Mixing Digital and Analog Audio Signals

By

David Robison

Senior Project

ELECTRICAL ENGINEERING DEPARTMENT

California Polytechnic State University

San Luis Obispo

2010
# Table of Contents

Table of Contents .................................................................................................................. 2

i. List of Tables and Figures ................................................................................................. 4

ii. Acknowledgements.......................................................................................................... 6

iii. Abstract .......................................................................................................................... 7

I. Introduction ....................................................................................................................... 9

   Overview ............................................................................................................................ 9

   Motivation .......................................................................................................................... 10

   Context and Environment ................................................................................................. 10

   Description of Customer ................................................................................................. 11

   Alternative Solutions ....................................................................................................... 12

II. Background ...................................................................................................................... 14

III. Requirements ................................................................................................................. 17

IV. Design Approach Alternatives ....................................................................................... 19

V. Project Design .................................................................................................................. 30

   Description of User Interface .......................................................................................... 30

   Blackbox Diagram .......................................................................................................... 32

   Block Diagram ................................................................................................................ 33

   Detailed Design Schematic of the Switch Circuit ............................................................ 34

   Detailed Design Schematic of the Audio-Handling Circuit ............................................. 35

   Description of Major Subsystem Designs ....................................................................... 36

VI. Physical Construction and Integration ........................................................................... 44

   Physical Layout of the Device .......................................................................................... 44

   Project Packaging ............................................................................................................ 45

   Photos of the Finished Product ....................................................................................... 46

VII. Integrated System Tests and Results ............................................................................ 47

   Demonstration of Specifications....................................................................................... 47
## i. List of Tables and Figures

<table>
<thead>
<tr>
<th>Table/Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 1</td>
<td>Terminology required to understand the project</td>
<td>14</td>
</tr>
<tr>
<td>Figure 1</td>
<td>MP3 Trigger Datasheet</td>
<td>20</td>
</tr>
<tr>
<td>Figure 2</td>
<td>MP3 Trigger Commands</td>
<td>20</td>
</tr>
<tr>
<td>Figure 3</td>
<td>Diagram showing the microcontroller inner-workings</td>
<td>26</td>
</tr>
<tr>
<td>Figure 4</td>
<td>Writing the Software to Control Digital Processing</td>
<td>27</td>
</tr>
<tr>
<td>Figure 5</td>
<td>Picture of the Completed System</td>
<td>27</td>
</tr>
<tr>
<td>Figure 6</td>
<td>Alternative Mixer Design</td>
<td>28</td>
</tr>
<tr>
<td>Figure 7</td>
<td>Alternative Buffer Design</td>
<td>28</td>
</tr>
<tr>
<td>Figure 8</td>
<td>Front Panel Design</td>
<td>30</td>
</tr>
<tr>
<td>Figure 9</td>
<td>Back Panel Design</td>
<td>30</td>
</tr>
<tr>
<td>Figure 10</td>
<td>Blackbox Diagram of Overall System</td>
<td>32</td>
</tr>
<tr>
<td>Figure 11</td>
<td>Block Diagram of Entire System</td>
<td>33</td>
</tr>
<tr>
<td>Figure 13</td>
<td>Switch Circuit Magnitude Response</td>
<td>34</td>
</tr>
<tr>
<td>Figure 12</td>
<td>Switch Circuit Schematic using LTSpice</td>
<td>34</td>
</tr>
<tr>
<td>Figure 14</td>
<td>Audio-Handling Circuit</td>
<td>35</td>
</tr>
<tr>
<td>Figure 15</td>
<td>Complete System with Circuits and Pictures of Devices</td>
<td>44</td>
</tr>
<tr>
<td>Figure 16</td>
<td>Front Panel Design</td>
<td>45</td>
</tr>
<tr>
<td>Figure 17</td>
<td>Back Panel Design</td>
<td>45</td>
</tr>
<tr>
<td>Figure 19</td>
<td>System and Bread Board Circuit</td>
<td>46</td>
</tr>
<tr>
<td>Figure 18</td>
<td>Drum Trigger Set-up</td>
<td>46</td>
</tr>
</tbody>
</table>
Figure 20: System and Bread Board Circuit Close-up........................................... 46

Figure 21: Drum Trigger Testing for Voltage Levels................................................. 49

Figure 22: Turning the Mixer Knob Counterclockwise to Clockwise with a 0.1VDC Voltage as one signal, and a 1kHz Sine Wave as the Other ........................................ 50

Figure 23: Transmission Test of the Entire System Using a 1kHz Sine Wave ............ 51

Figure 24: Transmission Through a Buffer ............................................................ 51
ii. Acknowledgements

I would like to thank all my professors and friends who have guided me throughout my college experience. In specific, I would like to thank Jeramie Bianchi who individually helped me with any problems and concerns I may have had along the way. As one of my best friends, he has been the most understanding and helpful electrical engineering friend anyone could ask for. I would also like to thank my Digital Signal Processing professor Dr. Pilkington for helping me understand the digital realm and the power it holds. I also want to thank Dr. Pilkington for being my advisor for my senior project. My Analog Filter Design professor, Dr. Derickson, provided me with an understanding of the analog world and really gave me a good perspective on how to think about real-world problems and potential solutions. Through his teachings, I was able to think of the many tools I have learned over the years, and apply them to working-solutions. Finally, I would like to thank all musicians, especially my brother, Adrian Robison, for motivating me to make my project succeed.
iii. Abstract

I have been playing the drums for thirteen years. I started out on an electric drum set and then progressed to an acoustic set. I have always wondered what it would sound like to combine the two types of sounds and this senior project dissected that interest to its raw form. Delving into the problems and solutions of combining digital and analog audio signals intrigued me and thus, became my senior project. The design process included research into the analog and digital realms, computer simulations, and actual implementation successes and failures.

The signal path begins at the bass drum where a piezoelectric drum triggering transducer converts vibration into a specific voltage level. This voltage level flows into a voltage comparator circuit. When the threshold voltage, pre-determined through various test methods, of the voltage comparator is exceeded, the rail voltage flows into a voltage-controlled, open-collector N-MOSFET circuit. This acts as a switch which closes when the threshold voltage has been exceeded, completing the circuit of the external trigger pins that directly trigger pre-selected MP3 tracks on an MP3 Trigger circuit. This signal flows into a voltage follower circuit which acts as buffer, eliminating distortion. Finally, the signal flows into a summing circuit with an adjustable gain.

