Glove Drummer: A unique and portable drum simulator

Aaron Rice, Michael Moran, Timothy Cox

Dept. of Electrical Engineering and Computer Science, University of Central Florida, Orlando, Florida, 32816, U.S.A

***Abstract* — Glove Drummer gives you the power of an entire drum set in the palm of your hands. Glove Drummer is an affordable electronic drum simulator with better playing response compared with similar products. The fingertip sensors are velocity sensitive which mean the harder you tap, the louder the drum will sound. There are also pedals for controlling the hi hat and bass drum. Glove Drummer can be taken with you to practice anywhere, and you're able to attach your headphones to avoid disturbing neighbors. The system allows for other configurations such as drum fills, and the switch to a piano controller is very possible. Here we will discuss what went into our design, as well as what we took into account, and how some necessary changes helped to sculpt our system into what it is now.**

***Index Terms* — Piezoelectric sensor, non-contact sensor, operational amplifier, microcontroller, field-programmable gate array, music, MIDI, analog to digital conversion**

I. Introduction

Glove Drummer aims to provide an intuitive, user-friendly, compact, and mobile alternative to the modern electronic drum kit. The hand modules will allow the user to play Glove Drummer on any hard surface. Utilizing intuitive sensor mapping, Glove Drummer will be an easy transition for any advanced drum player, while also being approachable to new users interested in electronic drums.

One major goal of Glove Drummer is ease of use. To this end, the gloves will be comfortable and must not restrict freedom of motion. Also, the gloves are intuitive enough that experienced electronic drum users will have an easy time making the transition to Glove Drummer, but also will be simple enough to allow new users to simply put on the gloves and play. By utilizing a compact table-top module, Glove Drummer is will be portable enough to take anywhere, and can turn any space into a music studio.

Another major goal of Glove Drummer is versatility. One way this is achieved is by allowing users to load their own personal audio library for use. Although Glove Drummer is designed to be a replacement for electronic drum kits, the user can make it play any sounds they desire. This allows Glove Drummer to be used to rtiger any sound, be it a trumpet, voice, etc. The ability to quickly swap SD cards and load new audio files allows the user to switch between Glove Drummer functionalities in an instant. Also, by utilizing a USB connection and the same MIDI messages found in other electronic drum kits, Glove Drummer will be able to work with the same Digital Audio Workstations (DAWs) that experienced electronic drum users will already be familiar with. One possible mapping from glove sensors to drums sounds is shown in the figure below. This mapping will be highly reconfigurable inside the users DAW of preference.

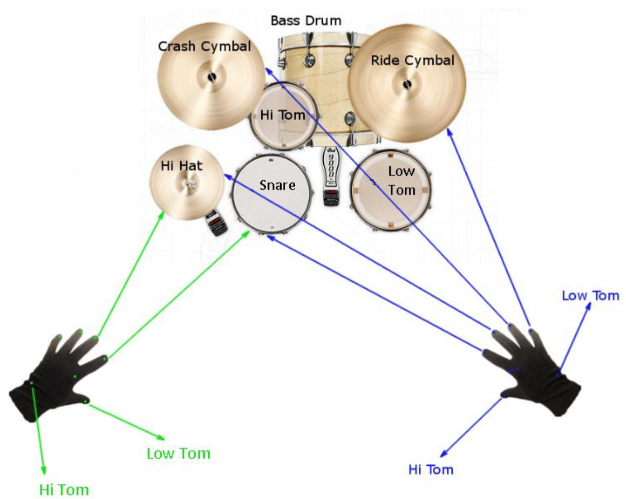


Figure 1: One possible sensor to drum mapping

A third major goal of Glove Drummer is accuracy of emulation. This means that Glove Drummer closely mimics the sounds of a standard drum kit. To this end, the sensors in the left/right hand modules are velocity sensitive, to mimic loud and soft hits on a drum kit. The bass pedal also utilizes a velocity sensitive sensor for the same reason. The hi-hat pedal will function much like a standard hi-hat pedal. Depressing the hi-hat pedal partially will cause the hi-hat sensor on the hand modules to make sounds corresponding with partially closed hi-hat. Fully depressing the hi-hat pedal closes the virtual hi-hat, and releasing it will open it. Also, for emulation accuracy, quickly opening or closing the hi-hat will produce its own unique sound. Most importantly, Glove Drummer is able to read sensor information and playback audio with low latency, specifically under at most 15ms. This is generally considered the threshold for noticeable latency but may be closer to 10ms or less for drummers and musicians. Jitter is also controlled which mean keeping the latency at a constant value under 15ms.

