Inductive Metal Identification

By

Paul Maggi

Senior Project

Electrical Engineering Department

California Polytechnic State University

San Luis Obispo

June 2012

Table of Contents

List of Tables and Figures

Abstract

This project focuses on creating a device to differentiate between various types of metals via magnetic induction. Magnetic induction is used because it is both non-optical and non-contact. It achieves this by generating a varying magnetic field, which induces eddy currents in the metal. These eddy currents produce small disturbances in the surrounding magnetic field, which can be sensed with Hall Effect Sensors (HES). The primary challenge in this project was generating a sufficiently strong magnetic field to detectable disturbances. In order to achieve better results, a stronger magnetic field must be used as well as more sensitive Hall Effect Sensors should be employed. The current device has two user modes, a continuous read mode and a single-sample read mode.

Acknowledgements

I would like to thank the faculty of Cal Poly’s EE department, especially Dr. Pilkington who advised me on this project and helped me narrow it into a manageable goal. I would also like to thank the wonderful folks at TI, for providing funding for the project. Lastly, I would like to thank my fiancé, as she stood by me even when I spent long hours soldering or writing code.

I: Introduction

Introduction/Overview

The sensor is a device that is able to identify a range of materials based on their interactions with an external magnetic field. This report outlines the basic theory of operation of the sensor, as well as providing background for how it operates and potential uses. The goal of the sensor is to provide a method for material identification that does not require contact with the material, or direct optical access. The device also has two user modes, a continuous scanning mode which outputs the relative strength of the magnetic disturbance generated from the metal, and a one-shot mode which attempts to match the value with a known metal. For this device, it will just feature a small sensing area in which a metal sample can be placed.

Motivation for Project

This project initially started due to my interest in magnetic fields and their interactions with various materials. The project went though several various phases/ideas, ranging from trying to make a small magnetically levitating device, to an array of magnetic sensors in order to image an object. However, due to the time limited scope of senior projects, a smaller, simpler device was settled on. This is the sensor in its current state. It still involves magnetic fields, yet tries to create a device that has a practical application.

Context/Environment in which it will be used

The sensor outlined in this project can be used when metals need to be identified. The two likely environments for use are industrial/manufacturing, and vending machines.

Description of potential customers

While this device may have a narrow range of uses, it can still provide valuable information to certain end users. For example, if a recycling facility that handles various types of materials (such as metals and paper) needed a way to identify and sort objects. This would allow them to more easily identify steel, aluminum, copper, etc, even if the materials were not previously sorted. This potential use would require some scaling of the project, because in its current state, the sensor is a much smaller device than would be required for an industrial operation. Another possible use for this project is metallic integrity tests. A suitably sized and calibrated device operating under this inductive identification principle may be able to detect defects in metals [2]– for example cracks in the skin of an airplane. This is because areas with defects will react differently to more uniform areas. Additionally, devices like this are currently employed in some vending machines to differentiate between metallic slugs and coins. [3] These various uses would require different levels of sensitivity and location identification, along with overall size. My device is a small device for to identify a single sample in one location.

Alternative Solutions

Devices similar to my sensor already exist and are encountered almost daily. The primary example of this is in modern vending machines. Vending machines identify coins not only by size/weight, but also material makeup. This is used to reject metallic slugs, or fake coins that are the correct size. The reason my proposed sensor offers advantages to this, is it includes user feedback and a lookup table of materials. This allows for identification of unknown materials, and provides more information about the material properties. Additionally, if the inductive method of identification is used, it can provide many benefits in terms of scaling accessibility. For example, in a recycling facility if there were a conveyer belt with metallic pieces running through an inductive identifier, it would be able to determine what type of material it was without needing direct physical access, or without interference from things like dirt. Additionally, when compared to optical identification of metals, many metals looks similar, and the processing power and code needed for optical identification is much greater than using several analog signals and some simple logic.

II: Background

Eddy Currents

Eddy currents are small currents that are induced in a metal or conductor that result from the material being exposed to a changing magnetic field. In general, the power loss due to eddy currents without the skin effect can be calculated from the following equation:

$$P = \frac{\pi^2 B_p^2 d^2 f^2}{6k\rho D}$$

Where

P is power dissipation in W/kg

k is 1 for thin sheets, 2 for thin wires

B_p is peak flux density in Tesla

d is the thickness of the material in meters

f is frequency in Hertz

ρ is resistivity

and D is density in kg/m^3

Based off this above equation, if the material size and shape is constrained for testing purposes, then each metal will have a specific ρ*D value. This will allow for determining between different metals.

Instead of measuring power lost directly, the strength of the magnetic field is measured. The strength of the resulting field is $B = \frac{\mu_0 I}{4\pi} \int \frac{dl\ x\ \hat{r}}{r^2}$ from Biot-Savart's Law. The I directly proportional to the power lost via $P = I^2R$.

Also, 1Tesla = 10,000 Gauss.

Helmholtz Coil

Helmholtz Coils are coupled solenoids designed to provide uniform magnetic fields within their interiors. This is due to the radiometric spacing and size of the two solenoids used to generate the magnetic fields.

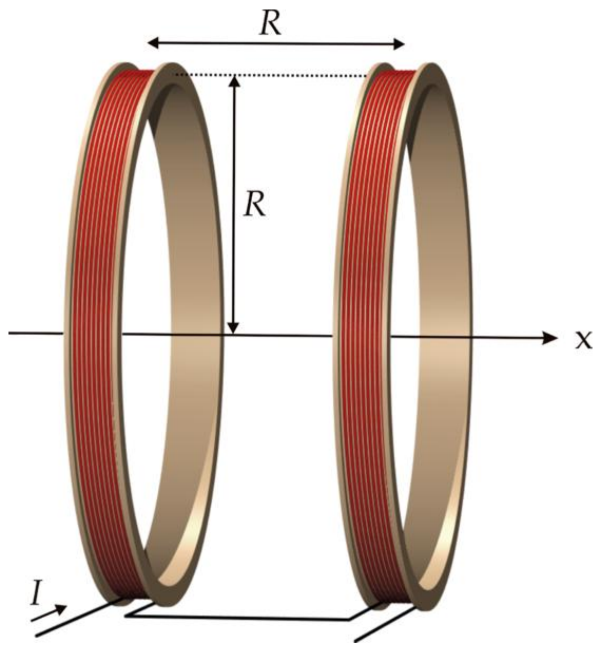

Figure 2.1: Helmholtz Coil Diagram[1]

The governing equation for Helmholtz coils at the center of point between the coils is:

$$B = \left(\frac{4}{5}\right)^{3/2} \frac{\mu_0 n I}{R}$$

Where B is the magnetic field strength in the x direction, μ_0 is the magnetic permeability of a vacuum, n is the number of turns in each coil, I is the current through both coils, and R is the radius, in meters. The Helmholtz coil is the coil structure used to generate the changing magnetic fields for this project.

Hall-Effect Sensor

Hall-Effect Sensors (HES) are magnetic field strength sensors. They act as a transducer, providing a voltage output in relation to an applied magnetic field. HES can either actuate a switch is a magnetic field of a certain strength is detected, or provide information about the strength and the direction of the magnetic field. For this project, a single axis, ratio-metric HES will be used, meaning that the sensor output is proportional to the strength of the magnetic field in a single direction.

Method of Operation

The sensor works by first using a changing magnetic field generated from the Helmholtz coil to induce eddy currents in the material for sensing. The eddy currents generate a magnetic field that is in the opposite direction of the changing magnetic field. This reduced field strength will be detected by the Hall-Effect sensors, which will be amplified, filtered, and fed into the ADC. The MCU will then do a calculation based on this value to determine the ρ*D of the material. This data will be output to the LCD screen where the user can easily read it.

III: Requirements

Functional Requirements:

The MMI must be able to achieve the following:

- Be able to detect an object within the sensing area
- Be able to measure the changed magnetic field due to eddy currents.
- Be able to identify the type of material based on the changed field strength.
- Be able to provide user feedback about the previously mentioned functions a small LCD screen.
- Must include a calibration function.
- It will also have a one-shot identification function, and a free-running function.
- Indicate the mode of operation via 2 LEDs.

Performance Specifications:

- Coil Specifications:
  Given an R of 4.5cm (0.045m), 15 loops of wire, and a peak current of 1.2A, the expected max magnetic field is 3.6G. Additionally, the field should be relatively uniform within the sensing areas, in that a 2cm radial movement or a 1cm axial movement should change the field strength by less than 10%.
- Hall-Effect Sensor Specs:
  Given the sensitivity of 5mV/G in the Hall-Effect Sensor, this should generate a voltage difference of 18mV at full B-Field. Additionally, depending on the direction of the field, the voltage difference will either be positive or negative. If the HES is turned off axis, it should read the no field value.
- ADC Specifications:
  V_{CC} = 3V, with a 10bit ADC, this gives a resolution of 2.9mV, and a max quantization error of 1.45mV. The max sampling rate is 200ksps, this gives a max operating frequency of 100KHz. The ADC will be sampling at 5kHz.
- Function Generator IC and High Current Buffer :
  The function generator and high current buffer must be able to supply four different waveforms to the coil: a sine wave (0V DC offset), a square wave, a triangle wave, and a DC value. The maximum current that will be supplied is 1.2A.
- LCD Screen:
  The LCD screen should have two primary display modes: free running and one shot. Both modes will have two lines of text. In the free running mode, the top line will say “Free Run Mode”, the bottom line will output the measured value. In the one shot mode, the top line with display “One Shot Mode” and the lower line with display the supposed material type.

IV: Design Approach Alternatives

The primary critical subsystems are the function generator, the high current buffer, the coil winding, and the Hall Effect Sensor. The other systems were fairly straightforward as will be explained.

For the function generator connected feeding into the high current buffer, several options were available. The options were to use an external function generator, to create a simple function generator out of discrete components such as op-amps, and to find a function generator IC. Using an external function generator needlessly complicates the design in that it makes it removes a level of autonomy to the project. While having an external function generator may provide a wider range of options, it is not needed and the trade off of actually requiring an expensive, external piece of equipment was prohibitive. With regards to using the discrete component option which was used for the project demonstration, it allows for similar functionality as compared to the function generator IC, but is less compact. It would require several ICs and passives, and not allow for the simplicity of the single function generator IC. Depending on the complexity of the design, it could match performance of a full function generator IC. The XR2206 Function Generator IC was initially selected because it provides the three main waveforms (sine, square, and triangle) in one simple package, and only required minimal external components. This allows for compactness and simplicity that is not as achievable with the other design alternatives.

With regards to the high current buffer, there were several options: finding a single high current op-amp or creating a high current op-amp using a normal current op-amp and several transistors. While the transistor approach (see figure below) was viable and produced viable simulations (see figure below), it required the use of additional transistors. By finding the OPA-561 high current op-amp and configuring it as a standard non-inverting buffer, it would be able to provide a 1.2A current limit without the need of additional components.

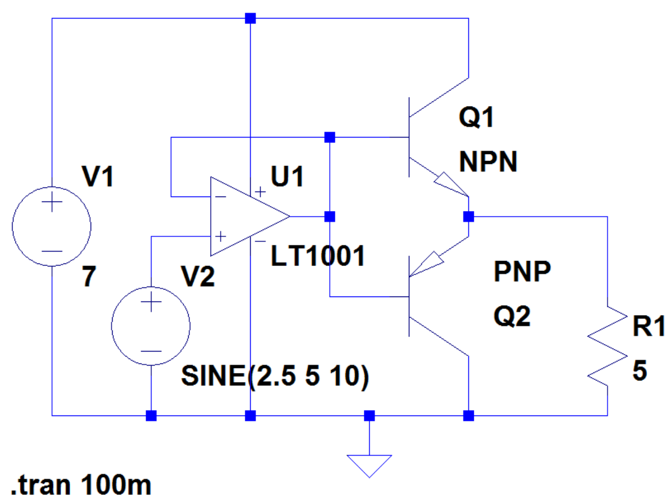

Figure 4.1: High Current Buffer using Discrete Transistors, Schematic

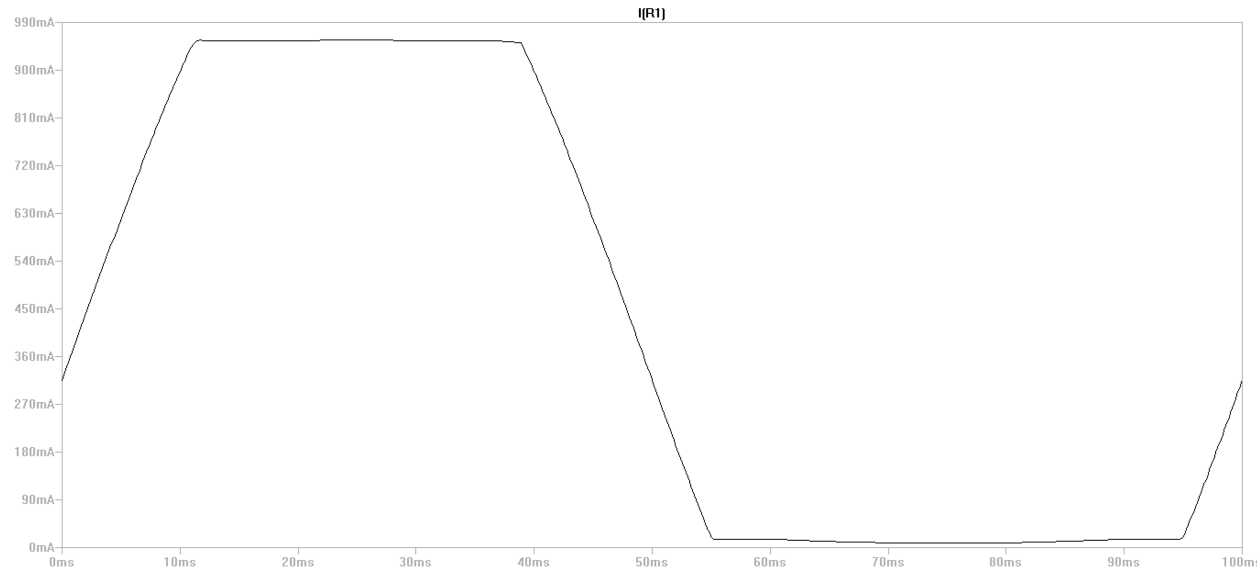

Figure 4.2: Resistor Current in High Current Buffer from Figure 4.1.

