Senior Design Final Documentation
DEC14-10
Honeywell Polarity Detection Device

Chen Cheng, Jeremiah Janssen,
Kailey McGuire, Tori Sorensen, Tymothy Wood

Table of Contents

Introduction 4
- Problem Overview 4
- Deliverables 4
- Specifications 4

System Level Design 5
- Functional Requirements 5
- Non-Functional Requirements 5
- Input and Output Specifications 5
- Interface Specifications 5

Problem Statement 6

Design Approach 8
- Powering the coils 8
- Reading the Magnetic Field 9

Implementation Details 10
- Circuit components 10
- Arduino 11
- Magnetic sensors 11
- Final Steps 12
- Cost 13

Testing 14

Conclusion 15

Appendix I: Operation Manual 16
- About the Reading 17

Appendix II: Alternative Designs 18

Appendix III: Other Considerations 19

Appendix IV: Arduino Code 20

List of Figures

Figure 1: Example test unit description 6
Figure 2: Honeywell mock testing units 7
Figure 3: Prototype cores used for lower level testing 19
Figure 4: Powering coil 1 8
Figure 5: Powering coil 2 19
Figure 6: Wand containing two sensors on the mock unit 9
Figure 7: Printed circuit board layout 10
Figure 8: Comparison of magnetic sensors 11
Figure 9: Prototype of design 12
Figure 10: Cost Breakdown 13
Figure 11: Testing of Honeywell mock units 19
Figure 12: Testing setup 16
Figure 13: Example output (1) 17
Figure 14: Example output (2) 17
Figure 15: Example output (3) 19
Figure 16: Initial design 18

Introduction

Problem Overview

Honeywell has requested a tool to check the polarity of magnetic fields produced by two coils that are components incorporated into a larger unit. The goal is to reduce the cost of finding a discrepancy in later assembly steps. Confirmation that coils are correctly wound and positioned is required at an early stage of production. Currently, testing is not performed until a high level of assembly and the cost to fix a mistake is very expensive. Employing a tool that confirms correct polarity at an early stage in production reduces costs by a factor of ten. The exact configuration of the coils is unknown; therefore, CAD renderings have been created using Inventor software of an example unit.

Deliverables

The team has produced a testing unit that is capable of precisely identifying magnetic field polarity. This unit was required to be adaptable and versatile, as it will serve as a base model for a similar testing unit. Adaptations will be made based on the specific use. For this reason, the tool was designed to be easily adaptable.

The Arduino platform is extremely easy to access, and the code was written with consideration given to the user who would like to make adjustments. The current provided to the coils is also very easily increased if a stronger field is required, or decreased if it is found that the current will damage the coils.

Other adaptations could address issues including:

- Reading a magnetic field through metal
- Examining effects from equipment on a magnetic field
- Redesigning for differing coil configurations
- Editing the format of data reporting
- Reducing the size of the testing unit
- Powering the testing unit from a wall outlet
- Sending a pulse to the coil using a capacitor

The team has produced an efficient tool with the ability to adapt to a variety of situations. In depth documentation is also provided to aid any user that wishes to adapt the tool.

Specifications

This device is required to supply power to the coils, read the magnetic field, and provide a “fool-proof” output to the technician to confirm correct polarity. In addition, the design must offer ease in adaptability, preferably using commercially available devices.

System Level Design

Functional Requirements

- Less than 10 V, DC
- Less than 100 mA, DC
- Easy to calibrate
- Operate at room temperature, 75° F
- Should not be affected by the earth's magnetic field or local sources
- Simple user interface
- Will not damage product during contact
- Bench top set up
- Indicates battery level

Non-Functional Requirements

- Easy to keep clean
- Use a battery (lithium ion) that can connect to a charging circuit
- Portable
- Light and easy to handle
- Utilize commercially available devices
- Can be custom fit to various product configurations

Input and Output Specifications

The input to the device will be obtained through two magnetic chips, which can be programmed to report output in any desired format. The magnetic chips collect readings on magnetic field strength along three axes. Only the data along one axis will be analyzed and reported to the LCD as output.

Interface Specifications

The interface is required to be user friendly and adjustable. For this reason, commercially available components were chosen. The Arduino Uno is a popular microprocessor that is extremely user friendly and easily adaptable. Other circuit components will also be chosen to make the design as flexible as possible.

Problem Statement

Due to the nature of Honeywell's work with government contracting, the details of the units containing the coils were not fully disclosed. The problem statement provided was intended to be vague, not corresponding to the actual unit (shown below).