Meanwhile, a microphone picks up the bass drum’s acoustical energy (sound waves) and converts this energy into electrical energy (the audio signal). This signal
flows into a preamplifier which boosts the low-voltage signal to a reasonable level. This signal flows into a voltage follower which acts as a buffer for the microphone signal. Finally, this signal also flows into a summing circuit with an adjustable gain.

The summing circuit combines the MP3 audio signal with the microphone signal. The output of this signal connects to an output jack and a quarter-inch cable connects this output jack to a PA system which has a power amplifier. This is necessary in order to drive the speakers.
I. Introduction

Overview

My project aims to improve live audio for acoustic drummers. The drummer usually gets a maximum of 8 channels to accommodate 8 microphones. Standard for this are one (1) bass drum microphone, one (1) top snare microphone, one (1) bottom-snare microphone, one (1) hi-hat microphone, three (3) tom microphones, and two (2) overhead microphones for the crash cymbals. My senior project will show the capability of one (1) bass drum microphone and one (1) drum trigger which could later be applied to the snare and toms along with any drum that would want to be used purely for effect. As a drummer who plays at local venues, I have noticed that no matter what venue I play at, the only microphone the drummer is guaranteed is the bass drum microphone.

The goal of the project is to allow the bass drum to sound great, even if the original sound of the drum doesn’t sound as good as you would like. If the drum does sound good, my project will allow the user to blend that sound with a drum sample. Drum samples are almost always used in the production of bands’ albums, and my senior project would allow the audience to hear a better representation of the actual CD they have been listening to.

Right now, a company called Roland makes drum triggers which the drummer places on the acoustic drums. These cost $90 each. The drum triggers plug into a
MIDI converter ($350) which plugs into a drum module (the brain containing factory drum samples costing around $500+). Finally, the drum module plugs into a mixer which outputs the sounds to the loudspeakers.

This project describes the development of a unique audio idea which combines digital and analog audio signals. The project goes into detail of how the product was made with explanations of each step from start to finish. After going through this report, you will be able to understand the many building blocks of the product and how they work individually and together with other components.

**Motivation**

The drummer is the epitome of the saying “What you see is what you get.” Typically, if the drum set looks cheap, it sounds cheap. Electric drum sets sound good but very digital as well. They are very expensive (in the thousands of dollars) and disallow the drummer to fully get into the music. As a drummer, I really wanted to combine the live, punchy sounding bass drum microphones provide at venues with the digital sound of perfection. This would sound unique to the listener, providing an extra layer of interest.

**Context and Environment**

Any drummer who wants their kit to sound much better without paying a lot of money would want to use my senior project. Many people do not want to purchase electric drums because it is not the “real thing” and it is usually a lot more pricy. My idea would allow the people that fall into these categories to have professional-
sounding drums during live-shows. The project is geared towards drummers that play at local venues, the band’s jam studio, or parties. The circumstance that the project would be used is when the band wants to add a little extra detail or texture to a song. If effects are needed for a certain part of the song, my project would work perfectly. If there is time to prepare for the show, there is time for the miniscule amount of time it takes to set up the project.

The percussionist needs an acoustic drum set or other percussion instrument. The instrument has to be played in a dry, covered location so that water does not damage the electrical components of the set-up. An AC outlet needs to be available for the project for it to function.

The project assumes that the percussionist has a computer and has drum samples to load onto the micro SD card. If not, the Drum Box could come with a micro SD card already pre-filled with drum samples available for use. A cool feature about the use of the project is that the drummer can load his/her own drum samples by simply dragging the files to the micro SD card from the computer.

Description of Customer

Any person that plays the drums and is looking for a new and unique sound may look at this product and buy it based on the intriguing idea of incorporating digital samples to the drummer’s overall sound. Furthermore, it is a cost-effective alternative to buying expensive digital machines that may or may not have the most desired sounds. Many drummers play acoustic drum sets because of the feel and the price. Many people do not want to purchase electric drums because it’s not the “real
thing” and it is usually a lot more pricy. My idea would allow these people to have professional-sounding drums during live-shows.

The project is geared towards drummers that play at local venues and parties. Bands that are on tour playing at huge venues have the money to purchase the more expensive version of my project and usually have great equipment at the front of house sound booth.

**Alternative Solutions**

The customer’s alternative is to buy a drum trigger ($15-$100), that feeds into an electronic computer drum module that has factory samples already loaded onto it (The “brain” – $300-$5000), which finally goes to the PA system.

My idea enables the user to choose which sound he/she wants prior to the live show and, together with the actual microphone amplified drum sound, a fuller and fatter sound is heard. My design bypasses the purchase of a drum module that may contain samples that the drummer may not want. It is also much more affordable to purchase my drum module and provides only what the drummer feels is necessary.

Another reason why the project would be worth the money is that the sound is unique. Instead of an electric set or acoustic set, the drummer now has both. The drummer gets the feel of a drum set, an analog sound which is more dry and punchy, and a digital sound the acts as another layer to the drummer’s overall sound.

Even though drummers will most likely load samples according to the drum that he/she wants to accent, effects and interesting sounds can also be loaded. This could potentially be used for certain parts of songs where the drummer isn’t just
“holding down the beat”. The song could be played with the Drum Box 100% DRY (just the microphone sound), and when a certain part of the song comes, the drummer could turn the knob to 100% WET and use the effect when necessary.

An application of this is if there is a rap in the middle of the song and the drummer wants to change his/her kit from rock to hip-hop. There could be a little break (for the drummer to turn the knob) and then the next time the audience hears the drummer, he/she could have 808’s (deep bass that is mostly heard out of subwoofers) which makes that part of the song completely different, interesting, and unique.

Many drummers play acoustic drum sets because of the feel and the price. Many people do not want to purchase electric drums because it’s not the “real thing” and it is usually a lot more pricy. My idea would allow the people that fall into these categories to have professional-sounding drums during live-shows.
II. Background

There are a few components used in the project that require knowledge of electrical engineering and circuit design. When discussing signal processing in respect to music, there are a few technical terms to understand. Table 1 below provides important, frequently used words that may come up throughout this report.

