Abstract
This study will demonstrate the feasibility of re-creating an expensive piece of analog hardware, in this case the Roland TR-808 Drum Machine, using a low-budget Arduino-based microcontroller and related inexpensive circuitry. The following proposal will outline the information and strategies necessary to achieve feasibility.

Significance
In the realm of music production, fundamental analog equipment has become largely superseded by cost-effective software based solutions, which have succeeded in making the idea of owning a physical instrument a mere novelty. Although this study has the ability to make technology of the early 1980's easily accessible and customizable, its major significance lies in the culture of music production. The modeled drum machine (TR-808) has always been an expensive and hard-to-come-by instrument. This is the same case with many other historic analog music devices in which software applications have trumped the physical presence of the instruments. This study, when completed and documented, could allow an entire new demographic of people to implement and customize their own music machine of the type, using easily modified code and inexpensive hardware. The technological advancement is rudimentary, yet the cultural impact could be huge.

Background
The Roland TR-808 Rhythm Composer is a drum machine that was produced in the eighties with the groundbreaking ability to program the entire percussive pattern of a song. The instrument has a row of 16 buttons with corresponding indicator lights, which are at the core of the 808's functionality. The buttons are mapped 1-16 indicating their respective time-signature locations (in our case 4 musical bars composed of 16 beats). The lights blink on beat across the row of buttons so the user knows what part of the beat is currently playing. After a particular channel and its related sound are selected, the user can program the sound into the drum pattern at particular time locations using the buttons to produce a looped beat. As opposed to what I will study, the 808 machine used internal sound synthesis to produce robotic sounding drums (I will determine the feasibility of playing back pre-recorded samples). Other notable features of the 808 machine that I will replicate include: channel selector buttons, sound selector knobs, tempo adjuster knob, and volume knobs.

I plan to prove the feasibility of using an Arduino Uno as a cost-effective way to control an analog drum machine. First, I will define core capabilities of the drum machine to make it simple yet powerful:

- The sixteen main buttons will each represent ¼ musical bar time-segments and respond to the user by playing the sound at that particular moment.
- The tempo knob will allow the user to find an appropriate musical beats per minute value (in a range deemed reasonable).
- The channel buttons will allow the user to toggle through different sounds.
- More knobs will adjust the playback volume of the individual channels.

- A 3.5mm auxiliary output will allow the user to connect to a speaker or headphones to hear the audio being played in real-time.
- For the current sound channel selected, the 16 indicator LEDs corresponding to the 16 buttons should appear on if the button has been toggled on, and off if the button has been pushed again and toggled off

There are many inherent challenges in determining the feasibility of this project. One of the biggest concerns will likely be sound quality. Real-time audio processing has been proven to work on the Arduino Uno; however, this has mostly been demonstrated using a low 8-bit sound resolution. Alternatively, this problem can be avoided with the use of an Arduino-compatible mp3 module. The Catalex Serial MP3 Player is an add-on module that allows the Arduino to communicate via a UART (Universal Asynchronous Receiver/Transmitter) IC on the module, which can read audio files from its onboard microSD slot and output the signal directly to a female auxiliary connection. This particular module communicates at 9600 baud, and it can read and generate audio signals up to 48kHz in sample rate. This sample rate is considered high-quality and fits modern audio formatting standards. Using commands over the serial port, the module can select particular files to play, adjust the volume, and pause playback audio. Demonstrating real-time audio playback using the serial mp3 player will aid in determining feasibility of replicating the TR-808 drum machine.

Another thing to note will be the scarcity of the I/O pins on the Arduino Uno. Without clever implementation, the project could easily use more input and output pins than available on the microcontroller, so it will be necessary to multiplex our push-button inputs in order to allow all 16 buttons to be functional on only 4 analog input pins. This can be done by wiring the buttons in a parallel design, with unique resistor values in series with each button. Upon being pressed, the Arduino can detect a state change on that pin, and accurately read the unique inputted voltage from each button on the singular input pin. We can use this in software to trigger specific audio playback events for each unique voltage value. In addition to the buttons, our LED indicator light outputs will need to be multiplexed as well. In a technique called Charlieplexing, we can utilize all three of the Arduino digital pin states (HIGH, LOW, INPUT) to target specific LED's in our circuit. We have already demonstrated functional LED Charlieplexing in our first lab. Push-button multiplexing will need to be properly demonstrated in order to determine feasibility of our project.

The programming of our drum machine will likely be tedious, yet achievable, using many conditionally toggled routines to allow for the complex functionality of the drum machine. A major concern with the code will be its ability to keep the audio playback sounding natural and musically quantized for the current tempo selected by the user. The main looping of the sound pattern must be consistent with the selected beats per minute value and not have any evident lag in the looped playback. Minimal disparities in playback time sound natural to a listener; if the resulting playback audio has a consistent rhythm, it can be considered music. Once the mp3 module is functioning, we will be able to test for discernable differences in the playback. We can create a tester drum pattern in software, and upon listening, we can manipulate the code to "push" it as close to perfect quantization as possible. We will likely have to time every

sample and stop the playback once that particular quarter bar in time has passed. The feasibility of accurately playing back the drum pattern can be confirmed if we hear a consistent rhythm that is enjoyable to the ear.

The last major barrier of the study is being able to playback multiple samples at one time. As defined above, the replicated drum machine could potentially have multiple different sounds selected to play at the same instant of time. Unfortunately, the Arduino Uno is limited by its 8-bit processor, so extensive byte by byte computations would need to be done in real-time to blend the audio signals in software. This would ruin the ability to keep our samples musically quantized. This also severely reduces the bit-depth and sample rate we would be able to achieve, and seems unfeasible. The alternative solution would be to merge multiple analog audio signals in a resistor circuit to produce the proper blended sound. If we can demonstrate the blending of multiple audio signals in an analog mixer circuit, then it follows that several mp3 modules could be chained together in a circuit as such.

Technical Objectives

- Ability to control audio playback using asynchronous serial communication with the Catalex Serial MP3 Player
    - We will demonstrate the ability to play a specific .wav file from the onboard microSD card and immediately deny any subsequent files from being played
    - We will demonstrate the ability to change the output volume of the mp3 module
- Ability to multiplex 4 push-buttons to one analog input pin and accurately interpret the button that was pressed
    - We will note the resistor values used in the circuit to maximize the accuracy and consistency of our analog input reading
- We will demonstrate that our software can properly handle the timing of playback events to produce a natural and consistent rhythm to the human ear
- We will show that multiple analog audio signals can be blended together using resistors in a mixing circuit

How to Measure Success

Feasibility will be demonstrated if the following outcomes can be achieved in the technical objectives:

- If the playback audio is quantized enough to be considered musically sound (this is subjective, but will be sufficient as long as the resulting audio pattern is pleasant and rhythmically consistent to the human ear)
- If the sampled audio files on the microSD card can be played back and toggled through using the Catalex Serial MP3 Player
- If *n* buttons or can be identified individually while multiplexing on fewer than *n* input pins (in our case 4 buttons to 1 analog input pin)
- If the audio looping is seamless enough to not be noticeable by the average person (the loop restarts with a small enough pause as to not be heard)

- If we can blend and play back 2 or more analog audio signals together through a mixer circuit