Project Final Report

Breathalyzer++

April 23, 2011

Ian Kenley

Project Abstract

This project takes advantage of the MQ-3 Alcohol Gas Sensor to create a working breathalyzer. Since the input is dependent on relative humidity as well as temperature, a humidity and temperature sensor are employed to adjust for these environmental factors. Both the environmental and alcohol gas stats will output to an LCD display, which updates the temperature and humidity at regular intervals while allowing for interrupt-driven breathalyzer scans.

Status

The final product did work reasonably well as planned. In particular, the environmental sensors allowed for more robust readings from different tests, which is something similar projects with a narrower scope did not accomplish. There are still problems with the alcohol gas sensor itself, though. What the data sheet describes as a “preheat time” of 24 hours does in fact mean that it requires an impractically long amount of time before it will report a trustworthy reading. Additionally, the sensor requires about 15 minutes after a positive reading to reset to the baseline resistance level.

Specification

Hardware Modules

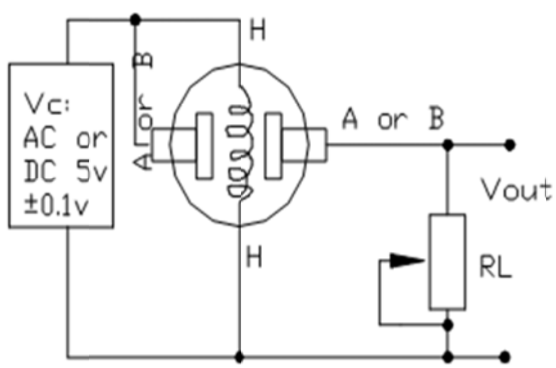
Zilog Z16 Microcontroller Contest Kit

The most important part of this project, the ZNEO board connects all of the other components and reads all the respective inputs. The ZNEO board is capable of converting analogue input to digital, scanning frequency output, and has software and registers for handling I2C interfacing. The board also has pins with voltage outputs ranging from 3.3V (the maximum for the HH10D humidity sensor) to 5V (the minimum required for the MQ3 alcohol gas sensor).

MQ-3 Alcohol Gas Sensor

The MQ-3 Alcohol Gas sensor can be acquired on sparkfun.com for about $5. The device heats the incoming gas and varies the load resistance based on the concentration, which can be read through an analogue to digital converter. While there are 6 pins on the MQ-3, only 4 need to be used. The voltage can be applied to either side provided that the heating and circuit voltage are wired to the same side. The datasheet recommends a 200kOhm pull down resistor, but the range of ADC readings is wider with a 10kOhm resistor.

Figure 1: MQ-3 Wiring



TMP102 Temperature Sensor

The TMP102 temperature sensor can be read from using the I2C interface. There is one additional pin, labeled ADD0, that determines the last two bits of the device address. To simplify operation, this is grounded in the breathalyzer system.

HH10D Humidity Sensor

The HH10D is a humidity sensor that can read the relative humidity to +/- 3%. It contains elements of both the I2C interface as well as an output frequency that can be read. The I2C allows access to the internal memory that stores offset and sensitivity values that need to be read only once and stored in the code. After reading these values the two I2C pins can be disconnected and only the FOUT pin is needed to read the relative humidity output as a frequency. The relative humidity can be read from the following formula:

(offset-FREQUENCY)*sens/2^12

Where offset and sens are predefined variables read from the device memory.