II. MIDI

MIDI was created in 1983 [1] and is commonly associated by some with the music from a Nintendo or other early gaming system. This could not be further from the truth. MIDI does not represent any certain type of sound, only commands to create and alter sounds. Glove Drummer will employ three different types of MIDI message, Note On, Note Off, and Continuous Control (CC). Note On and Note Off when combined create a drum sound, while CC messages give the current position of the hi hat pedal.

When an acoustic drum is struck with varying force, the sounds produced differ not only in volume, but also in timbre. This is the reason electronic drum sets store audio clips of each drum being hit with varying force or velocity. This 'velocity' term can be equated to the energy of the strike and is given in a 7-bit range from 0 to 127. In order to offset and shift the range of readings produced by Glove Drummer's different sensors, the team has used the equation shown below. Here A denotes the amplitude reading to be remapped, A' is the remapped amplitude value, and $A_{min/max}$ are the min and max amplitudes the sensor can generate respectively.

$$A' = \frac{(A - A_{min})127}{A_{max} - A_{min}} \qquad (1)$$

In the figure below, the messages needed to produce a single drum sound are shown. Two three byte messages are needed in order to turn the sound on and then turn it off. The Note Off message does not actually turn any note or sound off, but it is required to play the next sound for that "key" or particular drum. It could be said that when a MIDI synthesizer is configured to play drum sounds that the Note Off message is ignored. Note Off has an important role in creating other musical instrument sounds, but not in creating drum sounds.

msg1: note on, sensor[0], max velocity			msg2: note off, sensor[0],		
status	data1	data2	Status	data1	data2
0x99	0x00	0x7F	0x89	0x00	0x40

Figure 2: Serial messages needed to create a drum hit

III. ELECTRONIC DRUM KITS

Forty-four years ago the first electronic drum kit was created for Graeme Edge of the band Moody Blues [2]. Since that time great improvements have been made to electronic drums such as variable hi hat control, positional and cymbals, and the replacement of synthesized sounds with recordings of real sounds. Because Glove Drummer will attempt to condense a full electronic drum kit into two gloves and two pedals, the team gained much insight thorough investigating exactly how these kits function.

A. Drum Pads and Cymbals

The Drum pads and cymbals all have one or more piezoelectric disc transducers embedded somewhere within them, protected by foam or foam rubber. Some drums like the toms have only one, whereas the snare, hi hate, and other cymbals can have two for judging the position where the stick hit the drum or cymbal. The hi hat cymbal is also controlled by the hi hat pedal. Electronic hi hat pedals work in one of two ways. The more primitive way uses discrete open hi hat and closed hi hat positions. A better approach is called variable hi hat control which records the position of the pedal between open and closed so that the sounds created are more natural when compared to the sound of an acoustic hi hat. An acoustic ride cymbal can make three different distinct sounds and accordingly some of the better electronic ride cymbals include three transducers for recreating this effect. All of these features have been designed in order to make electronic drums more attractive to drummers. In the following section the team explores the MIDI messages created by Roland's flagship electronic drum set which includes all of the features mentioned here.

B. The Drum Brain

Early during Senior Design I, the team took a trip to Guitar Center, a music store in Winter Park, where the inner workings of the Roland TD-30 electronic drum set were explored. The Roland TD-30 is arguably the best electronic drum set there is with many features that help to make it respond close to the way an acoustic drum set would. The "drum brain" has input jacks for each of the sensors found throughout the drums, cymbals, and hi hat pedal. It takes the signals from those sensors and plays pre-recorded drum sounds through an audio output jack.

Surprisingly, many electronic drum players choose to record their kits using other pre-recorded sounds from computer software and not the sounds stored in the drum brain. This choice is usually made because the user feels that the sound libraries offered by computer software are of superior quality when compared to the drum brain's sound library. The team could only conclude from the preferences that the MIDI messages sent from a drum brain include all the necessary data to trigger drum sounds in a way that effectively emulates an acoustic drum set. Knowing this, the team studied the output of MIDI messages from the TD-30 drum brain in an attempt to understand exactly what data is needed by the drum software and why. One screenshot from MIDI Ox is shown and explained below. In the EVENT column, each Note On/ Note Off combo is preceded by a Control Change message. This configuration allows for the most realistic hi hat emulation as the exact position of the pedal (Data2 Column, Note On row) is given before the Note On message is received.

