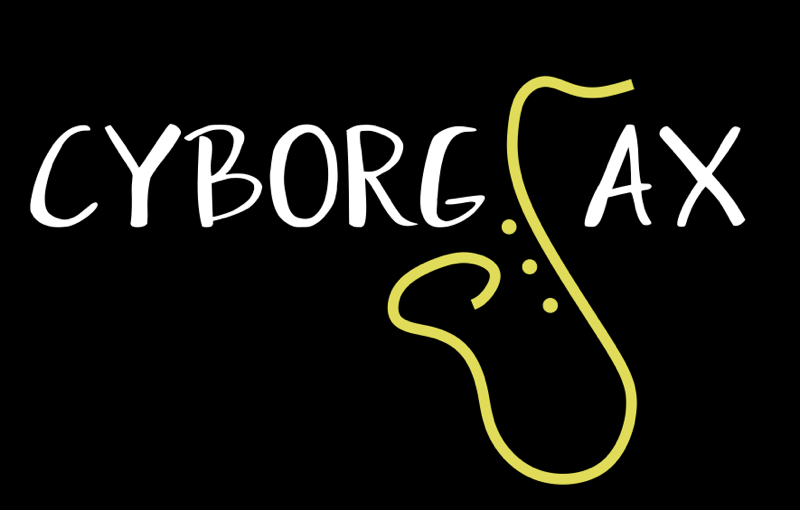
CyborgSax Test Plan for Final Prototype

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



Customer Requirements:

When considering the metrics of the device, our customer wants the device that we are designing to be portable, wearable and easy to use. Our customer wants a device that can perform audio effects and visual effects while they play the saxophone during live shows. The audio latency must not be more than 10ms while the visual latency can be 90ms-100ms. The customer wants the device to be powered by a battery pack that could power both the microprocessor and LED matrices in the system. Our LED matrices must also be able to fit around the saxophone. The customer also wants a way to have control over the parameters of the lighting and audio effects .

Significance of Deliverable in Meeting the Totality of Customer Requirements:

The system required something to generate visual effects and the approach the team went with to satisfy the requirement was to go with the 2 LED matrices in series. The choice to go with the LED matrices turned out to create the “wow factor” that our client was looking for when he used a prototype of our system at one his shows. People could now have a more enjoyable experience when watching these live performances.The choice of the battery pack of +5V was sufficient as it was enough to power the Teensy microprocessor and the LED matrices without burning out the two components. All the components in the system as a whole were able to be fitted to the client and his saxophone when he performed with it which showed the easiness and portability of the system as a whole. The controller for the device is going to be a TFT screen which will be controlled by encoder knobs and a switch which will give the client feedback on which mode and parameter he has chosen. Not only will he get to control what he wants for different settings but the controller will also satisfy requirements of being portable and easy to use. It will be portable because it can be worn around his waist when he is performing and will be easy to use as an interface with buttons and knobs will be less complicated compared to an interface with a touch screen. Additionally,audio effects will be fixed and implemented and will give the user the ability to manipulate the saxophone noise as the customer requested from the customer requirements.

Hardware Setup:

The CyborgSax has a complex hardware setup. Our hardware setup consists of many parts: the Teensy Microprocessor, a SD card, two 8 X 8 RGB LED Matrices, 2 encoder knobs, a laptop and a TFT screen. The Teensy 3.6 microprocessor is used to control the visual patterns on the LED matrices and the GUI output on the TFT screen. A micro-usb cable connects the Teensy to the laptop which will be used to code the Teensy for the LED matrices and the GUI for the TFT screen. Wires are connected from the Teensy 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 2 encoder knobs each will be connected to 3 digital pins on the 2 ground pins on the teensy . The TFT screen has various data pins, a LED pin, a VCC pin and a ground pin that will all be connected to the Teensy so there will be data transmission from the laptop to the TFT screen.

What To Perform For Hardware Tests:

When it comes to testing the hardware components, we will connect the Teensy micro-usb port to the laptop. 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. Based on the outputs of the FFT read off a WAV file stored on the SD card, 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 encoders and the TFT screen with the GUI code. In order to test the TFT screen, the screen must be able to display a visual output which will show which mode the user is on and the different parameters of the modes. Another goal of this hardware testing is to make sure each system (LED matrices, encoders and TFT screen) works synchronously with each other.

