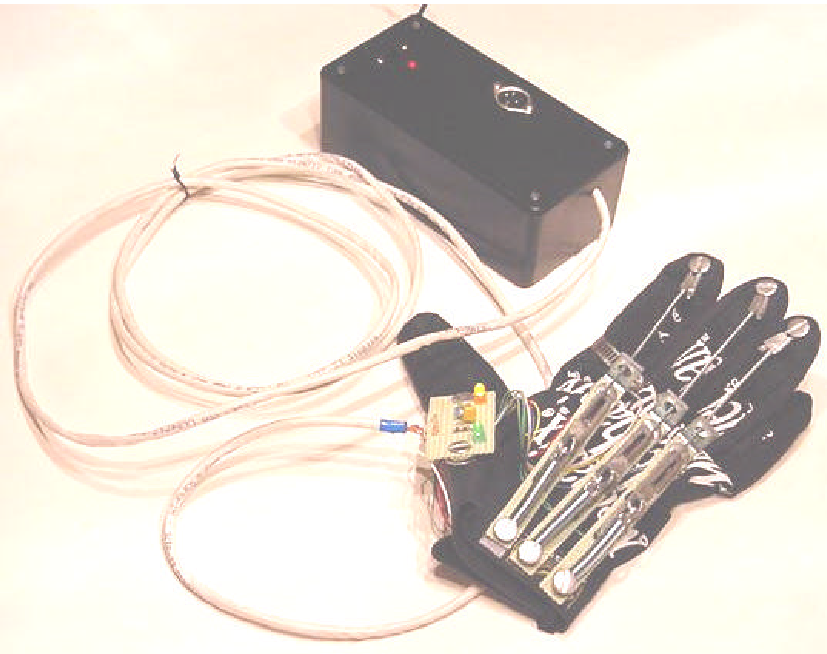
The MIDI Controller Glove
Matt Winkler

December 18, 2003

Physics 344: Digital Electronics
Independent Project
Professor Steven Errede



The Idea:

Electronics, no matter how simple or how complex, are only truly useful to human beings if there is an efficient way in which we can interface with them. This applies to both the digital and analog realms of electronics. This project is really an investigation into a practical interface between the human body and digital electronics; or more specifically, between the human hand and a MIDI synthesizer.

A Brief History of MIDI:

Some early commercial analog synthesizers utilized an analog control interface to send pitch information and note on/off events and to provide a timing reference in-between synthesizers. These interfaces, however, were not standardized and varied from manufacturer to manufacturer. (For example, different manufacturers used different voltage to pitch schemes and used different clock speeds for the timing references.) Also, these early interfaces only allowed for one note at a time to be controlled. Musicians using synthesizers from different manufacturers had to learn the differences between the various interfaces and had to adapt their work to fit within the constraints of each system. A definite need had arisen for a standardized control system that allowed for interconnectivity.

By the early 1980’s, commercial synthesizers had become extremely complex. Microprocessors and various other digital elements were finding their way into commercial synthesizers because digital components were becoming smaller and more reliable than analog components and the production cost of integrated circuits was at an all time low. This “digital revolution” in synthesizer design also led to a “digital revolution” of synthesizer control.

This digital revolution was MIDI 1.0. MIDI 1.0 was the brainchild of audio engineers from Sequential Circuits, Roland Corporation, and Oberheim Electronics. This groundbreaking system of standards was published in August of 1983. The MIDI (Musical Instrument Digital Interface) standard specifies connections at both the hardware and communications protocol levels. (All physical MIDI connections use the same cable and all MIDI devices use the same language to communicate.) These two simple points illustrate the true power of the MIDI protocol.

MIDI has been improved somewhat since 1983. The newer improvements include control over musical features like pitch bending, slurring, tremolo, and reverb. (Any new changes to the protocol must be passed by the MIDI Manufacturers’ Association (MMA) and the Japan MIDI Standards Committee.) The actual core of the protocol, however, has not changed a bit. MIDI synthesizers from 1985 still respond exactly as they should to MIDI messages sent from today’s latest-and-greatest sequencers and controllers.

