Glove-Duino Critical Introduction

Candidate Number: 148508
Module Name: Interactive Music Systems
Module Number: W3083
Academic Year: Autumn 2016

Glove-Duino Project – Sam Bilbow

Introduction

This project was about creating an Arduino powered glove that enables to wearer to engage with music performance and composition on a gestural level. Inspiration for me, came from watching Imogen Heap's demonstration of her Mi.Mu gloves at a conference. I was thoroughly impressed with the amount of flow that her instrument provided her in a performance setting. Learning about the New Instruments for Musical Expression conference also helped me set out to create not exactly a "new instrument" per se, but to allow me to explore my own compositional and performance based ideas through a gestural device of my own making. Joel Ryan proposed in his "Notes on STEIM" that the problem with computer music is the distance between the musician and the composition process (Ryan,1991). I saw the solution to this in the raw organic nature of the gestural glove.

Context and Research

As stated, I borrowed ideas from Imogen Heaps Mi.Mu gloves, but as my project progressed I looked at her own inspiration as well as the pioneers of glove technology and gestural music. Laetitia Sonami's instrument the 'lady glove' looked very similar to the glove I had in my head, with bend sensors running down the fingers. Her engineering ethos, or purpose of creation was very similar to mine, she says: "The sounds are now 'embodied', the controls intuitive, and the performance fluid. It has become a fine instrument" (Sonami, 2016). I set out to build this glove in order to make composition and performance of music a more intuitive and expressive process through gesture.

Lady Glove 4

Development

The development process began with drawing sketches of what I wanted the glove to look like as well as the functions I wanted it to have. I had a rough idea of having bend sensors over each finger and thumb of my right hand, sewed into the glove; with the Arduino being stored on a wrist strap of a thicker material. After the two sketches (one rough and one refined), I decided to look online for product images of the Arduino modules and sensors I'd need for the glove. I created a Photoshop mock-up of the glove that allowed me to plan the aesthetics and feasibility of the glove's features.

Photo 1: Foam Wrist Strap

The next stage of the development process was creating an Arduino 'sketch' in the Arduino coding environment that would allow me to pull data from the gloves bend sensors. I did this while waiting for my parts to arrive as to get a head start on what I thought would be the hardest part. I had used some Java script before, and the Arduino coding environment was very similar to it so I didn't have much trouble setting up a sketch. When the components arrived, I started by making the foam wrist strap (see photo 1). This involved stitching on Velcro for the Arduino, and sewing on snap fasteners, as well as measuring out a strip that would fit well around my wrist. Next came the soldering. I checked the hook-up guide for the bend sensors, and it seemed like there would be a lot of wiring involved, creating a variable resistor around the base of the bend sensor that allowed the data to enter the analogue input on the Arduino. I decided that if there was going to be five bend sensors, that this would be too much wiring to have loose, so I reduced the visible amount by soldering the resistor in, as well as the ground, voltage, and analogue data wires, before wrapping them in electrical tape. Photos 2-4 highlight this process. After this, came the procedure of stitching Velcro onto the nail part of each fingertip, securing each bend sensor, and sewing a ring around them at the centre finger joint to keep them in place, but not so tight as to inhibit their flex (see photo 5). Next, I decided to tidy up the aesthetic of the glove, by stitching felt over the top of the bend sensors, this also meant that they were protected from any accidental handling (see photo 6).

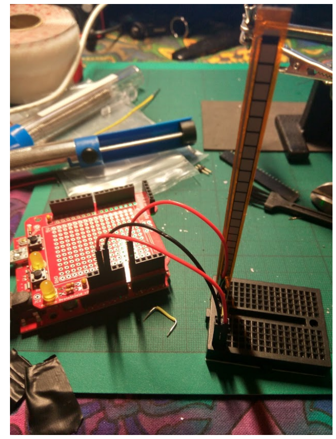

Photo 2: One sensor's hook-up

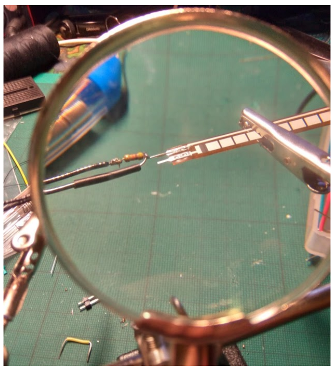

Photo 3: Streamlined wiring

Photo 4: One bend sensor

Photo 5: All bend sensors

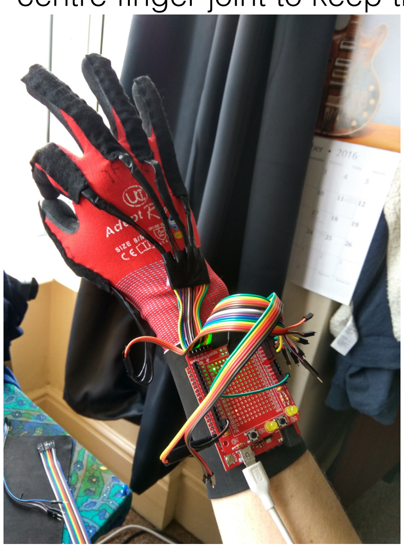

Photo 6: Felt stitched over fingers.

