
Project Final Report

Distance Sensor MIDI Controller

April 26th, 2011

Matthew Harlan

Project Abstract

The distance sensor MIDI controller is a MIDI output interface that can control any MIDI capable synthesizer, sequencer or other MIDI input based hardware or software device based on the position of a person's hand directly above a distance sensor. The project uses two ultrasonic distance sensors, a MIDI output circuit, and a firmware for the Zilog ZNEO microcontroller (as part of the ZNEO contest kit).

The firmware translates an analog voltage value into an integer distance value in inches. This value is then used to generate MIDI messages. These messages consist of note on and note off events and control change messages. The three buttons on the ZNEO contest kit can be used to put the controller into one of three modes: standard mode, controller mode, and polyphony mode.

Each mode configures what MIDI messages each sensor triggers. In standard mode, the right sensor sends note messages and the left sensor sends control change messages. In controller mode, both sensors send control change messages. Lastly, in polyphony mode, both sensors send note messages. The ZNEO contest kit LED array displays a number representing the distance interval last read from a sensor for note events.

Status

The project went as planned and all functionality was implemented. Potentiometers were dropped from the design when the distance sensors proved intuitive and easy to interface with. They were only initially included to provide a voltage source for testing if the distance sensors were difficult to use.

Buttons toggle device settings and the LED array displays note information to the player. The MIDI output protocol and circuit were implemented and built respectively. Analog distance sensors were used as planned. The project resulted in a useable MIDI device which was used during a live performance at a concert by students in the George Washington University's electronic and computer music class.

Parts List

Name	Source	Description	Qty.	Price
Maxbotix LV-EZ4	www.sparkfun.com	Analog distance sensor	2	\$27.95
ZNEOCTK0100KIT	www.mouser.com	Zilog ZNEO microcontroller "contest kit" development board	1	\$49.99
220-ohm Resistor	www.mouser.com	Resistors for the MIDI output circuit	2	\$0.12
SN74LS00NE4	www.mouser.com	Quadruple two input positive NAND gate	1	\$0.61
VAOL-3GCE4	www.mouser.com	Yellow through hole LED	1	\$0.16
57GB5FX	www.mouser.com	A 5-pin 180 degree female DIN port	1	\$3.17
Total Cost				\$110.07

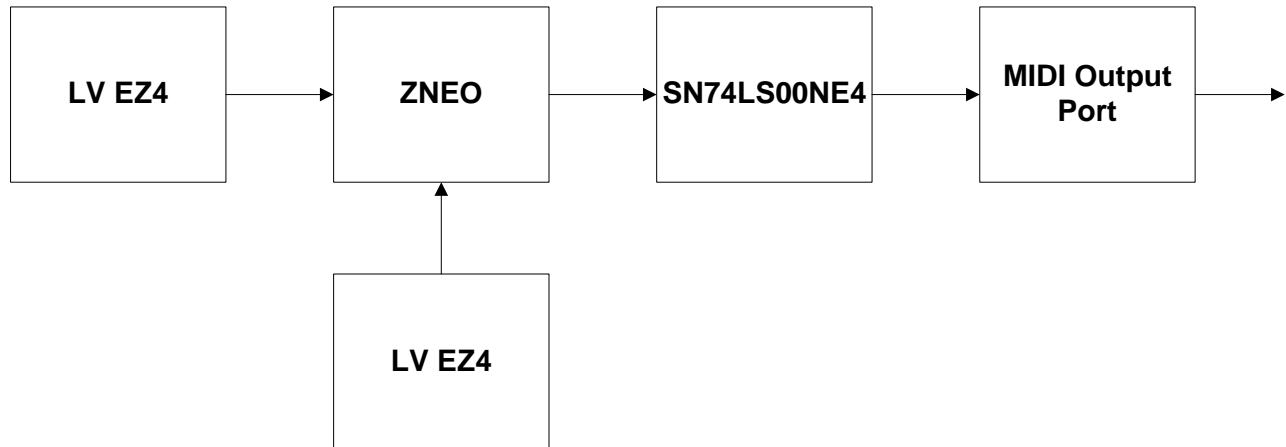
Hardware Specification

The ZNEO contest kit was selected as the development platform because the ZNEO was familiar and the contest kit was provided for free by the George Washington University Department of Computer Science. The ZNEO contest kit also provides an LED array as well as three hardware buttons. The LED array is used to display the note interval being played for note on events. The hardware buttons are used to switch MIDI output modes. In retrospect, another microcontroller should have been used due to Zilog discontinuing production on the ZNEO.