With regards to the coils and the coil supply, I could have used a single solenoid to generate the coil. However, this would cause more signal variability with regards to where the object to be sensed is placed within the coil. Additionally, it would decrease ease of use as the object would need to be placed within the windings. With the Helmholtz coils, not only is there a very uniform magnetic field, but it also allows for easier access to the sensing field, as it is an entirely open space. The one disadvantage of the Helmholtz Coil is that it has size dependence, and a small field reduction coefficient of 0.716. A normal air core solenoid produces a magnetic field of $B=\mu_0 n*I$, while a Helmholtz Coil produces a field of $B=0.716\ \mu_0 n*I/R$, where R is the radius in meters. However, this means that for small coil radius, there is an amplification effect of the field, which is not present in solenoids. This led to using the Helmholtz Coil with a radius of R=5cm, which gives a field strength of 14.3 times larger than a solenoid with the same current and number of windings.

When deciding on a sensing method, several options were available. The two primary options considered were basic wire loops and HES. The basic wire loops are the simplest solution to measure changing magnetic fields (due to the induced EMF), but they do not provide the level of accuracy required for this project. Additionally, in order to have sizeable signals induced in the wire, there would need to be stronger fields, more windings, or higher frequencies. These disadvantages resulted in picking a simple, 1-axis Hall-Effect Sensor, that will give a factory calibrated output based on the input field. This allows for easier signal processing and analysis.

With regards to the signal amplification stage, I could have used a single HES and an amplifier/filter, however this would have reduced the accuracy of the data. By using two HES and using an instrumentation amplifier, it not only helps reduce noise, but it also allows for more accurate signal

measurements. Based on the changing field directions, both HES should read different values, and the difference allows for more information to be obtained. By feeding the two HES outputs into the instrumentation amplifier, it eliminates the common DC component in both devices. The INA333 instrumentation amplifier was used because it provides differential inputs as well as gain setting with a single resistor.

For the Microprocessor unit (MCU), the MSP430G2452 was used because it provided all of the necessary features, such as a 10bit analog to digital convertor (ADC) and 10 general purpose input/output pins with interrupts. It was also offered in a DIP20 package.

The 2 line x 16 character I2C LCD screen was selected because it offered enough display space for the desired text. The I2C interface was selected instead of full parallel because it only requires 3 interface pins instead of 9. An alternative approach would have been to use a computer interface, though that would have required using an additional communications protocol to the computer as well as developing a GUI. The 2 line LCD provided the needed functionality, as well as reduced the code size/complexity, while allowing for additional portability.

V: Project Design

Definition of User Interface

The user will interact with the device through the use of one SPDT switch, and a SPST momentary switch. They will receive feedback from the device via the LCD screen and two LEDs to indicate the mode. There are two modes of operation, free running mode and a one-shot mode as well as a calibration function.

Free running mode is selected by setting the SPDT switch to connect to the 3V rail. This will be indicated by the green LED being turned on, and the red LED should be off. In free running mode, the LCD screen will continuously display the output value of the ADC. Additionally, when in free running mode, if the user presses the SPST switch, the device will run a calibration. When calibrating, there should be no object in the sensor field.

One-shot mode is selected by setting the SPDT switch to GND. This will turn off the green LED and turn on the red LED. In this mode, when the SPST switch is pressed, the device will take a reading, and attempt to identify the type of material that is present in the sensing area. On the LCD screen, it will display the material type.

One the coil winding assembly, there will be a slot where the user will insert the sample to be tested.

Blackbox Diagram

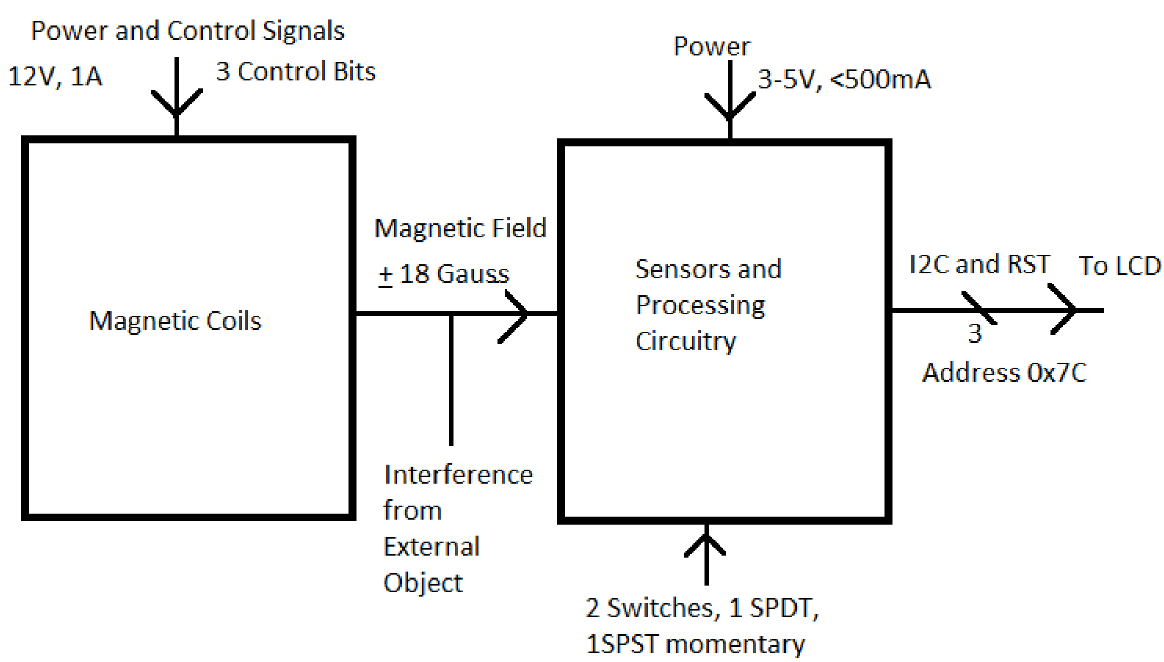

Figure 5.1: Overall Project Black Box Diagram

Overall System Functional Block Diagram

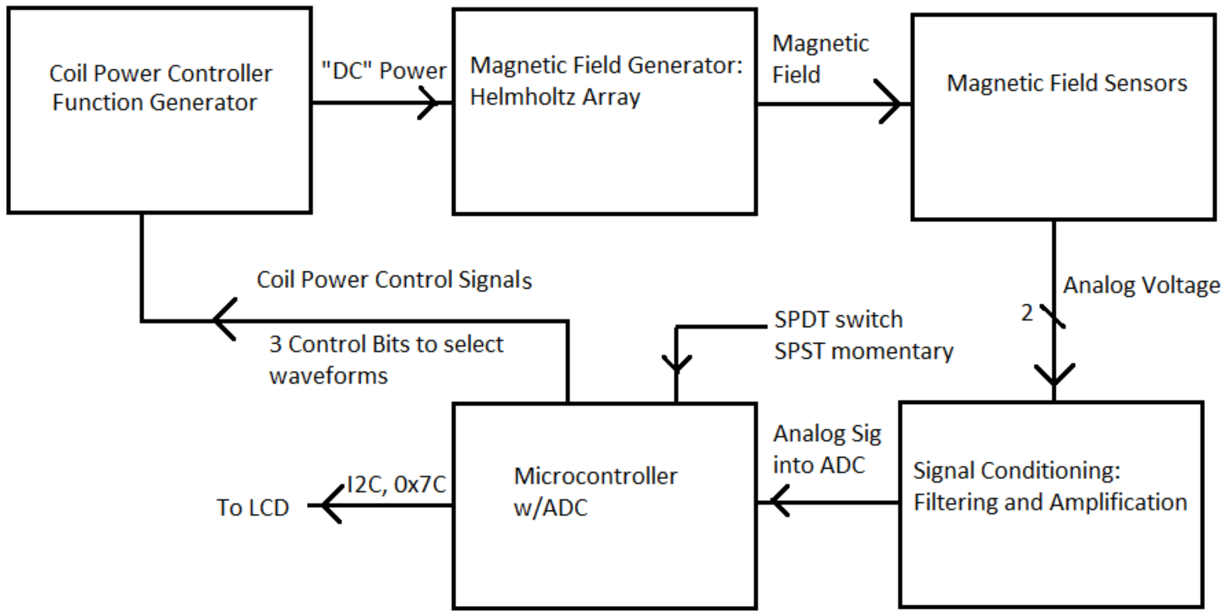

Figure 5.2: Functional Block Diagram

Subsystem Diagrams

<u>Coil Power Supply/Controller</u>

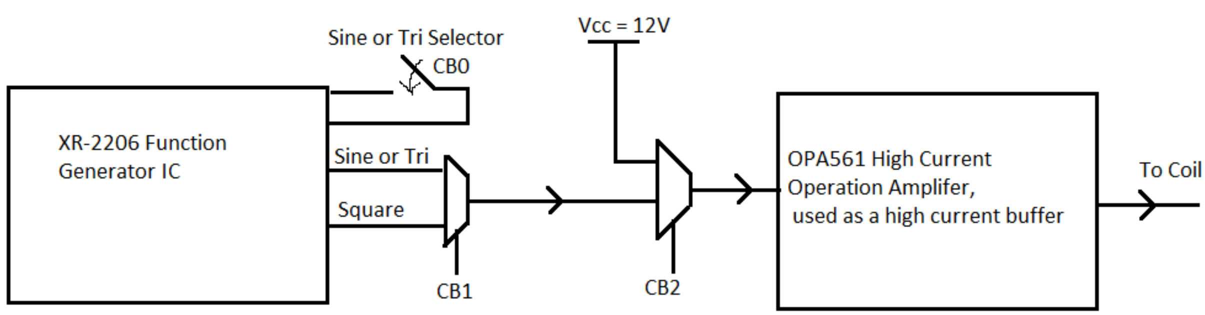

Figure 5.3: Coil Power Supply/Controller

The OPA561 will be supplied by a split $\pm$5.5V supply.

CB0, CB1, CB2 connect to the MCU. The different selectors are2:1 analog muxes, part TS5A23157, with bandwidth of 200MHz.

Model of Helmholtz Coils

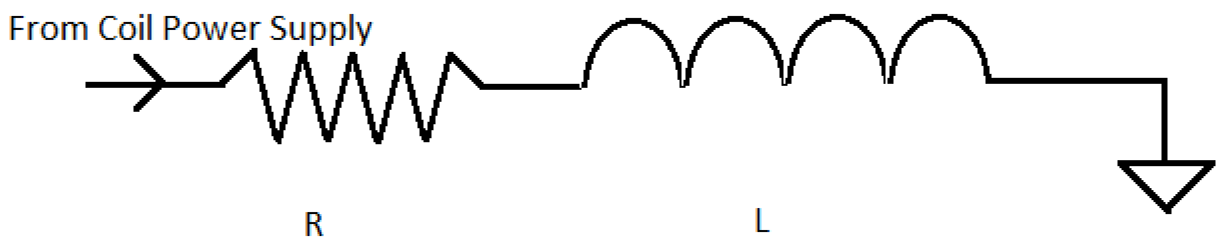

Figure 5.4: Helmholtz Coil Model

The coil is modeled as a resistance R in series with an inductance L. Given 10 turns of 21 gauge wire at a radius of 5cm, R is roughly .3Ω. Additional resistance may be added. This configuration also gives and inductance of roughly 70μH.

Hall-Effect Sensors (HES)

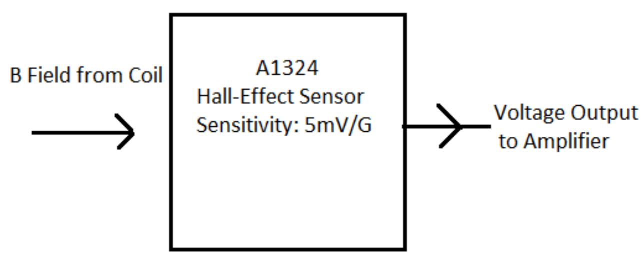

Figure 5.5: HES Model

There are 2 HES's used, one on each side of the sensing area.

Signal Conditioning

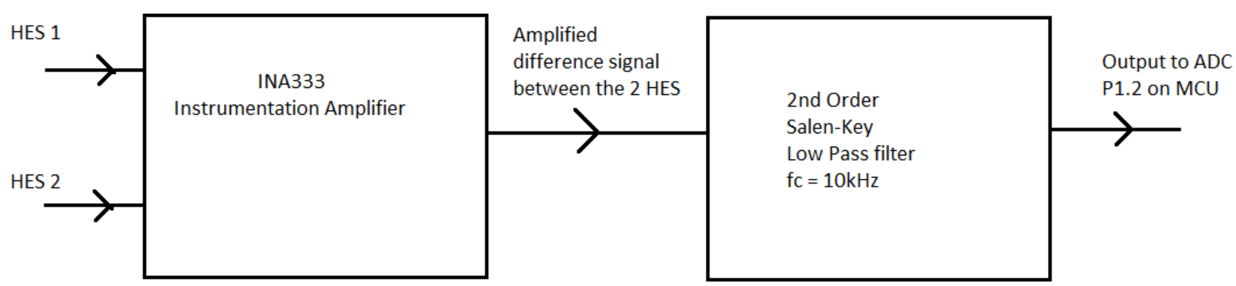

Figure 5.6: Signal Conditioning Model

HES 1 and HES2 denote the two Hall-Effect Sensors. The instrumentation amp will have a voltage gain of 40dB, or 100V/V. The amplified difference signal feeds into the filter before the ADC.