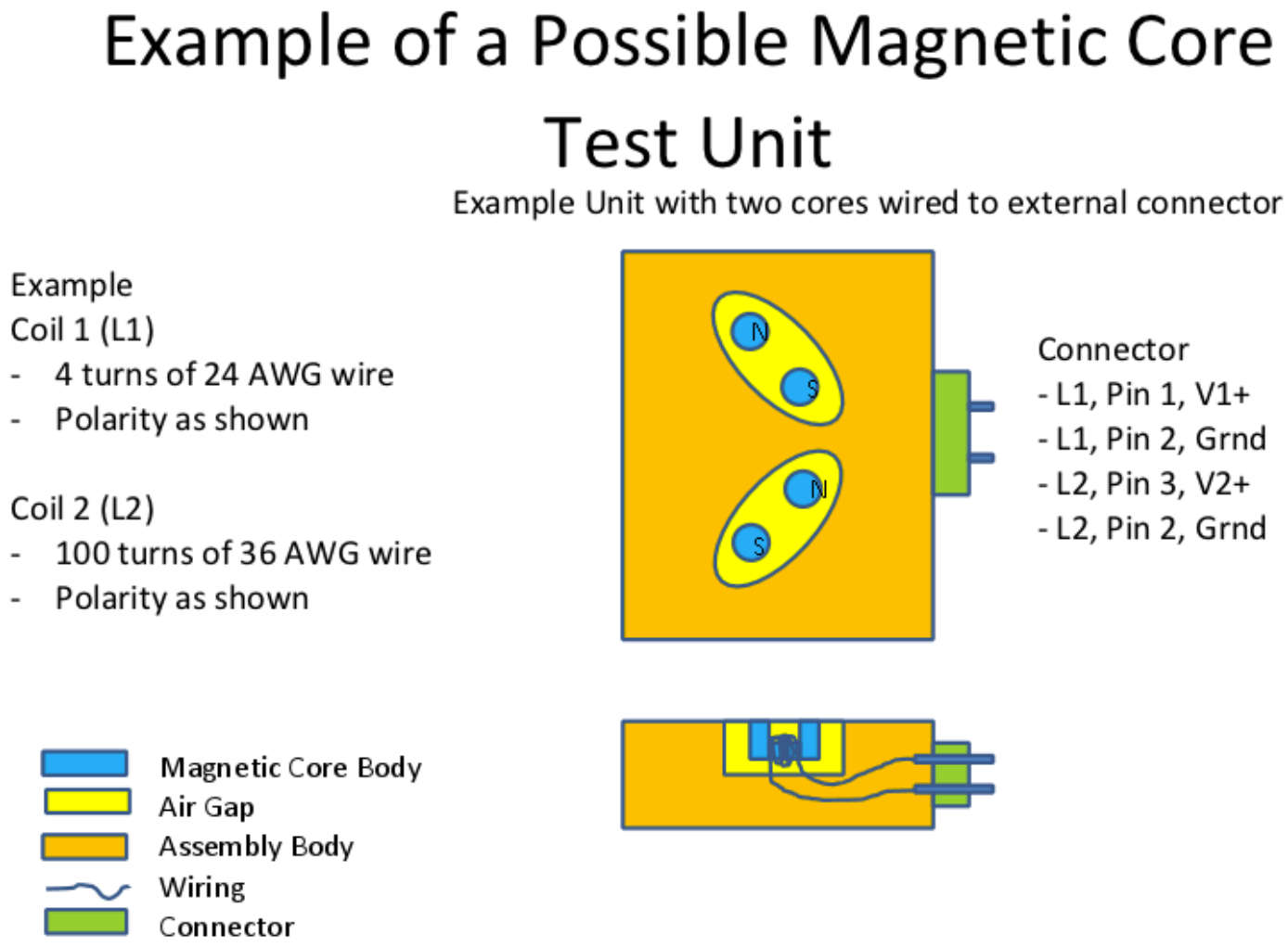

Figure 1 Example test unit description

This information was provided by Honeywell as a possible example, and the design was required to fit varying configurations of a similar set up.

For this reason, certain assumptions needed to be made regarding the unit. Mock units were provided for testing by Honeywell. The mock units contain two coils in an arbitrary configuration. The design of the polarity checker will be altered for use on the actual units.

Figure 2 Honeywell mock testing units

These mock units were only available for testing when a Honeywell representative was present to supervise. This presented a few challenges in performing necessary testing throughout the design process. For this reason, prototype cores were developed to perform low level testing using a simple C core and copper wire. The two cores were made to closely resemble the magnetic field strength of the coils.