After the bend sensors were attached to the glove, I decided to connect them all to the Arduino via the analogue inputs. Initially, only four of the sensors were displaying live data in the serial monitor, the index finger's sensor was displaying the maximum data point (1023), after taking off the electrical tape for that sensor, I found that the soldering had broken, so I re-soldered and hot-glued all of the connections to reinforce them. The next few stages in the development of the glove were unfruitful, I tried attaching a Bluetooth component to it, as well as a 9V battery in order to make the glove truly wireless, however the Bluetooth module would not connect to my laptop. As well as this, I tried adding an accelerometer to the glove, but I could only ever get either the accelerometer OR the bend sensors outputting data via serial for reasons I will cover later on in this write-up. At this point in the process, I had a working glove that output bend sensor data to the serial port on my laptop via USB. For the next step, I sought help from the PureData Facebook group, a group with over 10,000 people in, who give advice on various projects. I was having trouble taking the serial data from the Arduino and using it with PureData. Within two hours, someone had pointed me in the direction of an Arduino sketch and PureData subpatch that he'd made that seamlessly integrated with each other. I borrowed some of his coding, and updated my sketch, and suddenly I had the live sensor data in PureData, showing me the real-time positions of my fingers in a GUI I had made previously. The patch and various sub-patches are shown below.

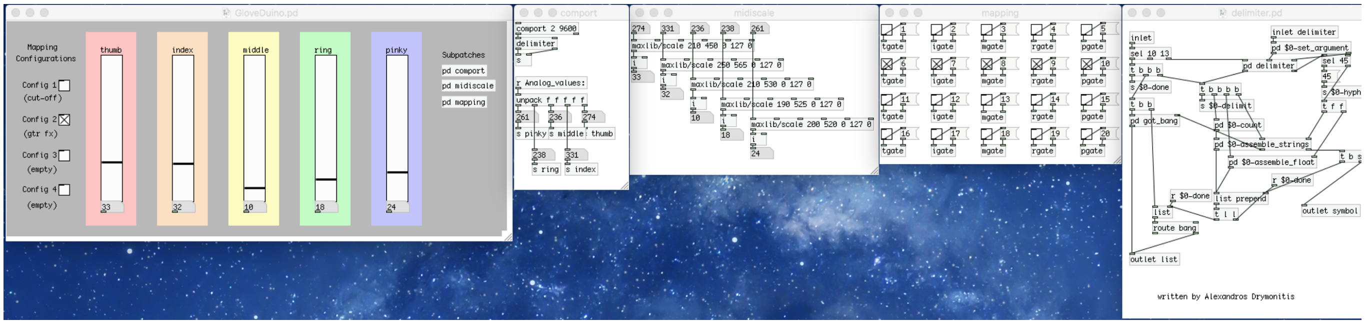

Photo 7: PureData patch that decodes data in the serial port and outputs it to number boxes, where they are scaled to 0-127 for usefulness with MIDI.

Each finger's bend sensor had separate minimum and maximum data values, due to the inherent positioning and flex of each finger. This meant I had to test out the minimum and maximum data values and input them into my patch to scale them between 0 and 127, this is what happens in the 'midiscale' subpatch. Each finger outputs one of five [ctlout] objects, which can be selected for different performances/compositions with the mapping matrix subpatch, I call them configurations. The data is then picked up by whatever DAW is listening to the IAC bus, and then mappings have to made inside the DAW.

Demonstration

Mapping of the data values to parameters within Logic Pro X is what I did first once I had the live MIDI data in PureData. I realised immediately that one of the biggest strong points of the glove was the expressivity it gave me. I experimented with a few different mappings, one to one, one to many, many to one, and found the usefulness of having 4 parameters mapped to my hand my favourite option. A problem that I found when mapping, was that I physically can't bend my pinkie finger without bending my ring finger simultaneously, and therefore I had to make sure I was aware of this when mapping them to different instruments or effects. I started out by mapping simply to instrument cut-off filters on a four track loop that I'd made, before experimenting with other effects like saturation, distortion, echo and delay. What I found was that the glove was very useful as a post-compositional expression aid, or a performance mixer aid and I demonstrate this in the videos I have provided.

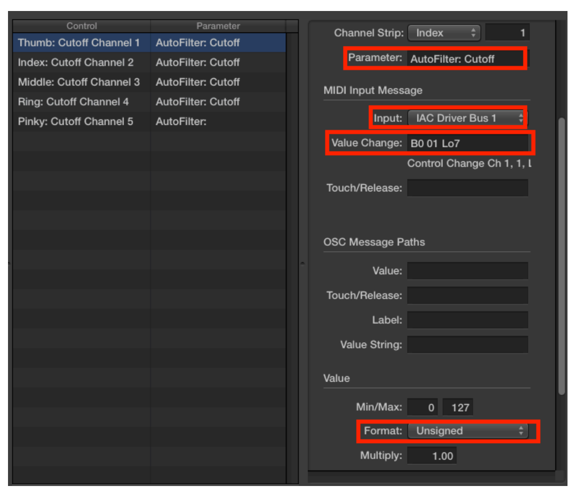

Photo 8: Mappings inside the controller assignments window of Logic Pro X

Evaluation

Having sound be affected in direct correlation to body movement means that the glove excels at being an extremely intuitive way to aid the composition and performance of music with digital audio workstations. If being used in a performance setting, mappings can easily be created in PureData and Logic, and mappings can be switched seamlessly through the patch with use of another MIDI controller, or just the computer keyboard/mouse. One of the major problems that I could not solve, due to time constraints was how to output both bend sensor data and accelerometer data. The PureData patch that decoded the space-separated values from the serial port that I borrowed was too advanced and above my level of knowledge to manipulate to decode the accelerometer data as well. As well as that, it would have been useful to have the glove be wireless, but having one USB cable connected does not seriously inhibit movement at all. Overall, I believe I have created a successful project, and it has spurred me on to explore other creative possibilities that microcomputers such as the Arduino have to offer in the field of music.

References

Ryan, J. (1991). Some remarks on musical instrument design at STEIM. Contemporary Music Review, 6(1), pp.3-17.

Sonami, L. (2016). Instruments – Lady's Glove | LAETITIA SONAMI. [online] Sonami.net. Available at: http://sonami.net/ladys-glove/ [Accessed 29 Dec. 2016].