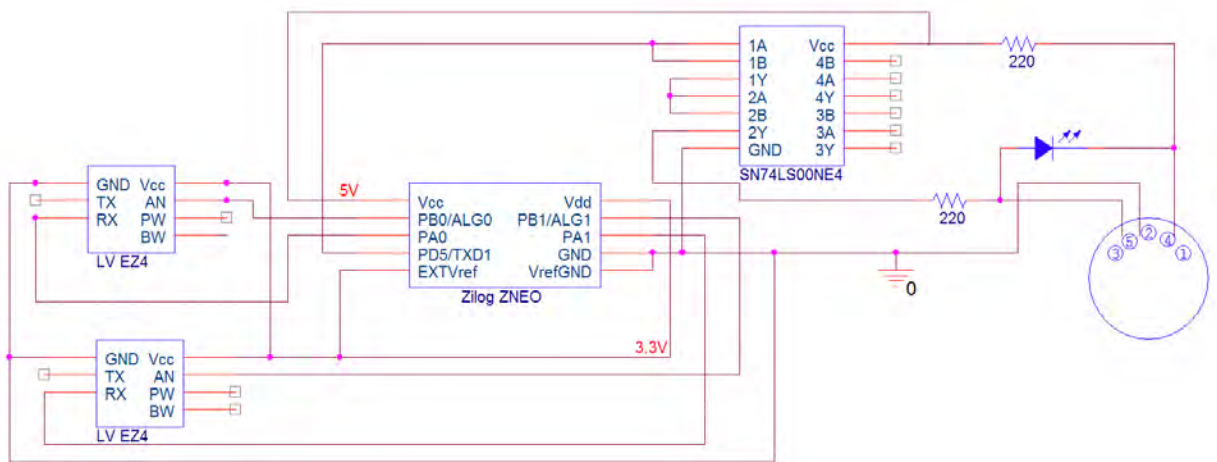
Distance sensors that provide an easy analog interface were selected to make development simpler. The Maxbotix LV-EZ4 provides a digital PWM output signal, a RS232 serial output, as well as an analog output. This provided backup options if the analog to digital conversion proved problematic. The LV-EZ4 also provides a narrow beam width, which is needed to only track motion directly over the sensor. A narrow beam width also prevents interference between the two sensors.

The rest of the hardware makes up the MIDI output circuit. The two NAND gates act as an inverter buffer to increase the maximum current that can be supplied by the UART output pin of the ZNEO. The resistors enable the circuit to conform to the official MIDI output circuit specification. The 5-pin 180 degree DIN port provides the standard MIDI port. A simple through hole LED acts as an activity indicator for the output port.