MCU and LCD screen

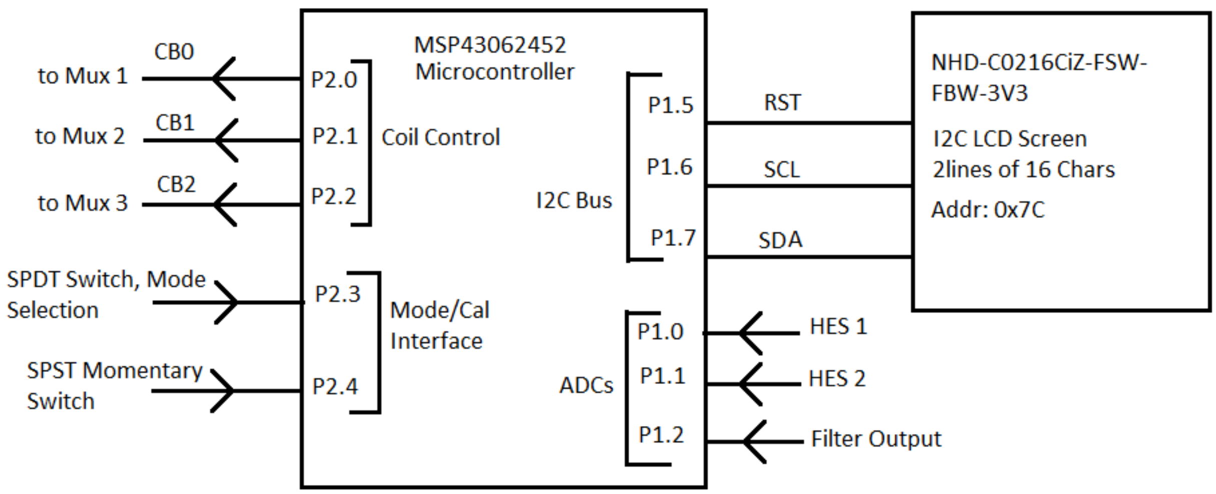

Figure 5.7: MCU and LCD Diagram

The MCU has 4 primary functions: 1) select what type of waveform the Helmholtz coil is receiving; 2) receive input through two different switches; 3) monitor the output of the sensors and filter output; 4) provide user feedback via the LCD screen over the I^2C bus. The MCU will contained the firmware for the project, written in C.

Function 1 is accomplished through interfacing to 3 analog 2:1 muxes, controlled through 3 GPIO pins
Function 2 is accomplished by reading the values of the 2 switches, which are connected to GPIO pins. Both GPIO pins connected to the switches will be interrupt-transition driven on state changes.
Function 3 is accomplished by reading the values of the 3 devices connected to the ADC pins. It will then process this information and determine the permeability value and/or material type.
Function 4 is accomplished by feeding the necessary information calculated after the ADC conversions to the LCD screen via I^2C.

SPDT Switch and Mode Indicator LEDs

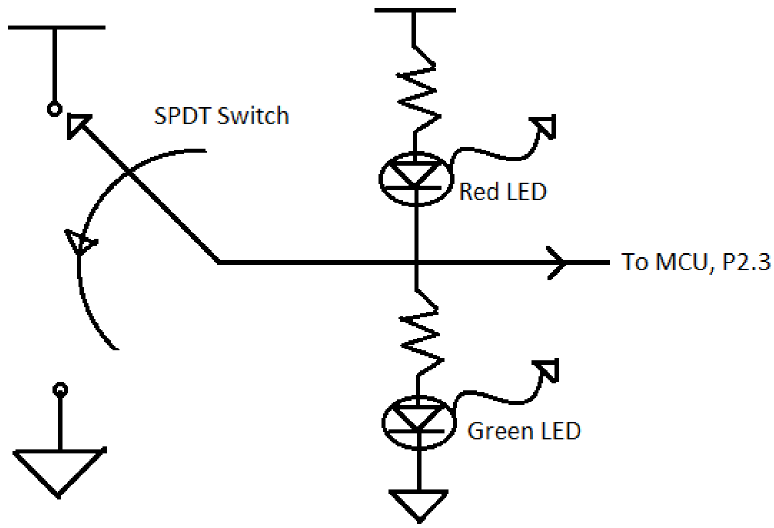

Figure 5.8: Mode Switch Model

If the switch is connected to 3V, the MCU will a high voltage on P2.3. This will set the sensor mode to continuous read mode. Additionally, the Green LED will be turned on. If the switch is connected to GND, the MCU receive a 0, and will set the operation to one-shot read mode. The red LED will also be turned on, indicating one-shot mode.

Critical System Parameter Selections / Settings

$f_{function\_gen}$ = 2kHz; function generator frequency feeding into the coil system.

f_{ADC} = 5kHz; ADC sampling rate

<u>Anti-aliasing Filter parameters (revised):</u>

Type: Lowpass, Sallen Key, Butterworth, 2nd order

20dB gain, 1dB max passband ripple, f_c= 10kHz

<u>Coil Specifications:</u>

r_{coil} = 4.5cm, at n (number of windings) = 15, total wire length is roughly 10m. Using 20 gauge wire, this gives R_{coil} = 0.35Ω. I_{coil} max = 1.2A, since Z_{coil} < 1, the output voltage will be relatively low.

Signal Amplification:

The two HES will feed into an instrumentation amplifier with a voltage gain of 40dB.

MCU settings:

The MCU will use 5 GPIO pins, 2 inputs, 3 outputs. It will also use 3 I^2C pins, and 3 ADC pins. For the ADC, it will use the internal reference of V_{CC} = 3V. The LCD screen is at I^2C address 0x7C.

I/O Map

Pin Number/Name	Input/Output	Function
P1.0	Input	ADC Input from HES1
P1.1	Input	ADC Input from HES2
P1.2	Input	ADC Input from Anti-aliasing Filter
P1.5	Output	RST pin of the LCD screen
P1.6	Output	SCL (I^2C clock) pin, connected to LCD screen
P1.7	Bidirectional	SDA (I^2C data) pin, connected to LCD Screen
P2.0	Output	CB0, selects between Sine or Triangle wave on function gen
P2.1	Output	CB1, selects between sine/tri or square wave on function gen
P2.2	Output	CB2, selects between previous wave or DC on function gen
P2.3	Input	Input from SPDT mode selection switch
P2.4	Input	Input from SPST momentary multipurpose switch

Table 5.1: MCU I/O Map

VI: Physical Construction and Integration

Physical Implementation & Packaging Plan (revised)

Circuit Board Concept:

The circuit board will ideally be a printed circuit board to reduce noise. There will be two primary circuit boards: the MCU main board, and a secondary coil power board. The MCU board will contain the MCU, LCD screen, switches, and signal conditioning circuitry. The coil power board will contain the muxes, the function generator and the high current buffer stage. The two HES will be independently mounted within the coils. Both boards will be 2 layer boards, with a mix of through-hole and surface mount components.

Enclosures :

Even though the coils will have 1A of current, they should not need any additional cooling, for the power dissipation is low. The only specific enclosure and mounting considerations are the coils with the HES, and the coil supply. The HES must be mounted in such a way that their sensing field is oriented correctly with regards to the magnetic field. Additionally, the wires connected to the HES should run parallel to the magnetic field to reduce noise induced by the fields.

Power:

This device with have 3 separate power supplies at 4 separate voltages: 3V, 5V and $\pm$7V. The $\pm$5.5V rails will be supplied by a split DC supply, while the 3V and 5V rails will be provided by separate LM317 adjustable voltage regulators. The 3V supply will power the MCU, LCD screen and LEDs; the 5V supply will power the HES, the instrumentation amplifier and the muxes; the $\pm$5.5V supply will power the function generator, buffer, and filter.

Hardware Configuration and Layout:

The following is a rough depiction of the hardware layout and arrangement of the aforementioned boards and components.

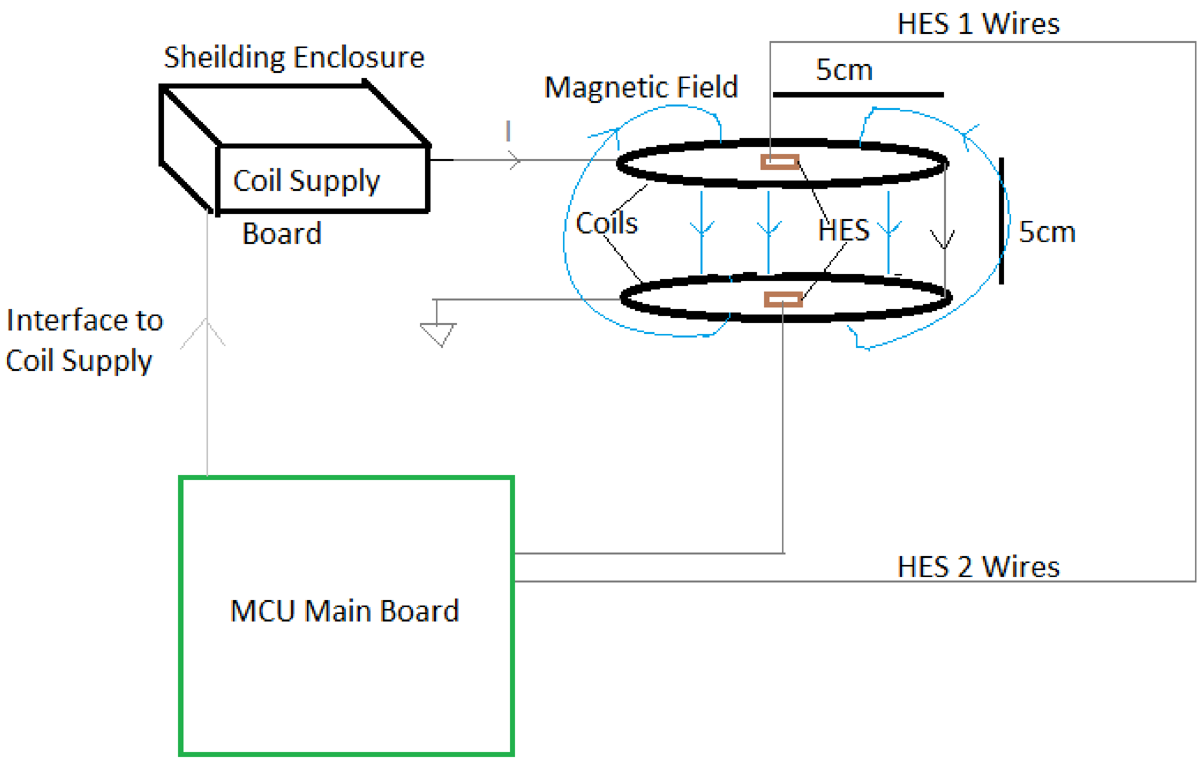

Figure 6.1: Physical Layout and Interconnects

The following pictures depict the final form of the project.

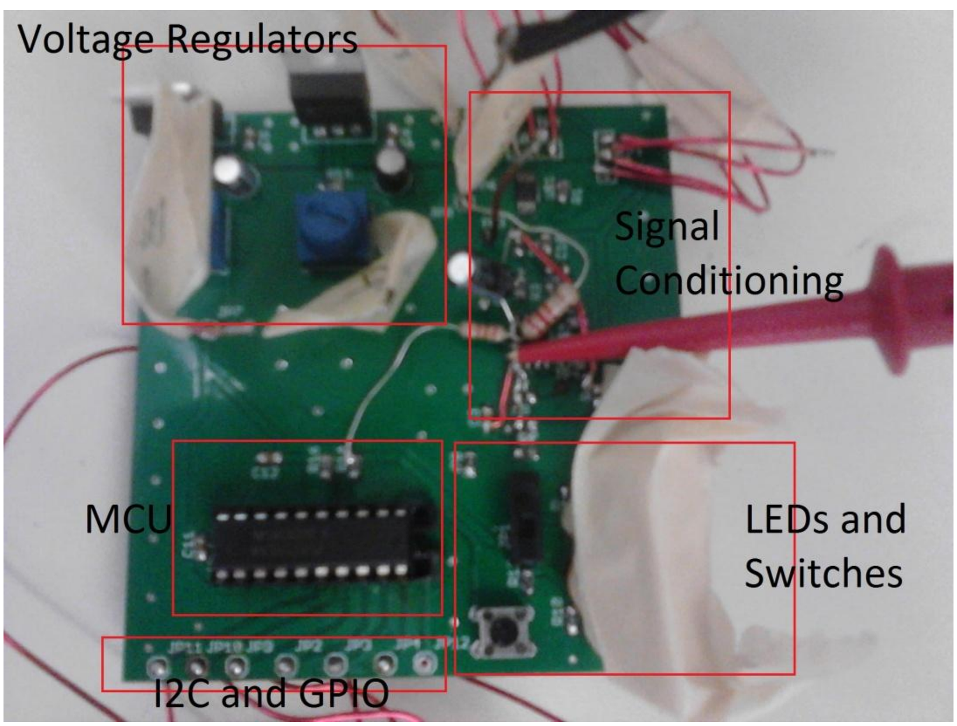

Figure 6.2: Main Board Picture

Figure 6.3: LCD Screen, connected at the bottom of the Main Board

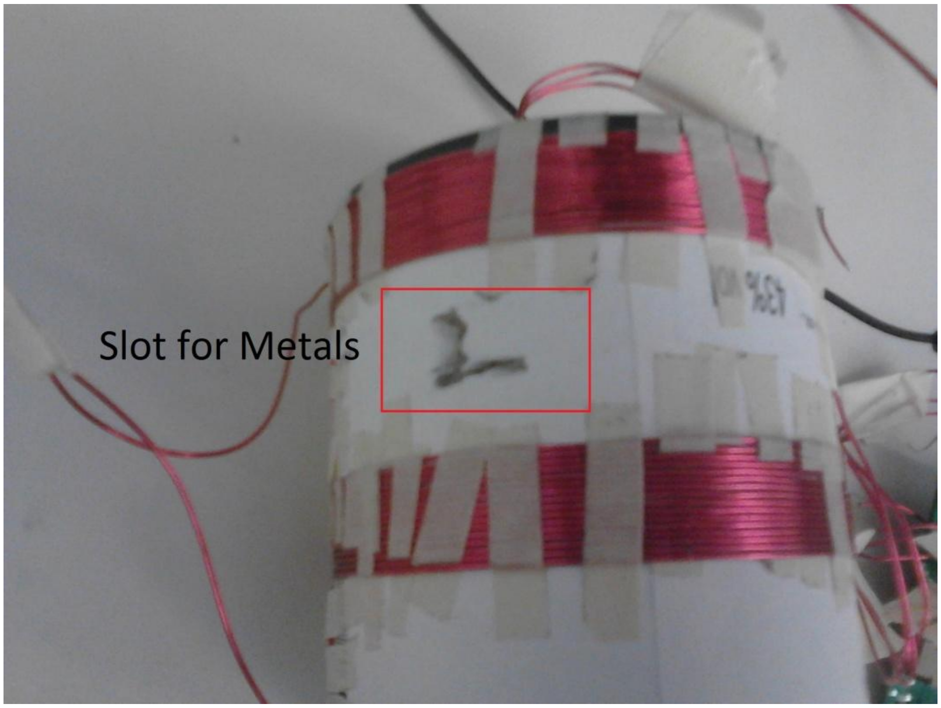

Figure 6.4: Helmholtz Coil, the slot on the left is for inserting the metal sample