An Overview of MIDI Messages:

MIDI messages are digital signals (in binary) sent between MIDI components for a number of reasons. There are three types of MIDI messages: channel voice messages, channel control change messages, and system messages. Each of these messages is comprised of a status byte followed by either one or two data bytes.

When a key is pressed on a MIDI keyboard, the keyboard outputs a "Note On" message. A "Note On" message is a channel voice message and in binary, looks like:

1001nnnn 0kkkkkkk 0vvvvvvv

where nnnn is the MIDI channel on which the message is being sent (MIDI channel 1-16 = 0-15 → binary), kkkkkkk is the note number (0-127 → binary), and vvvvvvv is the velocity, or volume, of the note (0-127 → binary). When the key is released, a "Note Off" message is sent. This message is the same as the "Note On" message with the exception of the first four bits being 1000 instead of 1001. "Note On" and "Note Off" are just two of the many channel voice messages.

Some MIDI components have built in sequencers that allow MIDI to be recorded in a song format. The bulk of system messages handle the control of these sequencers. They can, among other things, start, stop, and select these songs. In addition, system messages can control functions of a MIDI component like a system reset.

My project deals with channel control messages. These messages augment the values of any of the possible 128 different MIDI controllers. These controllers include pan, portamento, modulation, and channel volume, among many others. Some of these controllers are left undefined by the MIDI Manufacturer's Association and can be defined by a user depending upon the sophistication of his or her MIDI equipment. Control change messages look very much like channel voice messages. Below is a simple example. A control change message looks like:

1011nnnn 0ccccccc 0vvvvvvv

where again, nnnn is the MIDI channel on which the message is being sent (MIDI channel 1-16 = 0-15 → binary), ccccccc is the controller number (0-127 → binary), and vvvvvvv is the new value for the controller (0-127 → binary). For example, if we were working on MIDI channel 5 (= 4 → 0100) and changed our tremolo depth (c=92) to a value of 100, the control change message would look like:

10110100 01011100 01100100.

Creating Controller Messages:

To create the MIDI controller messages, I used the ManMIDI v. 1.02 firmware developed by Dylan Menzies of zenprobe.com. (ManMIDI stands for Multi-channel Analog to MIDI.) I burned the firmware to a Microchip PIC16C711, which is a microcontroller that has four channels of on-board analog to digital conversion. The homemade ManMIDI chip takes analog voltages at each of its ADC inputs, digitizes them, and appends the new digitized values (0-127 → binary) to the end of MIDI controller messages like the one above. Each of the four analog inputs (1-4) corresponds to a certain MIDI controller number (0-3) and are output on MIDI channel 1. Analog inputs that are equal to the power supply voltage of the chip will produce a controller value of 127. Analog inputs that are equal to ground will produce a controller value of 0.

Creating the Analog Voltages:

I used two primary methods to produce my analog voltage inputs for the ManMIDI chip. The first is an infrared distance sensor with an analog output (Figure 1) and the second is a simple voltage divider. The voltage divider consists of a fixed 500 O resistor and a 10kO linear fader potentiometer (Figure 2).

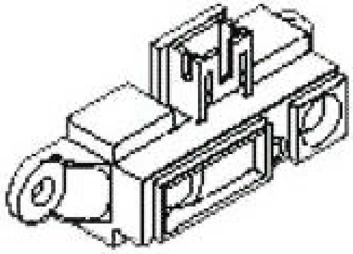


Figure 1: IR Sensor

Figure 2: Fader

Interfacing to the Human Hand:

As I stated above, the key element of my project was to build a practical interface between digital electronics and the human hand. The easiest way to do this is through a glove. I attached the infrared sensor (Sharp GP2D12 General Purpose Type Distance Measuring Sensor) to the palm of the glove so that it would output a voltage proportional to the distance of my hand from any object. (The sensor's range is from about 10cm to about 80cm.) I then designed and built a retractable cable system to move the faders. With the cables attached to the tips of the glove's fingers, an analog voltage can be output that is proportional to the amount that a finger is flexed. The retractable cable system and several pictures of the sensors on the glove are shown below.