Hardware Block Diagram



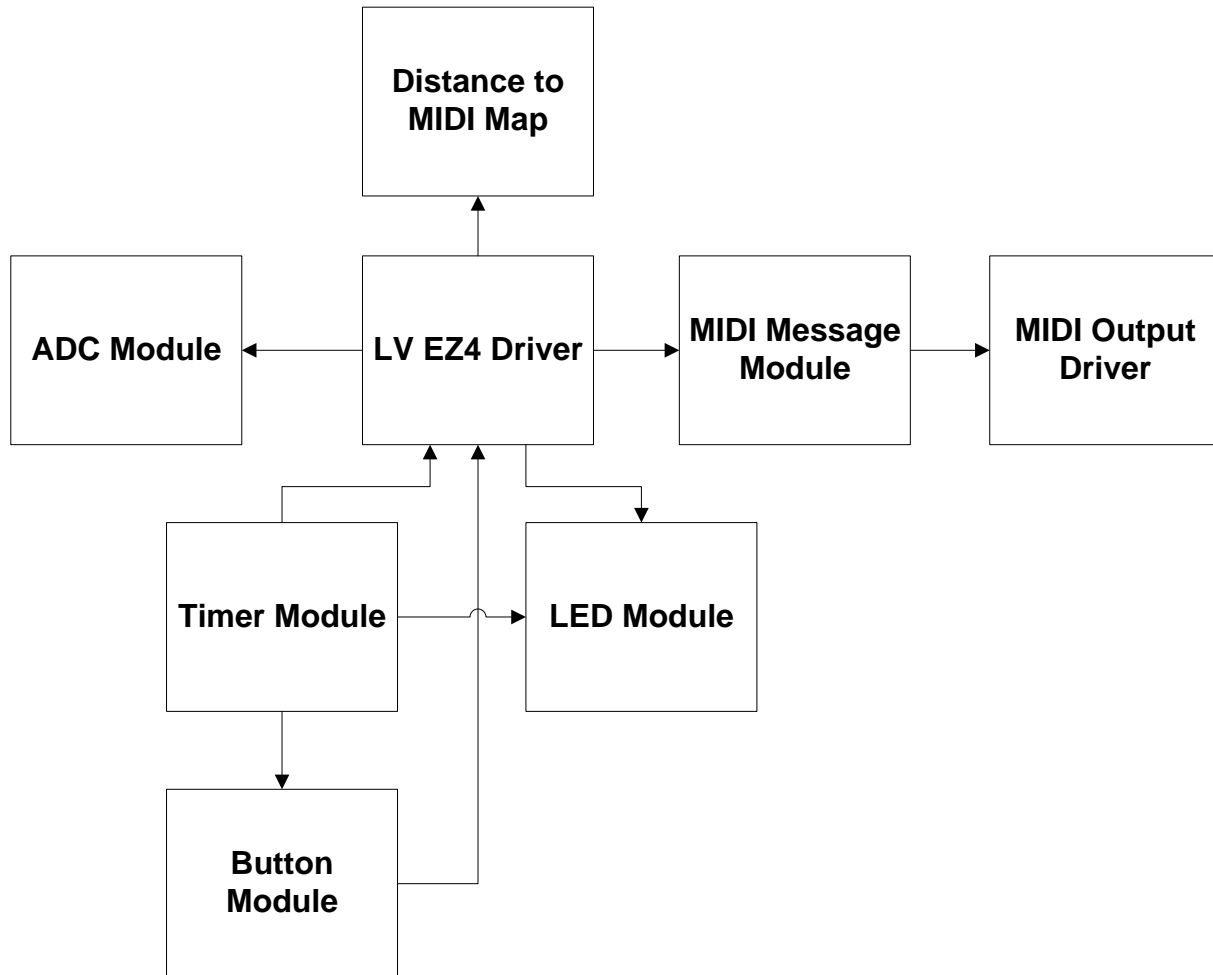
Hardware Schematic



Construction

Building the MIDI controller circuit was fairly simple. Wires had to be soldered to the pins on the back of the MIDI output port. Pins were also soldered into the distance sensor break-out boards. No more soldering was necessary to connect the sensors and MIDI port to the ZNEO development board. All remaining hardware was connected using two breadboards. Two breadboards were used to separate the distance sensors and prevent interference between them. The MIDI circuit was added to the right sensor's breadboard.