ADM1602K LCD Display

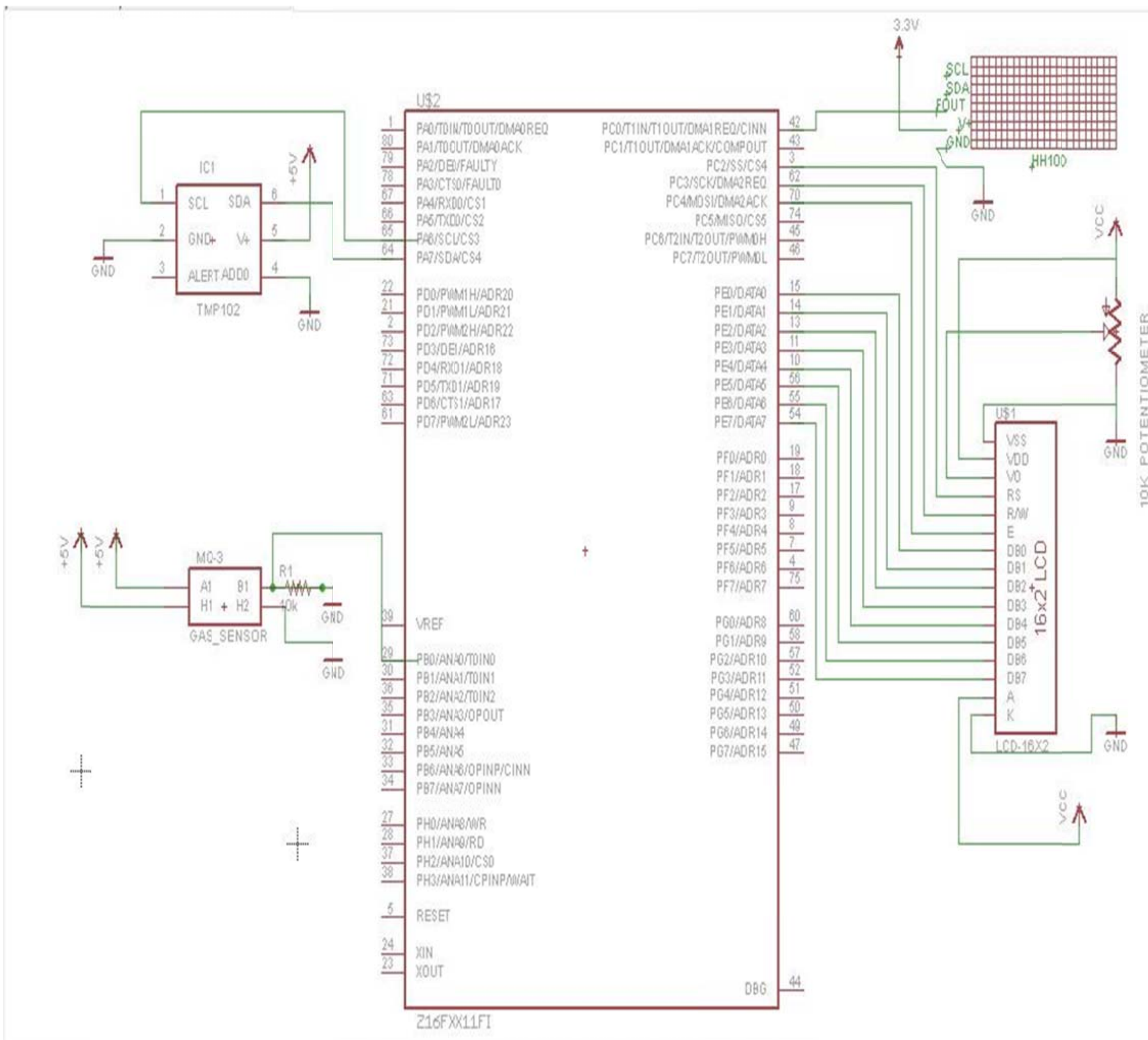
The LCD Display uses the HD44780 parallel interface that many other LCD displays employ. This particular model requires the Vcc pin on the ZNEO board, as the maximum voltage in is 3.5V. The Breathalyzer system uses the 8-bit interface for simple loading of instructions and character sending. Additionally, a 10k potentiometer is required for setting the contrast to a legible level.

Table 1. Pin Summary

Pin Number	Pin Description	ZNEO Pin Name	Notes
MQ-3			
1	V_c	V_{dd}	Circuit Voltage
2	V_h	V_{dd}	Heating Voltage
3	R_s	PD0	Sensing Resistance
4	V_s	V_{ss}	Heating Ground
TMP102			
1	SCL	PA6	
2	GND	V_{ss}	
3	Alert	-	Not used
4	Add0	V_{ss}	Not used- Held low
5	V+	V_{cc}	
6	SDA	PA7	
HH10D			
1	SCL	PA6	Only used once
2	SDA	PA7	Only used once
3	FOUT	PC0	Alternate Function 1 (T1_IN)
4	VDD	V_{cc}	
5	GND	V_{ss}	
ADM1602K (LCD)			
1	V_{ss}	V_{ss}	
2	V_{dd}	V_{cc}	
3	V_0	-	3rd pin on 10k

			potentiometer
4	RS	PC2	
5	R/W	PC3	
6	E	PC4	
7~14		PE0~PE7	8-bit data bus
15	LED+	V_{cc}	Backlight
16	LED-	V_{ss}	Backlight

Figure 2: **System Schematic**



Software Modules

Alc.c – Software driver for the alcohol gas sensor. Contains initialization routine and function for reading in analogue input.

