



Prifysgol Cymru
Y Drindod Dewi Sant
University of Wales
Trinity Saint David

Industrial Metal Detection Discrimination

By

Michael Sam O'Leary

Supervisors: Mr Gareth Jones & Dr Gary Jones

Project submitted as part of the requirements for the
award of BEng (Hons) Electrical & Electronic
Engineering Degree.

April 2018

Student Declaration

I, Michael Sam O'Leary declare that I am the sole author of this Project; that all references cited have been consulted; that I have conducted all work of which this is a record, and that the finished work lies within the prescribed word limits.

This work has not previously been accepted as part of any other degree submission.

Signed:.....

Date:.....

Form of Consent

I Michael Sam O’Leary, hereby consent that my Project, submitted in candidature for the BEng.(Hons) Electrical and Electronic degree, if successful, may be made available for inter-library loan or photocopying (subject to the law of copyright), and that the title and abstract may be made available to outside organisations.

Signed:.....

Date:.....

Abstract

Industrial metal detection systems are important in industry to protect consumers and to protect valuable industrial process plant equipment. Industrial metal detectors detect and remove metal contamination in products where the metal may cause harm to persons or damage to plant equipment. Some metals may not damage plant equipment or affect the product that it is contained in while others might. Therefore it would be more efficient to only remove the metals which are damaging than to remove all metals creating large amount of reject waste. An industrial metal detector discriminator is a solution to this where it can discriminate against ferrous and non-ferrous metals.

This report contains research on two different types of industrial metal detection methods and a solution to metal discrimination using an 8-bit microcontroller. A circuit is designed and built onto a printed circuit board with simulation testing and practical testing with successful results. The circuit design and software structure is documented and explained step by step. An evaluation and conclusion is carried out detailing the results found and possible future projects are included.

Contents

Chapter 1 – Introduction.....	8
1.1 Project Background.....	8
1.2 Aims & Objectives	8
Chapter 2 – Planning.....	9
Chapter 3 – Research & Review of Literature.....	10
3.1 Metal Detection Technology Overview	10
3.1.1 Pulse Induction.....	10
3.1.2 Balanced Coils	12
3.2 Market Research	14
3.2.1 Tectron 8050	14
3.2.2 Velleman CS150	15
3.3 Technology Selection	16
3.4 Coil Selection.....	16
3.5 Solution	17
3.6 Summary	20
Chapter 4 – Design & Implementation	21
4.1 Circuit Design	21
4.1.1 Channel 1 – High Gain Signal Conditioning.....	21
4.1.2 Channel 2 – Unity Gain Signal Conditioning.....	22
4.1.3 Attenuation/Channel 3 & Voltage Scaling	23
4.1.4 PIC Analogue Input protection & Anti-Aliasing.....	24
4.1.5 Digital Inputs	24
4.1.6 Digital outputs.....	25
4.1.7 1.25v Voltage Reference	25
4.1.8 Delay/Active Timers	26
4.1.9 Power Supply	27
4.2 PIC Microcontroller Selection	28
4.3 PCB Design	30
4.4 Software Design	32

4.4.1 Main Program	32
4.4.2 External ISR	35
4.4.3 Discriminator ISR.....	36
4.4.4 Delay Timer	39
4.4.5 Active Timer	41
4.5 Prototype Build	42
4.6 Final Build.....	43
4.7 Project Testing	44
4.8 Build Cost	50
Chapter 5 – Evaluation.....	52
Chapter 6 – Conclusion	53
Chapter 7 – Reflection	54
Chapter 8 - References.....	55
Chapter 9 – Appendix 1 – Software Code.....	56

List of Tables

Table 1 – TP6/TP8 Test Piece Signals	18
Table 2 – PIC Microcontroller Initial Specification.....	28
Table 3 – Minimum PIC Requirements	29