Software Block Diagram



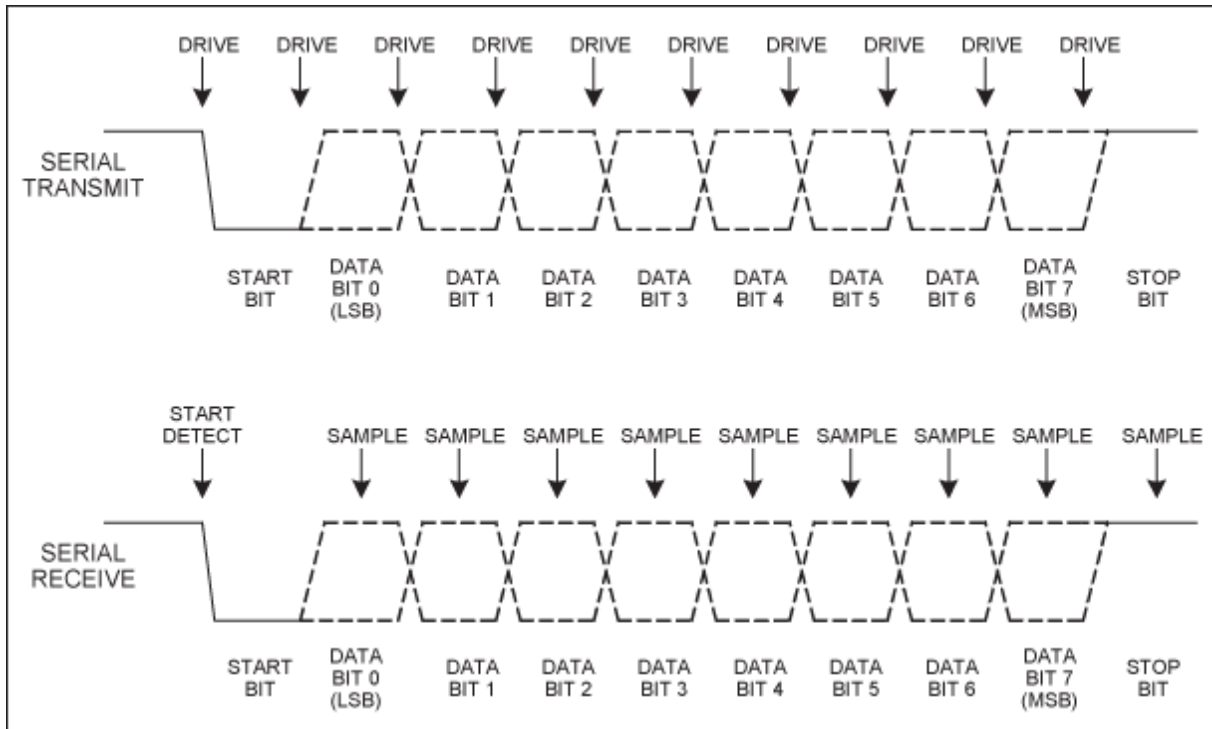
Software Specification

The MIDI controller makes use of an interrupt driven firmware to translate analog distance values into MIDI messages. The firmware was modularized to allow for a piecewise development process. Each module controls one aspect of the project and provides a concise public API. No pre-written APIs were used to provide any functionality for this project. However, standard C library functions are used for serial port debugging.

MIDI Output Driver

The MIDI output module provides an interface for sending MIDI messages. MIDI output is interrupt driven to provide a quick response. The MIDI output API currently supports sending one, two, and three byte MIDI messages. This was sufficient for this project because note messages and control messages, the only messages used, are each three bytes long. Support for longer messages can be easily added in the future if necessary. Each message is sent a byte at a time in serial with the LSB first. MIDI uses the 8-N-1 (eight data bits, no parity bit, one stop bit) serial output format at a baud rate of 31,250 bits-per-

second. A UART on the ZNEO provides the physical hardware for sending the MIDI messages.



MIDI Messages Module

The MIDI output module is interfaced with through the MIDI messages module. This provides an abstraction from the direct MIDI output code. This module can be used to quickly compose MIDI messages without needing to know the underlying specification. The MIDI messages module currently only supports control messages and note messages. The channel that each message is broadcast on is configurable.

Messages	Status Byte	Data Byte	Data Byte
Note On	1000xxxx	0-126 (Pitch)	0-126 (Velocity)
Note Off	1001xxxx	0-126 (Pitch)	0-126 (Velocity)
Control Change	1011xxxx	0-126 (Ctrl. #)	0-126 (Ctrl. Value)

ADC Module

The analog to digital converter module provides an interface for reading from analog sensors using the ADC on the ZNEO. When an analog value is available, an interrupt is triggered. The value is then processed by a processing procedure that is configured when the ADC module is first initialized. This allows the ADC module to be easily reused in future projects. The ADC module currently performs a single conversion and then stops each time a conversion is requested. It would be very easy to add free-run mode functionality to this module for future projects.

LV-EZ4 Distance Sensor Driver

The distance sensor module interacts with the ADC module. The raw analog sensor value is processed by this module and converted into inches. The value is then passed to the MIDI mapping module and converted into MIDI note or controller values. MIDI messages are then sent using the MIDI messages module. The distance sensor module also maintains the state of the sensors to determine what type of message each sensor generates.

In order to prevent interference between sensors, only one sensor is sampled at a time. A start up delay of around 400 milliseconds is performed to allow sensor calibration and warm-up to occur. After start-up, an analog conversion is performed every 50 milliseconds. Distances above and below the maximum and minimum recognized distances are ignored. This ensures that only a subset of the detectable distance range is capable of generating MIDI messages. Messages are only sent when MIDI output values change.

