Tyler Andrews
Partners: Matthew Seaton, Patrick McCelvy, Brian Bresee
For: P. Lehrman, ES-95: Electronic Musical Instrument Design
May, 2011

The New and Improved DJ Hands:
A Better Way to Control Sound

Introduction:

As three of our four group members fumbled with final programming during our first presentation day, trying to get something, anything, to come out of our prized "DJ Hands," one thing became very clear: we needed more time. After watching the other two groups present seemingly half-finished products, my heart sank as our turn came up. Missing one of our group members on the presentation day, and thus missing a key piece of programming for which he had been responsible, our project was similarly afflicted by a sense of incompletion.

In my first report, I had written that I felt like we all needed more time, and that being able to devote a few more weeks to these projects would likely yield impressive results. Thus, I was excited to find out, later, that we would indeed be able to continue working on the DJ Hands for the remainder of the semester. I was thrilled to have the opportunity to take a nearly-functional prototype and bring it to a fully-functional instrument.

As stated in the previous report, the original goal of the DJ Hands was to create a more natural way to control sound in a live DJ setting. Particularly, the goal was to solve the problem that so many DJs encounter of having "too many knobs and too few hands." To do this, the DJ Hands places the controls on the fingers, giving the user much more dexterity and fine control over sound parameters and levels.

With this extra time, the goals of the project changed from producing simply "something that worked" to "something that worked well." A great deal of changes were made to do this including an entirely new physical interface (outlined in **Hardware**) and a more sophisticated sequencer program for controlling samples (outlined in **Software**). The basic principles of the DJ Hands, however, remained the same. Momentary switches on the fingers of the left hand were used to control channel selection, while force sensitive resistors (FSRs) attached to the right fingertips were used as analog inputs to control parameters such as filter frequency and volume. Finally, the ten-year-old Doepfer Boxes were retired from use as these projects began the use of Arduino boxes. These allowed for wireless use, which made the DJ Hands that much more functional in a live setting. What started out as a rough prototype has been improved and modified to the point of being a fully functioning product.

The Basics

The DJ Hands feature left and right handed gloves which, together, work to select and manipulate pre-arranged tracks. On the left hand, a binary switch is attached to each finger-tip. Each switch is assigned to a specific midi-channel, which represents a specific instrument in Reason. By engaging one of the binary switches, the user is selecting which track he would like to manipulate.

Once selected, the DJ uses his right hand to manipulate the sound. FSRs are attached to each finger-tip on the right hand. The data from each FSR manipulate a specific parameter on the selected channel (outlined in **Software**).

Using only his fingers, the DJ now has the ability to control five different parameters on five different channels.

The Team

Tyler

As a mechanical engineer, Tyler is able to work with the physical construction of the instrument as well as the overall system design. With a strong background in music, he is also able to contribute to the creative design and brainstorming. Some programming experience also lends to the ability to contribute to the assembly of the Max and Reason patches

Patrick

As an electrical engineer, Patrick will be a valuable assets when designing the circuitry and wiring needed to create our instrument. A strong background in designing and wiring circuits will be particularly important when wiring sensors through the Doepfer boxes. Also, as a DJ with several years of experience working with Reason, Patrick will be a great resource when it comes time to programming the Reason patch.

Matthew

As a DJ, programmer, and a designer, Matt brings a wealth of experience and ability to the team. As a Computer science major, Matthew has gained considerable experience coding and handling data. Additionally, In his spare time he has worked to create midi controllers from old video-game controllers and other every-day items. He will be a valuable asset when it comes time to interpret data and program the Max patches.

Brian

With 15 years of piano experience, Brian brings an enormous amount of musical experience and expertise to the table. As an economics major, he is able to bring a different perspective on overall system design. With some programming experience, Brian will also be invaluable in writing the Max patches.

Hardware

In terms of hardware, the DJ Hands 2.0 are a marked improvement from the initial prototype. The Hands originally started as a pair of soccer goalie gloves. It was thought that the large surface area of the fingertips would provide a good amount of room for placing sensors and buttons. However, the material of the gloves – a soft, squishy type of plush – ended up being troublesome in the original prototype. The foamy material absorbed a good deal of the force when one pressed down on the sensors, and so it was difficult to get a consistent reading. A new pair of Saucony running gloves proved to be a much better fit.