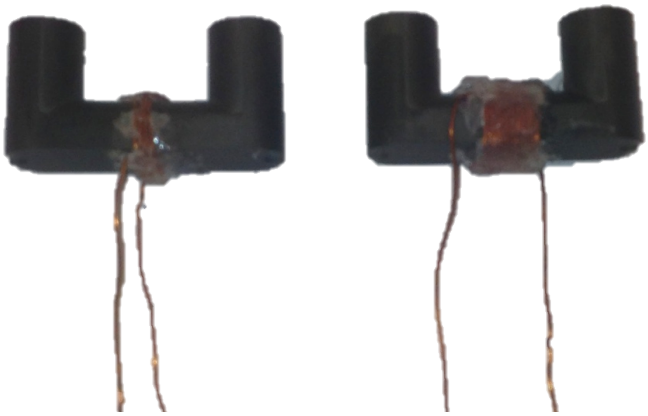

Figure 3 Prototype cores used for low level testing

Design Approach

Powering the coils

The coils are powered using one of two 9V batteries contained within the main unit by way of the DB9 connector cable. The original current requirement limited the current provided to the coil to 100 mA of constant current. This limitation was intended to protect the coil from potential damage. However, it was determined that the current was not enough to produce a field capable of being read by the selected magnetic sensors.

In order to provide a sufficient magnetic field while still ensuring the integrity of the coil, a pulse is sent to the coil by the 9V battery using two relays and a current limiting resistor on the PCB. Using Pin 1 and Pin 2 of the DB9 connection, the first relay powers the first coil with a 300 ms pulse from the 9V battery. This can be seen in the figure below in red.

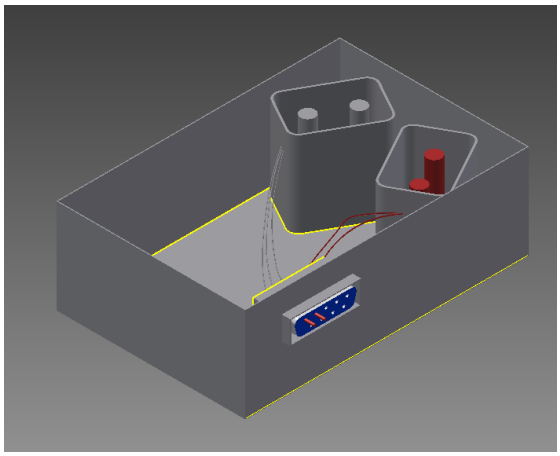

Figure 4 Powering coil 1

The first magnetic sensor takes a reading, and the field is allowed to decay before testing the second coil. Using Pin 2 and Pin 3, the second relay powers the second, center-tapped coil with a 300 ms pulse from the 9V battery. This can be seen in the figure below in red.

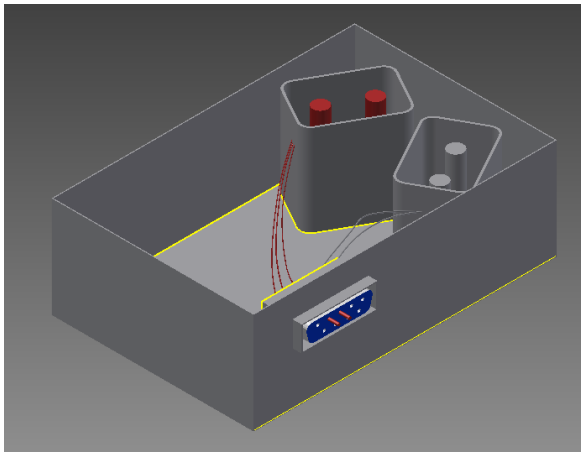

Figure 5 Powering coil 2

Reading the Magnetic Field

The magnetic field is read using two magnetic sensors contained within the wand. The sensors are placed in the wand to obtain optimal readings. Using the ribbon cable, the input is relayed to the Arduino for processing. The sensors are configured to read the magnetic field along the z-axis. The wand fits snuggly into the indentation on the top of the mock unit in order to ensure proper positioning.

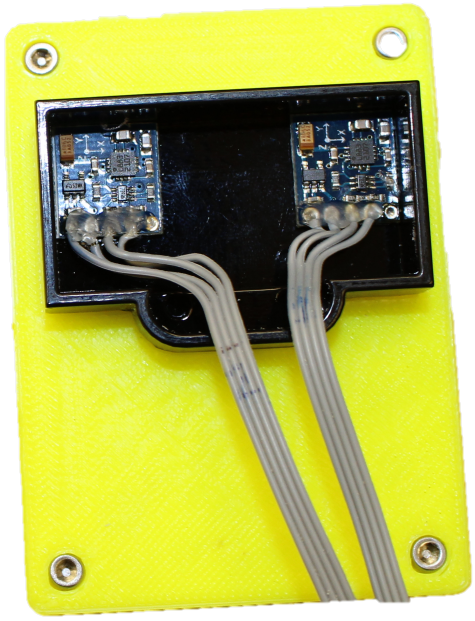

Figure 6 Wand containing two sensors on the mock unit

Implementation Details

Circuit components

The circuit is straightforward and contains two main parts. The left side of the printed circuit board handles routing power to the coils. The coil battery is connected to a current limiting resistor (16Ω). The resistor is connected to two relays (JZC-11F) that select which coil will be powered using two transistors (P2N2222A). The right side of the board contains the transistors for the sensors and buses for common connections between all components.