Distance to MIDI Mapping Module

Distance values are translated into either note values or control values. The current configuration uses a tolerance of three inches for each MIDI value. This allows a reasonable distance to represent a full musical octave. It also ensures that slight movements do not accidentally trigger new MIDI messages.

Timer Module

The timer module acts as a two millisecond timer that provides all timing functionality for the controller through the use of additional external counters. All timer events are called from the timer interrupt, thus all modules that have timed events must have a function that is called during this interrupt. Each module then manages their own time counters within their timer event function.

LED Module

The LED module uses the timer module to continually draw to the ZNEO contest kit LED array. The LED module is used to output the interval that is currently being played by a note on event. However, the LED module code is capable of displaying much more.

Button Module

The button module uses the timer module to poll and de-bounce button events. The three buttons on the ZNEO contest kit are used to switch the current mode that the MIDI controller is in. This allows the controller to be more versatile. In standard mode, the right sensor sends note messages and the left sensor sends control change messages. In controller mode, both sensors send control change messages. Lastly, in polyphony mode, both sensors send note messages.

Main Loop

The main application initializes each module and then loops infinitely. This is because the MIDI controller firmware is completely interrupt and event driven. However, a serial port based command line interface can run in the background to provide an additional debugging method if necessary. This debugging module was designed independently and is beyond the scope of this project.

Testing

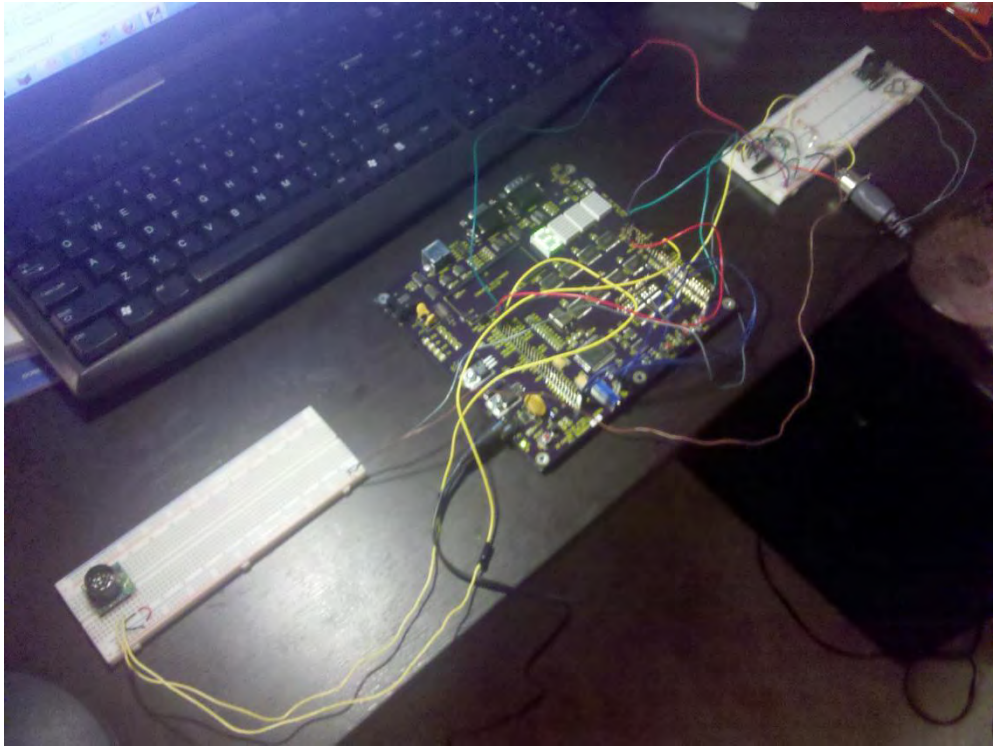
Testing was performed on each component independently. Modules were verified using small unit tests for specific test cases. The output from the MIDI circuit was also tested using a digital oscilloscope. The controller was connected to a computer using a MIDI to USB cable to debug the MIDI messages that were being sent. A Windows program, called MIDI-OX, was used for this verification. In addition, distance values were displayed on the LED array to verify that the distance sensors were working properly.