List of Figures

Figure 1 – Pulse Induction Transmitter Signal	11
Figure 2 – Pulse Induction Decay Curve.....	11
Figure 3 Pulse Induction Fe vs. NFe Decay Curve	11
Figure 4 – Balanced Coil Arrangement [3]	12
Figure 5 – Balanced Coil Metal Signal	12
Figure 6 – Balanced TX & RX Waveforms [3]	13
Figure 7 – Ferrous Signal [3]	13
Figure 8 – Non-ferrous Signal [3]	13
Figure 9 – Tectron 8050 Discriminator [4]	14
Figure 10 – Velleman CS150 Metal Discriminator [14]	15
Figure 11 - Velleman CS150 Display [6]	15
Figure 12 – Velleman CS150 Tuning Controls [6].....	15
Figure 13 - Eriez Magnetics E-Z TEC 9000 C Coil	16
Figure 14 - E-Z Tec 9100 Receiver Mod/Demodulator Schematic [13].....	17
Figure 15 – Oscilloscope Screenshot TP6/TP8	17
Figure 16 - Bar Graph showing Amplitude & Phase Voltage	18
Figure 17 –TP6/TP8/MS/DS Oscilloscope Screenshot - 3mm Fe Sphere	19
Figure 18 Circuit Block Diagram	20
Figure 19 – Op-Amp buffer Schematic.....	21
Figure 20 – Channel 1 Schematic.....	21
Figure 21 – Channel 2 Schematic.....	22
Figure 22 – Voltage Scaling Schematic.....	23
Figure 23 – PIC Analogue input Protection Schematic	24
Figure 24 – Digital Input Schematic	24
Figure 25 – PIC Output Schematic	25
Figure 26 – 1.25v Reference Voltage Schematic.....	26
Figure 27 – Delay/Active Timer Schematic	26
Figure 28 – Power Supply Schematic	27
Figure 29 – PIC16F1829 Specifications	29
Figure 30 – PCB Design – Top Copper	30
Figure 31 – PCB Design – Bottom Copper.....	30
Figure 32 – PCB Design – Proteus 3D View	31
Figure 33 – PCB Design – Populated PCB.....	31
Figure 34 – Software Main Loop Flowchart.....	32
Figure 35 – Software Code – Main While Loops.....	33
Figure 36 – Software Code - Main	34
Figure 37 – Software Code – External ISR.....	35
Figure 38 - External ISR Flowchart	35
Figure 39 – Discriminator ISR Flowchart.....	36
Figure 40 – Software Code – Discriminator ISR	37
Figure 41 – Software Code – Discriminator ISR 2	37

Figure 42 – Software Code – Discriminator ISR 3	38
Figure 43 – Software Code – Discriminator ISR 4	38
Figure 44 – Fe Delay ISR Flowchart	39
Figure 45 – Software Code – Delay Timer ISR.....	40
Figure 46 – Fe Active ISR Flowchart	41
Figure 47 – Software Code – Fe Active ISR	41
Figure 48 – Prototype Build	42
Figure 49 – Prototype Build 2	42
Figure 50 – Control Panel 1.....	43
Figure 51 – Control Panel 2.....	43
Figure 53 - Eriez Magnetics E-Z TEC 9000 Control Panel & Discrimination Indicators	43
Figure 54 – Simulation Testing – A Sig	44
Figure 55 – Simulation Testing – A Sig Oscilloscope Result	44
Figure 56 – Simulation Testing – Software Simulation	45
Figure 57 – Asig Testing	46
Figure 58 – A&P Sig 5.5mm Fe Sphere.....	46
Figure 59 – A&P Sig 10mm Fe Sphere.....	47
Figure 60 – A&P Sig 12mm Fe Sphere.....	47
Figure 61 – A&P Sig 40mm NFe cube.....	48
Figure 62 – A&P Sig 30mm Fe cube	48
Figure 63 A&P Sig 1.0mm NFe sphere	49

Chapter 1 – Introduction

1.1 Project Background

The author is a Metal Detection Service/Production Engineer for an industrial separation equipment manufacturing company named Eriez Magnetics. Industrial metal detection systems play a big role in the quality control of manufactured goods and protection of valuable industrial machinery by the use of detecting and removing metal contamination. Just some of the Industries metal detectors serve include food, pharmaceutical, recycling, ore mining and textiles. In the food and pharmaceutical industries metal detectors need to be the most sensitive in order to detect and remove the smallest pieces of metal possible to protect the consumer. In the ore industries metal detection equipment may not need to be as sensitive but needs to find large pieces of metal in order to protect expensive process equipment such as crushers.

After a detection there are a few ways to remove the contaminant, either manually or automatically depending on the application. For manual removal such as in ore mining where the product/contaminant is too large to be automatically removed, the metal detector will stop the process and may sound an alarm until the contaminant is removed and the system is reset by the operator. For an automated system where the product/contaminant is small or light an automated reject mechanism can operate to separate the contaminant into a reject bin. A problem with manual removal is the loss time between the process stopping and being reset again by an operator. A problem with automatic removal is good product surrounding the metal also getting rejected.

Industries often request a metal detector that can discriminate between different types of metal. The reason for the discriminator can vary such as to ignore or detect only certain types of metals, data logging or for separating different types of metals from one another. Currently Eriez Magnetics do not have a product which is capable of metal detection discrimination.

