# Modular Composition Arcade

Spring 2012 • MUS 66/ES 95: Final Project Report

Jeff Prescott

*Group Members: Ben Ewing, Eli Kohlenberg, Scott Winther*

May 9th, 2012

**Project Concept:**

The motivation for this project was driven by the goal of being able to have multiple musicians contribute to create a musical product. We wanted at least two or three people to be able to pick up our instrument and easily produce sound with an intuitive design. With the goals of what we wanted to accomplish, the design was constrained into a type of ensemble instrument with modular sections. We also wanted to be able to produce a repeatable sound with our instrument; a device that created a sound that could not be replicated was viewed to be inhibited by the randomness of the design.

Our design goals led us to exploring the possibility of using wireless technology to create completely separate modules for each section. We settled on the idea of having a main "melody" section that would be connected to the computer and able to control the melody notes of the instrument as well as the key that was being played. The other two modules would be able to control the bass line of the instrument and the drums section, including alterations to a drum loop sequence.

We went with an arcade theme with our controller selection; the bass section integrates a joystick, and the drum section utilizes arcade buttons. The arcade theme, paired with the modular nature of the instrument to create music, lead to the creation of the name, "Modular Composition Arcade" for our design. It is designed to be simple, fun, and portable to play in all environments.

**Construction:**

The design of this instrument allowed us to utilize lessons learned from the creation of the "Musik's Cube." We had the vision of integrating many sensors onto this device in a simple and organized pattern. This could be done in a very elegant fashion by using a computer program called Solidworks to create a three dimensional model of the design. The computerized model of the design allowed us to visualize the final product before spending a penny on material to create anything. This way we were able to make fine tweaks and adjustments to make sure the design was what we were looking for before we produced anything. The drawings of each module can be seen below in Figures 1-3.

