**Team S-Motion’s Project Progress Report**

**Brandon Wilde, Hanyu Liu**

**Project Goal Recap:**

The project is aimed to help DJs connect more with the audience while performing on stage. A common problem is that the audience is unable to see what the performer is doing while on stage, creating a disconnect. The goal of the project is to create a wearable for the artist that recognizes gestures to adjust the audio, rather than using a laptop or mixer, allowing for more interaction to the crowd.

Besides, it’s hard for live performers to control multiple parameters (e.g. 20 parameters) at the same time like an automation.

**Deliverables:**

**Video demo is uploaded on https://vimeo.com/327177342.**

* **Hardware:** The hardware deliverables consists of the wrist band that we’ve built using Arduino boards (Bluno Beetle SKU:DFR0339 ref: <https://www.dfrobot.com/wiki/index.php/Bluno_Beetle_SKU:DFR0339>). It already has a MCU which supports BLE. We used another Bluno beetle board to act as the receiver for accepting UART data and transmit them to the computer via UART over USB.

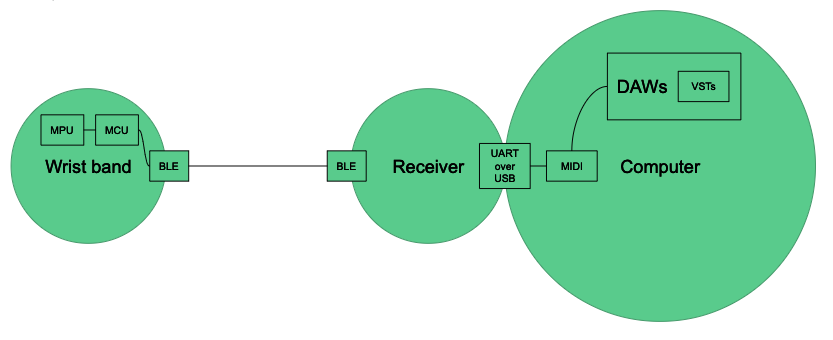


Fig 1. Overall layout and signal flow diagram

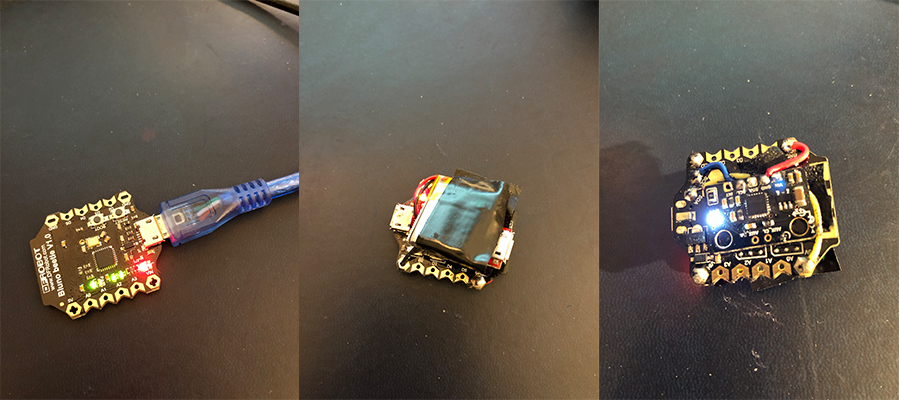


Fig 2. The receiver and the transmitter.

Transmitter is bundled with a Li-ON battery, a charger and a MPU6050 6-axis sensor

* **Software:** The software side contains two segments: The Serial-MIDI translator program and the VST plugin for sinusoidal modeling synthesis. The translator receives the Serial data from USB and parsed them into data which can be mapped onto the MIDI events. Then by using JUCE library to create a virtual MIDI device, it can then send MIDI events to any DAWs (Currently only support Mac OS X). The VST plugin is an implementation of a sinusoidal modeling process. Basically, it receives a frame of audio samples, converting them into STFT frame and analysis dominant frequency components in decreasing order based on their magnitude. By adding different sine waves with different frequency with respect to their corresponding magnitude in the STFT, we can resynthesize this frame of samples. We can also shift the frequencies of the sine waves or select number of sine waves we use to synthesize.

Both of these software components are built using JUCE library. We used Toth Laszlo’s FFT library for obtaining STFT frames.

The deliverables contains the source build file Xcode projects and built apps in ./Build\ Results folder. The code is also uploaded onto github: <https://github.com/proton1030/CS4590_Project>.