Milestone Chart

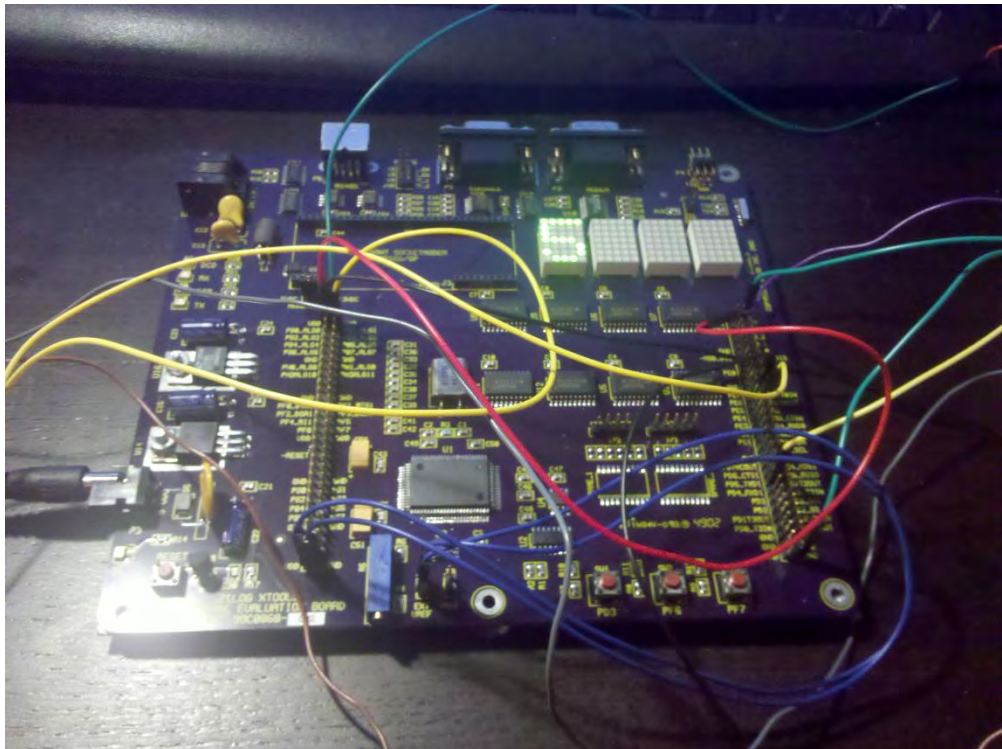
Milestone	Date Completed
Hardware Research	February 14th, 2011
MIDI Specification Research	February 21st, 2011
Built MIDI Output Circuit	March 7th, 2011
Built MIDI Output Module	March 21st, 2011
Built MIDI Message Module	March 21st, 2011
MIDI Output Testing	March 21st, 2011
Built ADC Module	April 4th, 2011
ADC Module Testing	April 4th, 2011
Built LV EZ4 Distance Sensor Circuits	April 8th, 2011
Built Distance Sensor Driver	April 8th, 2011
Distance Sensor Testing	April 8th, 2011
Built Distance to MIDI Module	April 11th, 2011
Full Software and Hardware Integration	April 15th, 2011
Full System Testing	April 15th, 2011
Tweaking and Polish	April 17th, 2011