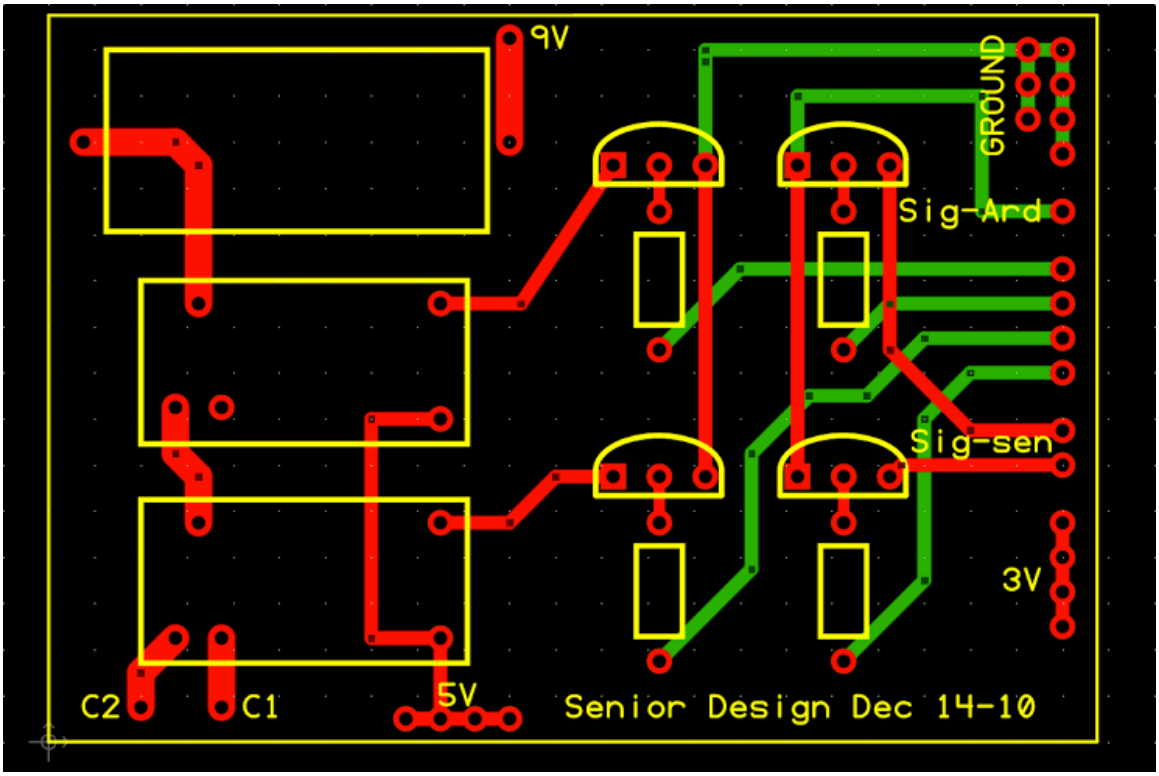

Figure 7 Printed circuit board layout

With the I2C interface, all data transfer is handled with 2 connections: SLC and SDA. Several sensors can be connected to the Arduino bus, but individual sensor selection is handled by sending and receiving data from each sensors unique address. The HMC 5883L sensor has the same address for each chip and is unable to be changed. This means when the code requests data from a specific sensor, both sensors will operate at the same time and the data is jumbled and unusable. To solve this problem, two transistors (P2N2222A) are used to disable the SDA line to one sensor while the other is being operated. On device startup we can enable both transistors and initialize sensors together with the same code.

Arduino

One of the main goals of this project has been to ensure the device is as easy to modify as possible. An Arduino is an economical choice and is simple to code. Honeywell engineers would easily be able to modify the Arduino code to suite their specific needs.

The magnetic field must be read at the moment the coil is being pulsed without interference from any outside source. To remedy this, the data from the sensor is read just before pulsing the coil and that value is subtracted from the reading taken. This provides fairly precise readings, even if large outside magnetic fields are present. If an extraneous magnetic field is larger than the capability of the sensor, the device will give an error message and the test can be rerun in a different environment.

Magnetic sensors

Two different Honeywell sensors were considered for this project. The differences between the sensors are outlined below.

	HMC 5883L	**HMC 2003**
Interface	PC	Analog
Resolution	730 µG	40 µG
Range	-8 G to +8 G	-2 G to +2 G
Cost	$7	$310

Figure 8 Comparison of magnetic sensors

The HMC 5883L sensor was chosen in the end. This sensor has a much larger resolution, but is available at a fraction of the cost of the HMC 2003.

