Ryan Lena
Joel Harley and Rob Seiden
5/9/08
EMID
Paul Lehrman

One Man Band v2.0

Figure 1. Joel wearing the OMBv2

Introduction

The world is full of musical instruments. Many people play them, and some play more than one. A few try to play many at the same time. On the streets, you might see an individual with an enormous apparatus on his back, a cacophony of sounds emanating from the area. This so-called one man band has an assortment of different musical instruments being played simultaneously. Stereotypically, the one man band consists of a massive bass drum on the back, cymbals between the knees, a harmonica mounted around the neck, and a guitar or some other stringed instrument in the hands. One man bands have been performing for hundreds of years with few updates. This year, the one man band got a major improvement, bringing it into and revamping it for the 21st century. The advent of the electronic, wireless one man band has the potential to change the lives of street performers worldwide.

Hardware

The One Man Band v2.0 (OMBv2) has quite an advantage over its predecessors. The main housing for the whole apparatus is simply an old jacket that was donated by Joel for the cause. It appears to be a normal jacket on the outside (Figure 1), but on the inside is contained a variety of sensors and wires that comprise the innards of the OMBv2.

The OMBv2 uses a variety of digital and analog controllers. In the left arm of the jacket there is an IR sensor on the inside of the wrist and a bend sensor in the elbow. Similarly, on the right arm there is a 3-axis 3g accelerometer in the wrist and another bend sensor in the elbow. A 3-axis accelerometer was used because we wanted to have as much control over the sound that was being played. Each of the three axes controls a different aspect of the sounds, to be explained later. Using a 3g accelerometer also enabled us to generate data only when we wanted to: tilt had only a minimal effect on the data sent to Max, and when the performer punched the air, the values changed considerably. Thus, an adequate threshold was established that was used in Max to generate bangs. On the left side of the jacket near the buttons we glued six force sensing resistors (FSRs) onto two pieces of wood, three on one, three on the other (Figure 2). These sensors are all that are needed for the OMBv2, and they allow for a great variety of sound.

Figure 2. Exterior of the jacket. Orange tape denotes locations of sensors. Numbers seen on the 6 pieces are the controller numbers used in Max.

The wiring of the jacket is very straightforward.
The FSRs and bend sensors only needed two wires, the IR three, and the accelerometer five. All of the wires were able to be routed to a single location in the jacket, since the lining was not in any way sewn to the outer layer except at the edges. This greatly facilitated the wiring of the instrument.

To send the data, a wireless MIDI transmitter was used, in conjunction with the MIDItron. A minor complication arose as a result of using the MIDItron, as 10k resistors needed to be wired in between the wires and the MIDItron. We amalgamated the wiring into a single wire, by taking all of the FSR voltage wires and soldering the resistors to them, and then soldering all of the resistors to a single voltage wire, which was routed to the MIDItron.

The MIDItron and the MIDair wireless MIDI transmitter were housed in a cardboard container that was built so that in the frenzy of the performance the wearer would not damage any of the expensive hardware essential to the functionality of the instrument. This caused a slight issue as to where to place it on the jacket, as the unit was too large to fit in the breast pocket as planned.

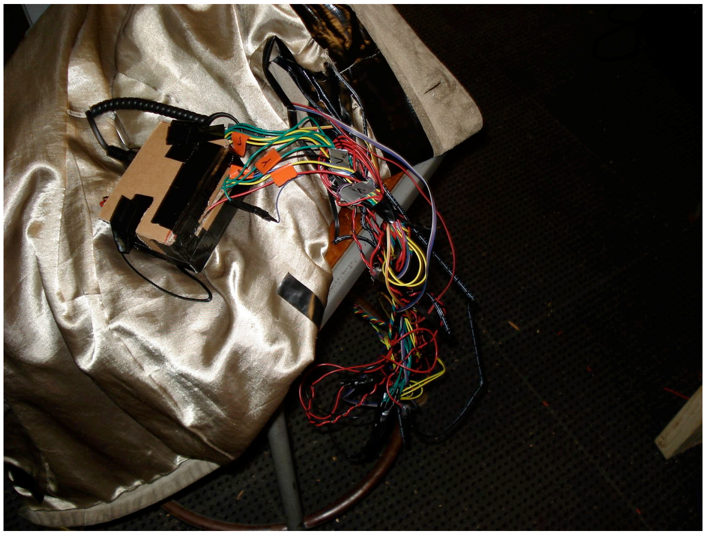

Figure 3. Wiring and combined MIDItron/MIDair apparatus

The problem actually turned out to be advantageous in that it could be placed in the right side of the jacket opposite the FSRs, and all of the wires could thus be hidden and contained in the back of the jacket, between the lining and the outer layer.

The first major challenge that presented itself in the development of the instrument was the issue of what sensors would be used to bang out notes. My initial idea was to use a combination of two sensors to accomplish this task: place an infrared (IR) distance sensor in the sleeve of one arm, near the wrist, the distance from the ground determining the note being played, and an accelerometer mounted in a similar manner on the opposite arm. A noteon command would be sent only when the performer punched the air. However, the IR sensor that was mounted was found to have a shorter range than originally thought, so that the distance from the user's wrist to the floor was beyond its capacity. This was overcome by using the bend sensor in the left elbow to determine the pitch to be played, and using the IR sensor to control pitch bend. The distance from the wrist to the performer's side was used to control pitch bend instead.