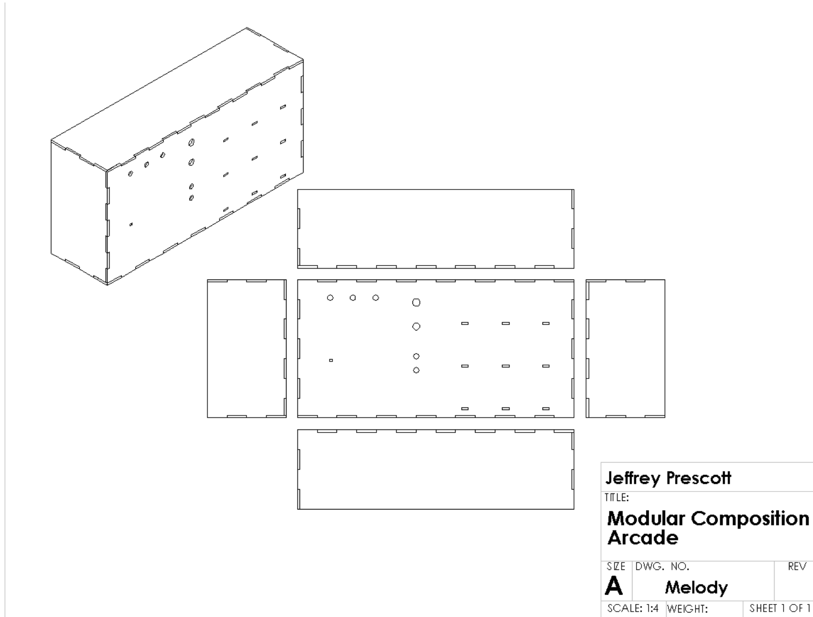

*Figure 1: Solidworks Drawing of Melody Section (image resized – neglect scale)*

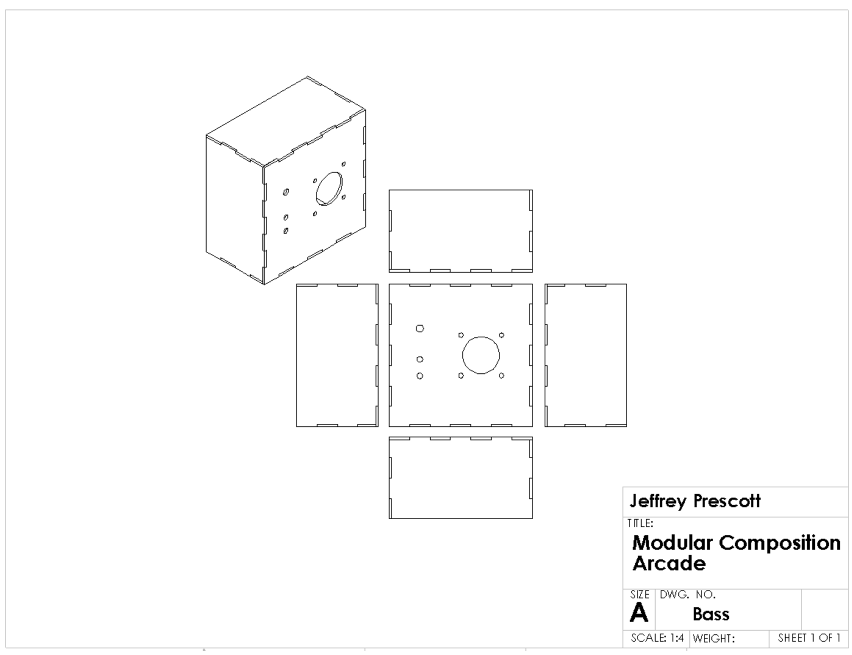

*Figure 2: Solidworks Drawing of Bass Section (image resized – neglect scale)*

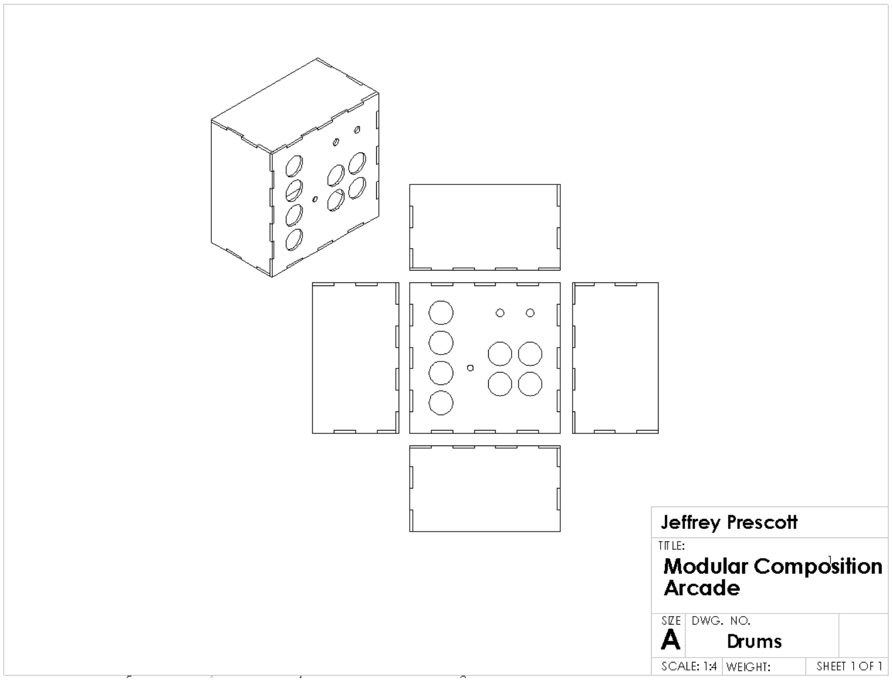


*Figure 2: Solidworks Drawing of Drums Section (image resized – neglect scale)*

It can be seen in these drawings that the three dimensional design boasted interlocking sides for a robust design. The notches in each of the sides allow for firm connections when using adhesive to assemble the modules. The three modules then connect to form a larger 14"x14" unit when the device is only being played by one individual. The melody unit is 14"x7" and the bass and drums units are both 7"x7". Magnets were attached to the insides of the connecting seams so that the modules can "snap" into place and held together when assembled.

Once the three dimensional models were created, each of the sides could be compiled in two dimensions in order to be cut out of the acrylic using the laser cutter machine. When the acrylic was cut, letters were etched into each side of the design using the laser cutter to label where each side should be assembled after production. This eased with confusion of dealing with 15 clear parts that could be assembled in different orientations. Once the entire design was cut out of the acrylic, it was easily assembled into the three dimensional design pictured in Figure 4. A small "MT" can be seen on the bottom left of the assembly; this is an example of an etched label (melody top).