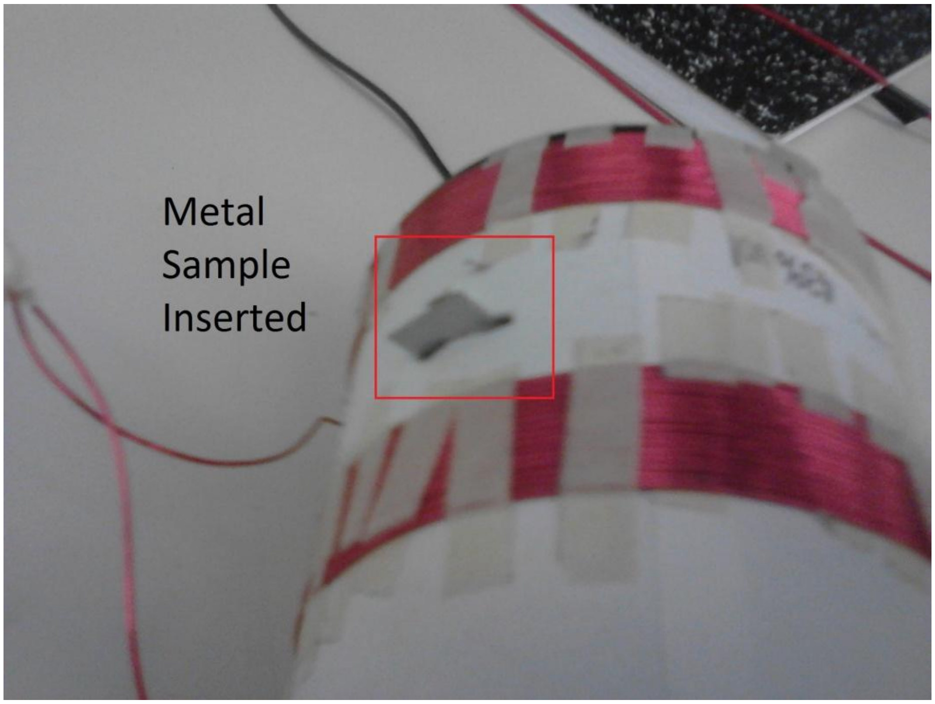

Figure 6.5: Example of Coil with Metal Sample in Place

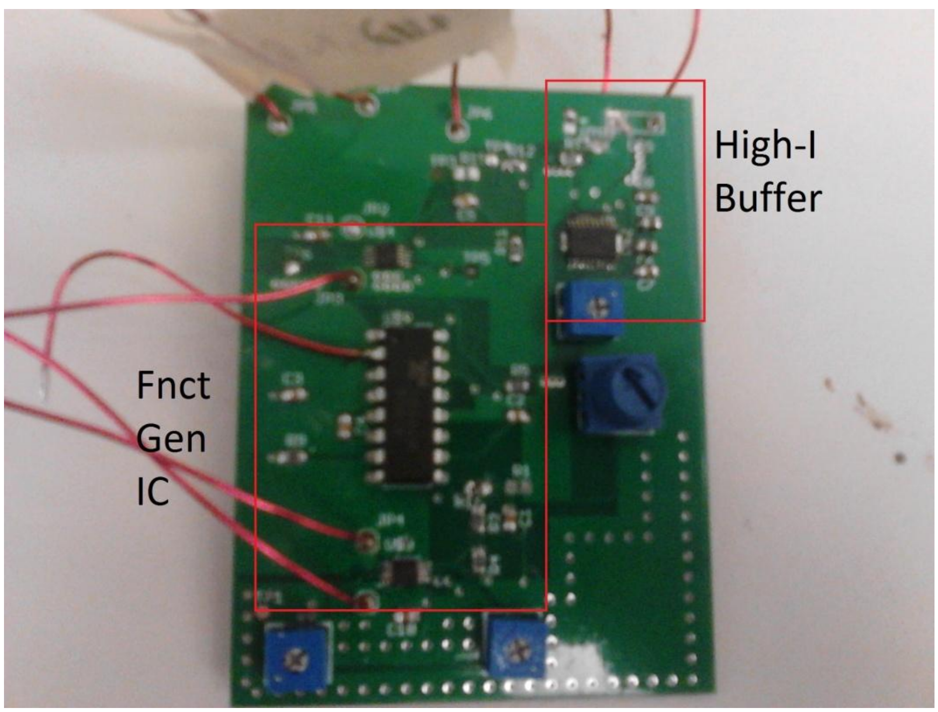

Figure 6.6: Coil Power Board

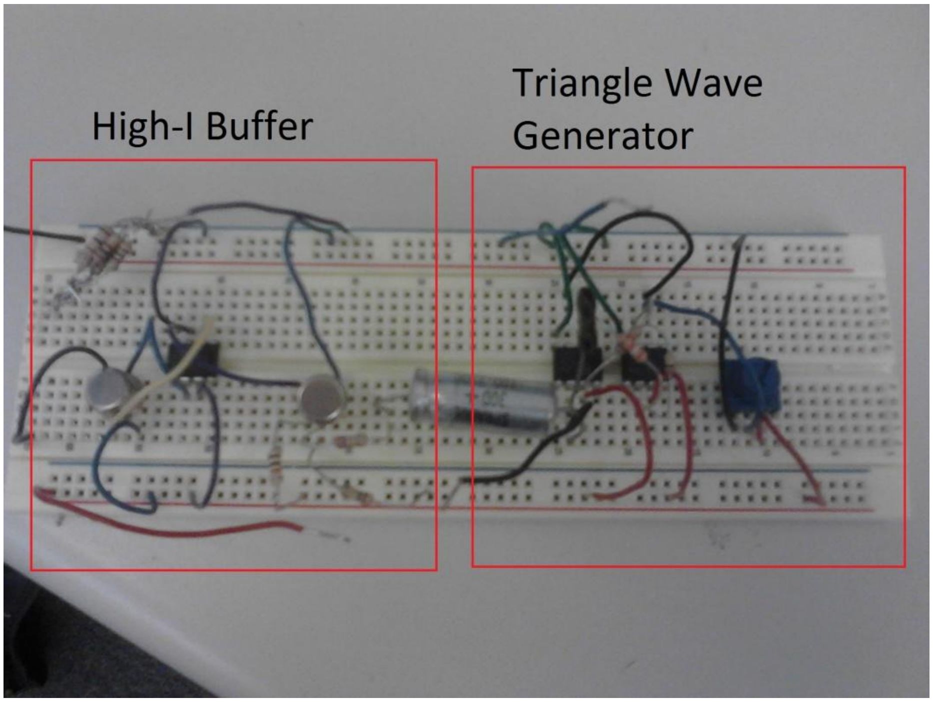

Figure 6.7: Auxiliary Coil Power Board

VII: Integrated System and Test Results

Coil Tests and HES tests:

The designed specification for the coil was a peak magenetic field strength of 3.6G at 1.2A. The coil had a R = 0.08m, and was supplied 0.48V and 1.198A. The HES was powered with 5.0VDC and 0.008A. In air, the output of the HES was 2.5242 V. When inserted into the coil with the positive sensing direction facing up, the output voltage was 2.5617 V, for a change of 17.4 mV. When the current direction in the coil was reversed, the output voltage was 2.5063 V, for a change of -17.5 mV. These two numbers suggest that the HES is working properly, and with the manufacturers' specified sensitivity of 5mV/G, the field strength inside the coil was 3.5G, which is a 2.8% error from the expected value. Additionally, the coil resistance was measured to be 0.5Ω, which was slightly higher than the expected 0.35Ω. However, this difference is negligible because a 5Ω load was added to the coil to increase voltage.

Instrumentation Amplifier

The instrumentation amplifier was tested by providing a 1.5V input to each terminal, then varying one of the voltages. For a 1.51V input, the expected output should have been (1.51-1.50) * 100 = 1V. The actual output value was 0.97V. For an input difference of 0.02V, the expected output was 2V, while the measured output was 1.95V. These two points suggest an actual gain of 97, slightly less than the 100 that it was designed for, though still within tolerance using a 5% resistor to set the gain level.

Function Generator IC

The function generator IC was not operating properly when used with the mux controls from the MCU. The ideal operation was that the muxes would be controlled using the MCU, which would in turn change the output of the function generator according to the following chart:

CB0	CB1	CB2	Output Waveform
0	0	0	Sine
1	0	0	Triangle
-	1	0	Square
-	-	1	DC

Table 7.1: Function Generator Output Table

However, due to limitations with the power supplies and grounding issues, the function generator was not working. This is because the coil board required ±5.5V rails and a current limit of 1.2A, while the main board was supplied from a 7V rail. The two power supplies that were used for testing were only able to supply +6V/2.5A and ±20V/0.5A. This meant that to achieve the proper power rails, both 6V outputs would be used, though the -5.5V rail would actually be the common port on the second generator. This then created an instance where different grounds were present for different boards, rendering then control signals unusable. The figure below depicts this.

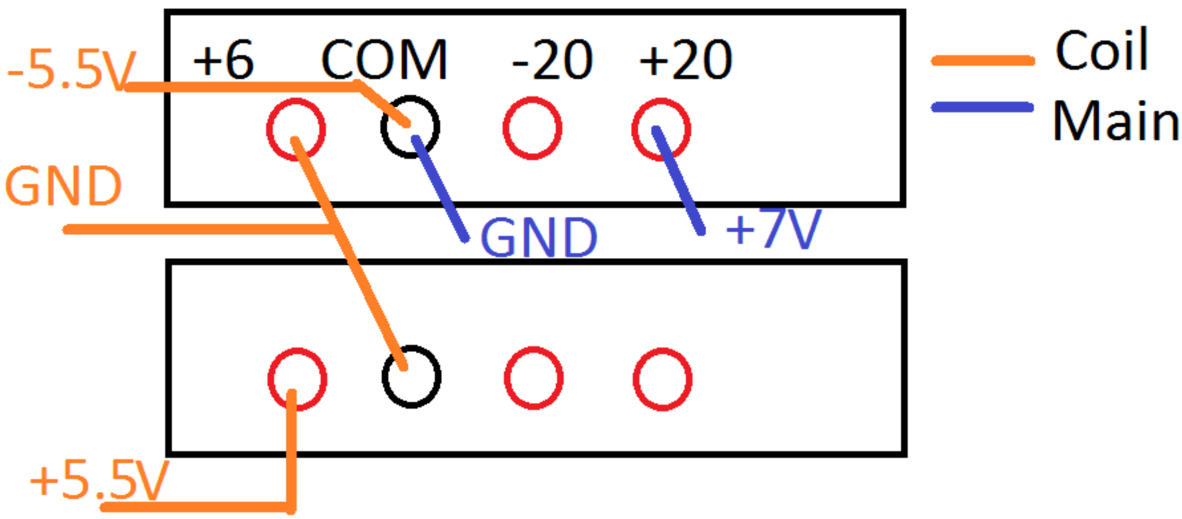

Figure 7.1: Power Supply Scheme

As a result of this issue, an auxiliary waveform function generator and high current buffer was used. Instead of having the ability to output the different wave forms, only a triangle wave was necessary. The triangle wave generator was able to be powered off a +7V rail so that grounding would not be an issue. Before being fed into the high current buffer stage, the triangle wave was amplified, which produced the clipping seen in the image below. This is not an issue because the exact waveform does not determine the response, it only matters that there is a constant rate of change. The first image shows with the clipped triangle wave and the output of HES1 without connecting the magnetic coil. The second image shows the same waves, only where the power is being delivered two the coil. The green line is the function generator output, the yellow is the HES output.