Software

Max Patch

The Max programming is fairly straightforward, although I personally did not have all that much to do with it. The layout is depicted in the following figures. To start the performance, the background tracks had to be triggered. All of the loops started playing at once when the on/off button in the Max patch was pressed. Each FSR triggers a different loop to be played in the background with a single tap. When the FSR is pressed, the track volume in Reason is brought up, and another tap turns the loop volume off. The original plan was to have four of the FSRs trigger multiple loops depending on the force that the user impacts upon the sensor, but this turned out to be extremely complicated to code for in Max, so all of the FSRs only trigger one loop. Four of the FSRs were wired as analog inputs, so the loop was triggered once a certain threshold was reached. To do this, a gate was used in the FSR module, and a receive toggle command was used in the main patch. In this manner, the switch was toggled and so only one bang was received, which started the loop playing.

The bend sensor in the left arm controlled which notes were being played by the performer. This sensor is a continuous controller sending values between 0 and 127, and we took the data sent by it and spread it out over a two octave range. We determined that the key of the loops we had chosen was E major (more specifically, an E7 chord was being played), so we had the bend sensor values correspond to the note values of an E major scale. However, once we started playing the instrument, we found that the whole scale sounded horrible with the loops. As a result, we changed the scale to be pentatonic (only using scale degrees 1, 2, 3, 5, and 6), and this sounded much better. Unfortunately, since the bend sensor is so small and sensitive, it became virtually impossible to play very accurately—note values jumped all over the place. It still worked effectively, however. The bend sensor in the right elbow controlled pitch bend and the modulation wheel, which was a very straightforward procedure in Max, merely sending a bendout command and a modwheel command to all three tracks.

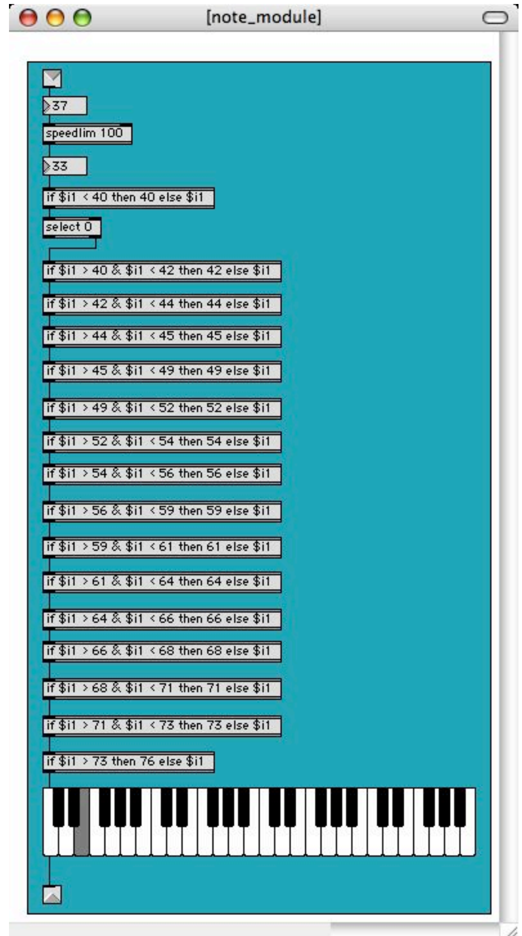

Figure 4. Scale determination in Max

Finally, to actually trigger the note to be played over the background loops, data from the three axes of the accelerometer had to be taken and made to control different aspects of the sound and send a bang at the same time. This was done by using if/then statements first to filter out all of the continuous data being generated just by moving the accelerometer. Using MIDIscope we found that when the accelerometer was punched through the air that the values jumped up to about 100 on every axis. Each axis corresponded to a different Subtractor module in Reason, each with a unique sound. We used different thresholds for each axis; that way we could selectively activate different Subtractor modules in the lead track through Reason. Normal punching activated the main track, punching a little harder activated the second track, and punching very hard activated the third track. The accelerometer also affected the sound by having the data generated determine the note velocity as well as the note duration that was sent to the noteout command. The signal was scaled appropriately so that the durations were reasonable—we scaled the data by a factor of 14, so that, for the base track, the minimum accelerometer value would give a duration of 80 * 14 = 1120 ms.

Reason Patch

The reason patch is a combination of Subtractor modules and Dr. Rex loop players. There are a total of 6 Dr. Rex modules for all of the FSRs. Each of these loops was chosen specifically to be in a funk theme. There are three different Subtractor modules, one for each axis of the accelerometer. Each of these modules has a completely different sound, so when all three combine it makes a very interesting sound. What we had wanted to do with the loops was have them all running at the same time and take advantage of the mute channel controller to turn loops on and off by merely muting them. The controller we needed wasn't defined, however, so we had to resort to a different method of activating and deactivating the loops, namely, changing the volume on each track.