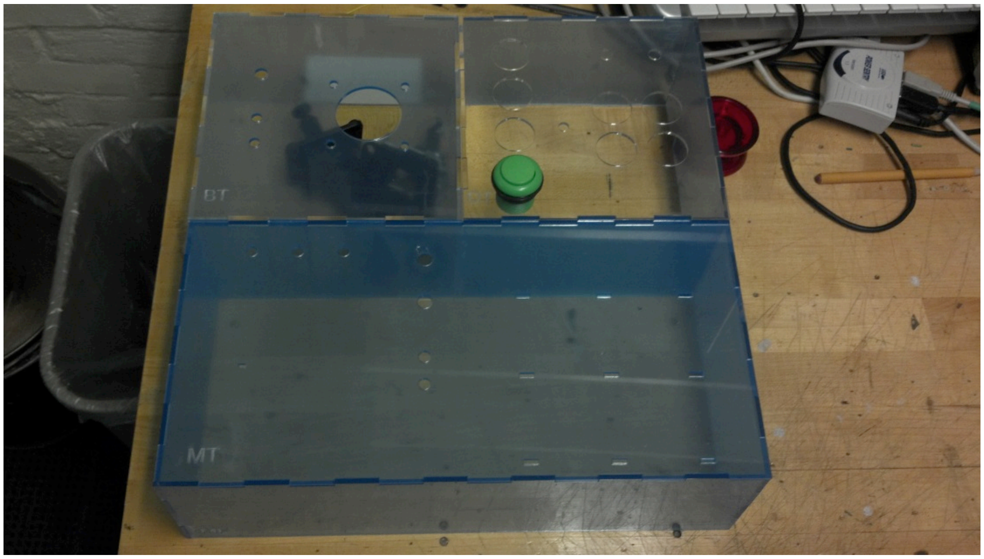

*Figure 4: Assembled Laser Cut Design*

**<u>Programming/Controller Functions</u>:**

With the assembled shell in place, the next task was the assemble all of the buttons and sensors onto the design. The buttons could simply be screwed into place after soldering wires onto their leads. They were each plugged into a digital input and a ground. Pins were soldered to the leads of the force sensing resistors (FSRs) so that ribbon cable could be used to packet up the inputs for ease of organized wiring. These were each wired to an analog input and a ground. The joystick functions using four binary buttons for each of the cardinal directions (with the possibility of the four diagonals when multiple buttons are depressed). This meant that the joystick could be wired with five digital inputs when the top click button was included. The rotary encoders function by sensing which direction they are rotating by sensing which order digital pulses are received in. Therefore the rotary encoders only use two digital inputs and one ground for each one. We used a 12 click rotary wheel on the melody module to select the key of the instrument. We were only able to program four discrete states due to the nature of the design, so this sensor utilized four digital inputs and one ground. The final input sensor was going to be a touch screen on the melody module. We were unable to retrieve meaningful data from this sensor after wiring it to the appropriate inputs and reading the information that it sent. The problem might have been a mechanical issue with the wiring, or it may have been a programming issue with the way the data was being read, but we were unable to interpret and diagnose this problem in order to get a working device. The assembled instrument can be seen below in Figure 5. This image shows all of the sensors in place without anything wired in place.

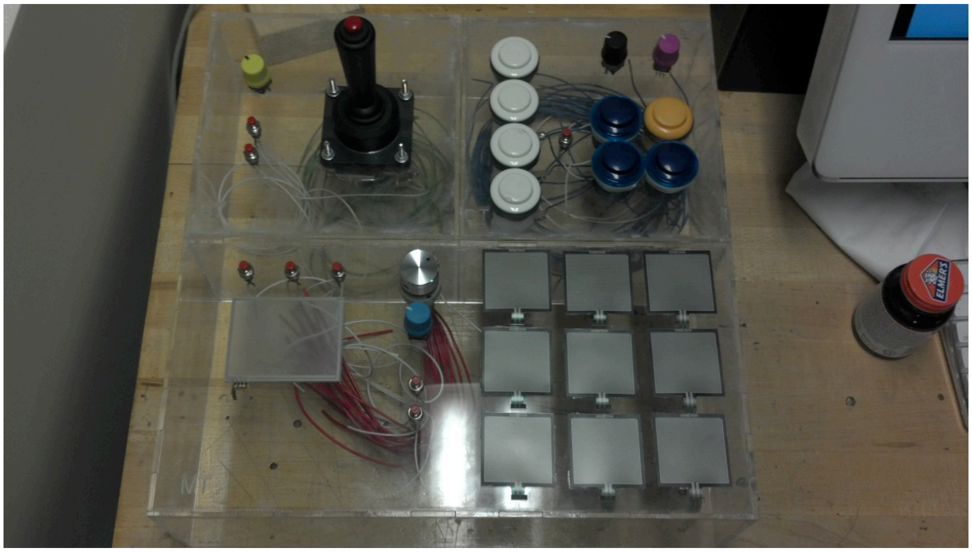

*Figure 4: Sensors installed*

The melody module features an Arduino Mega that reads the input from the sensors and sends the data to Max to be interpreted into commands that are then sent to Reason to produce sound. This main module is wired to the computer using a USB cable. The other two modules feature wireless technology to communicate wirelessly to the computer without needing to talk to the melody module. The sensors in each of the wireless modules are wired into a Wixel that is able to communicate over 2.4 GHz radio. Each of the Wixels is programmed to combine the digital button input, with an accompanying label to tell which Wixel they are sent from, into bytes that are then sent to a receiving Wixel that is connected to the computer through a USB port. The receiving Wixel is programmed to convert these decimal bytes back into binary data that can recognize which module is sending the data. The receiving Wixel will know the order that it receives the digital inputs in and can then determine which sensors are active based on which inputs are sending a digital "1." The completely wired and finished instrument can be seen pictured below in Figure 5.