The I2C interface of the 5883L was a very attractive feature for this project. Only two connections are needed between the I2C bus and the Arduino to read the data from the sensors. This eliminates several connections for each sensor when compared to analog sensors and simplifies the circuit. While the resolution of the 5883L is much greater, initial testing suggested it would still be sensitive enough for the scope of this project. The benefit of the I2C interface coupled with the economical cost, made the HMC 5883L made this sensor a very logical and attractive choice.

The HMC 5883L is a 3-axis sensor. The test unit configuration has the sensor positioned directly on top of the coil, meaning only one axis is needed to get a proper reading. As it has been stated before, the exact orientation of the coils on the real setup is not known. Using a 3-axis sensor allows for flexibility to correctly read the orientation of the magnetic field from anywhere around the sensor. The axis readings are taken from can simply be changed in the code depending on the coil orientation.

Final Steps

A prototype was developed using the required components assembled onto a bread board. This prototype can be seen in the figure below. Initial testing was performed with the prototype cores to confirm the correct functionality. Once the prototype was functional, the process of migrating towards a portable bench top setup began.

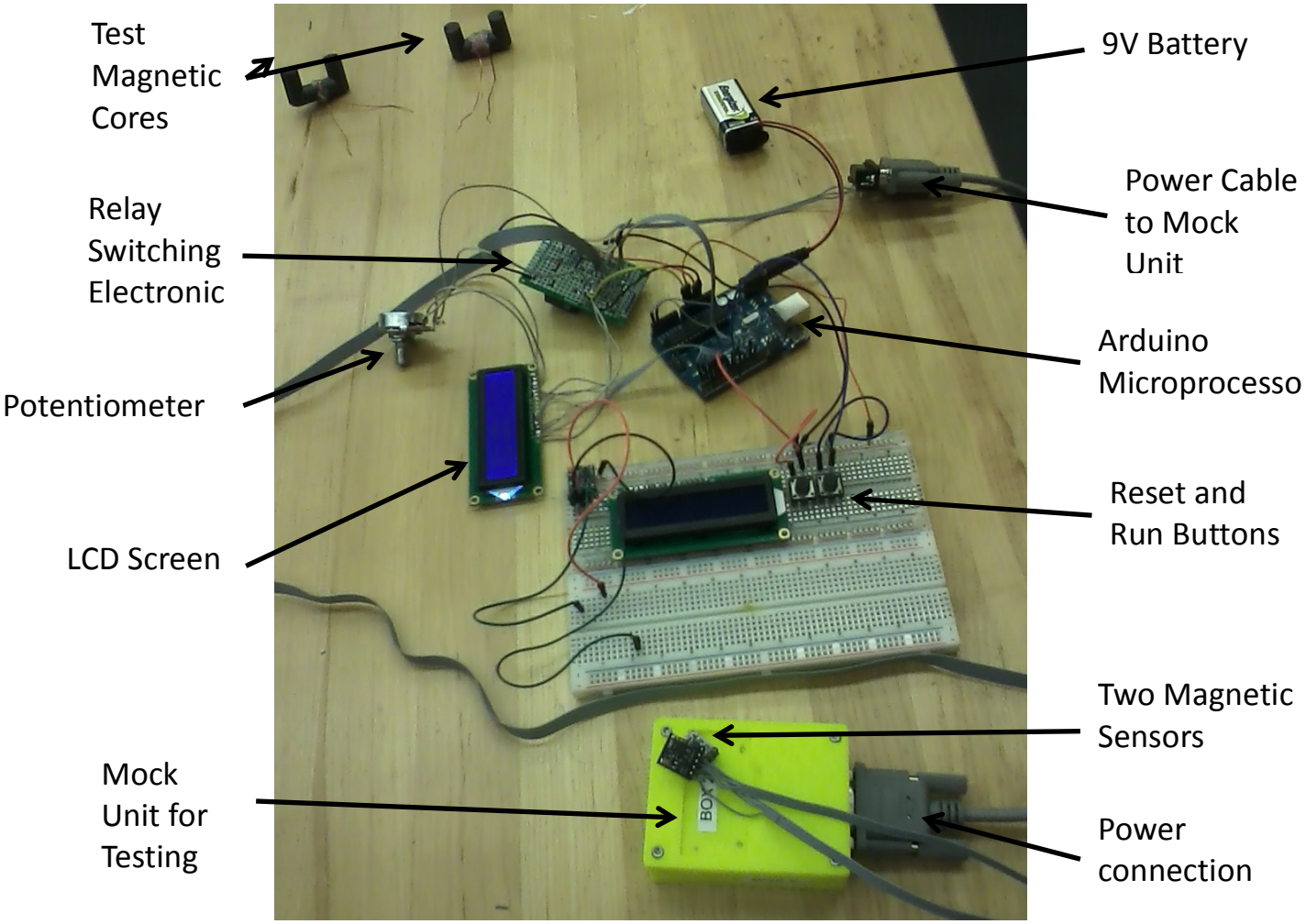

Figure 9 Prototype of design

Designing an efficient, foolproof, user-friendly device was a major goal of this project. An enclosure was chosen to house the Arduino, the PCB, two 9-volt batteries, a 2 x16 screen, a screen resolution potentiometer, specialty push buttons, a DB9 connection, a power switch, and all the wiring to connect them. The enclosure was modified to accommodate for the specific design.