1.2 Aims & Objectives

The aim of this project is to increase Eriez Magnetics customer base by providing a solution to industrial metal detection discrimination. The product to be designed throughout this report is to be an extension for an existing Eriez metal detector that is already out on the market today. The product must include two separate 'detect' relays that operate for ferrous (Fe) and non-ferrous (NFe) detections without affecting the original metal detectors settings and sensitivity. By using relays the customer can decide what they would like to do with the discriminator. For example, if they only want to detect NFe metals and ignore Fe metals, they only need to connect their process to the NFe relay. However, if they would like two separate reject bins then can connected their corresponding reject mechanisms to the two relays.

The project aims to research into current technology available on the market and to find a solution using a microcontroller. As the discriminator will be an extension for an existing metal detector the best metal detector model will need to be selected. Extensive simulation and testing is to be carried out with an analysis and conclusion on the reliability and feasibility of being put into production for real world use.

Chapter 2 – Planning

Below is a Gantt chart of the plan which was followed throughout the project;

Task Name	Duration	Start	Finish
Research	28 days	Mon 18/09/17	Sun 15/10/17
Market Research	7 days	Mon 18/09/17	Sun 24/09/17
Metal Detection Technology Research	7 days	Mon 25/09/17	Sun 01/10/17
Feasibility Study	14 days	Mon 02/10/17	Sun 15/10/17
Implementation	195 days	Sun 10/09/17	Fri 23/03/18
Circuit Design	30 days	Mon 16/10/17	Tue 14/11/17
Microcontroller Selection	7 days	Wed 15/11/17	Tue 21/11/17
Program Design	7 days	Wed 22/11/17	Tue 28/11/17
Write Program	60 days	Wed 29/11/17	Sat 27/01/18
Simulation/Debug	7 days	Sun 28/01/18	Sat 03/02/18
Order Required Components/Hardware	1 day	Sun 04/02/18	Sun 04/02/18
Build Prototype	3 days	Mon 05/02/18	Wed 07/02/18
Test/Debug	14 days	Thu 08/02/18	Wed 21/02/18
Design PCB	5 days	Thu 22/02/18	Mon 26/02/18
Build Panel	1 day	Tue 27/02/18	Tue 27/02/18
Extensive Testing	14 days	Wed 28/02/18	Tue 13/03/18
Analysis	10 days	Wed 14/03/18	Fri 23/03/18
Analysis of Performance	3 days	Wed 14/03/18	Fri 16/03/18
Possible Improvements	7 days	Sat 17/03/18	Fri 23/03/18
Finish	14 days	Fri 13/04/18	Thu 26/04/18
Write Final Report	14 days	Fri 13/04/18	Thu 26/04/18

Chapter 3 – Research & Review of Literature

This chapter includes information on research carried out and a literature review on the information found when investigating the possibility of industrial metal detection discrimination. To understand if the project is even achievable, brief background research on how metal detectors operate is included and the best type of technology is selected with reasoning. Market research is carried out to see if there are any solutions already available on the market and to gain ideas from those products. The overall goal of this chapter is to investigate if a solution can be obtained.

3.1 Metal Detection Technology Overview

There is a large variation in the types of metal detectors that are designed for specific operations depending on the environment they are operating in, size and type of the product and/or size of the contamination that requires detection. As this report is focusing on metal detection discrimination the general technical operation of metal detectors will be kept relatively simple and it should be stressed that the following information is an overview of metal detection and there may be variables that have been omitted in order to concentrate on the best type of metal detector to choose for discriminating.

Despite the variations of metal detection technologies they can all be generalised in terms of their structure and operation. All metal detectors have a transmitter (TX) which is in the form of a wound coil that generates an electromagnetic field when energised. In some cases the transmitter coil can also be used as a receiver (RX) coil switching from TX to RX to create a coil as what's known as transmit-receive (TR). In most cases however, a separate RX coil or multiple RX coils are used which couple the signals generated by the TX. The effect of metal objects passing through the electromagnetic field causes a change in the received signal. There are two main technologies which Eriez Magnetics metal detectors use which are known as 'pulse induction' and 'balanced coils'.