TIMESTAMP	IN	PORT	STATUS	DATA1	DATA2	CHAN	NOTE	EVENT
001EA1D2	1	1	B9	10	00	10	---	Control Change
001EA1D3	1	1	99	26	7F	10	D 2	Note On
001EA236	1	1	89	26	40	10	D 2	Note Off
001EA5E0	1	1	B9	10	7F	10	---	Control Change
001EA5E1	1	1	99	28	4E	10	E 2	Note On
001EA645	1	1	89	28	40	10	E 2	Note Off
001EA992	1	1	B9	10	15	10	---	Control Change
001EA993	1	1	99	26	7F	10	D 2	Note On
001EA9F6	1	1	89	26	40	10	D 2	Note Off
001EACFC	1	1	B9	10	7F	10	---	Control Change
001EACFC	1	1	99	28	4D	10	E 2	Note On
001EAD60	1	1	89	28	40	10	E 2	Note Off
001EB07D	1	1	B9	10	00	10	---	Control Change
001EB07E	1	1	99	26	7F	10	D 2	Note On
001EB0E1	1	1	89	26	40	10	D 2	Note Off
001EB39C	1	1	B9	10	7F	10	---	Control Change
001EB39D	1	1	99	28	55	10	E 2	Note On
001EB400	1	1	89	28	40	10	E 2	Note Off

Figure 4: MIDI data generated from hi hat cymbal and pedal

III. Sensors

Piezoelectric transducers made of Lead Zirconate Titanate (PZT) are ubiquitous in commercial electronic drum triggers because of PZTs high piezoelectric coefficient. For this reason piezoelectric disks comprise the majority of Glove Drummer's sensors. In addition, our team has utilized a form of non-contact sensor to sense the position of the hi hat pedal. This approach uses an IR proximity sensor in place the more traditional potentiometer found in eDrum pedals to create a longer lasting pedal-position sensing device.

A. *Piezoelectric Sensors*

Glove Drummer's 9.5mm diameter piezoelectric disk transducers create an AC voltage waveform when the piezo crystal inside is deformed. Positive and Negative pulses are produced as the piezo crystal reverberates as shown in the figure below. Additionally, the force exerted in striking the sensor against the table is proportional the peak magnitudes of those pulses. This is a very important characteristic for the sensor to have since Glove Drummer will play back different drum sounds for a certain drum hit with varying force. Surprisingly, the peak voltages were lower than expected based on electronic drum projects using larger piezo sensors. The team reasoned that since the size of the piezo crystal used here is smaller than in those projects, the resulting voltage waveform would also be smaller.

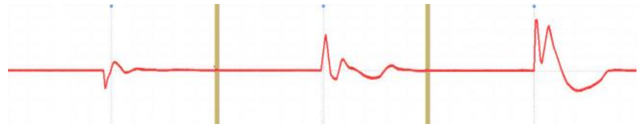


Figure 5: Signal generated from soft, medium, and hard strike

In addition to evaluating the magnitudes of the waveforms generated during a strike of the sensor, the length of the response was also noted. In the figure below, a waveform resulting from striking the index fingertip sensor repeatedly against the table at the fastest speed attainable by the members of the team is shown.

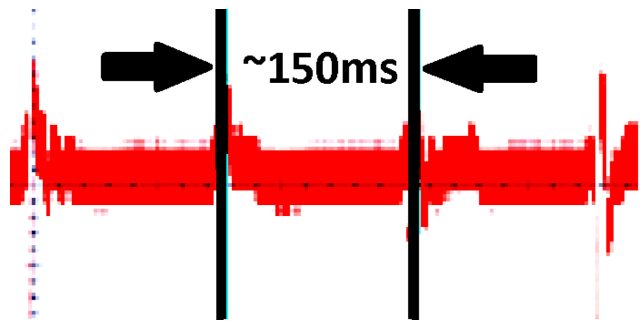


Figure 6: Verification of piezo's transient response and decay

Each individual strike of the sensor creates a pulse that dies out long before the next strike of the sensor. This is a very important characteristic for the sensor to have since users will be tapping their fingers very quickly at times. The ~150ms time period in between strikes correlates to about a 400 beat per minute (bpm) tempo for a single finger tap. Therefore with two fingers tapping alternately, a "drum roll" of approximately 800 bpm in tempo should be achievable. Also, noting the 150ms period between strikes will be useful in "debouncing" the sensor. Some strikes produce unwanted spikes in voltage long after the initial pulse. This would result in a "double trigger" where the drum sound plays more than once resulting in a machine gun type sound. This double trigger effect could be avoided using a counter that counts to 150ms after a valid strike before it allows another valid strike message to be sent. These preliminary tests proved to the team that while this exact sensor design has never been used in any 'glove controllers', it could be very effective.