<table>
<thead>
<tr>
<th>Table 1: Terminology required to understand the project</th>
</tr>
</thead>
<tbody>
<tr>
<td>Potentiometer</td>
</tr>
<tr>
<td>Voltage Comparator</td>
</tr>
<tr>
<td>Operational Amplifier</td>
</tr>
<tr>
<td>Line-level</td>
</tr>
<tr>
<td>Term</td>
</tr>
<tr>
<td>--------------------</td>
</tr>
<tr>
<td>Audio amplifiers</td>
</tr>
<tr>
<td>ADC</td>
</tr>
<tr>
<td>DAC</td>
</tr>
<tr>
<td>Latency</td>
</tr>
<tr>
<td>Line Level</td>
</tr>
<tr>
<td>Pulse Width Modulation(PWM)</td>
</tr>
</tbody>
</table>
time that the signal spends off. The duration of "on
time" is called the pulse width. To get varying
analog values, you change, or modulate, that pulse
width.

<table>
<thead>
<tr>
<th>DRY</th>
<th>100% microphone signal.</th>
</tr>
</thead>
<tbody>
<tr>
<td>WET</td>
<td>100% .mp3 audio file</td>
</tr>
</tbody>
</table>

Knowledge of music and how to gauge what sounds appealing is necessary for
the user of the product. Musicians and songwriters alike know how to control the
dynamics of their individual sounds so that the overall song sounds good. Although
my project could be used for fun by anyone trying to have fun on a drum set, proper
use of the project requires some recording and producing experience. Sound editing is
a skill that takes a long time to master, but sounding good using stock sounds found
on the internet or cut from a pre-existing song can still be accomplished easy enough
for a beginner wanting to sound unique on the drum set.
III. Requirements

When used by the customer, several functions must be demonstrated based on customer requirements. In the real-world, the marketing department would hold many different focus groups to find what would appeal to the majority of people that may use the product. I am a drummer for a band and have a good amount of experience recording, producing, and performing live for audiences. I know what I would think sounds good and made that into the customer requirements.

When powered on, the project must provide the ability to choose a ratio between two signals. This is accomplished by the knob labeled “DRY/WET”. The microphone signal and the .mp3 audio file must be in a constant ratio with one another so that if the DRY/WET knob is pointing in the middle, an equal amount of each signal, individually halved in volume, is heard from the loudspeakers. The drum sample blends with the bass drum microphone signal so that as one signal increases, the other signal decreases at a constant ratio. When the DRY/WET knob is turned to 100% DRY, the volume level of the bass drum microphone must be equal to the volume heard when plugged straight into the PA system. When the DRY/WET knob is turned to 100% WET, the volume level of the .mp3 audio file must be equal to the volume heard when plugged straight into the PA system.

The threshold voltage for the drum trigger must be user-defined based on how intense the drummer feels is necessary for the audio file to play. This is a trial-and-
error way of deciding the intensity and is decided upon by constantly hitting the bass drum with the bass drum pedal beater while turning the knob to the desired spot. This is an important feature because every drummer hits the drum differently.

The drum trigger must not have a noticeable delay from when the drum is struck to when the sound is heard. The drum trigger must not “double-trigger” which means the system must not be too sensitive to vibration. The threshold voltage of the drum trigger could prevent this.

Distortion must not be evident in the sound that comes out of the loud speaker. Along with this, feedback and humming should not be audible.

During live shows, the bass drum will sound professional, if not unique. This differs from the simple, punchy bass drum normally heard at smaller venues.

All processes must be done in a timeslot of about 65µs. If the allotted time is not met, the sample signal will not be synchronous with the microphone signal, and a distortion in the output signal could appear.
IV. Design Approach Alternatives

Many different design methodologies were considered which would have resulted in similar performance. At first, the drum trigger was going to be the input voltage to a microcontroller. The microcontroller was going to process this voltage and make it the threshold voltage needed to be exceeded in order for a drum hit to register.

The drum trigger plugged into a digital input and ground of the microcontroller. In order to protect the microcontroller from a voltage impulse that could potentially damage the microcontroller, a 1MΩ resistor in parallel with a zener diode rated at 5.1V was put into place. When the voltage exceeded 5.1V, the zener diode would turn on, allowing current to pass. This was a necessary component because any voltages above 6V could potentially damage the microcontroller.

*Drum Trigger Sensor Code—See Appendix A*

The microcontroller was then going to communicate with an MP3 Trigger which took care of playing an audio file previously placed on a micro SD card by the user via the computer. When the drum trigger felt a vibration intense enough to be considered an actual hit, the .mp3 audio file was suppose to play.
I realized how to somewhat code the interface between the MP3 Trigger and the microcontroller by the following datasheets in figures 1 and 2:

### Datasheets

**Using the Trigger Inputs**

The MP3 Trigger provides 7 input pins (TRIG1 – TRIG7) that can be used to trigger specific MP3 tracks on the microSD card. Use the following file naming convention to associate a particular track with a trigger:

- TRIG1: filename = TRACK001.MP3
- TRIG2: filename = TRACK002.MP3
- TRIG7: filename = TRACK007.MP3

These inputs are active low and are pulled high internally. Therefore, they can be activated either by digital outputs from another microcontroller (such as an Arduino) or by a simple contact closure (switch) to ground. The inputs support voltage levels of either 5V or 3.3V.

The trigger inputs are made available on the even-numbered pins of a dual row connector, and all the opposing (odd-numbered) pins are ground, making it easy to wire individual switches or contact closures directly to the MP3 Trigger board. Refer to the MP3 Trigger schematic for details. (The ground pins are those closest to the edge of the PCB.)

Activation of a trigger immediately causes the corresponding track to play (if it exists).

**Figure 1: MP3 Trigger Datasheet**

The following was the command summary for the MP3 Trigger:

- **Command: Trigger (ASCII)**
  - Number of bytes: 2
  - Command byte: 'T'
  - Data byte: n = ASCII '1' through '9'
  - Comments: If it exists, the track with the filename “TRACK00N.MP3” will be started, where N is the data byte.