Humid.c – Software driver for the humidity sensor. Contains initialization routine, functions for reading offset variables from device memory, and scanning the frequency to read the humidity.

Temp.c – Software driver for the temperature sensor. Contains initialization routine, as well as the function for reading the I2C input for temperature.

LCD.c - Software driver for the LCD screen. Contains initialization routine, as well as a generic function for displaying a string. Additionally, there are sensor specific functions that read in values from the sensors and display them to the LCD.

Saturn.c – Initializes the timer, which is uses for de-bouncing buttons, reading the frequency of the humidity sensor, and periodically updating the environmental variables.

Buttons.c - Initializes the buttons on the ZNEO, used to initiate a breathalyzer scan as well as switch to weather display mode when not in use.

main.c – The first function called on startup, this calls the initialization functions for all the other modules, sets the environmental variables to a known state, and then idles indefinitely.

Implementation & Construction

Figure 2: Hardware Layout

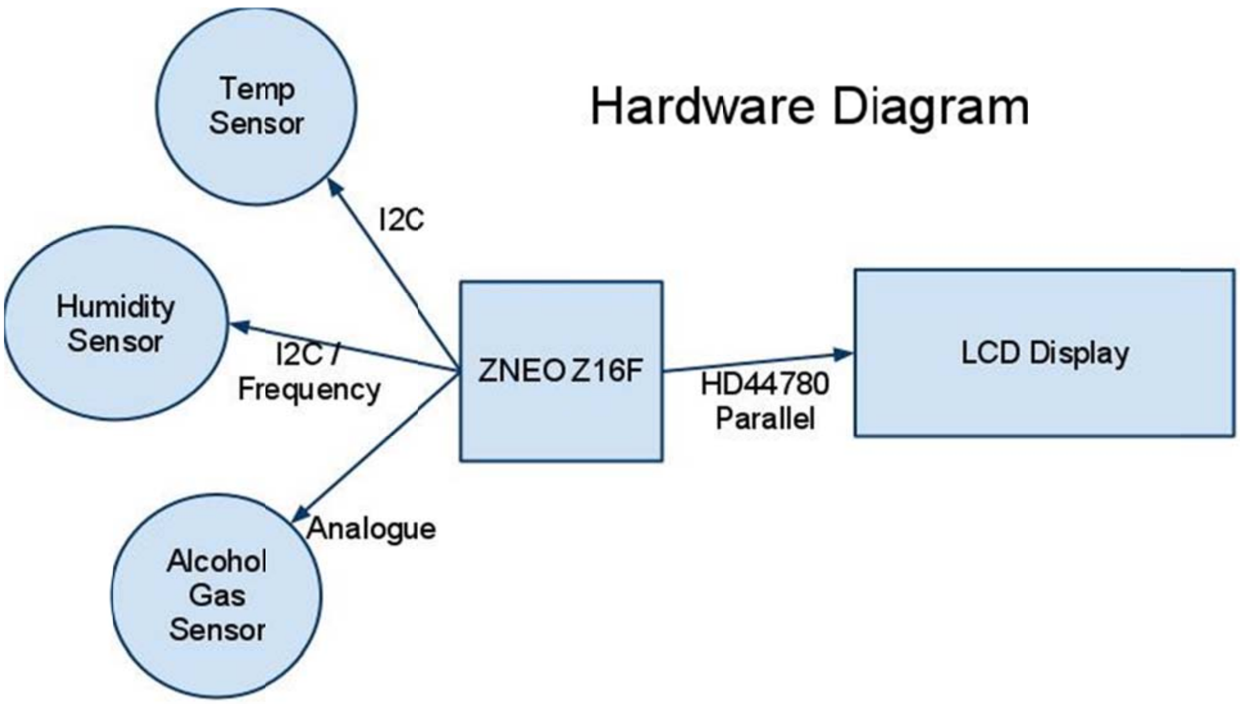
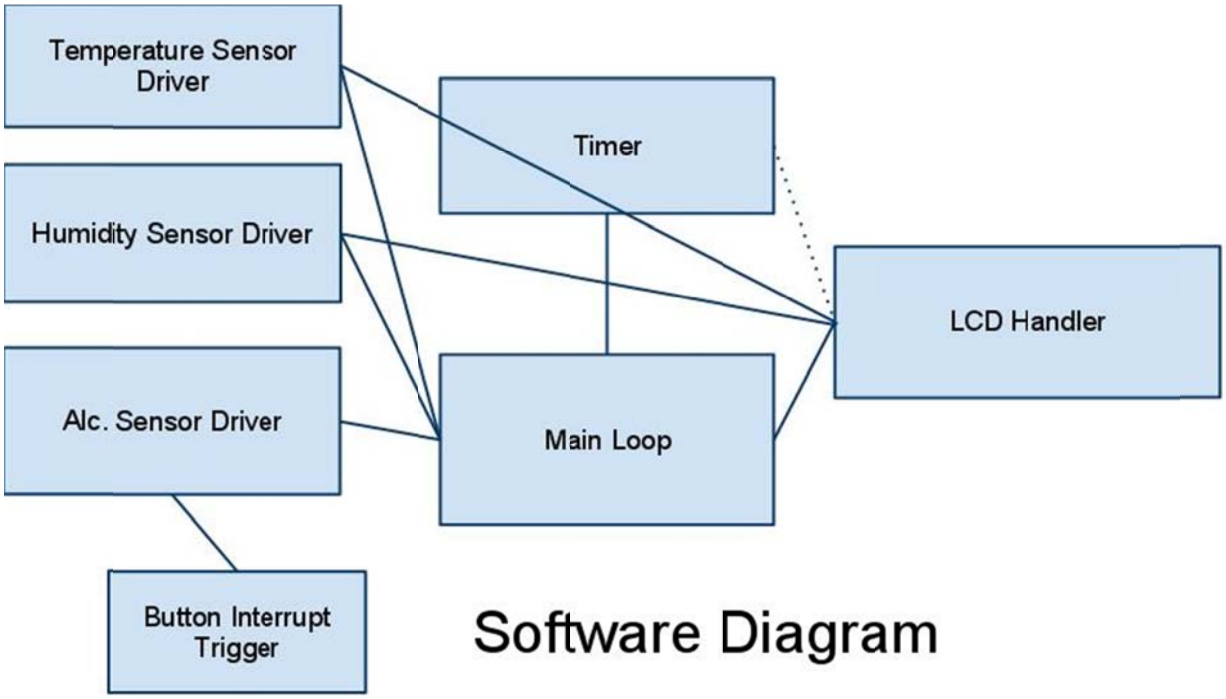