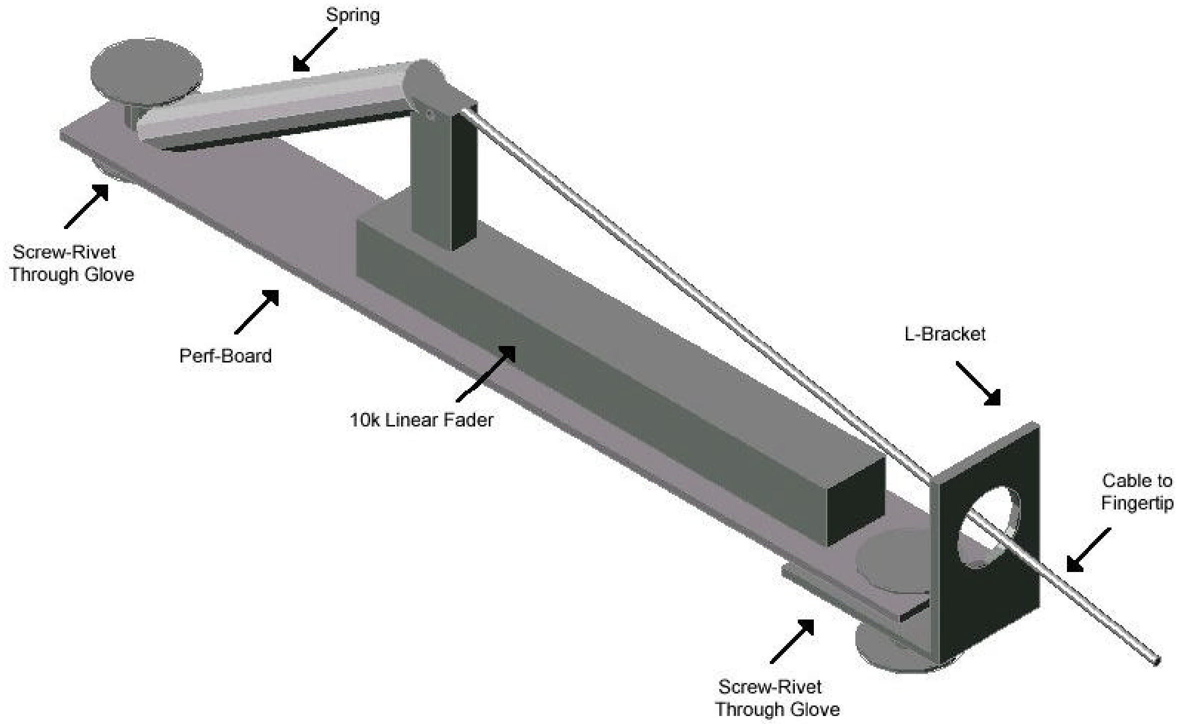


Figure 3: Retractable Cable System for the Faders

Figure 4: IR Sensor and Underside of Glove

Figure 5: Top of Glove and Cable/Fader Assemblies

The Circuits:

When digitizing any analog signal, it is best to amplify that signal so that its full range matches the full input range of the ADC. The output range of the IR sensor did not fill the entire input range for the ManMIDI chip. I had to offset the input from the IR sensor by –23mV and then give it a gain of about 1.94 to fill the entire 0V-5V range of the ADC. This was all done with op-amps.

In addition, I added a sample-and-hold circuit into the conditioned IR sensor output voltage line. This simple sample-and-hold circuit allows the user to lock in a certain hand position above an object and to then be able to move his or her hand without changing the value set by the IR sensor at that particular height. The “sample or hold” status is set by tapping a small momentary switch that I have mounted to the index finger of the glove. (This switch can be seen at the right-hand side of Figure 4 above.) The momentary switch provides a negative–going pulse that is fed through an RC constant/Schmitt inverter trigger and is then used as the clock pulse for a J-K flip-flop in toggle mode. The output of the flip-flop triggers the analog switch that controls the sample and hold circuit. One tap on the momentary switch sets “sample” mode. Another tap sets “hold” mode. I have also included a green LED indicator on the glove itself to show the user what mode sample-and-hold circuit is in. When the green LED is on, the circuit is in “sample” mode.