B. *The Piezo Conditioning Circuit*

Converting the raw piezo signal to a digital reading meant conditioning the signal to meet the input specifications of an Analog to Digital Converter (ADC). The signal needed to be at least half wave rectified using a diode, clamped between –Vref=0V and +Vref=5VDC of the ADC, and finally amplified due to the high output impedance of the piezo. Many different op amp choices were considered. The goal was to choose a rail to rail device with the ability to be DC biased from a single 3.3V power rail. A comparison of some of the different op amps available to the team is shown in the figure below.

Part #	Manufacturer	Technology	5V Single Supply ?	Rail to Rail ?
LM324	TI	BJT	N	N
TL072	TI	JFET, BJT	N	Y
AD822	Analog Devices	JFET, BJT	Y	Y
OPA350	TI	CMOS	Y	Y

Figure 7: Op Amp comparison for conditioning circuit

The team's final decision was to use the OPA350 op amp from TI. With CMOS technology, the conditioning circuitry will consume less power. The added cost of using CMOS transistors was of no concern since the team has sampled the part at for free from TI.

The final circuit for interfacing the piezos to the ADC is shown in the schematic figure below . This circuit was modeled after several circuits that have been built for the purpose of building an electronic drum kit. One in particular electronic drum project which very well documented was found at edrum.info [3]. This circuit not only amplifies the signal, but also filters out the components of the signal that are far from the piezos resonant frequency. Also shown below are screenshots of the fully conditioned signal.

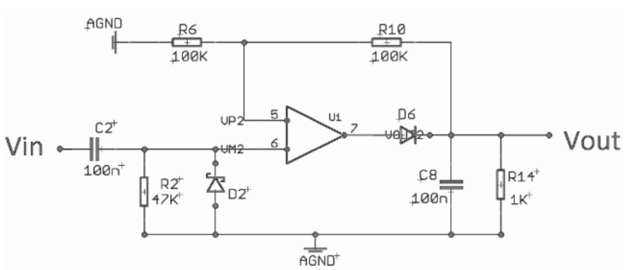


Figure 8: The piezo conditioning circuit

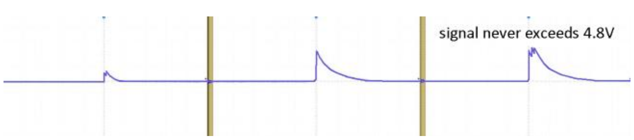


Figure 9 : The conditioned piezo signal

C. The Polulu IR Proximity Sensor

The purpose of using this IR proximity sensor in Glove Drummer's design is to allow for different 'playing modes'. Different playing modes are needed because drummers for the most part serve as the clock for other musicians. This task can be monotonous and needs to be broken up every so often with something a little more flashy, like a 'drum fill'. Therefore you might say there are two playing modes for a drummer, a 'clock mode' and a 'dynamic mode.' Tying each of the finger sensors to a single sound may be intuitive to only one of these two modes. If there were a way to switch very quickly from a 'clock mode' mapping, to a 'dynamic mode' mapping, the Glove Drummer system would be much more exciting to play.

Initial testing of the Polulu sensor showed that while it is an analog sensor, it basically gives two discrete voltages. When far from any object there is one voltage reading, and when within 30cm of an object it gives a much higher reading. No voltages are produced between these two low and high values making it a bad choice for a variable hi hat controller pedal. However this sensor was helpful in creating two different 'playing modes'. Placing these sensors one each wrist facing down allows Glove Drummer to sense if the wrist is hanging off the edge of the table or hovering over the table. As long as the fingers are within striking distance of the table, the IR sensor will register one of two statuses, either "table" or "no table." The sensor to drum sound mapping is then changed based on the status of the Polulu sensor, creating the two drum modes defined here. An example implementation of this idea conceived by the Glove Drummer team is illustrated in the two figures below. The mappings for each playing mode will need to be tested further once the prototype is working. Shown below is just one possible set of mappings.