- **Command: Trigger (binary)**
  - Number of bytes: 2
  - Command byte: 'Y'
  - Data byte: n = 1 to 255
  - Comments: If it exists, the track with the filename “TRACKNNN.MP3” will be started, where NNN is the ASCII equivalent of the data byte n.

- **Command: Play (binary)**
  - Number of bytes: 2
  - Command byte: 'p'
  - Data byte: n = 0 to 255
  - Comments: The nth track in the directory will be played. The total number of available tracks in the directory can be retrieved using Status Request command below.

- **Command: Set Volume (binary)**
  - Number of bytes: 2
  - Command byte: 'v'
  - Data byte: n = 0 to 255
  - Comments: The VS1053 volume will be set to the value n. Per the VS1053 datasheet, maximum volume is 0x00, and values much above 0x40 are too low to be audible.

**Figure 2: MP3 Trigger Commands**
I researched how to add reverb to a digital signal and was going to start programming for it once the MP3 Trigger and the microcontroller were fully interfaced. Because that never happened, I do not have any code to accomplish the reverb effect. I was going to use two ADC, 8-bit input slots for the audio coming in. One analog input would be used to sample the audio signal and one analog input would be used to control the audio effect. Reverb was going to be added to the audio sample using Digital Signal Processing. The output used would be the Pulse Width Modulation (PWM) output which handles the Digital to Analog Conversion digital output slot of the microcontroller. Timer 2 would be used to handle the timing, comparing, and interrupts that would have been needed to handle the task.

In order to do reverb, the following would have been needed to be implemented in order to create the reverb effect:

1) Store the audio sample in memory.
2) Read the delay buffer.
3) Subtract the DC offset which is controlled by the user.
4) Scale the delayed sample based on how much the user turned the potentiometer (it was just in memory and is now being scaled as it leaves its memory location).
5) Subtract the offset from the new sample.
6) Add the delayed sample to the new sample.
7) Limit the audio.
8) Add the offset.
9) Store the final sample in an audio buffer.

10) Limit the buffer index.

11) Blink an LED concurrently to make sure the reverb is working correctly.

12) Sample the value of the sample to the PWM output.

13) Test the output on a pin.

Although the list is a rough explanation of how to create reverb, the coding would be nothing more than setting registers, writing and reading samples, doing simple math, and outputting the signal. It would have been a great feature but would have taken much more programming experience. The problem that would have risen if I had gotten the reverb working would have included the level of delay from the A/D conversion time and the D/A conversion time. This is referred to as latency. The delay could be improved upon by limiting the amount of memory used. The reverb would have to be worked out until the desired effect is achieved. The time it takes to do these conversions, the digital processing necessary to achieve reverb, and the time it takes for the .mp3 file to output from the MP3 Trigger after the drum trigger threshold voltage is surpassed, would all have to sum together and be less than a noticeable amount of time. This would have been very unlikely.

Using the microcontroller would have been ideal, because I could have interfaced an LCD display which could have hopefully selected which drum sample I would want to use from the micro SD card. In the real-world, a digital display would be available for the user to see which sample is currently loaded. The ability to scroll through various different samples would be much more convenient than changing the
individual samples one-by-one on the computer. This would have added a very popular, user-friendly interface because customers love visuals while using products. This was going to be a long-shot, but it still would have been a really cool feature to the project.

All of the features mentioned require pretty good programming skills and proper interfacing techniques. I had only taken two courses on computer programming and did not have anyone to teach me the programming language or how to use it. After countless hours of reading and researching, I realized that the capabilities I desired could be done in the analog world.

Much of the time I took researching the programming language and realizing the potential of using programming seemed wasted when the program failed to do what I had desired. There were two main problems that I could not overcome. The first problem I had was interfacing the microcontroller with the MP3 trigger. I had a few steps of the interfacing complete and almost had the whole system working. This stage was ultimately replaced by analog circuitry in the final design.

Most of the digitally-controlled part of the project was luckily possible in the analog world. A microphone inside the bass drum would input to a pre-amp to obtain line-level, mix the signal with the digital audio, and sum the two using a summing amplifier.

This method of controlling the process digitally allowed me to learn so much about the capabilities of digital control and taught me how to read programming code so that I can work with Digital Signal Processing coworkers in the future.
Controlling the transfer of signals using analog components limits the capability of the project but completes the requirements of the customer. It was also easier to build and test than learning the programming language, somewhat from scratch.

Choosing the analog approach makes the product much more affordable, but it has the drawback of non-ideal components which may affect system performance. Because the device is more geared towards the aspiring musician who is looking for a cheap way of adding a new sound to his/her arsenal, the miniscule degradation of the signal quality is not of paramount importance.

To recap, the .mp3 audio file feeds into two inputs of an Arduino Microcontroller. Here, a user-defined amount of reverb is added to the audio signal. The microcontroller has input and output external circuitry. The input circuit controls the DC offset and the amount of reverb desired via potentiometers. The output circuit is essentially a low-pass filter that smoothes out the analog signal that was just converted from digital to analog. The output circuit also prevents aliasing.
The following is a diagram showing the old design and its corresponding blocks:

- **Drum Trigger**: (Intensity converts to a voltage level)
- **1/4” mono female input jack**
- **MP3 Trigger**
  - Rx  Tx
  - Audio Out
- **Micro SD flash card**
- **1/8” cable splits into wires that connect to the audio input line in jack and ground.**
- **Microphone**
- **XLR balanced cable**
- **Pre-amp**
- **Arduino Microcontroller**
  - Rx  Tx
- **Serial 16x2 LCD Display**
- **PA System**: Includes speakers and a mixer
- **DRY/WET Mixer**
- **XLR balanced cable**
- **MOBROW miniMiX**
  - http://diystrat.blogspot.com
- **Serial 16x2 LCD Display**
- **PA System**: Includes speakers and a mixer
Figure 3 below is a schematic showing the components necessary to achieve a user-defined amount of reverb:

![Figure 3: Diagram showing the microcontroller inner-workings](image)