In addition to this LED, there is a yellow LED on the glove itself and a red LED on the project box. These two LED’s indicate power to the glove and power to the box, respectively. Speaking of power, I would also like to note that the power supply that I used for this project was a simple 9V, 210mA wall-wart transformer from a cordless phone and a Radio Shack 5V voltage regulator. The power supply lines were also coupled to ground with 1 μF capacitors at several strategic locations as to reduce the possibility of spurious MIDI messages being sent.

Most of the circuit is contained within a plastic Radio Shack project box. The little that needed to be on the glove was simply soldered to a small piece of perf-board that is attached behind the thumb. To connect between the electronics on the glove and that in the box, I used a length of eight-conductor networking cable.

The IC’s used in this project are all of the low-voltage, low-power type. I chose low-voltage components strictly because I already had the makings for a 5V power supply. Other similar non low-voltage components would work in their place with an appropriate power supply. Below is a list of the IC’s that I used. (I must also note that ALL of these IC’s were obtained for free as samples from the manufacturers’ websites.)

Microcontroller	Microchip	PIC16C711
Hex Schmitt Inverter	Texas Instruments	CD74ACT14E
Dual J-K Flip-Flop	Texas Instruments	CD74AC112E
Quad Analog Switch	Texas Instruments	CD74HCT4316E
Low-V Quad Op-Amp	Burr-Brown	OPA4251PA

The following are the circuit diagrams for the MIDI Controller Glove. The first is a simplified circuit diagram and the second is the actual wiring diagram using the IC’s specified above. Please note that the 1 kO potentiometer at op-amp 1 should be set so that 23 mV appears at the non-inverting input of this op amp and that the 2 kO potentiometers at op-amp 2 should be set to 1.19 kO.