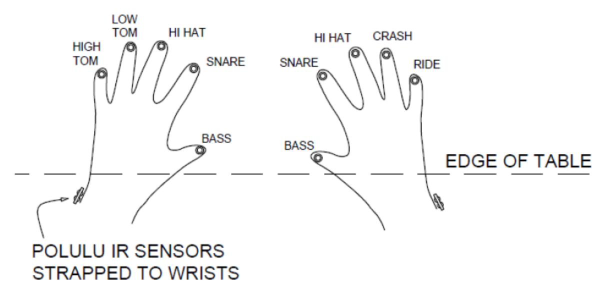


Figure 10: 'Clock mode' sensor to drum sound mapping

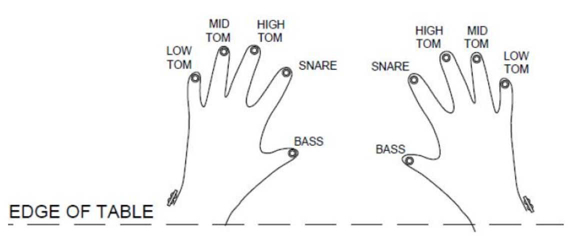


Figure 11: 'Dynamic mode' sensor to drum sound mapping

D. The TCRT5000 IR Sensor

The TCRT5000 IR proximity sensor is more basic and less plug and play when compared to the Polulu IR sensor. It requires the use of additional components like current limiting and pull up resistors and does not use amplitude modulation. Current limiting resistor values were calculated based on the supply voltage of 3.3V, Ohm's law, and the maximum continuous current the IR LED could withstand. The IR sensing LED feeds into a BJT amplifier inside the device. Here again, resistor values were calculated to obtain an acceptable sensitivity. As expected it cannot sense objects from as far as the Polulu sensor, but has other important characteristics. One of these characteristics is the ability to give a continuous range of voltages versus just a low value and a high value. This makes the TCRT5000 more suitable for use in the hi hat pedal. Once fastened to the bottom of the hi hat pedal housing, it can resolve a range of voltages between the open and closed pedal positions. This non-contact design is superior to the more traditional potentiometer-based pedals that fail after excessive usage. The physical contact of the potentiometer's wiper arm is the source of that failure. This hi hat pedal design is unique to Glove Drummer and altogether costs about $10 to make.

IV. Wireless Communication

Of course no user would enjoy playing the drums with wires attached to their hands. An acceptable wireless system was needed to send messages from the hand modules to the tabletop module. The nRF24L01 radio frequency transceiver was chosen for this purpose. It uses the microcontroller's SPI port to in order to receive commands and send data back to the MCU. This is a relatively new device, but has a wealth of open source support. Also the cost is less than $2 per unit. Other wireless and Bluetooth solutions were simply too costly, and some too slow, making the nRF24L01 the only viable option for Glove Drummer

V. Processing Hardware

Glove Drummer uses Field Programmable Gate Arrays (FPGAs), Microcontrollers (MCUs), and one single board computer to achieve a responsive, near zero latency drum playing experience. The selection of each of these items was based on both cost and performance. Keeping latency and jitter to an absolute minimum was of the utmost importance if real musicians are to be using Glove Drummer.

A. Spartan FPGA

MIDI Controllers are most often built with MCUs—however, for rapid development of a musical system to be used by the trained ear of a drummer, the use of FPGAs was a great advantage to the Glove Drummer team. The FPGA board used does not have many features found on other FPGA boards like displays, buttons, switches, various ports as they are not required in the hand modules. It features the Xilinx Spartan XC3S500E FPGA chip which has 500K equivalent logic gates and more pins than Glove Drummer could ever use. The cost of this board was only $35, making it the cheapest FPGA development board available on the market.

The FPGA dev board is connected through female headers to the hand module PCB. The PCB also holds two ADC0820 ICs. These two analog to digital converters (ADCs) are fed from a DG509 dual 4:1 multiplexer (MUX). The DG509 is used to time division multiplex six piezo sensor signals, the battery voltage, as well as the Polulu IR sensor signal. As the ADCs and MUX operate at Vcc=5V, logic level shifting is required when interfacing to the 3.3V FPGA. Accordingly three level shifting ICs are also found the hand module PCB.