Figure 3: Software Layout



The main loop initializes all the individual modules. From there, the timer periodically calls the LCD handler to update the temperature and humidity, which it does by calling each module in a subroutine. This can be interrupted by pressing the "SW1" button on the ZNEO board, which calls the alcohol sensor read function (until SW2 is pressed and it returns to weather variable state).

Milestones

4/5- MQ-3 properly wired, driver software finished.

4/10- TMP102 properly wired, driver software written and correct readings verified

4/14- Switched to 8-bit interface on LCD, acquired potentiometer, properly wired, driver software able to initialize and print strings.

4/16- MQ-3 calibrated based on linear regression fit of known BAC point series.

4/17- HH10D properly wired, driver software written and correct humidity reading verified.

4/18- Linked software modules together, wrote device specific LCD functions.

Figure 4: MQ-3 Wiring

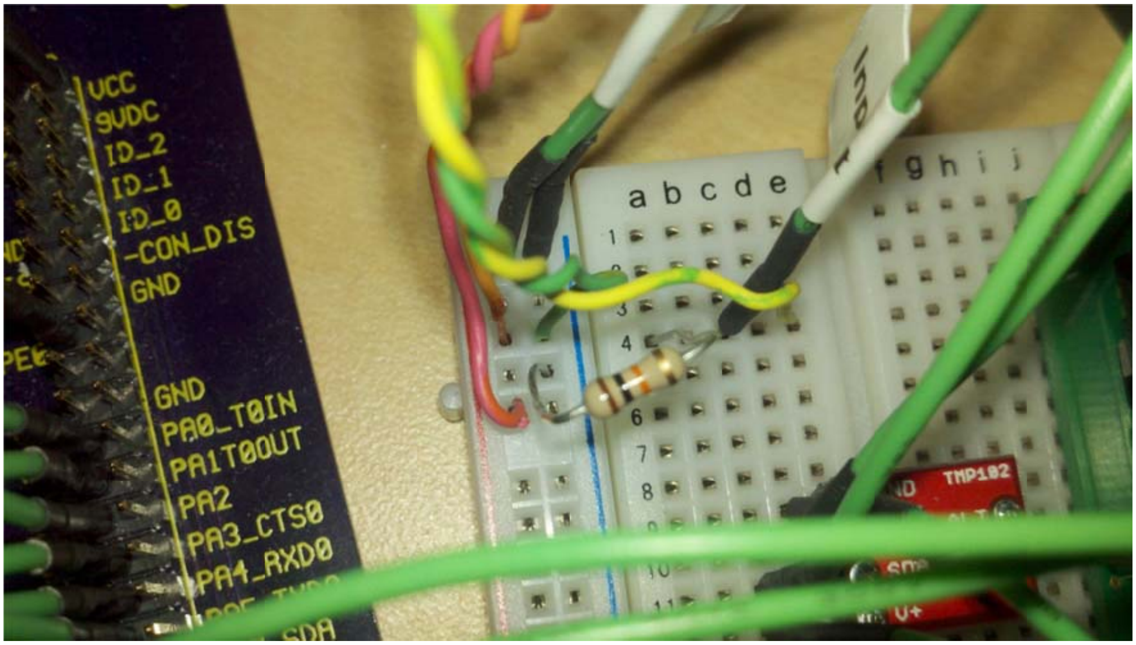


Figure 5: TMP102 Wiring

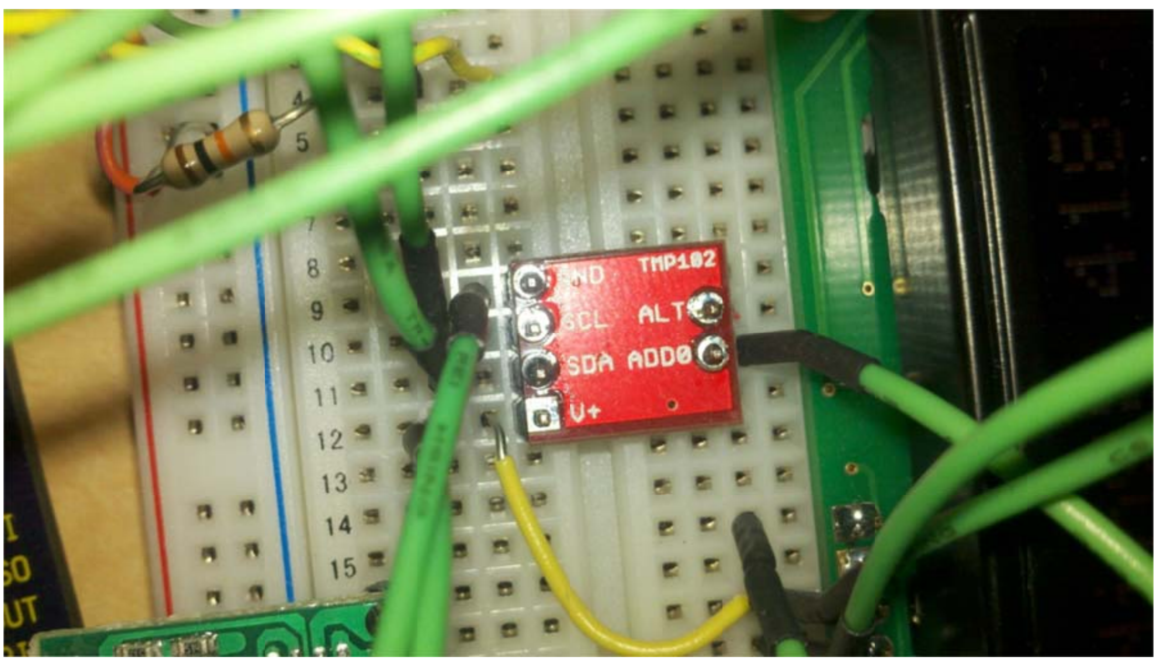


Figure 6: LCD Wiring

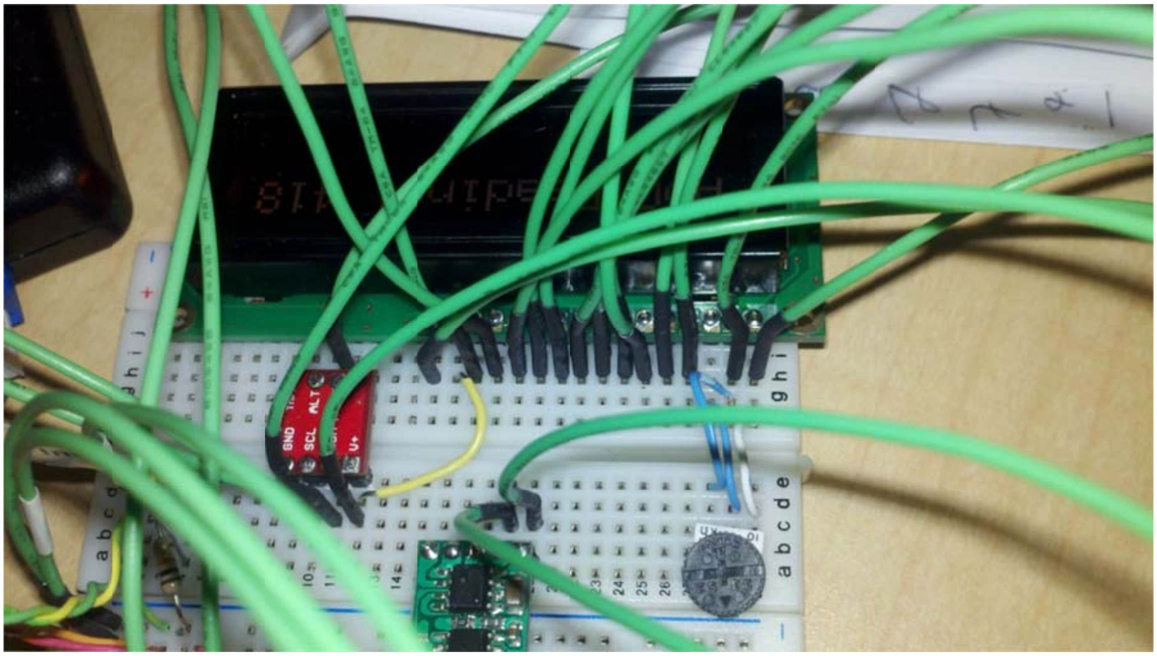
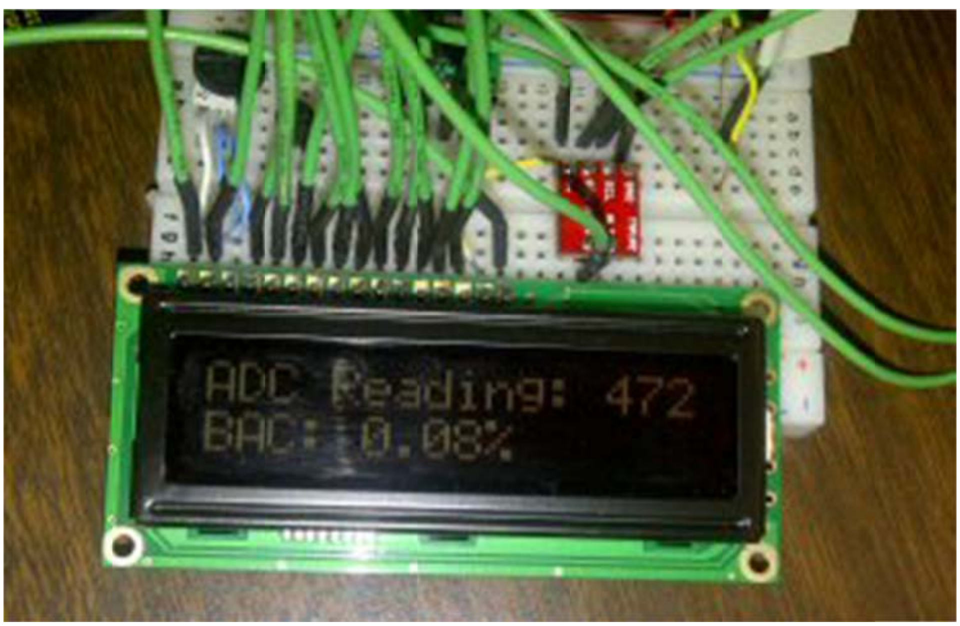


Figure 7: Finalized



Retrospective

One of the more important design changes I made was switching from 4-bit to 8-bit interface for the LCD display. Output issues turned out to be much harder to debug than the sensors, as there is no easy way to read the internal logic of the

LCD until the characters are actually displaying. This made the quick success of the 8 wire all the more miraculous.

The other major design decision that I had to make towards the beginning was which temperature and humidity sensors to order. Obviously, there are a variety of prices and interfaces to choose from. I wanted them both to operate on the same interface to make implementation less difficult, and the price range of the humidity sensors narrowed the selection down considerably.

One thing I would do differently would be to re-read the specifications in the datasheets. Until I discovered that the ZNEO board had a 3.3V pin near the modem connect section, I had no way of powering on the humidity sensor without setting up a voltage divider. Without this fortunate turn of circumstance, a component of the final design would have been unusable.