Honeywell's mock units have a unique shape cut into the top of their lids, so the wand was designed to fit perfectly into the molding. A custom 3D enclosure was created to house the magnetic polarity chips used to read the field. The box was designed in AutoCAD Inventor and printed using a 3D

printer on campus. The wand is beveled on the bottom so that the enclosure fits perfectly into the cut out on top of the mock units. This ensures the chips are placed on the boxes exactly the same way every time giving maximum accuracy in readings with minimal human error. Labels were also added to all important buttons or knobs and a set of instructions was also added to improve accuracy and reduce human error.

Cost

A large focus was placed on ensuring the tool is affordable. Honeywell has investigated having such a unit designed by a third party and the quote came in at approximately $40,000. To keep the cost minimal, low cost commercially available parts were chosen to be used.

Arduino UNO	$23.25
Potentiometer	$1.12
Backlit LCD screen	$9.99
DB9 cable	$4.00
USB cable	$4.67
Push buttons (2)	$9.90
Enclosure	$39.10
Wand	$16.09
Printed circuit board	$18.05
Relay (2)	$3.90
Power switch	$0.95
Battery (2)	$4.00
Transistors (4)	$1.68
3 axis magnetic sensor (2)	$13.98
Total	**$150.68**

Figure 10 Cost breakdown

With a grand total of $149, the project came in $551 under the budget.

Testing

All testing of Honeywell's units are to be supervised. The Honeywell liaison, Bob Dearth, was present during all formal testing. For lower level testing, the prototype cores discussed above were used.

The set of data below was collected during the final set of tests. The readings were confirmed with a gaussmeter.

Trial	Mock Unit 1 (mG)		Mock Unit 2 (mG)	
1	1058	28	-952	-34
2	1060	25	-947	-29
3	1055	27	-942	-31
4	1056	33	-938	-26
5	1054	27	-942	-31
6	1057	35	-934	-34
7	1054	29	-941	-31
8	1056	33	-938	-23
9	1053	35	-936	-23
10	1054	28	-933	-30

Figure 11 Testing of Honeywell mock units

By examining the test data, the first coil exhibits a standard deviation of 2.05 mG and the second coil exhibits a standard deviation of 5.57 mG. Testing was also performed using a metal plate to determine at what point in manufacturing the polarity testing should take place. At this point in development, it has been determined that accurate readings are not able to be taken through the metal housing.

Conclusion

Honeywell has approved the final design. The unit meets and exceeds the criteria that was originally defined. The tool fulfills Honeywell's request to find a way to check the polarity of coils assembled into larger units. Thorough testing was executed to determine which level of assembly testing needs to be performed in order to achieve the best results.

Appendix I: Operation Manual

1. Turn the unit on. The LCD screen will read “Unit Ready”.
2. Spin the dial on the right side of the device to adjust the brightness of the screen.
3. Connect the testing unit to the Honeywell unit using the DB9 connector cable.
4. Fit the wand into the indentation of the box. Adjust the wand so that it fits snuggly.
5. Press the “Reset” button. This will zero out the testing unit.
6. Press the “Run” button. This will read the direction and magnitude of each coil and display the results on the LCD screen.
7. Remember to press “Reset” before each test.