3.1.1 Pulse Induction

The transmitter of a pulse induction system consists of an electronic switch that switches a current to the TX coil on and off rapidly via a MOSFET transistor. When the MOSFET is switched on a current is passed through the coil which creates a magnetic field around the coil. Once the MOSFET is switched off the magnetic field around the coil collapses. If a metal object is passed through the magnetic field small electric currents are induced into the metal known as eddy currents. When the field has collapsed the metal object still contains an electric current and therefore its own magnetic field for a short period of time. This decaying magnetic field around the metal object is then induced into the RX coil in the form of very small signals. These signals are then amplified and processed to give an output which gives the indication that a metal object has been detected. [1] The following diagrams show what the signal looks like when measured with an oscilloscope;

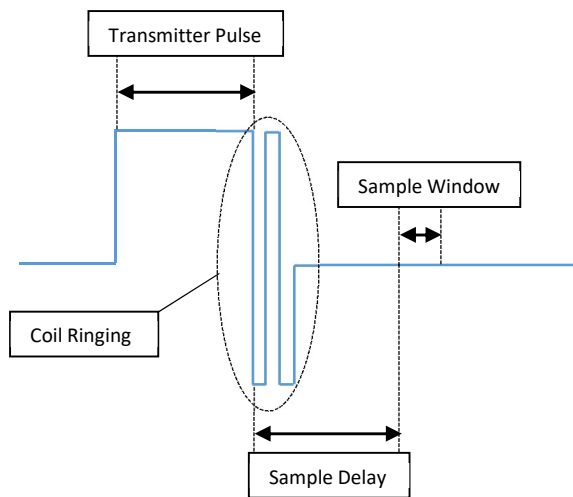


Figure 1 – Pulse Induction Transmitter Signal

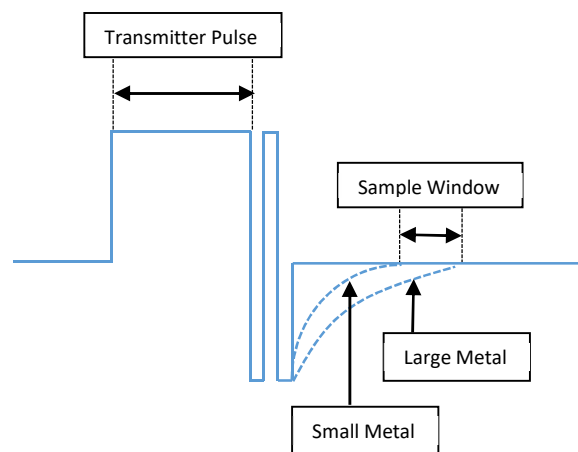


Figure 2 – Pulse Induction Decay Curve

The transmitter pulse is the duration of the TX coil being driven by current via a MOSFET. When the MOSFET switches off the coil oscillates or 'rings' until settling to 0v. Due to the ringing effect the signal to be captured is sampled at a point where the ringing has settled hence 'sample delay'. Once the sample delay has passed the RX captures any signal present for the duration of the 'sample window' which gets amplified and processed. The diagram in figure 1 shows the signal when there is no metal present within the field. Passing metal through the coils causes the ringing to be pulled up or down and if pulled up or down far enough to be captured in the sample window will generate a detect signal. The larger the size of the metal passing through the field the larger the effect due to the amount of energy the metal can hold, this is known as the 'decay curve'. [1]

Ferrous metals cause the inductance in the coil to increase creating a delay of the decay curve. To get the most accuracy from a microcontroller based discriminator the microcontroller must take the signal analysis as close to the decay curve as possible. This is difficult to achieve as amplification is required of 100 to 1000 times which will also amplify the noise in the signal. [2] The other problem is that the decay curve can change quite significantly depending on the design and installation of the coil. Surrounding metal work such as conveyor belt structures can cause a change in inductance which causes the decay curve to be delayed further. By moving the sample delay window further out discriminating becomes very difficult as the signal loses resolution. Below is a graph showing how ferrous and non-ferrous metals affect the decay curve;

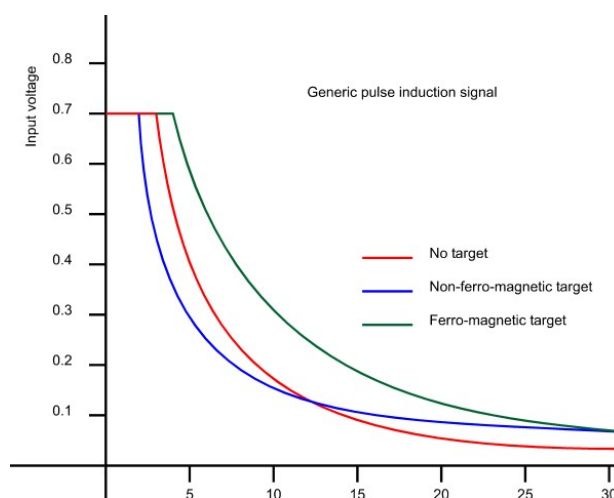


Figure 3 Pulse Induction Fe vs. NFe Decay Curve

3.1.2 Balanced Coils

Balanced Coil metal detectors use three coils which are wound around a 'tunnel'. A TX coil is situated in the middle while there are two RX coils that are spaced evenly either side of the TX coil. Rather than the TX coil being driven via pulse induction, balanced coils are driven by a continuous sine wave. The two outer RX coils are used to receive the signal generated by the TX coil and are connected in series so that their signals subtract from one another. The result of this subtraction means the received signals are in equilibrium or 'cancel out' which gives a signal that should be zero, hence the term balanced coils. [3] The image below shows this typical arrangement;