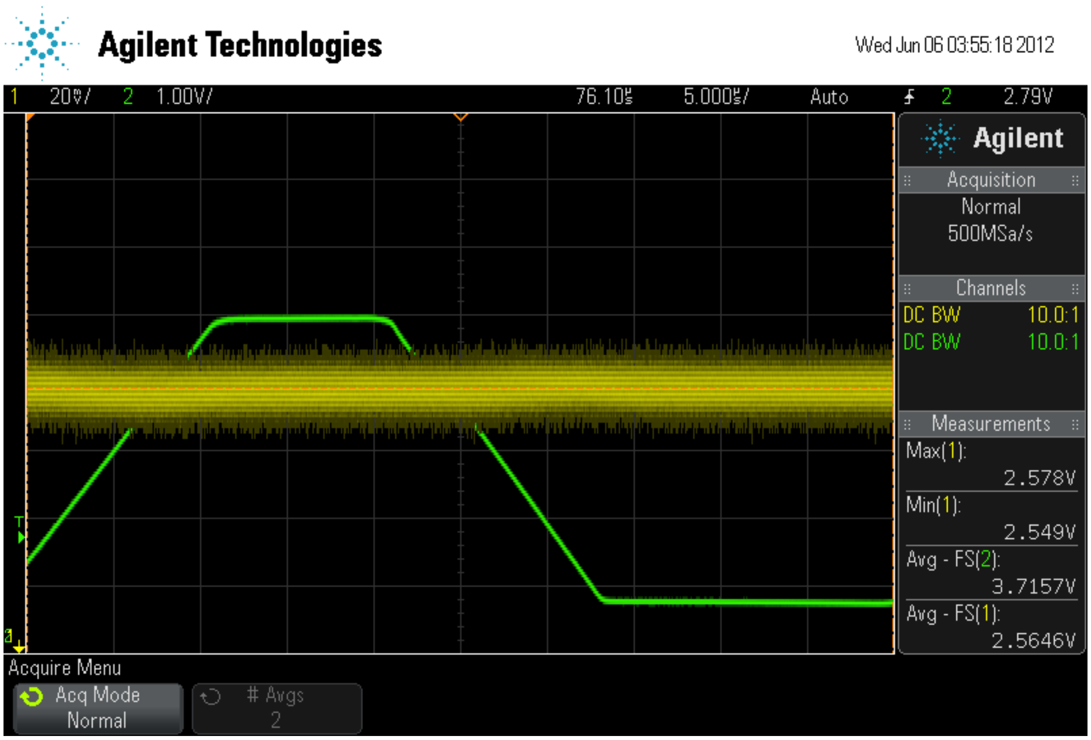

Figure 7.2: Triangle Wave and HES, coil not connected

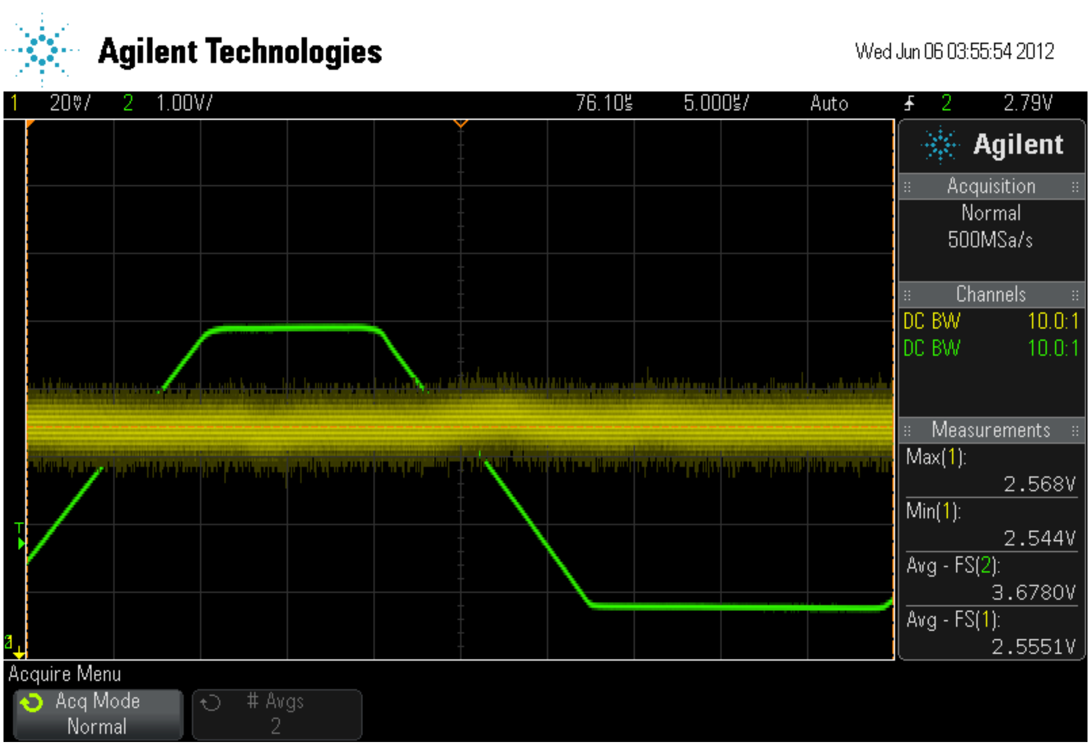

Figure 7.3: Triangle Wave and HES, coil connected

Comparing the two images shows the following:

The total coil resistance was measured to be 5.5Ω (0.5 Ω from the magnet wire plus and additional 5 Ω). This means that with the average (DC) voltage of 3.7 as shown in the plots, the current in the coil is I=V/R = 3.7V / 5.5Ω = 0.67A. This would generate a magnetic field of 2.05G. The average voltage change for the HES was 2.565V – 2.555V = 10mV. This would give a field strength of 2G at 5mV/G sensitivity. The expected valued based off the current and the value measured agreed within 2.5%.

The image below also shows that the HES is able to respond to subtle changes in the voltage. The green line is the input into the coil, and the yellow is the HES output. It should be noted that when there are large curerent spikes at 27µs and 78µs, the voltage on the HES drops. This indicates that the HES should be able to respond to a distrubace in the magnetic field generated by eddy currents.

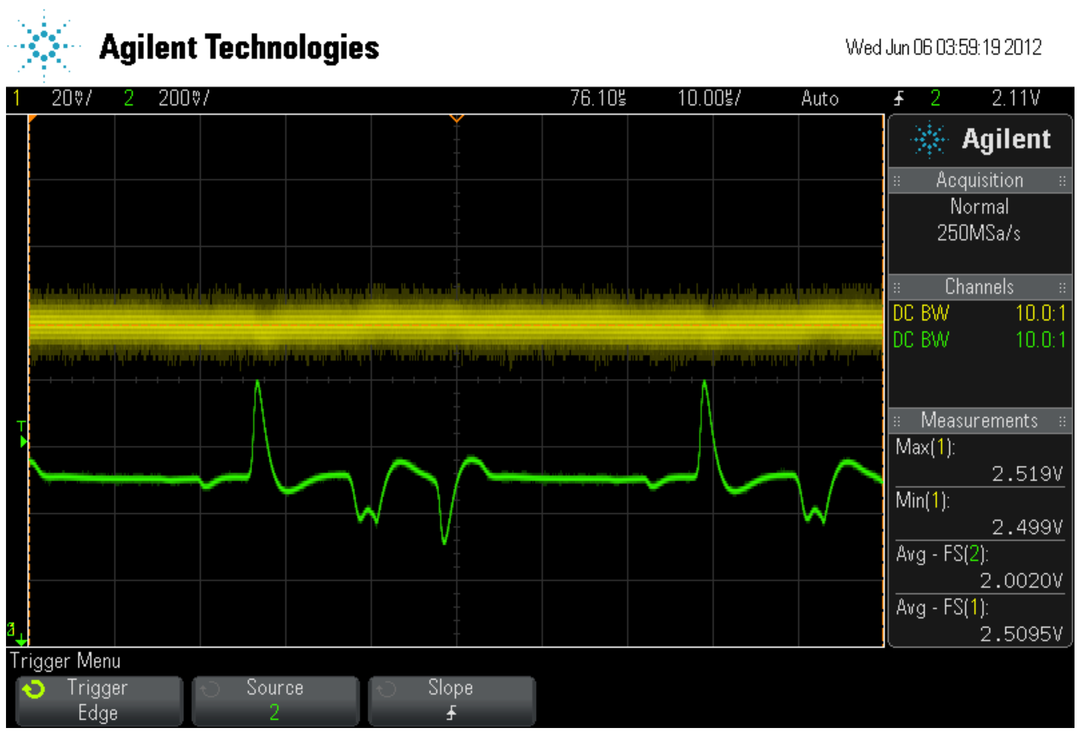

Figure 7.4: HES Disturbance Example

MCU and User Interface Functionality

The user interface should operate according to the following input table:

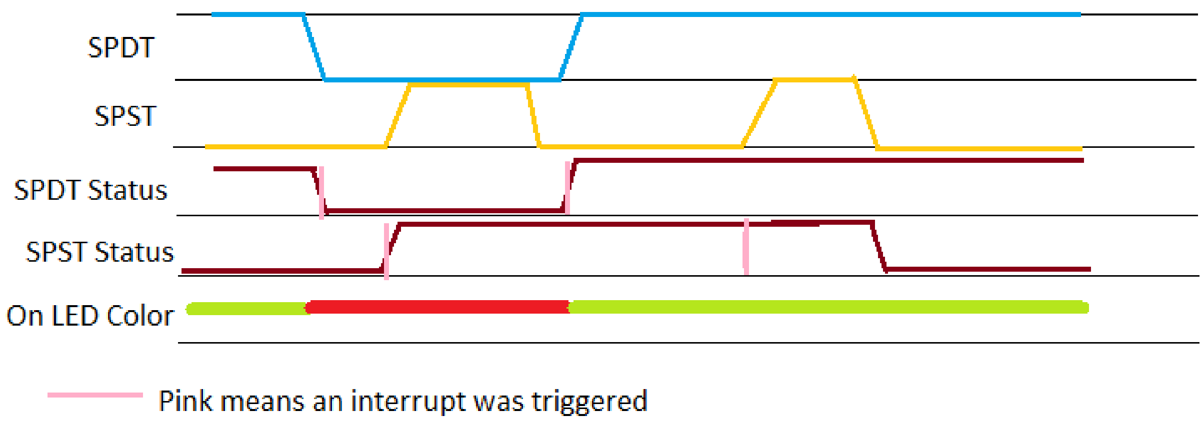

Figure 7.5: Switch Position and Interrupt Diagram

When the LED is green the text should read “Free Run Mode” on the top line, and display the value on the second line. When the LED is red, the text should read “One Shot Mode” on the top line, and display the type of metal on the second line. Additionally, the above table indicates that interrupts are triggered only during mode toggles, and when the momentary switch is pressed, not released.

In addition to the different operating modes, a calibration function was included. This is due to temperature variability of the HES. When the device is first powered on, it will require the user to run a calibration before it begins normal operation. This is done by pressing the momentary switch while in the free run mode. If the user tries to use the one show mode before calibration is run, the LCD screen will tell the user to run the calibration in free run mode. Additionally, the calibration is run whenever the button is pressed while in free run mode. The following images depict the possible messages.

Figure 7.6: One Shot Mode, before calibration

Figure 7.7: Free Run Mode, before calibration

Figure 7.8: Free Run Mode, after calibration

Figure 7.9: One Shot Mode, after calibration, no metal

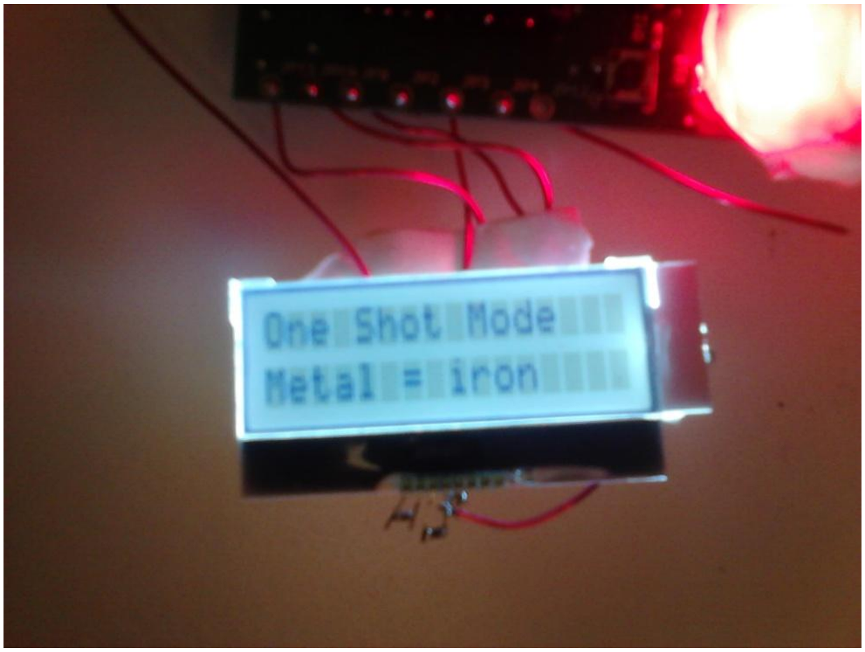

Figure 7.10: One Shot Mode, after calibration, iron example

ADC and Sampling

The ADC is designed to sample during one half cycle of the waveform, and isolate the largest data point. The reasoning behind this is that the peak value determines the disturbance level, not the overall average. This was done by initially setting the max value to 0, and then when the ADC samples, compare the new value to the old one. If the new value is a greater magnitude away from the current max, then it becomes the new max. This was tested by setting the ADC to do a 2s sample run, sampling at 5kHz. A function generator was connected to the ADC channel and set to output a triangle wave at Vmax = 2V, Vmin = 1V, f = 2kHz. The anticipated output value with a 3V AREF would be 2/3 * 1023 = 682. The actual output value was 680.

The ADC should have a resolution of 3/1023 = 2.9mV and a max quantization error of 1.45mV. As the previous example demonstrates, there was a sample difference of 5.8mV. A 5.8mV difference may result in misidentification of materials.