B. ATmega88 MCU

The ATmega88 was chosen to relay messages from the FPGA to the nRF24L01 wireless transceiver as there have been no drivers ever written for the nRF in Verilog. The Glove Drummer team simply did not want to be the first to try with so much on the line to create a working prototype. Instead, an open source library for nRF and Atmel MCUs was referenced from the web (source). The ATmega88 IC is found in its QFP form, soldered to the hand module PCB. In order to program the ATmega88, a special QFP programming adapter cable was used.

C. Tiva C Launchpad

Within the tabletop module, the Tiva C Launchpad was used to assemble all of Glove Drummer's sensor data into MIDI messages. This MCU was chosen for its abundant peripherals and low cost of $12. It receives the left and right hand module data wirelessly and combines it with bass drum and hi hat pedal data in order to create MIDI Note On/ Off and CC messages. This MIDI information is then decoded by a plug and play MIDI to USB converter. The USB output can then be connected to any computer or to the Beaglebone Black for the standalone operation of Glove Drummer.

D. BeagleBone Black Single Board Computer

Early in Senior Design I the Glove Drummer team set out to learn about SD cards and cd-quality digital to analog converters (DACs) to interface with a microcontroller. One development board in particular had both of these features but cost over $200, eliminating it from the list of affordable options. Very few examples using MCUs to generate cd-quality audio can be found on the web. Most of these examples simply use pulse width modulation (PWM) to generate sound at a lesser quality than 16bit 44.1kHz. Part of the reduction in quality stemmed from the maximum speed attainable in reading from the SD card with the open source FAT library. This library would need to be exploited for speed if the Glove Drummer team was to achieve the goal of cd-quality audio playback. Another downside to using an MCU to play back stored audio is the sheer amount of C code that would need to be written.

Using the BeagleBone Black allowed the team to abstract the inner workings of the DAC, SD card, and FAT file system in order to guarantee both cd-quality audio and near zero latency. Another reason for making this choice was found in benchmarking the ‘Wave Trigger’ by Robert Sonics. This device simply takes a MIDI input and plays back polyphonic cd quality audio, exactly what is needed by Glove Drummer. Mr Sonics has been developing this device for over two years and slowly increasing its performance to the point where it could be used with near zero latency. After viewing the processing power of the Wave Trigger and knowing the amount of time it took to develop, the Glove Drummer team decided on using the Beagelbone Black for polyphonic audio playback. Another noteworthy feature of the Beaglebone Black is the possibility of audio synthesis, where Glove Drummer can one day be expanded into another type of instrument entirely!

VI. Software

A. Verilog

Verilog coding began with an open source UART module obtained from open-cores.org [4]. Modules were later written by the Glove Drummer team to control the ADC and MUX, find transient peaks in the piezo signal, and to transmit multiple bytes through the UART. The state machine which uses the UART module to send those bytes is shown below. With each successful transmission of a one byte, a single cycle clock pulse in generated by combinational logic in order to advance to the next state. One cycle of this state machine produces the Note On and Note Off messages needed to create the sound of striking one drum.

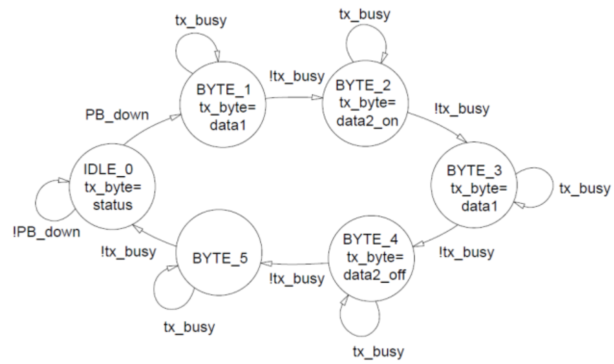


Figure 12: State machine that sends a five byte serial message

Other Verilog modules were also written to debounce a pushbutton mounted to the index finger, to disable the wrist-mount IR sensor with the pushbutton, and to flash an LED when the battery is close to depleted. Also, because the prototype includes five extra digital input headers, an additional module may be written to give Glove Drummer the ability to emulate a piano. This addition could most likely be achieved using a piezoresistive material spanning each fingertip. By applying conductive fabric electrodes to the piezoresistor and placing it in a voltage divider circuit, a thin digital switch can be constructed. If done correctly the signal will exceed the logical ‘1’ voltage when the fingertip presses down and fall back to ‘0’ when the finger is lifted. In this example the Note On message corresponds to the positive edge and Note Off to the negative edge of the piezoresistor signal. Here Note off is not ignored as it is in MIDI drum controllers. Also, there is a possibility that velocity information would still be obtainable through the piezo sensors.