Other than the change in gloves, the hardware on the Hands 2.0 is quite similar. FSRs are attached to each fingertip of the right hand and buttons are attached to each fingertip on the left hand. The attachment process and the types of button are slightly different in this updated version (outlined in more detail in **Problems Encountered**).

There were three significant additions to the hardware of the Hands. First, on the right hand, a two-axis accelerometer was attached to the back of the glove. This allows the user even more freedom when performing, since all he has to do is move his hand through the air to produce a signal. The pitch of the accelerometer determines whether the accelerometer is in use; that is, if the pitch is 0, the hand is flat and the data from the other axis is ignored. If the pitch is close to 90°, the user's hand is upright and the yaw of the user's hand (a back and forth motion) will control the pitch bend of the current channel.

The other major addition was that LEDs were added to each finger on the left channel-selection hand. These two-color LEDs give the user immediate, visible feedback about which channels are currently being controlled. A red light means a channel is not selected, while a green light means a channel is selected. In the original prototype, the user had to watch a computer screen to remember which channels he was controlling at a given time. The LEDs give more freedom of expression by letting the user rely only on the Hands, themselves, to tell him which channel is selected

Finally, the last major change in the hardware was the use of the Arduino midi box. These boxes replace the Doepfer Boxes and are much smaller and more accurate and would allow for wireless midi transmission. For the Hands, this was important because it meant that the user no longer had to be tethered to a computer.

The wires from all the sensors are run to a bread-board via ribbon cable, which was original secured to the top of the Hands (1.0). Since the new Hands are a sleeker, smaller design, it was decided to have the wiring all run to a belt pack to leave the actually Hands as light as possible. To do this, all of the wiring is run into the bread-board which then runs into the Arduino, all of which sits in a fanny pack, which the user can wear.

The wiring itself is all nearly identical to the original versions. Sensors which yield analog inputs (FSRs and Accelerometer) are run into the analog inputs of the Arduino and are set up in a voltage divider circuit between a +5V DC source (from the Arduino) and a 10kΩ resistor connected to ground (see **Appendix – Wiring Diagram)**. The digital sensors (buttons and LEDs) are run into the Arduino digital inputs (and outputs for the LEDs).

Software

(NB – I did not personally design much of the software code and so I will provide a basic overview of the programming which was done mostly by Brian and Matthew.)

The basic architecture of the Max patch remained the same, utilizing a great deal of the same code. The button data are sent into Max as note-on values through the binary inputs in the Arduino. These data are sent into a channel selection patcher. This patcher allows the user to select one of four channels to control. These channels correspond to four separate modules in Reason and are a Subtractor module, a Thor Module, a Malstrom Module, and a Re-Drum module. When a button is pressed, the channel to which it corresponds is selected for manipulation by the FSRs. This could be one channel individually or all four.

The data generated by the FSRs is routed through the Arduino continuous controller inputs and into Max. These data were of the scale 0-1023 coming from the Arduino, and so they were scaled to a 0-127 midi scale, with extra scaling on the lower end to eliminate any "phantom values" – data generated by noise or unwanted signal. These data are then transmitted as controller values for the parameters outlined in the **Appendix Table 1.** As was the case in the original design, the re-drum channel is slightly different. Here, the FSRs simply act as switches which are used to start different drum loops.

The sequencer was the most complicated piece of new code which was added to the design. This was programmed by Matthew and took data coming from a list of rhythmic and tonal data which was scrolled through by a metro object in Max. This allowed the user to easily change the tempo (via computer), while still using the same samples.

Personal Responsibilities

Once again, I found myself at the helm of our (sometimes unsteady) ship. I organized meetings and tried to make sure people were staying on top of their responsibilities. This was often frustrating, however, as some of our group members were notoriously bad at returning emails and showing up to meetings. This caused some unfortunate tension in the group, since it was unclear whether certain tasks were being accomplished (more in **Problems Encountered**).

In terms of tangible work, I found myself again in charge of the physical construction and aesthetics – or, as I think about it, the "playability" – of the project. As a mechanical engineer, I took pride in selecting the materials for assembly and then putting everything together. I think the physical design of these gloves is much better both in terms of playability and in terms of simply looking good. Along with Patrick, the group's electrical engineer, I helped with soldering, running cable, and wiring via the bread-board. Patrick was very helpful in this and certainly more knowledgeable than me, but it was a good learning experience for me to have to sort out a few of the kinks in the wiring.