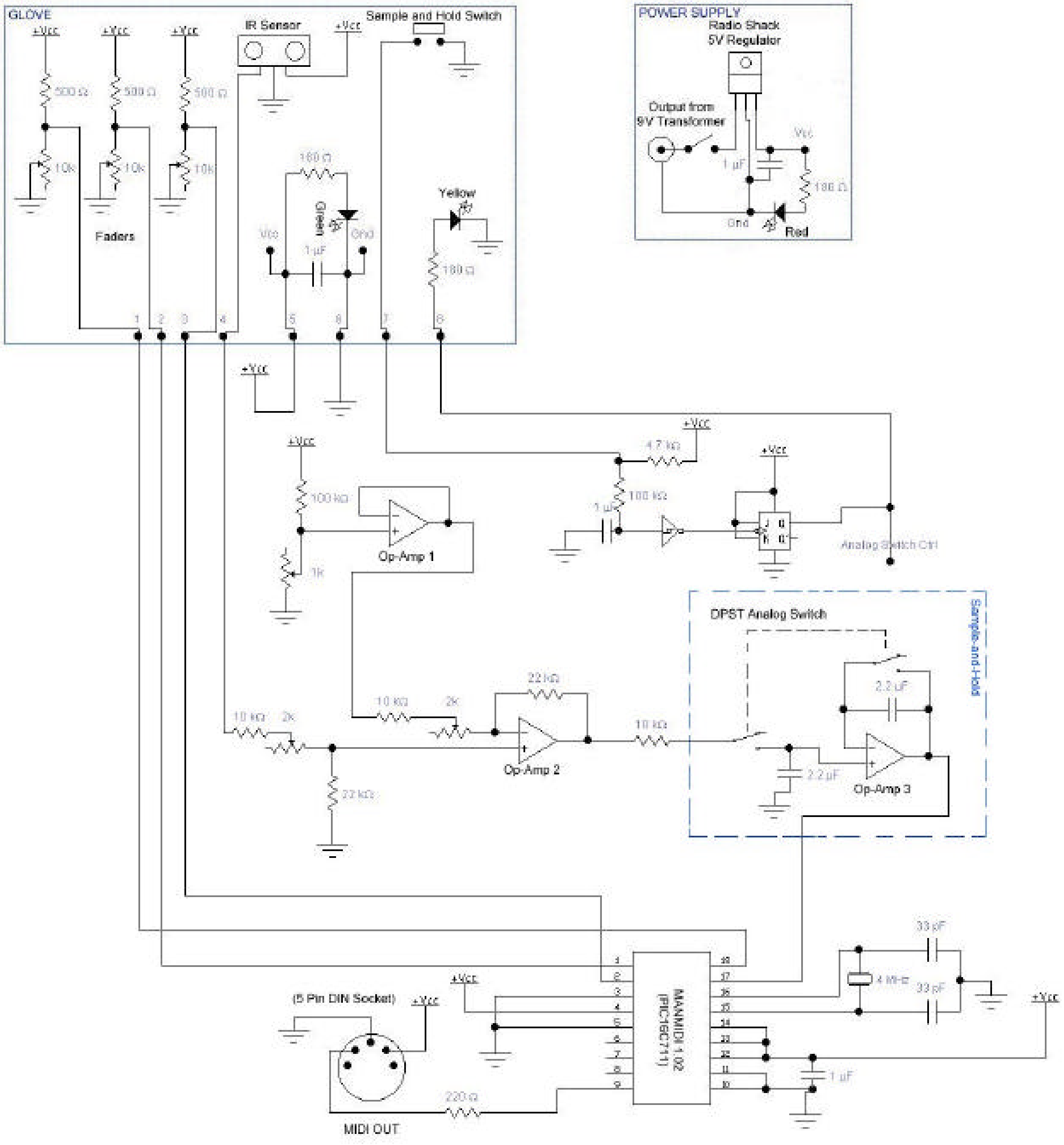


Figure 6: Simplified Circuit Diagram

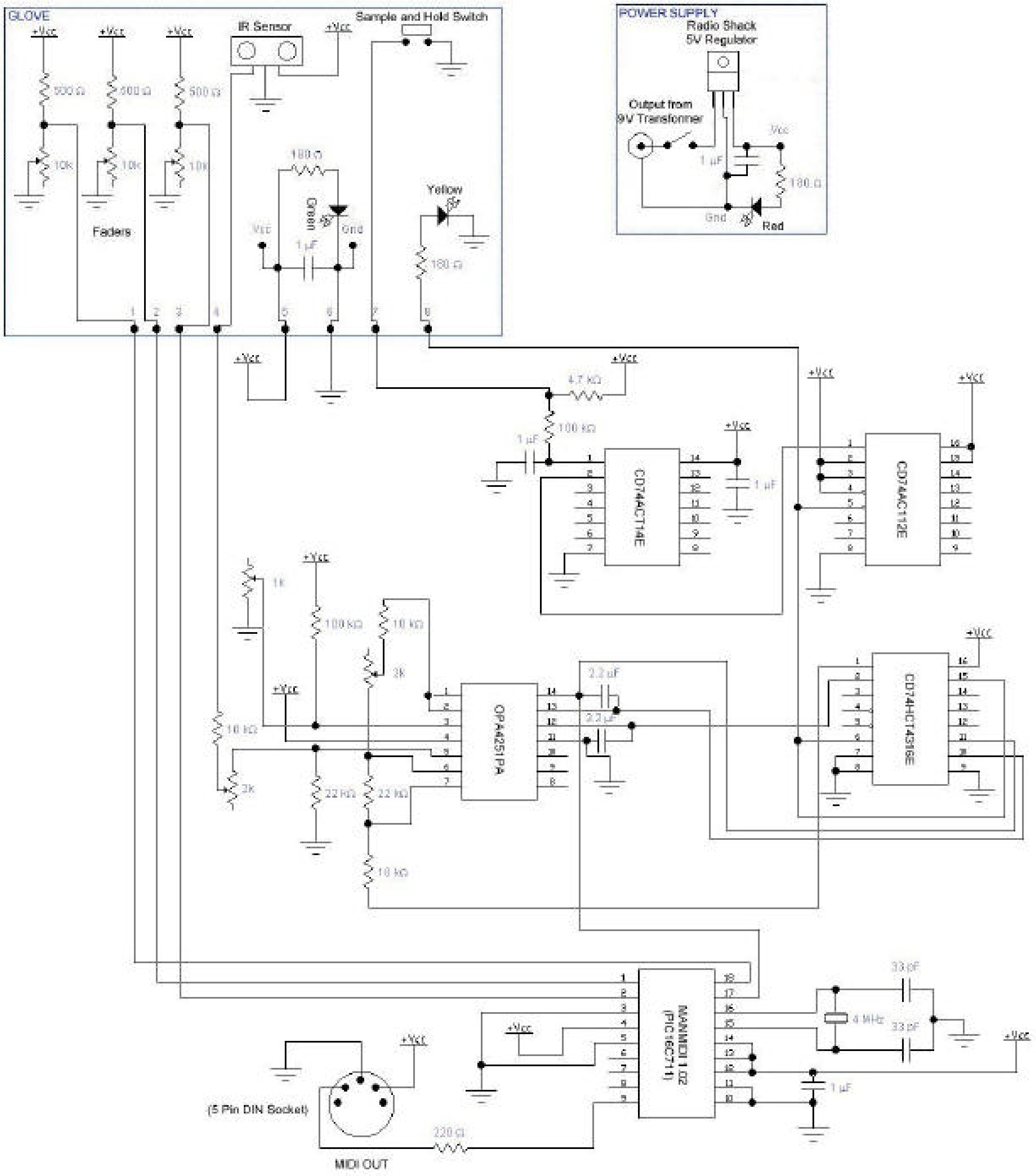


Figure 7: Wiring Diagram

Building the MIDI Controller Glove:

The glove itself is a mechanic's glove. This type of glove, I decided, had the perfect balance of durability and allowance for dexterity. Holes were punched through the spandex and leather of the glove to allow for screw-rivets to be placed through the material. The screw rivets hold each of the fader assemblies, the cables at the fingertips, and the network cable-attachment perf-board to the glove.

The fader assemblies are constructed by first drilling out holes in a length of perf-board to allow the screw rivets and the fader to be attached through it. The fader has clips on the bottom that are simply bent to attach it to the board. A hole must be drilled through the sliding part of each fader to allow the cable to be passed through it. One end of the spring is slipped over the sliding part of the fader before the cable is attached. The other end is attached to the rear screw-rivet. The cable must pass through an L-bracket that is attached to the front screw-rivet. Each cable-end is secured with a crimp.

The sample-and-hold switch is mounted to a corner bracket that I custom made out of sheet aluminum. This bracket is secured to a hose clamp that fits around the index finger of the glove. For simplicity, the IR sensor was sewn to the palm of the glove with heavy-weight plastic thread.

The plastic project box is easily drilled out and / or cut with an Exacto knife or a rotary tool. Three major holes and one slot must be cut in this box: a slot for the power switch, and one hole each to accommodate the DIN socket, the network cable, and the transformer plug. (Four other holes must be drilled out for the screws that hold the switch and DIN socket in place.)

Please refer to Figure 3 and other pictures throughout for visualization.

Performance:

I had originally wanted to augment the ManMIDI code to allow for "Note On" and "Note Off" messages to be sent. I quickly learned that this type of programming is out of my league and settled for controller messages being output. I figured that I could find some interesting way to use a tool like the MIDI controller glove in the hardcore electronic music that I produce in what little free-time I have. Many modern MIDI synthesizers like mine allow the user to remap MIDI controller numbers and to assign custom effects to them. I was able to accomplish this and was extremely surprised with what I can do with the MIDI Controller Glove. I can hold down a single note and change the sound so drastically with the movements of my hand through the MIDI Controller Glove that the same sound will not come out twice. This type of control and instrumentation has innumerable applications in the genres of ambient music, noise loops, and film scores, just to name a few. I spent approximately $100 building the MIDI Controller Glove and can safely say that I think I got my money's worth.