B. C Language Code

A good portion of coding went into altering an existing nRF24L01 library for Atmel processors to fit Glove Drummer’s needs [5]. From there, the library was altered again to work with the Tiva C. Both codes use a hardware SPI port to read from and write to the nRF’s registers. This is how setting the unit’s address, picking an RF channel, receiving data, and transmitting data are accomplished.

Additionally, tedious coding went into combining the Tiva C’s incoming data traffic in a timely manner. Readings from the hi hat pedal were taken by ADC0 every 4ms using a timer interrupt. Each sample was stored in a 25 element long array and compared to past values in order to obtain the rate of change of the signal. Whenever the pedal moves from open to closed (or vice-versa) in less than 100ms, sounds are created without hitting a piezo sensor. The figure below illustrates the hi hat pedal closing as fast as possible, or within ~25ms. This event will create a 'foot chick' sound at the maximum velocity of 127, similar to the two hi hat cymbals of an acoustic drum set being pulled together rapidly. Closing the pedal in greater than 25ms will create a foot chick message with a lesser velocity. If the pedal takes longer than 100ms to close, then no foot chick message will be sent. Conversely, opening the pedal quickly creates what is known as a foot 'splash' sound, again just as an acoustic hi hat would.

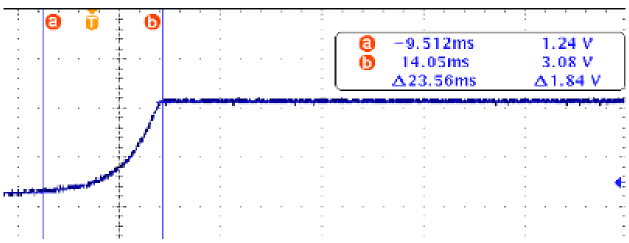


Figure 13: Closing the hi hat hat pedal as fast as possible

Additional C code for the Tiva accomplished finding the transient peaks in the piezo signal. This was also done using timer interrupts for efficiency. Whenever a peak is found in the signal, the peak finding algorithm is disabled and a timer begins to count down from 10ms. When the timer reaches zero, the peak finder is enabled again. This process allows only one sound to be produced per strike of a sensor which is critical to Glove Drummer's functionality.

In the Tiva C code main function a circular FIFO buffer is parsed whenever new data enters it so that information can be assembled into MIDI messages. This process is continually happening in between interrupts from the pedals and hand modules. Each hand module corresponds to its own nRF unit and SPI port the Tiva C. The nRF includes a pin to send interrupt requests so that the when a new message is received, the data can be accessed immediately.

C. Pure Data 'Patches'

Pure Data is an object oriented programming language similar to C++, only it is a *visual* object oriented language. Programs are called patches and are made up of object blocks connected by edges from their respective inlets and outlets. A simple example 'patch' is shown in the figure below. This patch takes an incoming MIDI signal and plays back a sound which is hard coded into the 'open' block whenever the Note On message has a 'key' of one. Here 'key' means the particular sensor which has been struck. One important thing to note is that dac~ object at the bottom can also mix many stereo audio signals together digitally with little effort.

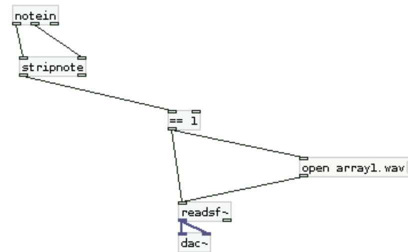


Figure 13: An example Pure Data patch

Portions of patches can be hidden inside of 'subpatch objects' to enhance the readability of the program, similar to functions in the C language. This feature was useful in managing the large number of short audio clips Glove Drummer uses. There are at least ten types of drum sounds our team wishes to store and each sound should have on average eight separate audio clips of different velocity hits. That means at least 80 separate short audio clips need to accessible to the program for realistic drum simulation. The more sounds stored, the more realistic the playing experience will be. Professional music software stores hundreds of different samples for a full drum kit. Each drum or cymbal may have up to 128 different velocity audio clips associated with it. Using Pure Data made it possible for Glove Drummer to achieve something similar and also opens the door for expansion into audio synthesis and other interesting musical topics.

VII. POWER SYSTEMS

A. Tabletop Module Power System

The tabletop module is powered from a 12V, 5A AC to DC supply which connects through a barrel jack. This 12V directly powers the speaker amp within the tabletop unit and then feeds separate 5V and 3.3V linear voltage regulators which power the Beaglebone, USB Hub, and Tiva C MCU.

B. Hand Module Power System

The hand modules are powered from what is known as an 18650 lithium ion battery. These batteries, like other single cell li-ions, have nominal voltage of 3.6V. The actual voltage varies between 4.2V and 3.0V, from fully charged to fully drained. Their current capacity is 2800mAh and they also have built-in over charge, over discharge, and thermal protection. To create the 5V and

3.3V power rails required for the hand modules, TI's TPS61032 was used to boost the varying battery voltage to a constant 5VDC. This IC is a switching boost converter that has been designed to work with 18650 batteries as well as other battery chemistries. The TPS61032 can source 2A, which leaves enough overhead for a low dropout (LDO) linear voltage regulator to supply the roughly 30mA required by the 3.3V nRF24L01 wireless unit.

VIII. Standards and Regulations

The FCC in the US dictates the wavelengths of electromagnetic energy an electronic can emit. The nrF24L01 wireless unit uses the 2.4GHz Industrial, Scientific, and Medical (ISM) band of the spectrum. The FCC allows devices to use the band from 2.4GHz to 2.4835GHz and the nRF has 128 1MHz wide channels. This means that the use of channels hat the use of channels 84 through 127 is illegal [6]. Being that the 'n' in nRF stands for Nordic, the company may not be too concerned with the FCC regulations.

Of course using PZT sensors is concern since they contain lead. Research is currently being done to replace PZT and other piezoelectrics containing lead. PZT's exceptionally high piezoelectric coefficient makes it the material of choice not only for drum triggers, but also electronic buzzers. Although the actual PZT wafer is surrounded by glue and sandwiched between brass disks, eliminating lead from any design is always a good idea. Each IC on the Glove Drummer PCB was chosen in a lead-free package.

IX. Conclusion

The Glove Drummer team set out to build a unique electronic instrument that might be fun and intuitive for musicians to play. The task required pulling from past knowledge gained through coursework as well as learning a significant amount of new material. Hours and hours were spent pouring through datasheets to first select and then completely understand the operation of each IC, hundreds of lines of code were written, and many circuits were breadboarded then soldered. In the end, some music was made and the team was left with a musical electronic device with the capability to altered and improved in many ways.

Acknowledgement

The authors would like to thank Guitar Center of Winter Park and its manager Ray, who were a great help in allowing the team to record data from the Roland TD-30. They would also like to thank their Senior Design Review Committee and Dr. Richie for taking the time to participate in the Glove Drummer final demonstration.

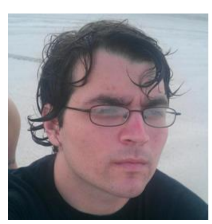
References

[1] Retrieved October 24th, 2014, from http://www.midi.org/aboutmidi/tut_history.php
[2] Retrieved November 2nd, 2014, from http://www.drummagazine.com/features/print/200-greatest-drumming-moments-events
[3] Retrieved October 10th, 2014, from edrum.info
[4] Retrieved October 17th, 2014 from open-cores.org
[5] Retrieved October 23rd, 2014, from https://github.com/maniacbug/RF24
[6] http://www.diyembedded.com/tutorials/nrf24l01_0/nrf24l01_tutorial_0.pdf

Meet the Engineers

Aaron Rice is a senior Electrical Engineering student at UCF, graduating with his Bachelor's in May of 2015. He enjoys playing music and has always wanted to own a drum set but decided to create something new instead. He enjoys working with his hands as well as designing and building everything from surfboards to electronics. He hopes to pursue a career somewhere in the electronics field after graduation

Timothy Cox is completing his Bachelor's of Science in Computer Engineering in May of 2015. He enjoyed the experience in both Computer Science and Electrical Engineering and his technical electives reflect this. He has been working as a College Tech at Lockheed Martin MST for the past year getting valuable experience, and he wants to pursue a career in flight simulation right out of college.



Michael Moran is completing his Bachelor's of Science in Electrical Engineering in May of 2015. He had interest in both Electrical Engineering and Computer Science/Programming, which is reflected in the technical electives he chose. He has been working as an Electrical Designer at Joseph, Lawrence, and Co. (JLC), a small MEP construction engineering firm, for the past 2 years. After college he intends to pursue his P.E. license.