The audio signal connects to the microcontroller via an input circuit. A 10μF capacitor buffers the signal to the analog input of the microcontroller. Two resistors and a potentiometer are adding a DC offset to the audio signal. Another potentiometer takes care of adding effects to the digital audio signal. The pulse width modulated signal outputs from the PWM audio output and passes through a low-pass filter. Originally, this low-pass filter was an RLC circuit, but using knowledge of Analog Filter Design, I transformed the filter to an active equivalent. The signal has to be filtered to avoid the aliasing effect when the signal gets sampled. At the output of the PWM output, the signal is still ridged from the digital-to-analog conversion and the filter smoothes out the sharp breaks, creating a nice audio signal. This also prevents distortion. The Arduino Microcontroller needs 2.5V-peak at the input of the ADC for best quality.
Figures 4 and 5 below show the software and picture of the complete system, respectively:

Figure 4: Writing the Software to Control Digital Processing

Figure 5: Picture of the Completed System
Figure 6 below is another design for the line mixer that I was considering:

![Simple line mixer design by Tomi Engdahl](image)

I did not use this design because it would have added more cost to the project and the double potentiometer seemed like an easier choice. I wanted to investigate the problems with only using a potentiometer and was curious to see how much the signal strength degrades by using simple components. My design called for the easiest, cheapest solution for mixing digital and analog signals. Figure 7 below is the buffer I tried using:

![Alternative buffer design](image)
I also tried implementing a “Fast Voltage Follower” as a buffer for my audio signals. This did not work as well as I thought and in fact, it only added more noise to the signal if the signal even got through in the first place.
V. Project Design

Description of User Interface

Figure 8 below is a picture of the front panel:

![Front Panel Design](image)

**Figure 8: Front Panel Design**

The user can individually adjust the digital or analog gain, the level of intensity needed for the drum trigger to output the audio file, and the desired ratio of microphone level to sample level.

Figure 9 below is a picture of the back panel:

![Back Panel Design](image)

**Figure 9: Back Panel Design**

A microphone cable connects the microphone to the XLR input jack. Microphone cables are usually XLR-to-XLR which makes this a convenient input choice. The drum trigger input requires a standard ¼” mono cable. The ¼” output
means the signal will no longer be balanced but drives the cost down and allows the price to drop. Much more complicated circuitry would be necessary to ensure a good quality balanced signal which is unnecessary to the potential customer. The MP3 Trigger compartment is spring-loaded and is the most problematic aspect of the design. It induces user-error into the equation but protects the MP3 Trigger from the external environment. The Micro SD card can be removed to put more digital samples onto it by simply pushing lightly.
**Blackbox Diagram**

Figure 10 below are the inputs and outputs of the overall system.

![Diagram](image-url)

Figure 10: Blackbox Diagram of Overall System
**Block Diagram**

Figure 11 below are the components used in the project. It also shows what is connected to what.
Detailed Design Schematic of the Switch Circuit

Figures 12, 13, and 14 show the schematics and simulation of the circuits used in the project.

Figure 12: Switch Circuit Schematic using LTSpice

Figure 13: Switch Circuit Magnitude Response
Detailed Design Schematic of the Audio-Handling Circuit

Figure 14: Audio-Handling Circuit
**Description of Major Subsystem Designs**

*Drum Trigger*

The drum trigger consists of a piezoelectric sensor which converts pressure into a specific voltage. This particular sensor has a metal case to protect it and can be easily connected to a bass drum by screwing a lug nut through the corresponding hole on the drum trigger. I tested the output voltage range by connecting it to an oscilloscope and recording the voltage outputted when pressure was applied to the sensor.

*Voltage Comparator*

\[
V_{\text{ref}} = \left( \frac{R_2}{R_2+R_1} \right) V_{\text{in}}
\]
\[
V_{\text{ref}} = \left( \frac{1k\Omega}{1k\Omega + 4k\Omega} \right) 5V
\]

This allows \( V_{\text{ref}} = 1V \). This is the value that I came up with for the threshold voltage necessary to exceed for an audio file to play. I came up with this value experimentally using an oscilloscope to measure the output voltage of the drum trigger and a trashcan that acted as a bass drum head.
The voltage comparator sets the threshold voltage of the drum trigger. This is a key component of the project because it determines when the audio file will play. When the threshold voltage is exceeded (the voltage decided for the negative input of the op-amp), the rail-voltage flows through the output of the comparator, switching the output from OFF to ON. The inputs for this particular op-amp are very sensitive and a difference of a few millivolts between the two inputs will switch the output. The voltage comparator can operate on a supply of up to 32V. A problem may occur with the op-amp when input voltages change very slowly, but that problem does not exist in my project because the voltage is changing very abruptly.

I first tested the drum trigger and found that when connected to the bass drum, the minimum voltage necessary to register a drum hit was around 1V. Using voltage division, I determined that using a 4:1 ratio of resistors will set the threshold voltage at 1V. To make the project more user-friendly, a double potentiometer could be used so that as R1 increases R2 decreases and vice-versa. The ratio of the resistances is more important than their actual values.
This N-Channel, Power MOSFET is normally used in power applications, but the faster, smaller voltage MOSFETs are very sensitive to current and kept breaking. The fact that this MOSFET induces a little more delay to the overall system makes it a bad choice for my project. When the threshold voltage of the voltage comparator is exceeded, the rail voltage flows out of the voltage comparator and into the MOSFET. Since I am using the MOSFET as a switch, when the threshold voltage has not been exceeded, the MOSFET does not turn on because it is voltage-activated. This is a key component of the design because it allowed me to use analog parts rather than doing this part of the project digitally. The rail voltage easily turns the MOSFET on, completing the circuit of the MP3 Trigger which plays the digital audio file.
When the trigger pins located on the MP3 Trigger are connected by a wire, an .mp3 file is outputted from the 1/8” audio out jack. The circuit I constructed, using a voltage comparator circuit and a MOSTFET, acts as a switch for the trigger pins. This circuit allows the .mp3 file to play only when the drum trigger senses that the bass drum has been struck. This ability to control when the audio file is played is very important to the project’s success.

Once the drum has been struck at intensity greater than the user-specified level, the MP3 Trigger will output the .mp3 file to the analog circuitry in real-time. This is where the signals get buffered and combined. The levels of each signal and the amount of each signal that is outputted can all be changed by the user. The signals cannot take too long to go through the circuit because noticeable delay will not sound good.