Problems Encountered

I am happy to say that the problems encountered in the second half of this project were far less severe and far less frustrating than in the original proto-typing. I believe that there are several reasons for this. First and foremost, we are simply all more experienced with every aspect of the projects. That is, the first half of the project gave us an enormous amount of hands-on learning about everything from basics of design to soldering and wiring to computer programming. I also think that we all worked much more efficiently during this project. Instead of saving a lot of work for the last 72 hours, most groups seemed to work well and work gradually over the course of a few weeks. This kind of slow-and-steady work will almost always end up yielding better results. Finally, I think after experiencing the failure of the first half of the project, most groups set their sights on more realistic goals. Both the first and second half of these projects were done in fairly brief blocks of time. I think people realized that they had bitten off a bit more than they could chew the first time and set much more reasonable goals for the final project.

In my department, specifically, there were still a few problems which needed to be sorted out. Unlike last time, however, some of these problems were less specific ("I can't get a button to fire") and more general ("How do I make the Hands feel better?"). As mentioned before, the original goalie gloves were archy and hard to use. To correct for this, I selected new pair of gloves for use. These gloves, a pair of Saucony running gloves, were markedly different from the previous goalie gloves. They were thin and tight and did not absorb any of user's force. Having a tighter fitting pair of gloves also meant that the overall dexterity was much greater. When using the goalie gloves, the Hands simply felt clumsy and hard to manage. These new running gloves felt much more natural, as they fit snugly and motion felt more natural.

Next, it was necessary to move the sensors and wiring from the original gloves to the new gloves. In the original gloves, I had come up with an elaborate attachment scheme in which the FSRs were attached to a hard rubber backing which was then attached to the glove and then a rubber nubbin was taped over the FSR to provide a pressure point. This ended up looking ridiculous and not being particularly functional. I decided to go the opposite direction with these gloves and try to make the attachments as simple as possible. I had originally used rubber cement, which took a long time to dry and had not stuck particularly well (and made the whole lab smell terrible!). I decided, instead to try using hot glue. I tested the glue on both the FSR and the gloves to make sure that it wouldn't melt either material and, once it was deemed safe, I simply glued each FSR to the gloves fingertips. This worked remarkably well; the attachments were very strong and simple with no extraneous parts.

The buttons were the next problem on my plate. Originally, we had used relatively tall black plastic buttons. These had seemed like a good idea, but were really too tall for the gloves, which made pressing them awkward and difficult. After scouring the lab, I came across a much more appropriate set of buttons. These were much flatter than the original buttons, which meant they lay much closer to the finger time and were easier to press. Also, they offered a satisfying click when pressed, a valuable piece of tactile feedback. After testing each to make sure they were still functional, I soldered each to ribbon cable. I then tested the buttons with the hot glue to be sure that they wouldn't melt, and proceeded to simply glue them onto each finger.

Honestly, these were more design decisions than real problems. The biggest problem that our group faced was communication and accountability. As previously mentioned, I tried to act as a group leader and organize meetings and make sure that everyone understood what his job was. However, when it came time for these meetings, I would sometimes be the only one to show up (including our timeslot with our professors, where we were supposed to go over trouble-shooting of the project). Since I was really only on the physical side of the project, I often unclear as to how far along the programmers were. I would have been happy to help, but again, it was a lack of communication which prevented me from even knowing if help was needed.

I'd like to add an addendum, as the previous paragraph was written a few days before the end of the project. After speaking with the professor, the communication of the group greatly increased. I was happy to receive updates on the programming side of things letting everyone know how things were going. This was a huge improvement and I only wish it had occurred sooner.

Finally, our presentation was a bit of a struggle. This was the first time that all of us had been in the same room in quite a while, and so it took a bit of time to get everything smoothed out. Like everyone's project, the Hands were really more of a prototype than a finished product. There was a lot of trouble-shooting going on during the presentations, which ideally would have been done before. Still, given the time frame of the projects and everyone's other commitments, I think the projects were a great success.