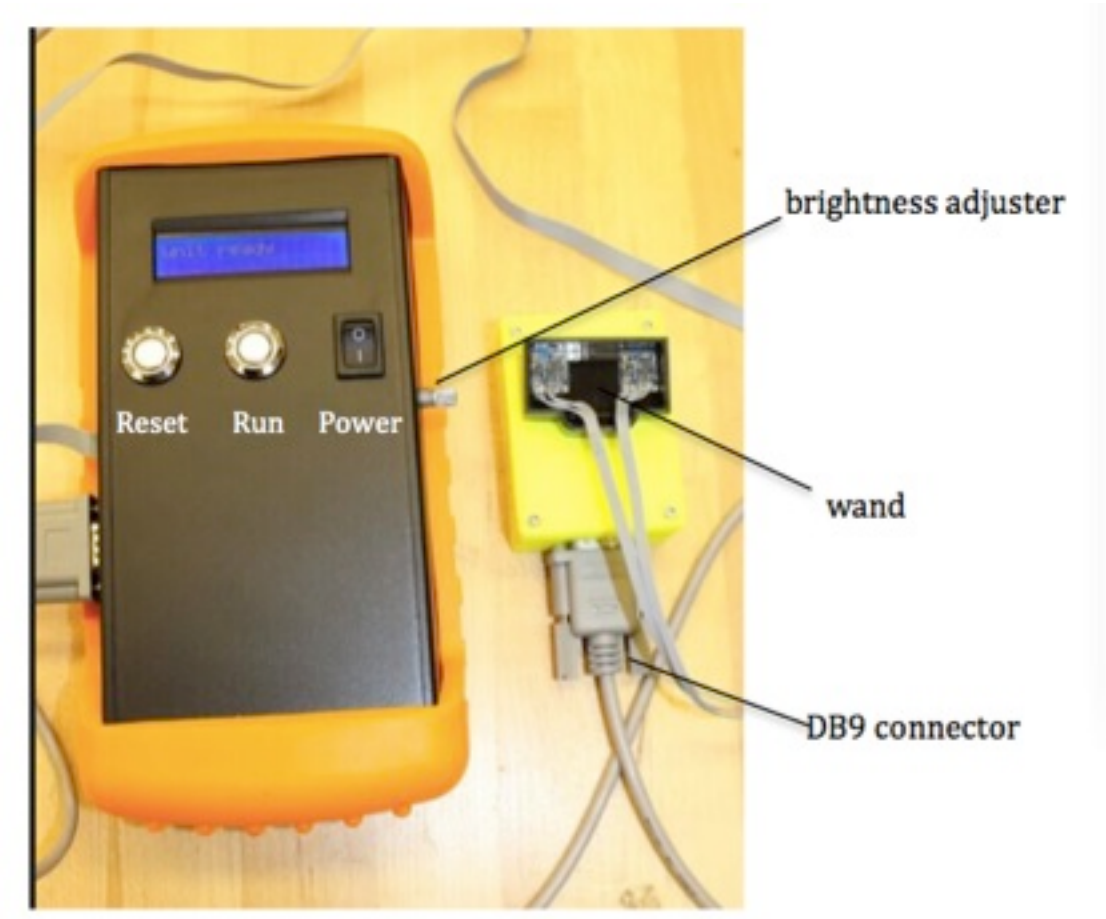

Figure 12 Testing setup

About the Reading

After each test, two sets of measurements are displayed on the screen. These readings correspond to each coil. The results represent the magnitude of the magnetic field in milligauss. The sign of the reading indicates polarity. The names of the coils are shown on the screen for easy detection of a mistake.

Below are some examples of the output given by the tool.

Figure 13 Example output (1)

Figure 14 Example output (2)

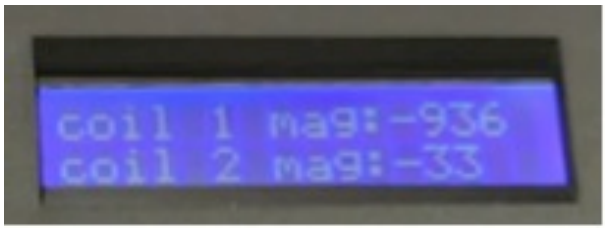

Figure 15 Example output (3)

Appendix II: Alternative Designs

The initial design was based on the two coils being coupled much like a transformer and using a DC to AC converter. By powering one coil, current would be induced in the other coil and the LEDs would flash in or out of phase with each other depending on current flow. As the project progressed, it was determined that the coils were shielded in a way that would not allow for this method of powering the coils.