Because the process of choosing which audio files will be used is up to the user, some experience clipping and altering audio files is necessary to use the project. Although there are many different techniques on how to obtain desired digital sounds,
the preferred way of installing drum samples is as follows: Drum samples are edited on the computer using a simple, audio-editing program. The editing program can be downloaded for free online (Audacity) or the user could already own a program such as Pro Tools. This program is used for editing the audio files. This enables the user to zoom-in and trim the samples so that the sample starts at exactly zero. Pro Tools also provides the capability to change the sound by adding digital filters and effects to the sound. If the trigger is asserted before the sample is finished, the .mp3 file stops and immediately plays the .mp3 file from the beginning.

The samples are then transferred to the micro SD flash card (1GB is $3) by dragging and dropping the files from a computer to the card. The MP3 Trigger has the capability of reading the files on the flash card using the FAT16 format. When the drum trigger is asserted, the MP3 Trigger outputs the .mp3 file in real-time.

![Buffer Diagram](image)

The first time I built the working prototype project, I hadn’t yet put into place a buffer for the audio signals. In theory, placing a buffer at the input of the summing junction should get rid of any distortion that may be present. The buffer is made of a simple voltage follower with a bypass capacitor. This component of the project really
makes a difference in the overall sound of the project because it almost acts as a compressor for the amount of current flowing through the circuit.

**Double Potentiometer**

The potentiometer on the top controls the microphone signal and the potentiometer on the bottom controls the .mp3 audio file. As the knob is turned clockwise, more resistance is added to the microphone signal and less resistance is seen by the .mp3 audio file. The opposite effect occurs when the potentiometer knob is turned counterclockwise.

There were a couple problems I noted about the double potentiometer. I noticed nulls when the knob was cranked all the way to one side or the other. This had the effect of not letting either of the signals pass to the output. This component is very far from ideal, especially when used in my project where the quality of the audio signal is only as good as the weakest link. The double potentiometer still does a pretty good job of passing one signal while not passing the other signal. The ratio between signal levels in-between caused problems because resistors alone do not do a very good job by themselves in attenuating signals accurately.
The double potentiometer allows one of the main functions of the project to be accomplished. A user-decided amount of the microphone signal over the .mp3 audio file allows the DRY/WET mixer to operate.

The DRY/WET knob on the box creates a ratio between the original microphone signal and the drum sample. When the knob is at DRY, 100% of the output will be the original, amplified microphone sound. When the knob is at WET, 100% of the output will be the digital sample. Anywhere in the middle provides a ratio of the blended sounds. A dual potentiometer makes this knob work and allows one signal to be heard over the other signal based on a resistance ratio.

The levels of both signals must be at line-level at the input of the DRY/WET Mixer. This way, at 50/50, the individual levels will be half as strong as when at 100/0 or 0/100. The levels will add together in the summing circuit to create a signal that is as loud as when the ratio is 100/0 or 0/100.

**Summing Amplifier**

![Summing Amplifier Diagram](image)

This stage of the signal path was originally not in the overall design but it does the task of combining the two signals efficiently rather than just taking the two outputs of the double potentiometer and tying them together. Operational amplifiers set up as summing amplifiers have a virtual ground which makes the input currents
linearly proportional to the corresponding source voltages. The summing amplifier prevents the two sources from interacting with each other which lowers noise distortion and makes for a better signal transmission. Summing amplifiers are often used in audio mixing.

Using the summing amplifier also has the benefit of altering the gain of each signal. Two 10kΩ potentiometers can be used in place of R1 and R2 with Rf fixed at 10kΩ. This provides additional functionality to the project and could be desired for adding an additional boost to the signals.

**Public Address (PA) System**

![Public Address (PA) System](image)

This section could potentially be added to future projects to see the effect of every single component of the system. The power amplifier enables the resulting signal at the output of the summing amplifier to be heard through the loud speakers. This is possible because there is a major increase in the output current capability. Proper power packaging is necessary to handle the increased dissipation of heat.

I am currently in a band that has all the necessary equipment to play a live show, and the PA system was readily available to use in the project. Because of this convenience, I found it unnecessary to design this part of the project.
VI. Physical Construction and Integration

Physical Layout of the Device

[Diagram showing physical layout with labels for Microphone, Drum Trigger, Switch, MOSFET, Buffer, PA System, MP3 Trigger, Audio Out, Micro SD flash card, XLR balanced cable, Pre-amp, Voltage Comparator, and DRY/WET Mixer.]

Figure 15: Complete System with Circuits and Pictures of Devices
Project Packaging

Figures 16 and 17 below are potential designs for the case the circuitry will be inside.

**Figure 16: Front Panel Design**

**Figure 17: Back Panel Design**
Photos of the Finished Product

Figure 18: Drum Trigger Set-up

Figure 19: System and Bread Board Circuit

Figure 20: System and Bread Board Circuit Close-up
VII. Integrated System Tests and Results

**Demonstration of Specifications**

My own approval of whether or not the system sounds good was needed to meet the customer’s requirements. The system works, but I noticed noise and other types of distortion. This could be prevented by using high-quality cables or implementing the pre-amp into the project so that there isn’t a weak point. Both audio signals go through and are definitely audible over each other.

When powered on, the project has the ability to choose a ratio between two signals. The accuracy is a little off, but the signals attenuate like they’re supposed to. The knob labeled “DRY/WET” accomplishes this task. The microphone signal and the .mp3 audio file, however, are not in a constant ratio with one another. The .mp3 audio file is audible even when the potentiometer is turned all the way to DRY. This shows inaccurate resistances and a new method may need to be looked into. When the DRY/WET knob is pointing in the middle, the two signals are about equal. This satisfies the customer requirement. The drum sample blends with the bass drum microphone signal so that as one signal increases, the other signal decreases at somewhat of a constant ratio. When the DRY/WET knob is turned to 100% DRY, the volume level of the bass drum microphone is about equal to the volume heard when plugged straight into the PA system. When the DRY/WET knob is turned to 100%
WET, the volume level of the .mp3 audio file is about equal to the volume heard when plugged straight into the PA system.