Metal Identification

There should be two methods of metal identification, one by looking at how the value of free run mode changes when a metal is inserted into the testing area, or by using the one shot mode for matching to a lookup table. However, the current device is not able to identify metals. The following table illustrates some sample calculations for signal values based on three different metals.

	iron	silver	copper
P (W/kg)	3.30E-02	1.02E+00	1.14E+00
Bp (T)	3.60E-04	3.60E-04	3.60E-04
d (m)	2.00E-03	2.00E-03	2.00E-03
f (Hz)	2.00E+04	2.00E+04	2.00E+04
k	2.00E+00	2.00E+00	2.00E+00
p (ohm*m)	6.90E-07	1.59E-08	1.68E-08
D (kg/m^3)	7.50E+03	1.05E+04	8.94E+03
pD (ohm kg/m^2)	5.17E-03	1.67E-04	1.50E-04

u0 (N*A^-2)	1.26E-06	1.26E-06	1.26E-06
r(m)	2.00E-02	2.00E-02	2.00E-02
I (A)	2.70E-08	2.28E-08	2.47E-08
Volume (m^3)	1.60E-06	1.60E-06	1.60E-06
P*V (W)	5.27E-08	1.64E-06	1.82E-06
pd	1.38E-08	3.18E-10	3.36E-10
B (tesla)	4.24E-11	3.58E-11	3.88E-11
B (Gauss)	4.24E-07	3.58E-07	3.88E-07

Response (V)	0.0212	0.0179	0.0194

Table 7.2: Signal Strength Calculations

These calculations were performed using the formula for eddy current power dissipation (top box)and then the magnetic field strength resulting from an electric current (bottom box). These equations can be found on pages and respectively. While the final response is within the realm of identification for the ADC as the values are greater than 2.9mV apart, the issue lies in the initial response B (Gauss). This is because while the final response includes the gain of the amplifier, LPF, and conversion from G to mV, the initial response as generated from the HES is minute. The SNR was not great enough at the input to the amplifier. Possible solutions will be discussed in the next section.

However, the sampling and identification methods while in both modes of operation still functioned. This was done by verifying that if the input to the ADC was connected directly to a power supply, it would give the correct response. In free run mode, as the power supply was varied, the value displayed on the screen also changed. In order to test one shot identification, some arbitrary values were assigned: 100 for iron and 200 for silver. This means that if the difference from the calibrated value is between 0 and 100, then one device would read none (for no metal), between 100 and 200 it would read iron, and greater than 200 would read silver. With the calibration to 1.5V, a 100 variation would be (3/1023)*100 = 0.29V. When the supply was set to 2V or 1V and a measurement was taken, the device output iron (as seen in figure above). Above 2.1V or below 0.9V, the device read silver. This is the correct operation behavior.

VIII: Conclusions

The inductive metal identifier was not able to successfully identify metals. However, the individual components were able to function as it was intended.

The reason the device did not work was because of the low sensitivity of the HES. Based on the current constraints of the design such as power and size, creating a working device may not have been feasible with the available resources. In order to reach usable signal levels, the magnetic disturbance created by the metals would need to be at least 1000 times great. The use of an overall gain of 1000 did not generate usable signals due to the initial signals not being above the noise floor. Luckily, there are several variables that can be changed in order to increase the SNR.

First, the coil and coil power source can be redesigned. The three variables that are easily controllable are the number of windings, the current in the coil, and the size of the coil. Increasing the number of windings or increasing the current will increase the strength of the magnetic field. However, in order to increase the number of windings or the current, the size of the coil is likely to change. This will have a detrimental effect on the device, because as the radius increases, the field strength decreases. A possible solution to this would be to use a standard solenoid, which are not size dependent. However, this would create the physical access problem. Additionally, increasing the rate of change of the field will increase the power dissipated in the field, which will have a direct increase in the disturbance field strength.

A simpler and possibly quicker solution would be to replace the HES. The current only had a sensitivity of 5mV/G. Increasing the sensitivity by a factor of 10 to 100 would be greatly beneficial.

Further general improvements to the device would be the ability to add unidentified metals to the devices memory. In its current state, in order to add new limits for the metal lookup table, the code needs to be modified and re-flashed onto the MCU. Including a more robust user interface would simplify this process. Also, automatic identification in free run mode is possible, though it was not implemented.

IX: Bibliography

[1] Image source - http://en.wikipedia.org/wiki/File:Helmholtz_coils.png

[2] Hu, David. "2.972 How a Coin Tester Works." *2.972 How a Coin Tester Works*. MIT, 1998. Web. <http://web.mit.edu/2.972/www/reports/coin_tester/coin_tester.htm>.

[3] Jessop, Joe. "Evaluating Product Integrity with Eddy Current." The Fabricator, 11 Oct. 2010. Web. <http://www.thefabricator.com/article/testingmeasuring/evaluating-product-integrity-with-eddy-current>.

Appendix A: Schematics

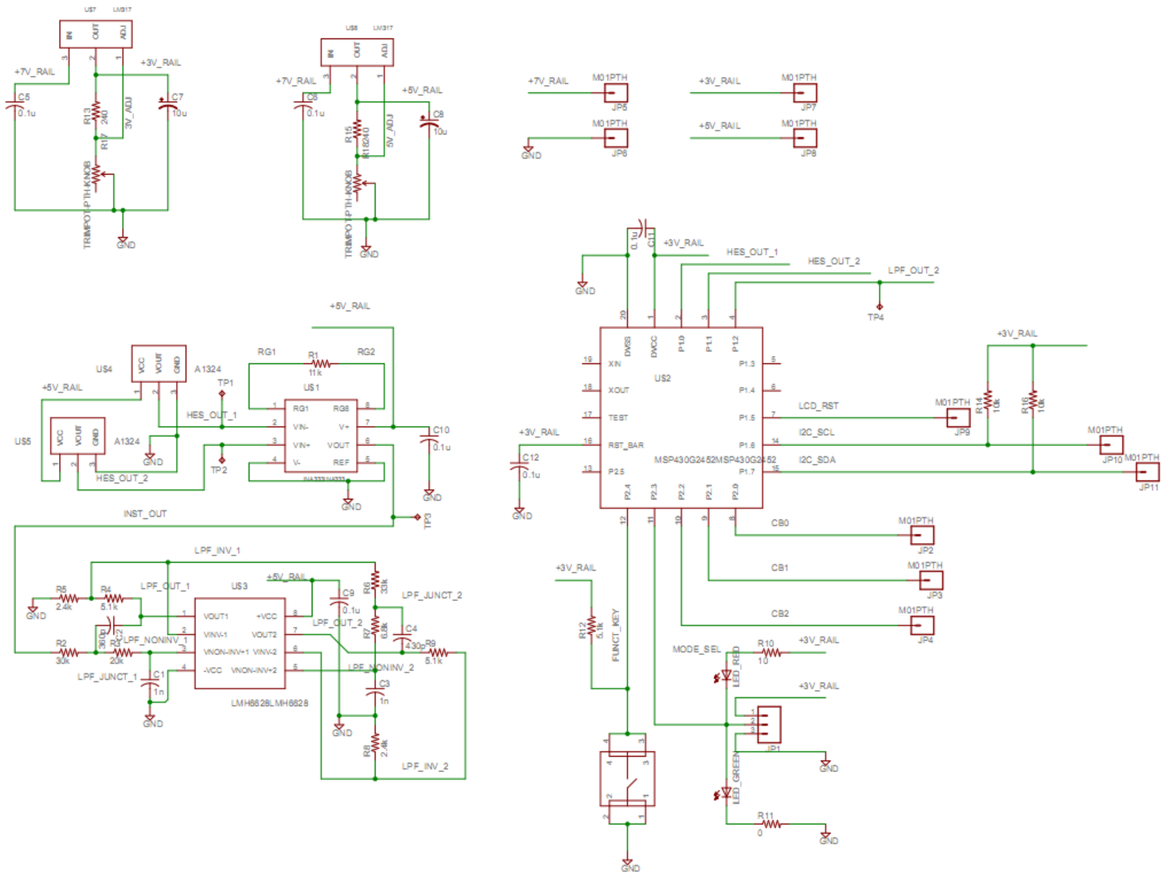

Figure A.1: Main Board Schematic

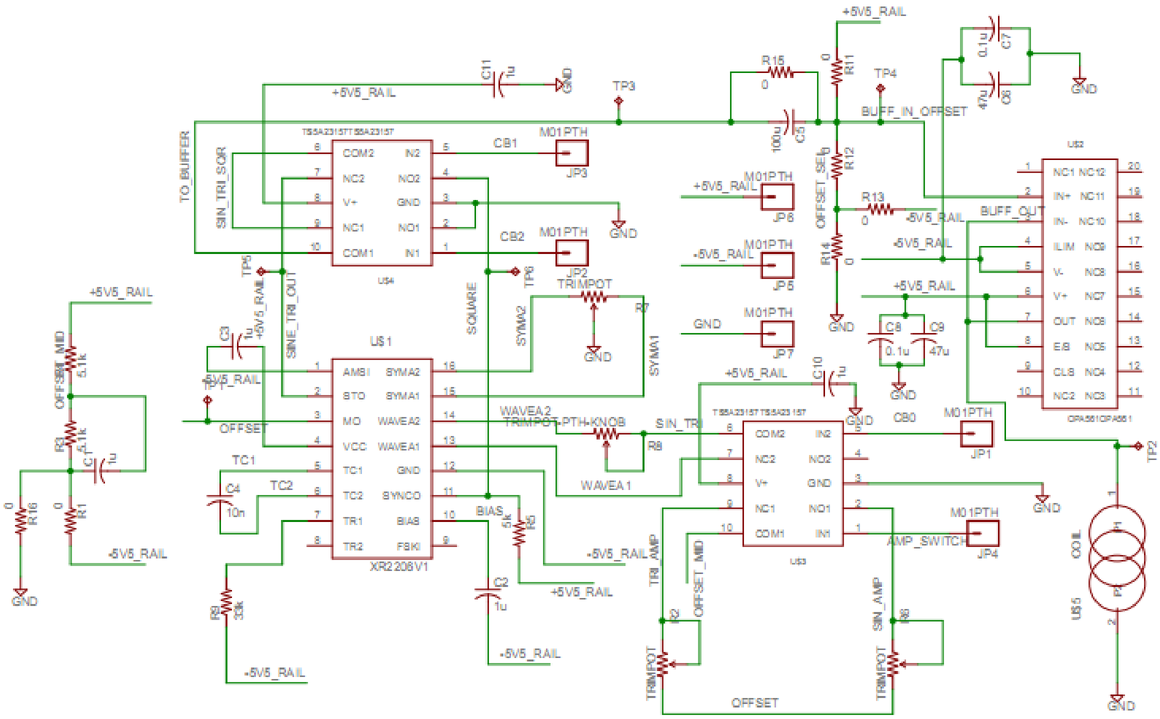

Figure A.2: Coil Board Schematic

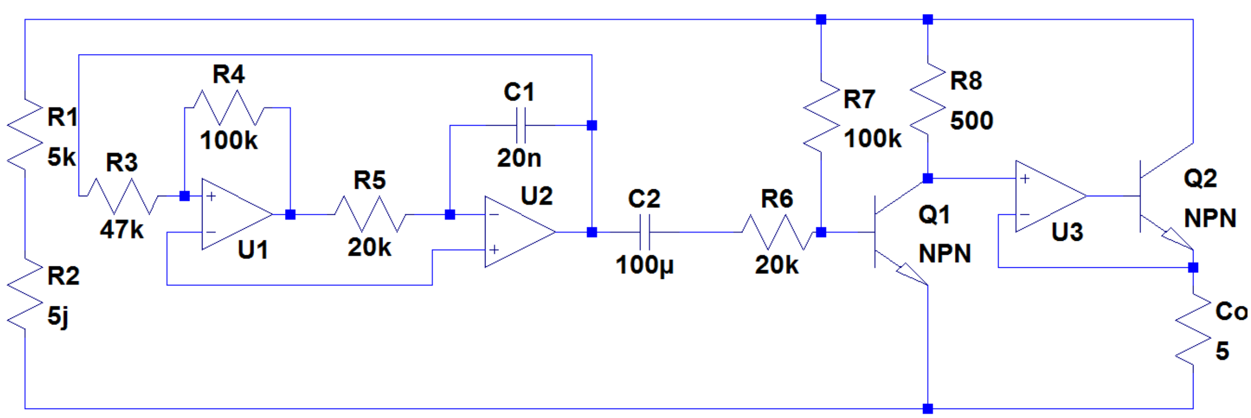

Figure A.3: Auxiliary Coil Schematic

Appendix B: Parts List and Costs

Part	Value	Device	Package	Description
C1	1n	C-USC0805	C0805	CAPACITOR, American symbol
C2	360p	C-USC0805	C0805	CAPACITOR, American symbol
C3	1n	C-USC0805	C0805	CAPACITOR, American symbol
C4	430p	C-USC0805	C0805	CAPACITOR, American symbol
C5	0.1u	C-USC0805	C0805	CAPACITOR, American symbol
C6	0.1u	C-USC0805	C0805	CAPACITOR, American symbol
C7	10u	CAP_POLPTH2	CPOL-RADIAL-10UF-25V	Capacitor Polarized
C8	10u	CAP_POLPTH2	CPOL-RADIAL-10UF-25V	Capacitor Polarized
C9	0.1u	C-USC0805	C0805	CAPACITOR, American symbol
C10	0.1u	C-USC0805	C0805	CAPACITOR, American symbol
C11	0.1u	C-USC0805	C0805	CAPACITOR, American symbol
C12	0.1u	C-USC0805	C0805	CAPACITOR, American symbol
JP1		M03PTH	1X03	Header 3
JP2	M01PTH	M01PTH	1X01	Header 1
JP3	M01PTH	M01PTH	1X01	Header 1
JP4	M01PTH	M01PTH	1X01	Header 1
JP5	M01PTH	M01PTH	1X01	Header 1
JP6	M01PTH	M01PTH	1X01	Header 1
JP7	M01PTH	M01PTH	1X01	Header 1
JP8	M01PTH	M01PTH	1X01	Header 1
JP9	M01PTH	M01PTH	1X01	Header 1
JP10	M01PTH	M01PTH	1X01	Header 1
JP11	M01PTH	M01PTH	1X01	Header 1
LED_GREEN		LED10MM	LED10MM	LEDs
LED_RED		LED10MM	LED10MM	LEDs
R1	11k	R-US_R0805	R0805	RESISTOR, American symbol
R2	30k	R-US_R0805	R0805	RESISTOR, American symbol
R3	20k	R-US_R0805	R0805	RESISTOR, American symbol
R4	5.1k	R-US_R0805	R0805	RESISTOR, American symbol
R5	2.4k	R-US_R0805	R0805	RESISTOR, American symbol
R6	33k	R-US_R0805	R0805	RESISTOR, American symbol
R7	6.8k	R-US_R0805	R0805	RESISTOR, American symbol
R8	2.4k	R-US_R0805	R0805	RESISTOR, American symbol
R9	5.1k	R-US_R0805	R0805	RESISTOR, American symbol
R10	10	R-US_R0805	R0805	RESISTOR, American symbol
R11	0	R-US_R0805	R0805	RESISTOR, American symbol
R12	5.1k	R-US_R0805	R0805	RESISTOR, American symbol
R13	240	R-US_R0805	R0805	RESISTOR, American symbol
R14	10k	R-US_R0805	R0805	RESISTOR, American symbol