Project Photos

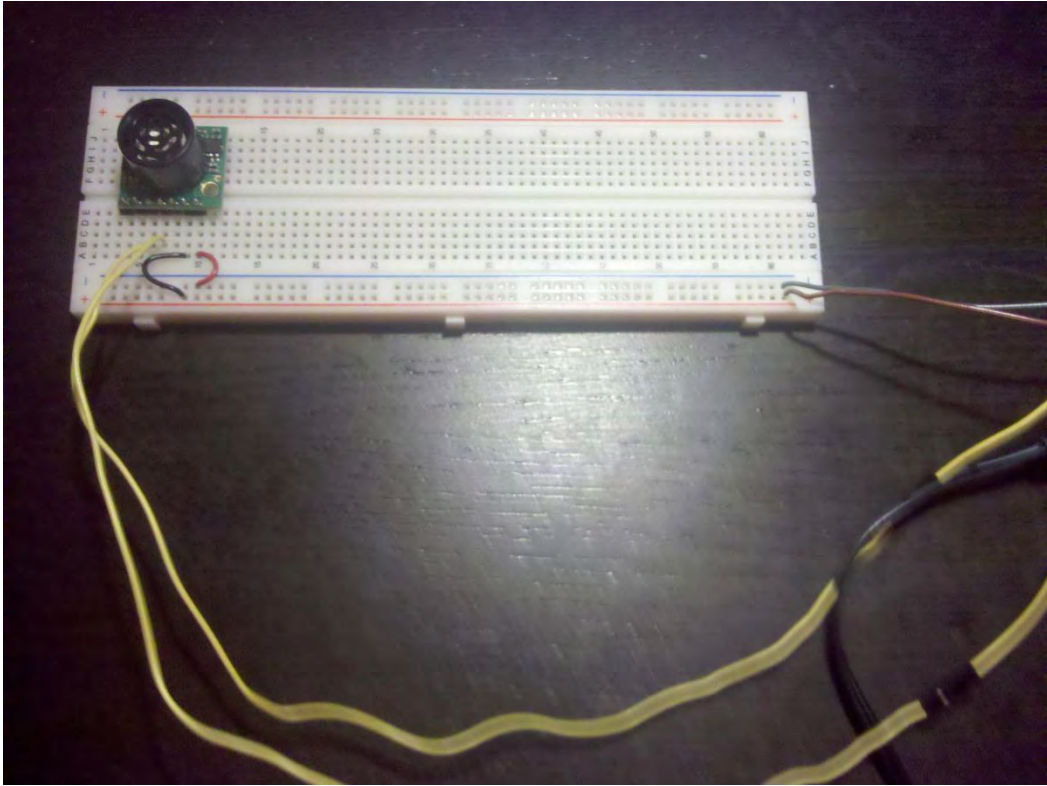
The Complete Device



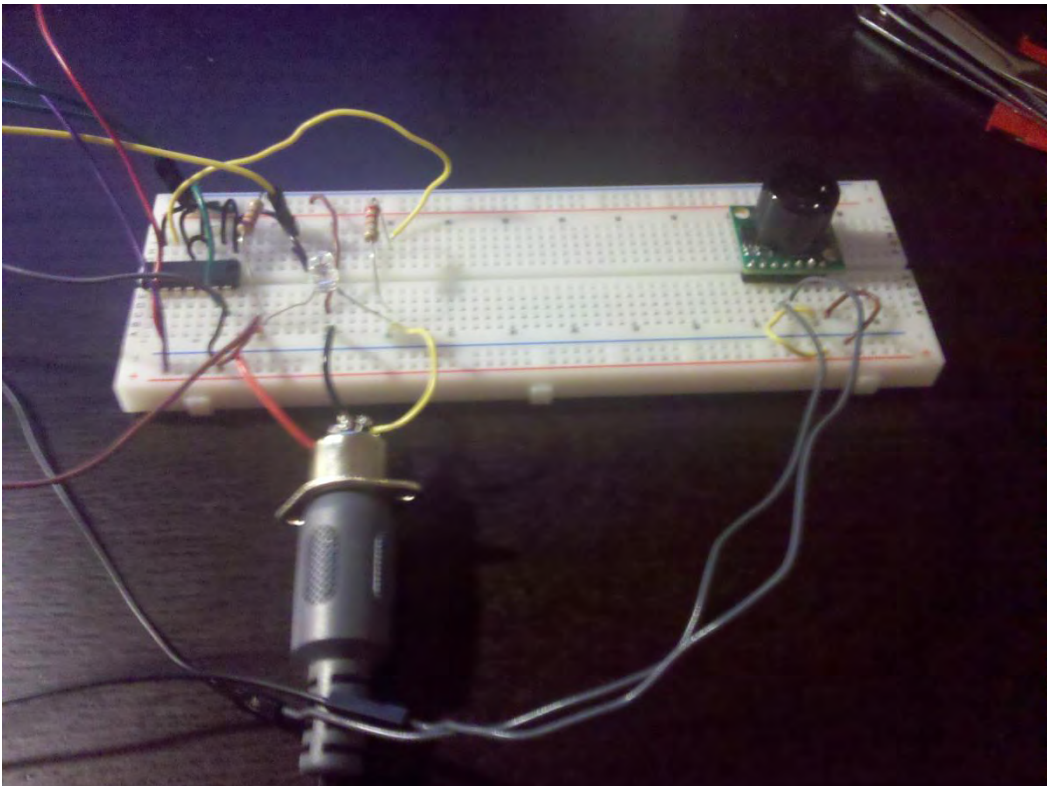
ZNEO Contest Kit Wiring



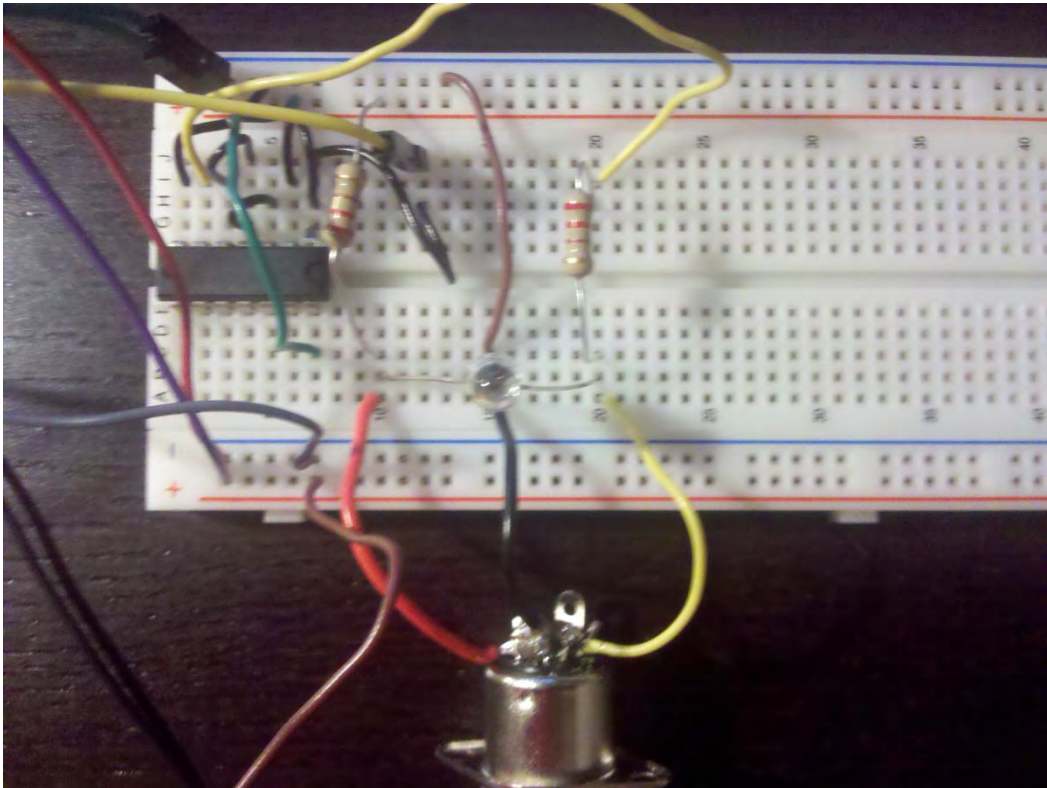
Left Distance Sensor



Right Distance Sensor



MIDI Output Circuit



Retrospective

Deciding to use high quality analog distance sensors that have a narrow beam length was a very important early design decision. This made it easy to write the software that read from the sensors. It was also important that the sensors could be easily enabled and disabled. Ultrasound from one sensor interfered and caused false positives to trigger on the other sensor. It was a simple fix to alternate which sensor was reading each time and eliminate this interference.

Code modularization made development go very smoothly. It was very easy to write the software for this project because all of my design goals were clearly outlined and properly abstracted. In addition, testing each module individually helped make it easy to isolate and identify bugs.

This project taught me a lot about MIDI controllers, the MIDI hardware and software specification, and ultrasonic distance sensors. I am confident I could produce a much more complex MIDI controller as a result of the experience I gained from this project. The only thing I wish I had done differently was my choice of microcontroller. I enjoyed developing for the ZNEO, but my code has less re-usability because ZNEO microcontroller production was discontinued by Zilog in June 2010.

Project Source Code: <http://www.github.com/mharlan/UltrasonicMIDI>