The threshold voltage chosen for my project worked, but it may be inconsistent for other drummers. A more sensitive MOSFET would be needed for faster switching, and a more efficient way of triggering the .mp3 audio file is desired. The drum trigger only triggers one volume-level of the .mp3 audio file, so better techniques must be researched to make the project more realizable in its functionality. When playing the bass drum, I noticed a small amount of inconsistent triggering. The trigger did its job.

The drum trigger does not have any “noticeable” delay from when the drum is struck to when the sound is heard. The drum trigger sometimes “double-triggers” which means the system is a little too sensitive to vibrations.

Distortion was mildly evident in the output sound and could be slightly heard out of the loud speakers. Along with this, feedback and humming were also slightly audible.
**Test Descriptions**

Figure 21 below shows drum trigger tests. I had a friend hold the drum trigger up to a garbage can and kicked the garbage can. This seemed like a reasonable way of testing the drum trigger voltage limits.

*Figure 21: Drum Trigger Testing for Voltage Levels*
Figure 22 below shows the test for the DRY/WET mixer. I turned the mixer double-potentiometer clockwise from its starting point and saw how a 0.1V DC signal mixed with a 1kHz sine wave.
**Transmission Tests**

The following plot shows a signal passing through the input (top signal) and output (bottom signal) of the entire circuit:

![Figure 23: Transmission Test of the Entire System Using a 1kHz Sine Wave](image1)

The following plot shows a signal passing through the buffer stage:

![Figure 24: Transmission Through a Buffer](image2)
## Summarized Test Results

<table>
<thead>
<tr>
<th>Description</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mixer provides a user-defined ratio of signals</td>
<td>Yes</td>
</tr>
<tr>
<td>Distortion is not noticeable</td>
<td>No</td>
</tr>
<tr>
<td>No significant delay is present from when the drum trigger senses a drum hit to when the audio comes out of the loud speakers</td>
<td>Yes</td>
</tr>
</tbody>
</table>
VIII. Conclusion

The design specs were met to a certain degree. The project investigated cheap solutions to what is now a very pricy alternative. The reason why the current technology is so much more expensive than my project is simply that the system is only as good as its weakest component. Products on the market today use high quality components and make sure the signal isn’t getting distorted by noise or other unwanted distortion. In order to keep signals balanced, circuits must be complex, costing much more money. My product failed to return a quality signal, but it did return the signal.

The main function of my device, the DRY/WET mixer, works in outputting either the digital or the analog signal. When trying to output 50% of both signals, a drastic loss of signal was noticed. To hear the signal, the volume would have to be turned up, inducing distortion and high levels of noise to the system. Feedback was also noticed in high quantities when trying to make the signal audible. Many different problems are associated with these results.

The main source of signal disruption is most likely the potentiometer controlling how much of each signal passes through. The means of using resistors to provide the ratio of the two signals is not efficient and causes loss of signal strength. A more appropriate method for obtaining the desired results would have included
many more components to ensure the signal stays at the same level of strength throughout the circuit path.

Lead length and the components used have the inherent property of inducing inductance and capacitance to the circuit. These can cause problems including, but not limited to, unwanted filtering or unwanted noise.
IX. Bibliography

*Design with Operational Amplifiers and Analog Integrated Circuits (Third Edition)*


Arduino Real-time Audio Processing, Peter-Welter-Platz

http://interface.khm.de/index.php/lab/experiments/arduino-realtime-audio-processing/

(March 20, 2010)

Arduino USB Board (Duemilanove) (Microcontroller), Solarbotics Ltd.


MP3 trigger, © SparkFun Electronics


(April 6, 2010)

MP3 Trigger volume control, phpBB Group

Appendices

**Appendix A—Details**

**Lab Equipment Used:**
- Dual DC Power supply
- Function Generator
- Oscilloscope
- Digital Multi-meter
- Grabber Wires
- Breadboard
- Soldering Iron
- Solder

**Time Schedule:**
- Research: 60 Hours
- Design: 15 Hours
- PSpice Simulation: 10 Hours
- Lab Testing and Troubleshooting: 35 Hours
- Layout Design and Ordering: 6 Hours
- Soldering and Mounting: 4 Hours
- Audio Testing: 20 Hours
Appendix B—Alternative Solution Code

The following code enabled the microcontroller to realize when the drum trigger had passed the threshold voltage necessary to play the .mp3 audio file:

```c
const int knockSensor = 0; // the piezo is connected to analog pin 0
const int threshold = 60;  // threshold value to decide when the detected pressure is a hit or not – Decided upon experimentally

// these variables will change:
int sensorReading = 0; // variable to store the value read from the sensor pin

void setup() {
    Serial.begin(9600); // use the serial port
}

void loop() {
    // read the sensor and store it in the variable sensorReading:
    sensorReading = analogRead(knockSensor);

    // if the sensor reading is greater than the threshold:
    if (sensorReading >= threshold) {
        play(); //function made to play an audio file
    }

delay(100); // delay to avoid overloading the serial port buffer
}
```
The following is the H-File code to play the audio file:

```c
/*

MP3Trigger.h
*/

#ifndef MP3_TRIGGER_H
#define MP3_TRIGGER_H

#include "WProgram.h"

class MP3Trigger
{
    public:
        MP3Trigger();
        ~MP3Trigger();
        void Trigsetup(HardwareSerial* serial);
        void Trigsetup();
        void play();
        void stop();
        void trigger(byte track); //0–255
        void play(byte track); //0–255
        void forward(); //move ahead one track
        void reverse(); //move back one track
        void setVolume(byte level); //0–255
        void statusRequest();
```
void setLooping(bool doLoop, byte track); // turn looping on/off

void setLoopingTrack(byte track); // select the track to loop

void updateTrig(); // make sure to call this during your loop()

private:

bool mDoLoop;
byte mLoopTrack;
bool mPlaying;
void loop();
HardwareSerial* s;