More Pictures:

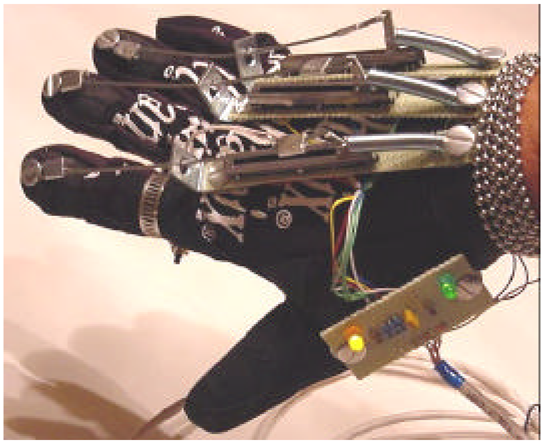


Figure 8: Top View, on Hand


Figure 9: Bottom View, on Hand

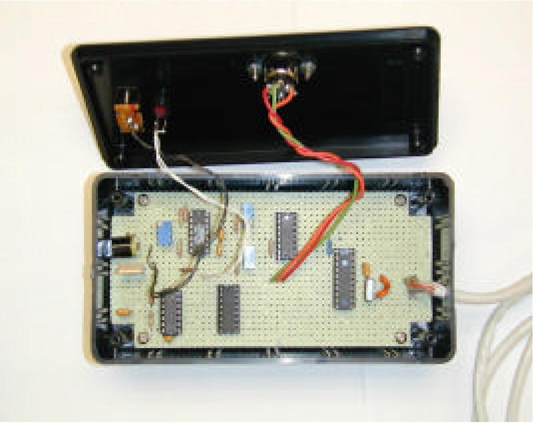


Figure 10: Project Box Inside


Figure 11: Project Box Outside

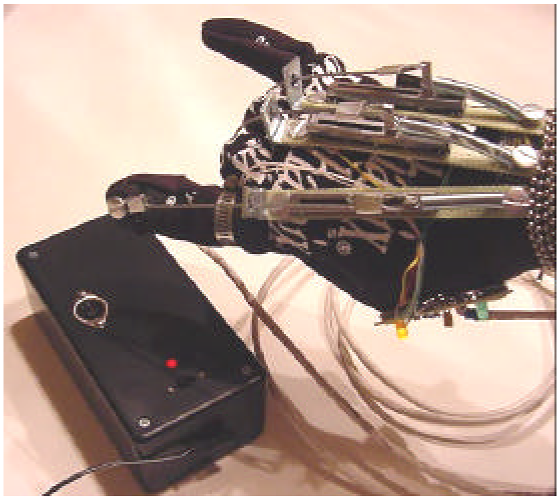


Figure 12: The MIDI Controller Glove
In Action

Acknowledgements:

This project would have never been possible without the help of several people. Thank you to Dylan Menzies of zenprobe.com for developing the code and allowing me to use it. Thank you to Todd Moore for helping me burn the code. And finally, thank you to Patrick Marks and Professor Steven Errede for the help and guidance when necessary.

MIDI Controller Glove Helpful WWW References:

http://www.hobbyengineering.com	(IR Sensor)
http://www.digikey.com	(Faders)
http://www.radioshack.com	(Misc. Parts)
http://www.microchip.com	(Microcontroller)
http://www.ti.com	(For TI and Burr-Brown Parts)
http://www.zenprobe.com	(email → manmidi@zenprobe.com)
http://www.midi.org	(MIDI Manufacturers' Association)
http://www.ece.uiuc.edu/ecestores/	(Misc. parts)
http://www.mechanix.com	(Mechanic's Gloves)