Conclusion

All in all, this was a great project and a great course. I learned far more here than I have in most of my other courses here at Tufts and I believe an enormous amount of that is due to the hands-on nature of the last two months. This has been the most "real" engineering that I have done as a student: given no real instructions other than turn gestures into music. The open-endedness of the original project assignment allowed for an enormous amount of creativity, which can be seen by how different each project is from one another.

Personally, I think that continuing on the same project was a great idea. After the failure to produce in the first round, it would have been depressing and disappointing to have to table all of our un-finished projects. Also, continuing to work on these projects forced us to look back critically at the previous few weeks and ask ourselves what we had done well and what we had done poorly. A lot of what we had done poorly was time management, which I think just about every group improved on during the second half of the project. I also think that we were able to take ideas which were quite complex – probably too complex for the original 2-3 week timeframe – and actually complete and execute them. To me, it is more rewarding to make one really complex, interesting project than two smaller-scale ones.

Overall, this project served as a great challenge and a great culmination to the year and the course. I was thrilled to be a part of this class. I have always been passionate about science and music; this class served as the perfect bridge between the two. I hope to be able to expand that connection over my next two years at Tufts.
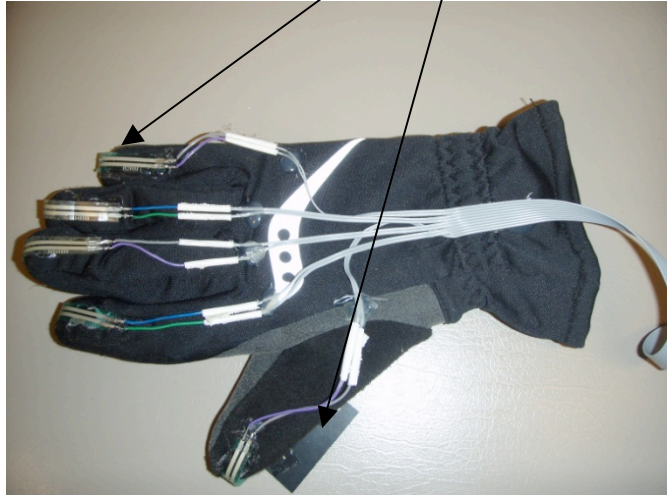
Appendix