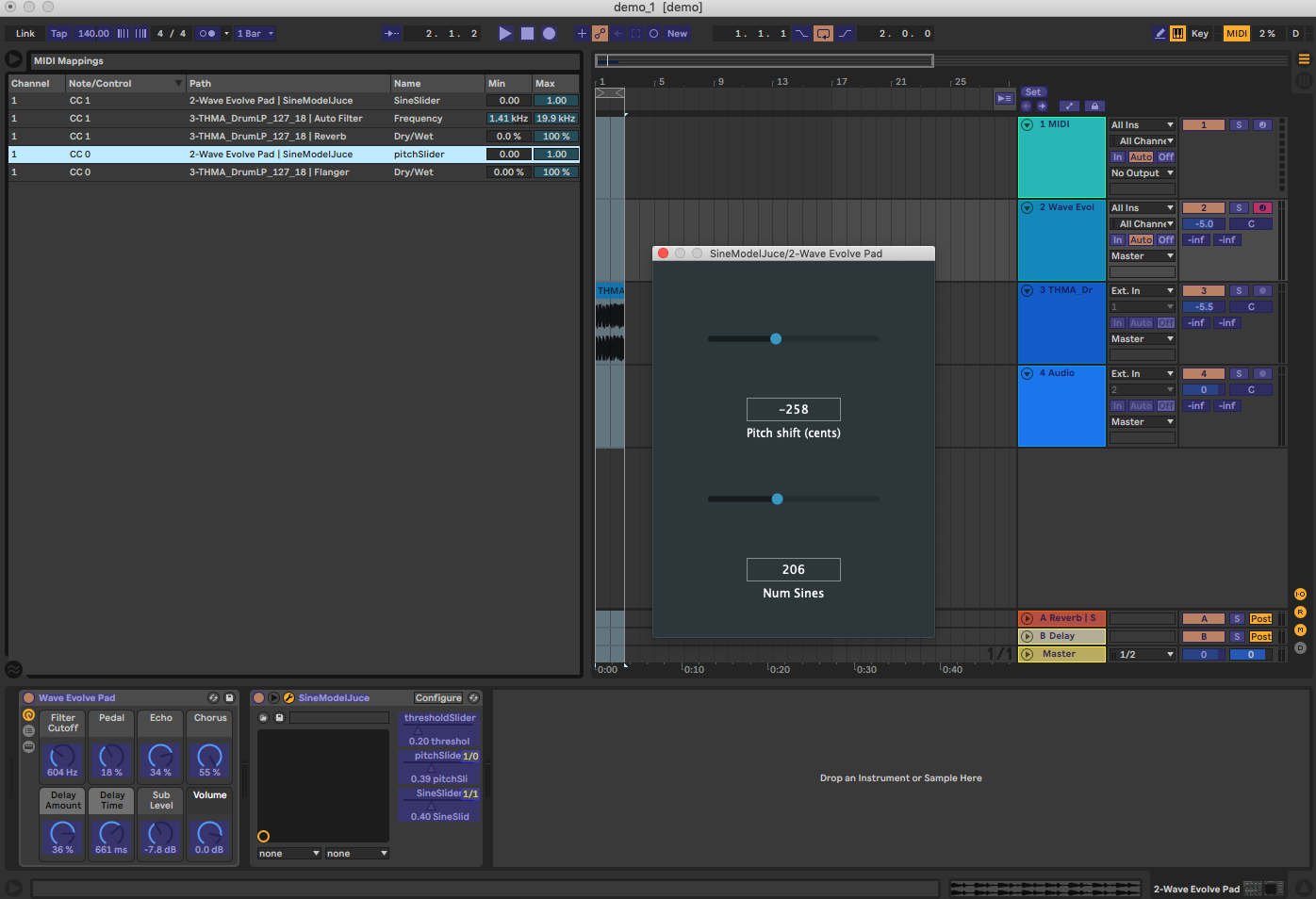


Fig 3. A screenshot of the VST plugin working in Ableton Live and current active

MIDI mappings.

**Current Progress comparing to Proposed Timeline:**

|  |  |
| --- | --- |
| Week 1 | Gesture recognition algo development |
| Week 2 | Continue Gesture recognition |
| Week 3 | ~~UART to MIDI interfacing~~ |
| Week 4 | ~~UART to MIDI interfacing~~ |
| Week 5 | ~~Sinusoidal modeling algo implementation~~ |
| Week 6 | Granular synthesis implementation |
| Week 7 | Refinements |

**Contributions:**

* UART to MIDI interfacing: Brandon & Hanyu
* Sinusoidal modeling: Hanyu