Hardware Measurable Criteria:

1. Be able to check whether the wires are connected properly with solder by making sure the lights on the LED light up.
2. 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.
3. Be able to have the GUI output to the TFT screen showing parameters and mode.
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 laptop 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 connected with the power source (Laptop or Battery Pack) - If powered by Battery pack, the USB cable 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 values 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 the software through the arduino monitor, we can see the different FFT value n shown in the display, and it will refresh every 0.2 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 pushed count should switch from 0 to 1 depending on whether a function is changeable or not.
3. The number that indicates the position of the rotary encoder ranges from 0 to 99, with different constraints depending on the modes.
4. All the measurable numbers on the simulator will not fluctuate with hardware interrupt feature.
5. GUI Test:

When the device is connected to the power source, the GUI on the TFT screen will light up. The design features that needed to be tested are as follows:

1. The current mode, the two different encoders’ current status, and their corresponding values are properly displayed to the TFT screen.
2. When encoder one is on (RED), and encoder 2 is off (WHITE), it indicates that the device is currently on function one; when encoder one is off (WHITE), and encoder 2 is on (RED), it indicates that the device is currently on function two.
3. When the first encoder is rotated, either clockwise or counterclockwise, the text box to indicate the value will change accordingly. Also, the volume bar will make corresponding changes as well depending on the range of the parameter values.
4. When the first encoder is rotated, either clockwise or counterclockwise, the text box indicating the value will change accordingly. Also, the volume bar will make corresponding changes as well depending on the range of the parameter values.
5. When the device is set on Mode 2 (Lighting Mode) and the first encoder is ON (Function One), it indicates that the device is now on the “Color switching mode” the range for the parameter values for encoder one is constrained from 1 - 5 to indicate four different combinations of colors. When the encoder value is set to 1, the color combination on the matrices will turn to Green, 2 for Blue, 3 for Pink, 4 for Yellow accordingly.

Software Measurable Criteria for Success:

1. Program can be run properly once the device is connected with a power source and the GUI on the TFT screen should work with the LED matrices at the same time and interact in real-time.
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 depending on the value passed by the color\_switching function. If the encoder value is set to 1, then the colors combination on the matrices will be set to “Green”. if it is set to 2, the colors combination will be set to “Blue”; if it is set to 3, the colors combination will be set to “Pink”; if it is set to 2, the colors combination will be set to “Yellow” accordingly.
4. Once the encoder knob is pushed down and new FFT values are presented by the Teensy.

Description of Data Collection

To test the accuracy of the system, the following parameters need to be tested by running the serial monitor on the arduino software:

1. Encoder values:

In terms of setting the encoder value, the parameter can only be changed if the current status of the encoder is set to 1 (ON) and the corresponding encoder status on the GUI will be set in RED. When the encoder is rotated clockwise, the simulator will display a text with its current value, and the value will be incremented accordingly. When the encoder is rotated counterclockwise, the current value will be decremented accordingly.

1. Information of the current mode and the corresponding function:

When the encoder is pushed down, a text will display in the simulator and clearly indicate its current status (F1 is 1 - ON; F1 is 0 - OFF). The text will also display the status of the other encoder (F2 is 0 - OFF), vice versa. Also, the text will also include the current mode of the device by showing a text “current mode: ?” (1 - Sound; 2 - Lighting; 3 - Playback; 4 - General).

1. FFT value: These are the values used to correspond to the number of lights per column (bin). After testing with a sine wave we determined our fft readings and output were very accurate.

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. New case/enclosure for entire project must be produced in CAD and printed.
4. New encoders/switch configuration for enclosure.
5. Fix the connection issue as the encoder values are not always getting outputted to the screen.
6. Adding a switch to control GUI modes.
7. Find a way to relate the parameters of each mode of the system to different encoder values.
8. Microphone needs to be fixed as faulty audio shield damaged the real time analysis portion of the system.
9. Noise in the microphone needs fixing.