R15	240	R-US_R0805	R0805	RESISTOR, American symbol
R16	10k	R-US_R0805	R0805	RESISTOR, American symbol
R17	TRIMPOT-PTH-KNOB	TRIMPOT-PTH-KNOB	3386U	
R18	TRIMPOT-PTH-KNOB	TRIMPOT-PTH-KNOB	3386U	
TP1	TPTP18R	TPTP18R	TP18R	Test pad
TP2	TPTP18R	TPTP18R	TP18R	Test pad
TP3	TPTP18R	TPTP18R	TP18R	Test pad
TP4	TPTP18R	TPTP18R	TP18R	Test pad
U$1	INA333INA333	INA333INA333	MSOP-8	
U$2	MSP430G2452MSP430G2452	MSP430G2452MSP430G2452	DIL20	
U$3	LMH6628LMH6628	LMH6628LMH6628	SOIC-8	
U$4	A1324	A1324	SIL_OFF_BOARD	
U$5	A1324	A1324	SIL_OFF_BOARD	
U$6	MOM_SWITCH	MOM_SWITCH	MOM_SWITCH	
U$7	LM317	LM317	TO220-ADJ	
U$8	LM317	LM317	TO220-ADJ	

Table B.1: Maine Board Parts Lists

Part	Value	Device	Package	Description
C1	1u	C-USC0805	C0805	CAPACITOR, American symbol
C2	1u	C-USC0805	C0805	CAPACITOR, American symbol
C3	1u	C-USC0805	C0805	CAPACITOR, American symbol
C4	10n	C-USC0805	C0805	CAPACITOR, American symbol
C5	100u	C-USC0805	C0805	CAPACITOR, American symbol
C6	47u	C-USC0805	C0805	CAPACITOR, American symbol
C7	0.1u	C-USC0805	C0805	CAPACITOR, American symbol
C8	0.1u	C-USC0805	C0805	CAPACITOR, American symbol
C9	47u	C-USC0805	C0805	CAPACITOR, American symbol
C10	1u	C-USC0805	C0805	CAPACITOR, American symbol
C11	1u	C-USC0805	C0805	CAPACITOR, American symbol
JP1	M01PTH	M01PTH	1X01	Header 1
JP2	M01PTH	M01PTH	1X01	Header 1
JP3	M01PTH	M01PTH	1X01	Header 1
JP4	M01PTH	M01PTH	1X01	Header 1
JP5	M01PTH	M01PTH	1X01	Header 1
JP6	M01PTH	M01PTH	1X01	Header 1
JP7	M01PTH	M01PTH	1X01	Header 1
R1	0	R-US_R0805	R0805	RESISTOR, American symbol
R2	TRIMPOT	TRIMPOT	3362U-2	Various small potentiometers for "set-and-forget" applications
R3	5.1k	R-US_R0805	R0805	RESISTOR, American symbol
R4	5.1k	R-US_R0805	R0805	RESISTOR, American symbol
R5	5k	R-US_R0805	R0805	RESISTOR, American symbol

R6	TRIMPOT	TRIMPOT	3362U-2	Various small potentiometers for "set-and-forget" applications
R7	TRIMPOT	TRIMPOT	3362U-2	Various small potentiometers for "set-and-forget" applications
R8	TRIMPOT-PTH-KNOB	TRIMPOT-PTH-KNOB	3386U	Various small potentiometers for "set-and-forget" applications
R9	33k	R-US_R0805	R0805	RESISTOR, American symbol
R11	0	R-US_R0805	R0805	RESISTOR, American symbol
R12	0	R-US_R0805	R0805	RESISTOR, American symbol
R13	0	R-US_R0805	R0805	RESISTOR, American symbol
R14	0	R-US_R0805	R0805	RESISTOR, American symbol
R15	0	R-US_R0805	R0805	RESISTOR, American symbol
R16	0	R-US_R0805	R0805	RESISTOR, American symbol
TP1	TPTP18R	TPTP18R	TP18R	Test pad
TP2	TPTP18R	TPTP18R	TP18R	Test pad
TP3	TPTP18R	TPTP18R	TP18R	Test pad
TP4	TPTP18R	TPTP18R	TP18R	Test pad
TP5	TPTP18R	TPTP18R	TP18R	Test pad
TP6	TPTP18R	TPTP18R	TP18R	Test pad
U$1	XR2206V1	XR2206V1	DIL16	
U$2	OPA561OPA561	OPA561OPA561	HTSSOP-20	
U$3	TS5A23157TS5A23157	TS5A23157TS5A23157	MSOP-10	
U$4	TS5A23157TS5A23157	TS5A23157TS5A23157	MSOP-10	
U$5	COIL	COIL	COIL_CONNECT	

Table B.2: Coil Board Parts List

Cost Breakdown:
Magnet Wire: $18.54
Board Order: $54.27
Other Parts: $75.36
Total Cost: $146.17

Appendix C: Schedule

Milestones (by topic, revised):