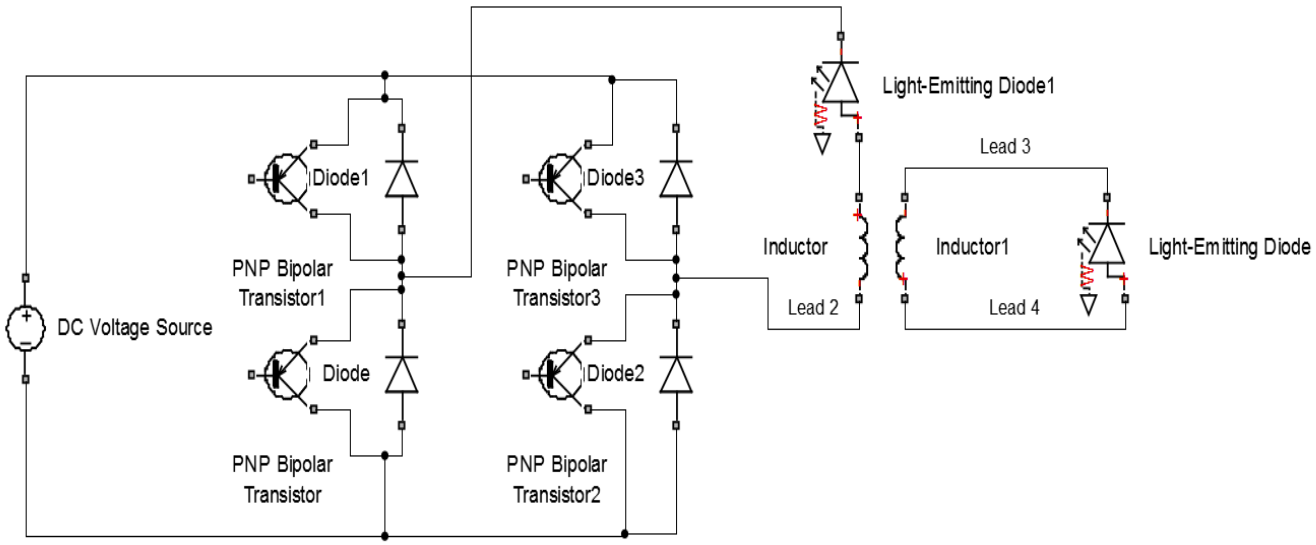

Figure 16 Initial design

Appendix III: Other Considerations

The polarity checking tool will serve as a base model for a similar testing unit. Adaptations will be made based on the specific configuration of the coils. For this reason, the tool was designed to be easily adaptable. The Arduino platform is extremely easy to access, and the code was written with consideration given to the user who would like to make adjustments. The current provided to the coils is also very easily raised if a stronger field is required or lowered if it is found that the current is damaging the coils.

Other adaptations could address issues including:

- Reading the magnetic field through metal housing
- Examining effects from surrounding equipment on a magnetic field
- Redesigning for differing coil configurations
- Editing the format of data reporting
- Reducing the size of the testing unit
- Powering the testing unit from a wall outlet

Appendix IV: Arduino Code

```
// Reference the I2C Library
#include <Wire.h>

// Reference the HMC5883L Compass Library
#include <HMC5883L.h>
#include <LiquidCrystal.h>

// Store our compass as an object.
HMC5883L compass;
 LiquidCrystal lcd(5, 4, 3, 2, 1, 0);

int final1,final2,s,w,run,magx1,magx2 =0;

// Out setup routine, here we will configure the microcontroller and compass.
void setup()
{
  pinMode(6, INPUT);
  pinMode(7, INPUT);
  pinMode(8, OUTPUT);
  pinMode(9, OUTPUT);
  pinMode(10, OUTPUT);
  pinMode(11, OUTPUT);

// Start the I2C interface.
Wire.begin();
 lcd.begin(16, 2);
 digitalWrite(10,HIGH);//turn on sensor 1
 digitalWrite(11,HIGH);//turn off sensor 2
  compass = HMC5883L(); // Construct a new HMC5883 compass.

  compass.SetScale(1.3); // Set the scale of the compass to 1.3Ga
  compass.SetMeasurementMode(Measurement_Continuous); // Set the measurement mode to
Continuous
}

// Our main program loop.
void loop()
{
  if(digitalRead(6)==LOW)
  {
    run=1; //start program
  }
```

```
  else
  {
    lcd.setCursor(0,0);
    lcd.print("unit ready"); //display start screen
  }

 /* loop that checks coils */
 while(run==1)
  {
    digitalWrite(10,HIGH);//turn on sensor 1
    digitalWrite(11,LOW);//turn off sensor 2
    w= valuerefresh(); //will have to have function for each sensor and pin connections change
    digitalWrite(8, HIGH); //  (change pin) connect relay 2 to battery, //relay operate 20ms, release
10ms
    digitalWrite(9, LOW); // relay 2 power coil 1
    delay(300);
    magx1= valuerefresh(); //get values from coil 1
    final1=magx1-w; //magnitude of coil 1
    digitalWrite(8, LOW);
    digitalWrite(10,LOW);//turn off sensor 1
    digitalWrite(11,HIGH);//turn on sensor 2
    delay(1000);
    w=valuerefresh();//will have to have function for each sensor and pin connections change
    digitalWrite(9, HIGH);
    digitalWrite(8, HIGH); // relay 2 power coil 2
    delay(300);
    magx2= valuerefresh(); //get values from coil 2
    final2=magx2-w; //magnitude of coil 2
    digitalWrite(8,LOW); //disconnect coil power
    digitalWrite(9,LOW);

/*-------------------------------------------------------*/

/* display results for coils and wait for program reset */
  lcd.clear();
   while(1)
 {
  lcd.setCursor(0,0);
  lcd.print("coil 1 mag:");lcd.print(final1);
  lcd.setCursor(0,1);
  lcd.print("coil 2 mag:");lcd.print(final2);
  delay(100);
  if(digitalRead(7)==LOW)
  {
```

```
    lcd.clear();
    break;
  }
 }
/*-------------------------------------------------------*/
  run=0;//break while loop
  }
}

int valuerefresh()
{
   MagnetometerScaled scaled = compass.ReadScaledAxis();
  // Values are accessed like so:
  int mGaussXAxis = scaled.ZAxis;// (or YAxis, or ZAxis)
  return mGaussXAxis;
}
```