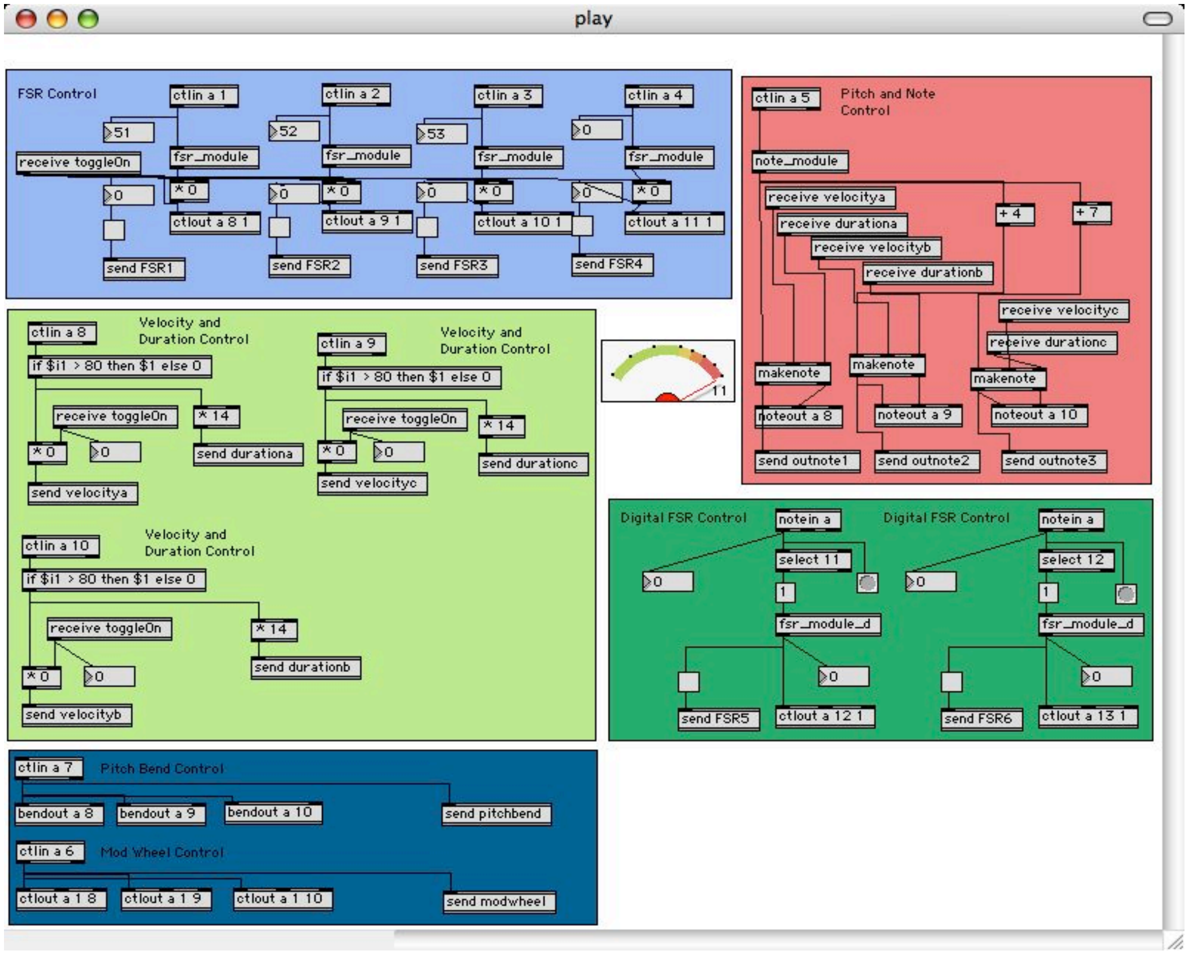

Figure 5. Main Max patch. The dial in the middle is for decoration, and is a Spinal Tap reference.

Conclusion

The OMBv2 was a great success. All of the parts came together nicely, and each member of the group did his part to ensure its timely completion. The jacket is an effective outlet for musical expression, and functions almost entirely as planned. The only thing that I would have liked to see is the IR sensor controlling the pitch played rather than the bend sensor. The IR sensor would have been able to give the performer a more fine control over the pitch, rather than the bend sensor, the values of which jump erratically. To expand on the project, I would like to make it a whole jumpsuit, with more sensors on the legs, and even some in the feet to control the drum patches better. The great thing about this instrument is its versatility. The user can easily program what the instrument sounds like through Reason, simply by changing the loops activated by the FSRs, as well as the sound of the lead played over the loops by changing the Subtractor patches. The OMBv2 was a satisfactory instrument that was easy to play and fun to assemble.

I dealt mainly with the hardware, soldering all of the wires and mounting all of the FSRs, as well as figuring out what to do with all of the wires. Joel helped out with the electrical aspect of the jacket, and spent a lot of time in the lab troubleshooting and programming the Max patch. Rob dealt mainly with the Max and Reason patches, and he pretty much made the whole thing work.