Coil Construction and Testing

  Power Supply Construction – Completed May 22nd

  Coil Windings and Field Testing – Completed May 1st

Sensor Testing – Completed April 27th

Signal Conditioning (Filter and Amplifier) – Completed May 23rd

MCU Programming

  ADC Input from Sensor – Completed May 29th

  Data Analysis – Completed May 29th

  DAC Output to Coil Power Supply – Completed May 24th

  Output to LCD screen – Completed May 29th

  Switch Interface Testing – Completed May 24th

Assembly, such as creating HES mounting – Completed May 25th

PCB Design – Completed April 12th

Additional Circuit Design – Completed May 30th

Project Completion Date: May 30th

Appendix D: PC Board Layout

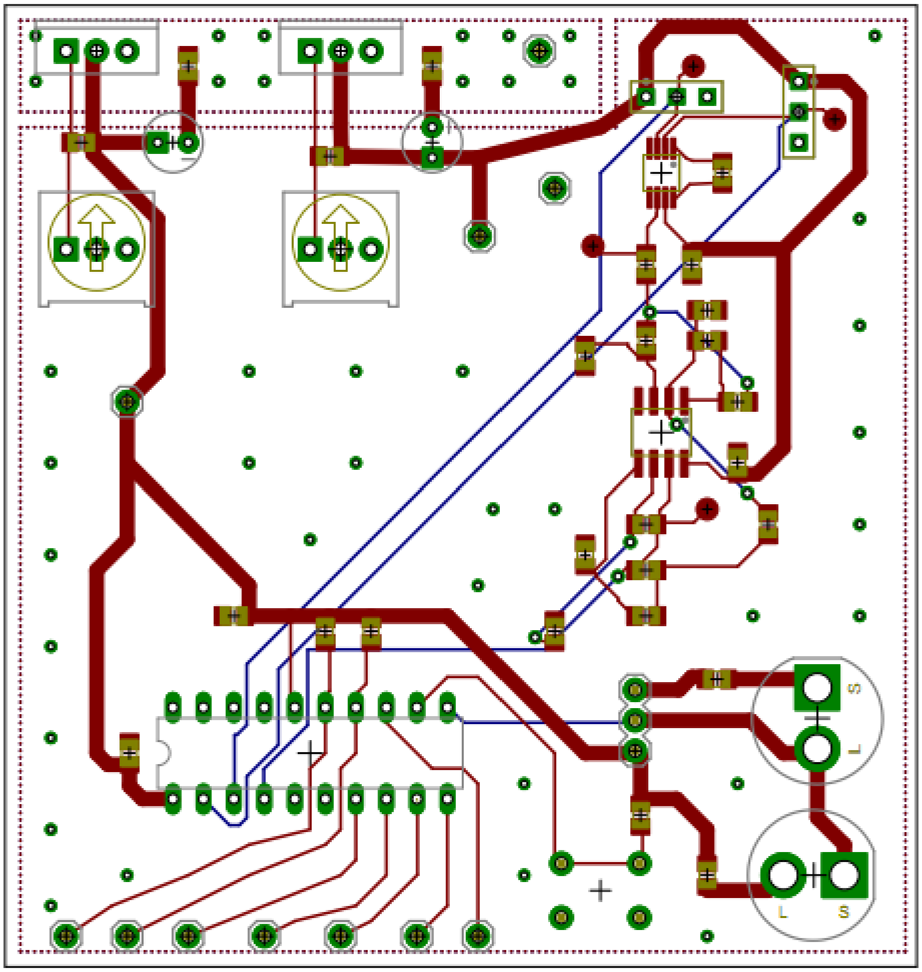

Figure D.1: Main Board Layout

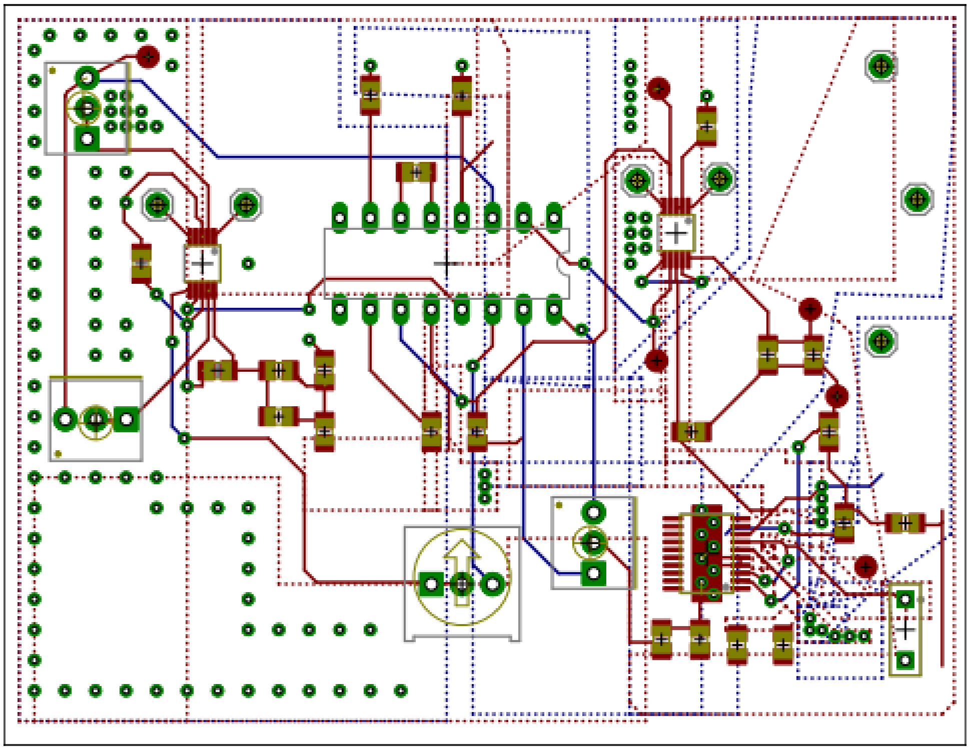

Figure D.2: Coil Board Layout

Appendix E: Microprocessor Code

```c
// Senior Project
// Paul Maggi
// I2C code used from the LCD data sheet, with guidance from
http://dbindner.freeshell.org/msp430/lcd_i2c.html
//
// Pinouts:
// P1.0 - INPUT - ADC - HES-1
// P1.1 - INPUT - ADC - HES-2
// P1.1 - INPUT - ADC = LPF-Out
// P1.3 - NC
// P1.4 - NC
// P1.5 - OUTPUT - LCD_RST
// P1.6 - IN/OUT - I2C_SCL
// P1.7 - IN/OUT - I2C_SDA
// P2.0 - OUTPUT - CB0 - tri/sine selector - 1 for tri, 0 for sine
// P2.1 - OUTPUT - CB1 - square or other selector - 1 for square, 0 for sine/tri
// P2.2 - OUTPUT - CB2 - Amplitude select - 1 for tri, 0 for sine
// P2.3 - INPUT - MODE_SELECT - low = one shot mode, high = continuous read
// P2.4 - INPUT - FUNCTION_BUTTON - Active Low (default high)
//
// LCD Slave Address = 0x7C


#include <msp430g2452.h>
#include <string.h>

#define CB0 BIT0 // CBO line for tri/sine selector
#define CB1 BIT1 // CB1 square or other selector
#define CB2 BIT2 // CB2 amplitude select for sine/tri
#define I2C_SDA BIT7 // I2C data line
#define I2C_SCL BIT6 // I2C clock line
#define LCD_RST BIT5 // LCD RST Line, active low
#define MODE BIT3
#define BUTTON BIT4
#define SLAVE_ADDR 0x7C // I2C slave address


volatile int mode_status; // 1 = continous read, 0 = one-shot
volatile int button_status = 0; // 1 means pressed
volatile unsigned int ADCValue;
volatile unsigned int calibration = 0;
volatile int ADCCount;
volatile unsigned int tens;
volatile unsigned int thousands;
volatile unsigned int hundreds;
volatile unsigned int ones;
volatile unsigned int buffer;
volatile int change_top = 1;
volatile int change_bot = 1;
volatile int change_button = 1;
volatile int change_switch = 1;
volatile int cal_done = 0;
```

```c
char disp_top[16];
char disp_bot[16];

// a delay function for timing
void delay( unsigned int n) {
	volatile int i;

	for( ; n; n--){
		for( i = 0; i < 50; i++);
	}
}

void data_read(void) {
	P1DIR &= ~I2C_SDA; // float to get ready to read, sets SDA as input
}

void data_high(void) { // floats the data line, external pullups bring line high
	P1DIR &= ~I2C_SDA;
	delay(5);
}

void data_low(void) {
	P1OUT &= ~I2C_SDA; // assert low
	P1DIR |= I2C_SDA;
	delay(5);
}

void clk_high(void) {
	P1DIR &= ~I2C_SCL; // float pin to go high from external pullups
}

void clk_low(void) {
	P1OUT &= ~I2C_SCL; // assert low
	P1DIR |= I2C_SCL;
	delay(5);
}

void I2C_Start(void) {
	clk_high();
	data_high();
	data_low();
	clk_low();
}

void I2C_Stop(void) {
	data_low();
	clk_low();
	clk_high();
	data_high();
}

void I2C_out(unsigned char d) {
	int n;

	for( n = 0; n < 8; n++) {
```

```
			if (d & 0x80) {
				data_high();
			}else {
				data_low();
			}

			clk_high();
			clk_low();

			d <<= 1;
		}

		data_read();
		clk_high();

		// wait for ack

		while(P1IN & I2C_SDA) {
			// toggles clock line and checks again for ack
			clk_low();
			clk_high();
		}

		clk_low();
}

void init_LCD(void) {
		I2C_Start();

		I2C_out( 0x7C ); // slave address of LCD panel
		I2C_out( 0x00 ); // control byte for commands
		I2C_out( 0x38 ); // 8bit but, 2 line display, extended instruction mode
		delay(10);
		I2C_out( 0x39 ); // 8bit but, 2 line display, extended instruction mode
		delay(10);
		I2C_out( 0x14 ); // sets bias to 1/5
		I2C_out( 0x78 ); // contrast set
		I2C_out( 0x5E ); // icon display on, booster on, contrast set
		I2C_out( 0x6D ); // follower circuit on, amplifier=1
		I2C_out( 0x0C ); // display on, cursor off
		I2C_out( 0x01 ); // clear display
		I2C_out( 0x06 ); // entry mode set to cursor moves right
		delay(10);

		I2C_Stop();
}

void clear_display(void){
		I2C_Start();

		I2C_out( SLAVE_ADDR );
		I2C_out(0x00); // control byte for commands
		I2C_out(0x01); // clear display command

		I2C_Stop();
```

```
}

void show_top(void) {
	int length;
	int n;

	delay(10);

	//if(change_top == 1){
		length = strlen(disp_top);
		while(length <= 15){
			disp_top[length] = ' ';
			length++;
		}

		I2C_Start();
		I2C_out( SLAVE_ADDR );
		I2C_out(0x80);
		I2C_out(0x80);
		I2C_out(0x40); // control bye for RAM data

		for( n = 0; n < 16; n++) {
			I2C_out(disp_top[n]);

		}

		I2C_Stop();
		change_top = 0;
	//}

}

void show_bot(void) {
	int length;
	int n;

	delay(10);

	//if(change_bot == 1){
		length = strlen(disp_bot);
		while(length <= 15){
			disp_bot[length] = ' ';
			length++;
		}

		I2C_Start();
		I2C_out( SLAVE_ADDR );
		I2C_out(0x80);
		I2C_out(0xC0);
		I2C_out(0x40); // control bye for RAM data

		for( n = 0; n < 16; n++) {
			I2C_out(disp_bot[n]);

		}
```

```
		I2C_Stop();
		change_bot = 0;
	//}

}


void init_interrupt(void){
	_BIS_SR(GIE); //enables interrupts

	P2DIR &= ~MODE;
	P2DIR &= ~BUTTON;

	mode_status = (P2IN & MODE);
	if(mode_status != 0){
		mode_status = 1;
	}else{
		mode_status = 0;
	}

	button_status = (P2IN & BUTTON);
	if(button_status != 0){
		button_status = 0;
	}else{
		button_status = 1;
	}

	if(mode_status){
		P2IES &= ~MODE; // sets interrupt to H->L triggered
	}else{
		P2IES |= MODE; // sets interrupt to L->H triggered
	}


	P2IES |= BUTTON;

	P2IE |= BUTTON; // enables interrupt on button pin
	P2IE |= MODE; // enables interrupt on mode pin

	P2IFG = 0x00;
}

void init_ADC(int chan);

void num_parse(char* stuff);

void ADCRead(void);

void sine_out(void);

void tri_out(void);

void square_out(void);
```

```
int main(void) {

	int wave = 0;

	WDTCTL = WDTPW + WDTHOLD; // stops the watchdog timer

	delay(5000);

	init_interrupt();

	sine_out();

	P1DIR |= LCD_RST;
	P1OUT &= ~LCD_RST;
	P1OUT |= LCD_RST;

	init_ADC(2);

	init_LCD();
	clear_display();

	change_top = 1;
	change_bot = 1;
	change_switch = 1;

	sine_out();


	while(1){

		if(mode_status & ~change_switch & cal_done){
			ADCRead();
			strcpy(disp_bot, "Value = ");
			if(ADCValue >= calibration){
				ADCValue = ADCValue - calibration;
			}else{
				strcat(disp_bot, "-");
				ADCValue = calibration - ADCValue;
			}
			num_parse(disp_bot);
			change_bot = 1;
			show_bot();
			delay(500);
		}

		if(change_switch | change_button){
			if(mode_status & change_switch){
				clear_display();
				strcpy(disp_top, "Free Run Mode");
				delay(5);
				change_top = 1;
				show_top();
				change_switch = 0;
				if(cal_done == 0){
					strcpy(disp_bot, "Pleas Run Cal");
```

```
                change_bot = 1;
                show_bot();
            }
            change_button = 0;

        }

        if(change_button & mode_status){ // calibration function

            strcpy(disp_bot, "Calibrating");
            change_bot = 1;
            show_bot();

            ADCRead();
            calibration = ADCValue;

            strcpy(disp_bot, "Air = ");
            num_parse(disp_bot);
            change_bot = 1;
            show_bot();

            cal_done = 1;

            change_button = 0;
        }

        if((~mode_status & change_switch & ~cal_done) | (~mode_status &
change_button & ~cal_done)){
            strcpy(disp_top, "Please Run Cal");
            strcpy(disp_bot, "In Free Run Mode");
            change_top = 1;
            change_bot = 1;
            show_top();
            show_bot();

        }

        if(~mode_status & change_switch & cal_done){
            strcpy(disp_top, "One Shot Mode");
            delay(5);
            change_top = 1;
            show_top();
            change_switch = 0;
            change_button = 1;
            button_status = 0;
        }
        if(~mode_status & change_button & cal_done){
            ADCRead();
            strcpy(disp_bot, "Metal = ");
            if(ADCValue >= calibration){
                ADCValue = ADCValue - calibration;
            }else{
                ADCValue = calibration - ADCValue;
            }
            if(ADCValue >= 200){
```

```
                                strcat(disp_bot, "iron");
                        }else if(ADCValue >= 100){
                                strcat(disp_bot, "silver");
                        }else{
                                strcat(disp_bot, "none");
                        }
                        change_bot = 1;
                        show_bot();
                        change_button = 0;


                }
            }
        }
}

// Port 2 ISR
#pragma vector=PORT2_VECTOR
__interrupt void Port_2(void){

        if(P2IFG & BUTTON){ // if was a button press, modify the button status pin
                if(button_status == 1){
                        button_status = 0;
                }else if(button_status == 0){
                        button_status = 1;
                }
//              P2IES ^= BUTTON; // toggles the direction pin.
                P2IFG &= ~BUTTON;
                change_button = 1;

        }else if(P2IFG & MODE){ // if it was a mode switch toggle, then change the
interrupt direction, set the new mode status, and clear interrupt
                P2DIR &= ~MODE;
                mode_status = (P2IN & MODE);
                if(mode_status != 0){
                        mode_status = 1;
                        P2IES |= MODE;
                }else{
                        mode_status = 0;
                        P2IES &= ~MODE;
                }
                P2IE |= MODE;
                P2IFG &= ~MODE;
                change_switch = 1;
        }

        change_top = 1;
        change_bot = 1;

}

//Initializes the ADC, Vref+ = Vcc, Vref- = Vss
void init_ADC(int chan){
        ADCValue = calibration; // sets the ADC to the calibrated value, defaults to 0
        ADCCount = 0;
```

```
	ADC10CTL0 &= ~ENC; //disable ADC
	ADC10CTL0 = SREF_0; // sets Vref+ = Vcc, Vref- = Vss
	ADC10CTL0 = ADC10SHT_1 + ADC10ON + ADC10IE; // 16 clk ticks, ADC on, interrupt
enable
	ADC10CTL1 |= ADC10SSEL_0; // sets clock to ADC10OSC at ~5MHz
	ADC10CTL1 |= ADC10DIV_4; // divides clk by 5
	if(chan == 2){
		ADC10CTL1 = INCH_2; // sets ADC to sample A2 on P1.2
		ADC10AE0 |= 0x04; // enables analog inputs on P1.2
	}else if (chan == 1){
		ADC10CTL1 = INCH_1; // sets ADC to sample A1 on P1.1
		ADC10AE0 |= 0x02; // enables analog inputs on P1.1
	}else if (chan == 0){
		ADC10CTL1 = INCH_0; // sets ADC to sample A0 on P1.0
		ADC10AE0 |= 0x01; // enables analog inputs on P1.0
	}else{
		ADC10CTL1 = INCH_2; // defaults to A2
	}
	ADC10CTL1 |= CONSEQ_0; // single channel sampled once
	ADC10CTL0 |= MSC; // sets so that it continues sampling until ENC is set to 0
}

void num_parse(char* stuff){
	int checker = 3;

	thousands = (ADCValue / 1000);
	hundreds = ((ADCValue - (thousands * 1000)) / 100);
	tens = ((ADCValue - ((thousands * 1000) + (hundreds * 100)) ) / 10);
	ones = (ADCValue - ((thousands * 1000) + (hundreds * 100) + (tens * 10)) );

	while(checker <= 3){
		if(checker == 3){
			buffer = thousands;
			checker = 2;
		}else if(checker == 2){
			buffer = hundreds;
			checker = 1;
		}else if (checker == 1){
			buffer = tens;
			checker = 0;
		}else if (checker == 0) {
			buffer = ones;
			checker = 4;
		}


		if(buffer == 0 ){
			strcat(stuff, "0");
		}else if(buffer == 1){
		strcat(stuff, "1");
		}else if(buffer == 2){
		strcat(stuff, "2");
		}else if(buffer == 3){
		    strcat(stuff, "3");
		}else if(buffer == 4){
```

```c
                strcat(stuff, "4");
            }else if(buffer == 5){
                strcat(stuff, "5");
            }else if(buffer == 6){
                strcat(stuff, "6");
            }else if(buffer == 7){
                strcat(stuff, "7");
            }else if(buffer == 8){
                strcat(stuff, "8");
            }else if(buffer == 9){
                strcat(stuff, "9");
            }else{
                strcpy(stuff, "Value = Unknown");
                checker = 4;
            }
        }
}


#pragma vector=ADC10_VECTOR // ADC ISR, just gets the max value during hte ADC read
__interrupt void ADC10_ISR (void){
        if(ADC10MEM > ADCValue){
                ADCValue = ADC10MEM;
        }
        ADCCount++;

}

void ADCRead(void){
                    ADCCount = 0; // resets the ADC counter
                    ADCValue = 0;
                    ADC10CTL0 |= ENC + ADC10SC; // enables ADC and starts conversion

                     while(ADCCount <= 4){
                        ADC10CTL0 &= ~ENC;

                        if(ADC10MEM > ADCValue){
                                ADCValue = ADC10MEM;
                        }
                        ADC10CTL0 |= ENC + ADC10SC;
                        ADCCount++;

                    }
                     ADC10CTL0 &= ~ENC;
}

void sine_out(void){
        P2OUT &= ~CB0;
        P2OUT &= ~CB1;
        P2OUT &= ~CB2;

        P2DIR |= CB0; // sets P2.0 to output
        P2DIR |= CB1; // sets P2.1 to output
        P2DIR |= CB2; // sets P2.2 to output
```

```
}

void tri_out(void){
	P2OUT |= CB0;
	P2OUT &= ~CB1;
	P2OUT |= CB2;

	P2DIR |= CB0; // sets P2.0 to output
	P2DIR |= CB1; // sets P2.1 to output
	P2DIR |= CB2; // sets P2.2 to output

}

void square_out(void){

	P2OUT &= ~CB0; // defaults to sine
	P2OUT |= CB1; // defaults to sine
	P2OUT &= ~CB2; // defaults to sine

	P2DIR |= CB0; // sets P2.0 to output
	P2DIR |= CB1; // sets P2.1 to output
	P2DIR |= CB2; // sets P2.2 to output

}
```