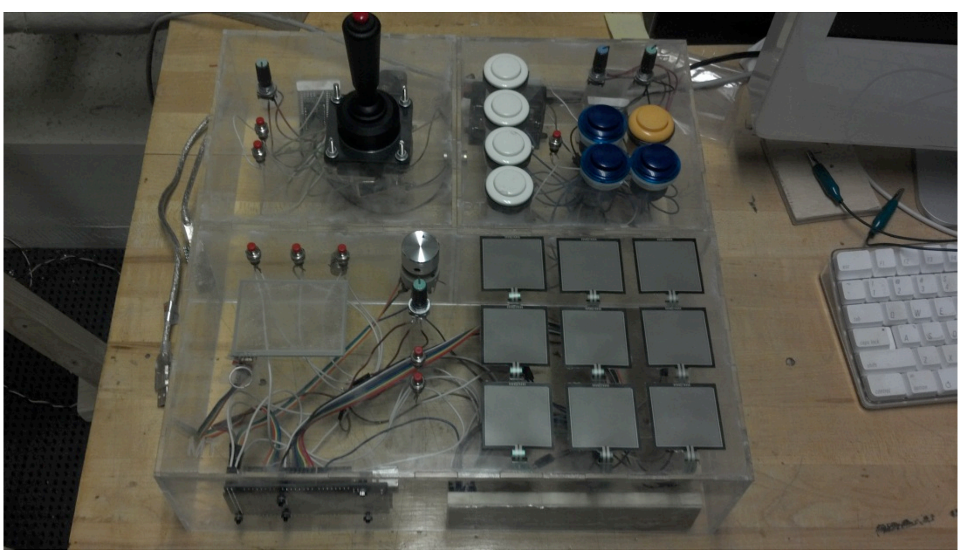

*Figure 4: Completed Modular Composition Arcade*

Once the sensors were read using the Arduino and Wixels, the data was sent to Max. The Max patch interprets this data and instructs Reason what to do musically depending on sensor changes. The patch for the melody FSRs simply interprets when the resistance on each FSR has crossed a certain threshold to determine when to send a note on and then a corresponding note off. This patch also samples the resistance 75ms after it has crossed this threshold to determine the velocity for the note. The Max patch for the bass joystick simply determines which combination of buttons is being depressed at all times to determine the corresponding note to play. The sustain button can toggle a hold on for the notes that will then be flushed out when it is released. The octave up/down buttons simply add a MIDI octave to the notes that are being played. The key selection knob interprets which position it is in by determining which digital pun is active and selects the appropriate note combination for that corresponding key. The drum kit has "one-hit" buttons that simply activate a single hit of a drum or cymbal. It also has a drum loop possibility with various drum settings. This in interpreted by reading which white button is pressed to decide how many layers of the loop to activate, and Max then tells Reason which loop to turn the volume up on, while simultaneously turning the volume off on the other loops. This way, additional "layers" can be added to the loop so that the background beat can increase from a simple kick drum, to drums, a clap, cymbals, and a snare. The rotary encoders are read by Max to determine which direction they are being turned. They are then scaled to correspond to the parameters in Reason so that an appropriate degree of turn changes the controller a reasonable amount.

**Moving Forward:**

Given more time to work on this project, the first focus would certainly to be to get the track pad working. The goal for this sensor was to give the user (musician) the ability to continuously change two related controller parameters (for example, LFO frequency and LFO rate) while playing the melody notes. The vision for this device would be to enable the option to change which two parameters were being controlled by using buttons above the track pad. This would give a very useful additional control to the musician for expression during play.

A second improvement that could be down the road is moving the system to a completely wireless model. This first prototype was kept with the melody module wired to the computer simply because there were at least 11 analog inputs (9FSR, 2 track pad) in the design and this data would be hard to transmit wirelessly via Wixels. However, with future design it might even be advantageous to use two Wixels for the melody module if necessary to enable the musicians to play completely wirelessly.