};
The following is code to interface the Arduino Microcontroller with the MP3 Trigger:

```cpp
#include "MP3Trigger.h"

MP3Trigger::MP3Trigger()
{
    mDoLoop = false;
    mPlaying = false;
}

MP3Trigger::~MP3Trigger()
{
    s->flush();
    s = NULL;
}

void MP3Trigger::setup()
{
    setup(&Serial);
}

void MP3Trigger::setup(HardwareSerial* serial)
{
    s = serial;
    s->begin(38400);
}
```
//
// Looping functions
//
void MP3Trigger::setLooping(bool doLoop, byte track)
{
mDoLoop = doLoop;
    
mLoopTrack = track;
    
if(!mPlaying && mDoLoop)
{
    
    loop();
    
}
}

void MP3Trigger::setLoopingTrack(byte track)
{
    mLoopTrack = track;
}

void MP3Trigger::update()
{
    
if(s->available() )
{
    
    int data = s->read();
    
    if(char(data) == 'X' || char(data) == 'x')
    {
        
        if(mDoLoop)
{  
    loop();  
}  
else  
{  
    mPlaying = false;  
}  
}  
else if(char(data) == 'E')  
{  
    mPlaying = false;  
}  
}  
void MP3Trigger::loop()  
{  
    trigger(mLoopTrack);  
}  
void MP3Trigger::stop()  
{  
    bool wasPlaying = mPlaying;  
    mDoLoop = false;  
    mPlaying = false;  
  
    if(wasPlaying)
{  
    play();  
}

// Two-byte functions  
// Found in the MP3 trigger datasheet  

void MP3Trigger::trigger(byte track)  
{  
    s->write('t');  
    s->write(track);  
    mPlaying = true;  
}

void MP3Trigger::play(byte track)  
{  
    s->write('p');  
    s->write(track);  
    mPlaying = true;  
}

void MP3Trigger::setVolume(byte level)  
{  
    // level = level ^ B11111111;  
    //Line level volume  
    s->write('v');  
    s->write(level)  
}
Appendix C—Analysis of Senior Project Design

**Student:** David Robison

**Advisor:** Dr. Pilkington

**Summary of Functional Requirements:**

When powered on, the project must provide the ability to choose a ratio between two signals. The drum sample blends with the bass drum microphone signal so that as one signal increases, the other signal decreases at a constant ratio.

The threshold voltage for the drum trigger must be user-defined based on how intense the drummer feels is necessary for the audio file to play.

The drum trigger must not have a noticeable delay from when the drum is struck to when the sound is heard. The drum trigger must not “double-trigger” which means the system must not be too sensitive to vibration. The threshold voltage of the drum trigger could prevent this.

Distortion must not be evident in the sound that comes out of the loud speaker. Along with this, feedback and humming should not be audible.

**Primary Constraints:**

Using cables and non-ideal components definitely hinder the products abilities. The simplicity of my design is another limiting factor because careful signal propagation is not taking place. Because the MP3 Trigger is hard-wired, multiple
tracks cannot be uploaded to one drum trigger. This means that you’re stuck with the sound you picked. A computer is needed to make the .mp3 files sound good.

**Economic:**

<table>
<thead>
<tr>
<th>Before</th>
<th>After</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MP3 Trigger</strong></td>
<td><strong>MP3 Trigger</strong></td>
</tr>
<tr>
<td>100K Dual-Ganged Stereo Volume Control</td>
<td>100K Dual-Ganged Stereo Volume Control</td>
</tr>
<tr>
<td>1/4&quot; MONO FEMALE INPUT JACK</td>
<td>1/4&quot; MONO FEMALE INPUT JACK</td>
</tr>
<tr>
<td>.75&quot; Black Bakelite Knob w/ Fluted Grip and Chrome Insert</td>
<td>.75&quot; Black Bakelite Knob w/ Fluted Grip and Chrome Insert</td>
</tr>
<tr>
<td>Behringer MIC100 Tube Ultragain Microphone Preamp with Limiter</td>
<td>Behringer MIC100 Tube Ultragain Microphone Preamp with Limiter</td>
</tr>
<tr>
<td>SanDisk 1GB micro SD Card</td>
<td>SanDisk 1GB micro SD Card</td>
</tr>
<tr>
<td>Miscellaneous Parts</td>
<td>Miscellaneous Parts</td>
</tr>
<tr>
<td>Serial Enabled 16x2 Black on Green 5V</td>
<td>Serial Enabled 16x2 Black on Green 5V</td>
</tr>
<tr>
<td>Arduino Main Board</td>
<td>Arduino Main Board</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>Total</strong></td>
</tr>
</tbody>
</table>
If Manufactured on a Commercial Basis:

The number of these products, if working properly, sol in a year would be 5000. This is because the product is very specific and has limited functionality. Musicians walking into a music store might buy it for its cheap cost and unique idea.

Environmental:

Because the system only operates on two 9V batteries, a very small environmental impact will be seen. The system draws very little power, so the batteries will not have to be replaced often. The metal box the system will go in is small and compact so not a lot of metal will be used.

Manufacturability:

The product can be manufactured very easily, and a unit can probably be produced as a prototype in one day. Most of the cost will be in the design process and testing equipment used to see where the weakest node is for the signals.

Sustainability:

The system will be protected by a metal case which could be dropped a few times and still maintain workability. The SD card must be loaded and retrieved with care.

Ethical:

No ethical issues come to mind.
Health and Safety:

If too much signal bombards the system, a current over-load may damage the op amps, but the system will not catch fire. A fuse rated at a certain resistance, could be placed at the input of the system as a circuit protector. This could also prevent the system from blowing op amps.

Social and Political:

I do not think anyone will get mad that I am integrating analog and digital sounds unless it is used to create horrible music.

Development:

I became very efficient at using both the oscilloscope and the soldering iron throughout the development of the project. My problem solving skills were also put to the test, because I was finally on my own without any guidance. Everyday skills such as time management and problem solving were tested and passed. I am now very comfortable at attacking a problem, and applying the knowledge I have obtained with my time at Cal Poly.
Appendix D—Additions for Future Projects

An aspect of the project that could have been added is the pre-amp. This would have made the system much more complete, allowing the user to just plug a microphone into the project’s case without worrying about another set of wires. A built-in pre-amp boosts the signal to line-level. Future designs could have features such as input gain, output gain, dB limiter, etc. These additions would allow the user to worry about fewer components during set-up time. The following could have been a good design, providing a low-noise balanced microphone preamp ideal for portable use, operating on two 9V batteries: