Timothy Cox
Aaron Rice
Mike Moran

Glove Drummer

Senior Design Project

September 4, 2014

Introduction

With Glove Drummer, the power of music is in the palm of your hand. Play the drums anywhere with Glove Drummer and a desk or table.

We would like to thank Ray K. and Guitar Center for allowing us to research modern electronic drumsets at their Winter Park location.

Group #15 Members: Aaron Rice, Mike Moran, Tim Cox

Project Description

Electronic drum sets have been around for 40 years or more and have evolved to closely mimic the nuances of acoustic drums. We believe there is a way to intuitively map an electronic drum set to a pair of gloves. By intuitively, we mean that the placement of the sensors would make sense to most drummers, allowing them to easily learn to use Glove Drummer. This will allow users to feel like they are playing a real drum set inside an office cubicle or other small space. Also, transporting the "glove drumset" is not an issue when you compare the size of an acoustic or electronic drum set to the size of a pair of gloves and two pedals.

Glove Drummer will feature velocity sensitive piezo sensors sewn into the fingertips giving you the ability to play a full drum kit on any hard surface. The lightweight glove material and tiny sensors will allow the hand to move freely while playing. Hardware design will support low latency, responsive playing of drums. Left and right hand glove modules along with one two-pedal audio playback module comprise the entire Glove Drummer system. The audio playback module is added for hi hat control, bass drum hits, and audio playback with a line-level ⅛" audio output and standard MIDI output. Simply connect the gloves the audio playback module, plug it in , and start jamming! Alternatively, you can connect audio playback modules MIDI out to your computer MIDI interface, start your Digital Audio Workstation and route the sensors to the sounds of your choice for more flexibility.

Requirements and Specifications

What we are designing is known as a “MIDI Controller.” The very first midi controllers were music keyboards that made no sound, instead they controlled another module that created the sounds. Today there are many MIDI controllers on the market that can emulate almost any musical instrument or alternatively a mixing desk of any size. Digital Audio Workstations (DAWs) on all operating systems have become powerful enough to use many MIDI Controllers at once with very little latency and even feature templates for mapping of the software controls to the hardware knobs, drums, sliders, etc.

MIDI is simply a protocol for playing musical notes, or in this case, short audio clips of drum sounds. A MIDI controller serves the purpose of relaying information about how a certain musical instrument was played to a MIDI sequencer (usually PC software) which will determine what is actually done with that information. MIDI messages must be sent as quickly as possible to avoid any noticeable latency. Of course any EE student knows there is always some latency when using a microcontroller. Our team will evaluate the threshold for “too much” latency via oscilloscope. The reason for doing this is to obtain a more objective viewpoint of the threshold of noticeable latency than what is given by computer musicians, hardware/ software companies, etc. Jitter is also an issue that should be avoided where different actions will have different latencies, which can even be worse than having a uniform latency above the noticeable threshold divided equally among the sensors.

In addition to being just a MIDI Controller, our design will allow for audio playback of pre-selected drum samples. An SD card interfaced to the mcu or FPGA will be loaded with .wav audio files for this purpose. The digital samples must be “mixed” if more than one sound plays at a time so that the volume of the audio playback is remains constant. The mixed samples will be converted by a digital to analog converter (DAC) at a rate of 44.1 kHz. Most likely, the signal would then play out through a small speaker. If we have the time though, we would like to implement an audio preamplifier circuit. This preamp would bring the signal to audio line-level where it can be fed through a ⅛ inch jack into a powered speaker.

Commented [1]: I added this paragraph. From what I have seen, a small speaker attached to the audio playback module may be easiest, but the speaker will have no bass. The preamp and separate powered speaker (my keyboard amp) would sound WAY better, but take more time and effort.

Timothy Cox
Aaron Rice
Mike Moran

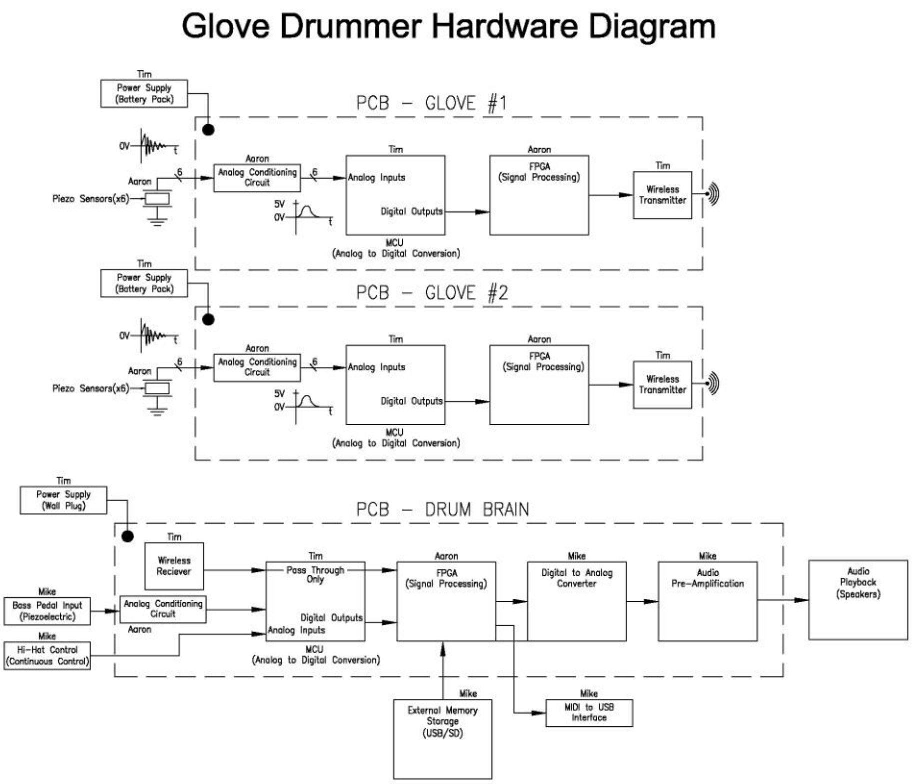
Piezoelectric transducers with a diameter of 9.5mm will be sewn into the fingertips and possibly elsewhere on the glove in order to send MIDI “note on” and “note off” messages in rapid succession to the music software. In order to keep the analog signal cable runs as short as possible, the mcu’s will be strapped to the wrists. The piezo sensor is not inherently compatible with an ADC, therefore some analog signal conditioning must be done before interfacing to the mcu. An active bandpass filter using an op amp will be used to amplify the high impedance piezo signal and filter out the unwanted frequencies. In addition, this circuit will contain diodes forming clipper and half wave rectifier circuits to keep the analog voltage between 0V and 5V.

In our C program written for the mcu, the ADC will continuously poll each input channel and wait for a signal to break a preset threshold. This means that a sensor may have been “struck”. Once that threshold is exceeded, the mcu will begin trying to decide if it should actually send a MIDI note on message. Before it sends a message, the mcu stores subsequent ADC readings for the sensor that broke the threshold, until it recognizes a peak in the signal. This peak value is known as velocity and is sent along with the note on message to convey how loud the drum sample should be played back through the speaker. Still before the message is sent, the time since the last valid note on message was sent on that channel must be assessed in order to prevent “double triggers”. Whether we use this method or not, we will try to find the most efficient algorithm for detecting the peak of the strike signal to send with the note on message.

As with any acoustic drum set, Glove Drummer will include bass drum and hi-hat pedals which are wired to the tabletop module. The bass drum pedal will send standard note on/off messages with velocity like the piezo transducers in the gloves while the hi-hat choke pedal can send messages in a few different ways. The most basic example would be that if the hi hat pedal was closed and finger corresponding to the hi-hat was struck against a hard surface at the same time, a note on/off message would be sent to play the closed hi-hat audio sample. Conversely, if the pedal was open (not pressed) then an open hi hat sound would play through the speaker. This method uses the pedal as a digital switch and is not optimal. If the pedal is fitted with a potentiometer or better yet some sort of contactless photo sensing component, an analog signal can then be read giving values in between open and closed. The mcu must send out what is known as a Continuous Control message in this case, where the music software determine which hi hat sound will play according to the pedal’s position at that moment.

We would like to incorporate a “continuous hi hat control,” possibly an FPGA, wireless gloves, and possibly other features to our design. This will of course be dependent on many factors like the time to implement the analog conditioning circuit, the time to write and debug our mcu program to reduce latency, and the fact that probably less than ten MIDI glove controllers have ever been created-- none of them ever using the exact sensor we will be using. There will most likely be some factors we have overlooked at this point as well. By the end of Senior Design II, we are proposing that we will create a velocity sensitive, low latency electronic drum MIDI controller with audio playback composed of two gloves and two pedals.

Block Diagram



Commented [2]: Looks really nice Mike. You might want to make the drum brain box bigger and put the sd card on the pcb and add an mcu or adc between the analog inputs and the fpga if its not too much trouble. And like we said the assignments will be arbitrary so no worries on what the diagram says about who's doing what.

Commented [3]: Aaron: FPGA, piezos, analog signal conditioning

Mike: SD card, DAC, preamp

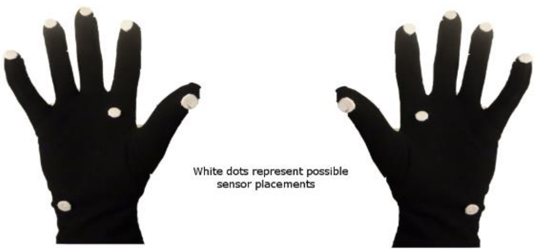
Tim: wireless, mcu's for c code, (since Tim's computer engineering), power supply and batteries

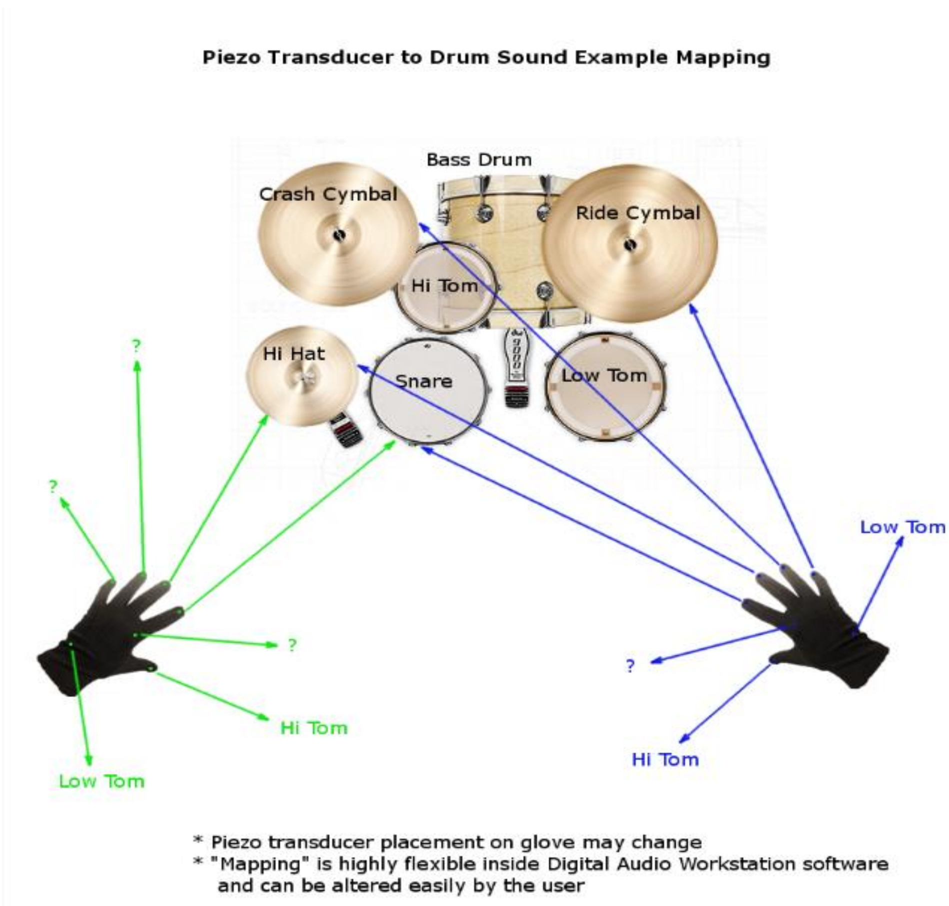
But like we said we will all kinda work on everything at least for a while then once we know what we're doin maybe we can break things up

Commented [4]: Looks good! Ya we will definitely all work together, Im trying to get an idea what the code is gonna be by looking at past projects, Im sure you guys are way better at the assembly language part of it all. Im interested to see how this is all gonna come together!

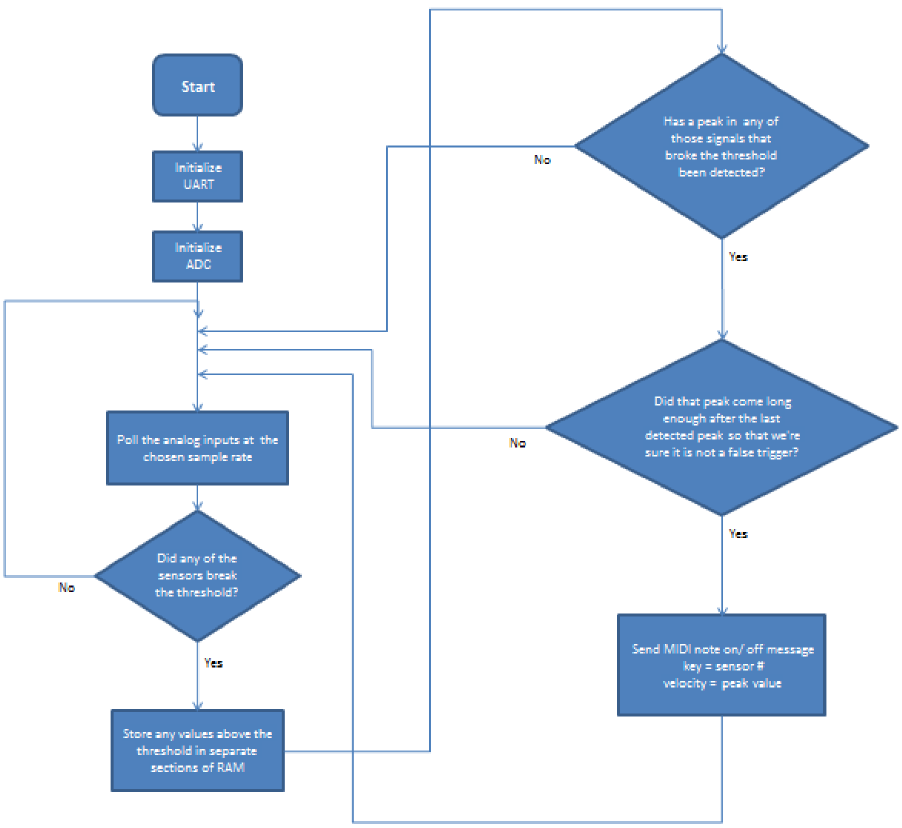
Commented [5]: I say we just say no to assembly! C programming only

Timothy Cox
Aaron Rice
Mike Moran





Timothy Cox
Aaron Rice
Mike Moran



Flowchart for C Program shown above without hi hat control.

Financing and Budget

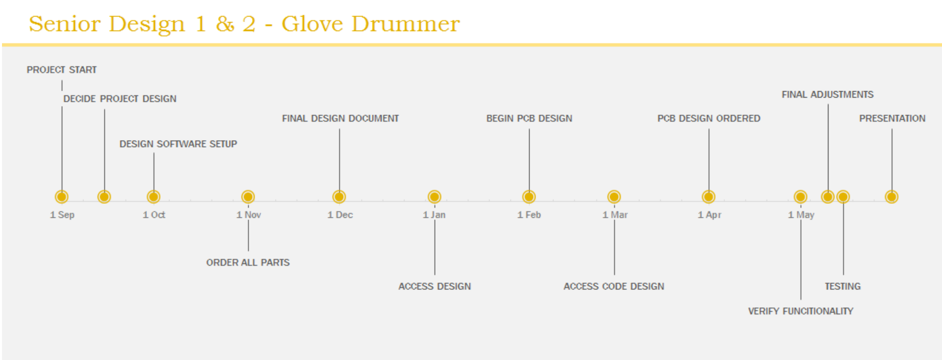
Timothy Cox
Aaron Rice
Mike Moran

Aaron has purchased most of the items we will require in order to hopefully implement a single glove and a single pedal. He conceived the idea for Glove Drummer and wanted to get acquainted with the type of components used in some MIDI projects found online. Throughout the course of Senior Design I and II, there will be more parts/ components that need to be purchased in order to realize the final design.

Our budget will be roughly $150 per team member, as financial circumstances will allow, in addition to Aaron's initial purchases. $150 per team member is based on the cost of what we each might have spent on textboook for the course and is not set in stone. With three team members, our total budget will be about $450 to purchase everything we decide we will need to make a low latency, responsive glove drum system with audio playback. Major purchases will include PCB order from Osh Park, additional mcus/ FPGAs, and a large number of op amps for piezo conditioning.

Implementation Plan

Commented [6]: I corrected the timeline as well



Senior Design I:

- Order all the parts we may need by midpoint of semester
- Start with building analog signal conditioning circuit. Test the circuit with piezo sensors sewn into glove. Use oscilloscope for testing.
- Write C code for one piezo sensor and verify its functionality by connecting to music software.
- Expand C program to N # of piezo sensors, determine appropriate sample rate for that number of sensors. Again, verify functionality with music software.
- Contact Guitar Center, hopefully do some research on the electronic drums (mainly continuous hi hat control) before end of semester

- Begin determining if any of the extra features found in "Things to Consider" are feasible.

Senior Design II:

- Determine feasibility of any extra features we have thought up through experimenting and decide which, if any, will be incorporated into the final design.
- Develop an almost complete understanding of what the final product will be so that we can start designing the PCB
- Continue to refine our design before ordering the PCB
- Have PCB design completed and ordered the PCB by midpoint of the semester
- Solder components to PCB and verify functionality

Things to Consider

- Will our sensors work? (they should, but let's find this out first)
- Whether or not to use continuous hi hat control or digital switch
- Whether or not to try infrared idea with white and black surfaces for drum fills
- Whether or not to use FPGA in conjunction with the mcu for improved response - Aaron is taking FPGA Design this semester
- How to design our circuit board
- Whether or not to incorporate something where making a fist quickly and releasing will switch the functionality of the glove drum (could have single click, double click like a mouse) Also, holding a fist could deactivate the sensors so that you could hold a real set of drum sticks. Double click of the fist could also deactivate the sensors so you could play a keyboard until another double click of the fist turns the glove sensors back on.
- Whether or not to try bluetooth or any other wireless technology
- If we are getting any sort of EM interference, we will try wrapping the sensors in tin foil and grounding the tin foil. If this works to shield the EMI, then we can make two gloves of conductive fabric and sandwich the wires and sensors between the two gloves.
- The cost of making Glove Drummer will be much less than if we tried to create an Edrum set. We also think it will lead to a more portable design than regular Edrums.