Photos

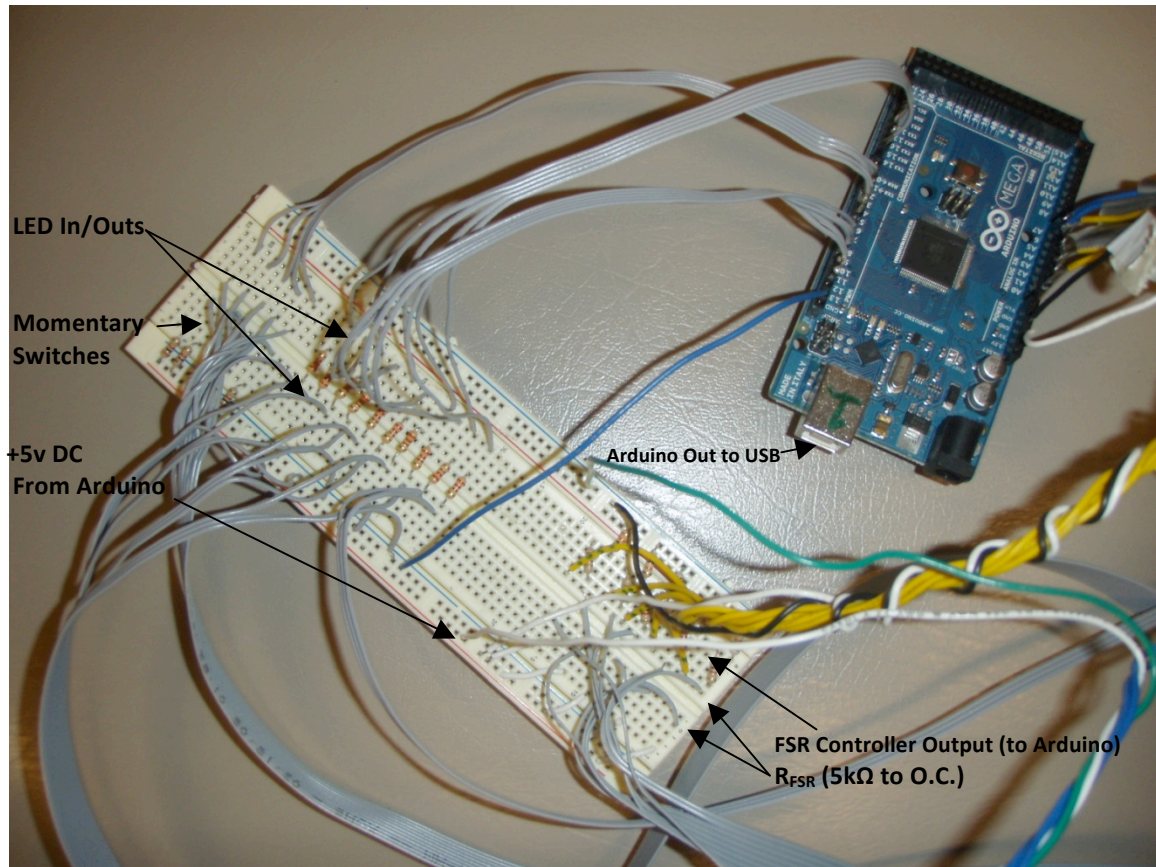Left Hand with Momentary Switches

Right Hand with FSRs

Left Hand Back with LEDs and Accelerometer

Wiring Photo
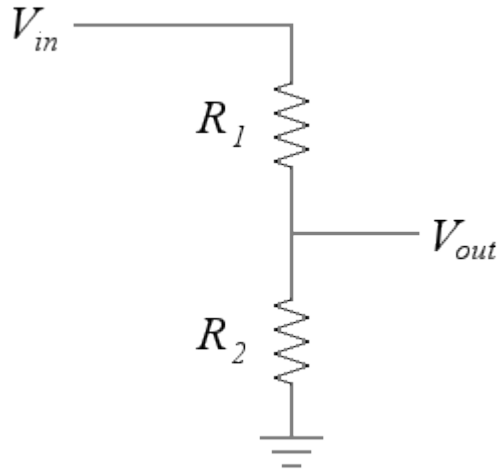
(NB – All wiring is contained within the Fanny Pack)

LED In/Outs

Momentary Switches

+5v DC From Arduino

Arduino Out to USB

FSR Controller Output (to Arduino)

R_{FSR} (5kΩ to O.C.)

Final Product Photo

Wiring Diagrams

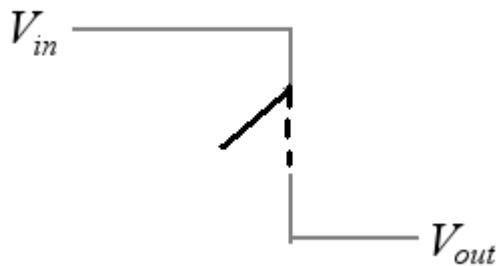## Diagram 1 – FSR Voltage Divider

V_{in}

R_1

V_{out}

R_2

Legend:
V_{in} = +5V DC (From Arduino)
V_{out} = Output Voltage (To Arduino analog input)
R_1 = FSR (5kΩ to $\infty\Omega$)
R_2 = 10kΩ

Diagram 2 – Momentary Switches

V_{in}

V_{out}

Legend
V_{in} = +5V DC (From Arduino)
V_{out} = Output Voltage (To Arduino digital input)

Table 1 – Channel Selection in Max

Channel	Reason Instrument	Parameter 1	Parameter 2	Parameter 3	Parameter 4	Parameter 5
1	Subtractor	Volume - 7	Mod Wheel - 1	LFO Rate - 26	Filter Resonance - 71	Filter Frequency - 74
2	Thor	Volume - 7	Mod Wheel - 1	LFO Rate - 26	Filter 1 Resonance - 71	Filter 1 Frequency - 74
3	Malstrom	Volume - 7	Mod Wheel - 1	Modulator A - 26	Filter A Resonance - 78	Filter A Frequency - 79
4	Redrum	Volume - 7	Pattern Select - 3	Pattern Select - 3	Pattern Select - 3	Pattern Select - 3