED will flash green if on)
   2. LEDs on matrices will turn on and off based on the value of the FFT value we get from the simulator. For example, if the range of n is at 0.2 < n < 0.3, the first two LEDs in the given column should light up green; if the range is at n > 0.4, the same 2 LEDs will turn green, but the LED above the matrix will turn on and turn yellow.
   3. FFT functional test
      1. Running through the arduino monitor, we can see the different FFT value n shown in the display, and it will refresh every .2 seconds.
   4. The leftmost column on the LED matrices will go up or down to indicate the master gain (sensitivity level of the microphone) depending on which direction the rotary encoder rotates. If the rotary encoder is rotated clockwise, it will increase the master gain, so the leftmost column will go up; if the rotary encoder is rotated counter-clockwise, it will decrease the master gain, so the leftmost column will go down.
   5. By pushing down on the encoder knob, the color scheme of the FFT should change to a different color scheme. It would cycle through 5 different color choices with green, yellow, red being the default setting.
2. Debouncing Feature Test:

When the program is loaded in the simulator run by the Arduino Software on a laptop, the following features are the ones need to be tested:

1. The number that indicates the position of the rotary encoder should go up or down stably depends on the direction which the rotary encoder is rotated. (Up: Clockwise; Down: Counter-clockwise).
2. The number that indicates the button pressed count should count up once at a time stably once the rotary encoder is pushed down.
3. The number that indicates the position of the rotary encoder ranges from 0 to 100, and it will never go above or below the boundaries.
4. All the measurable numbers on the simulator will not be fluctuated with the hardware interrupt.

Software Measurable criteria for success:

1. Program can be run properly once the device is connected with a power source
2. LED will turn to different colors based on the FFT value from the Arduino simulation.
3. The color combination of the LEDs will change once the encoder knob is pushed down and new FFT values are presented by the Teensy.
4. The leftmost column on the LED matrices that indicates master gain will react accordingly once the gain changes in the program.

What needs to be improved:

1. Real time audio effects must be implemented.
2. Audio outputs for processed saxophone signal to be sent to speakers. (XLR out, 3.5mm out)
3. UI on TFT screen must be implemented.
4. New case/enclosure for entire project must be produced in CAD and printed.
5. New encoders/switch configuration for enclosure.