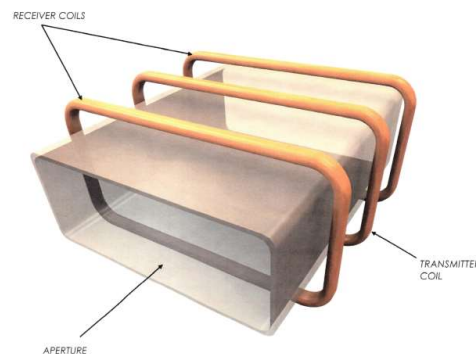


Figure 4 – Balanced Coil Arrangement [3]

An oscillator is used to generate the continuous sine wave used to drive the TX coil. This creates a high frequency electromagnetic field within the aperture. When metal is passed through the aperture it will cause changes in the balance of the coils which are amplified and processed to determine if metal is present within the metal detector. The balance of these coils can be affected by a particles conductivity as well as its magnetic properties. This means that all types of metals are detectable with these types of metal detectors. One of the great aspects of this arrangement is that the metal detector will receive two detect signals, one from each receiver coil. As it can be seen from the image below, as the metal approaches the first receiver coil a peak with positive amplitude appears. When the metal is at the centre of the aperture or directly below the TX coil the signal returns to its original balanced state, then once it reaches the second coil another peak but with negative amplitude appears. [3]

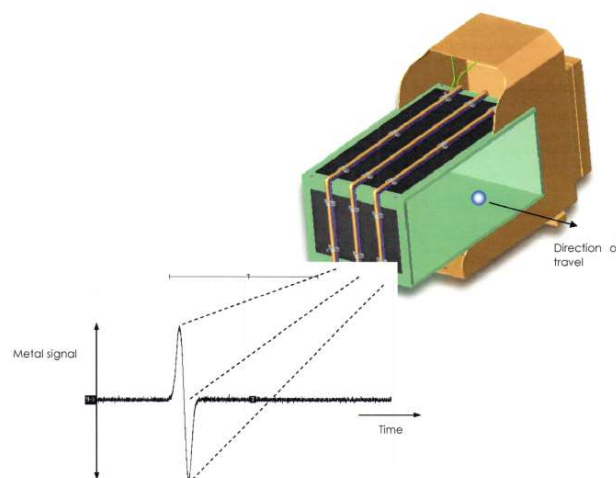


Figure 5 – Balanced Coil Metal Signal

Particles which have a high 'penetrability' such as ferrous metals causes the coupling factor between the TX and RX coils to increase. This is a 'reactive' effect and creates a signal with high amplitude. Particles with lower penetrability such as non-ferrous metals give a weaker signal with respect to amplitude. Non-ferrous metals are conductive materials which in a high frequency electromagnetic field absorbs some of the energy in the form of eddy currents. These eddy currents cause an energy loss which affects the resistance of the system. When metal passes through an RX coil the resistance increases and creates a phase shifted signal. [3] This is important when looking into metal discrimination as not only do different types of metals cause an amplitude change as seen in the pulse induction coils but they can also cause a phase shift. Looking at the relationship between amplitude and phase changes of different metals could be a very reliable way to discriminate between these metal types. The images below shows the relationship between the TX and RX signals;

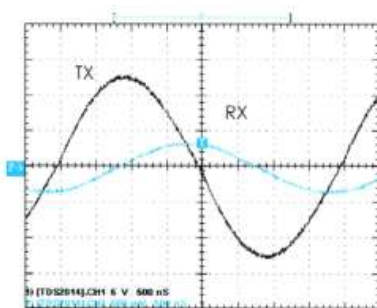


Figure 6 – Balanced TX & RX Waveforms [3]

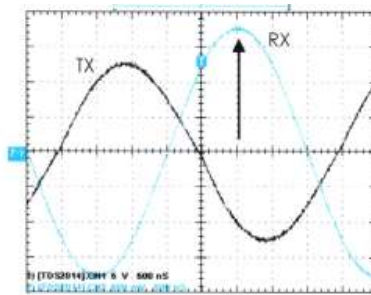


Figure 7 – Ferrous Signal [3]

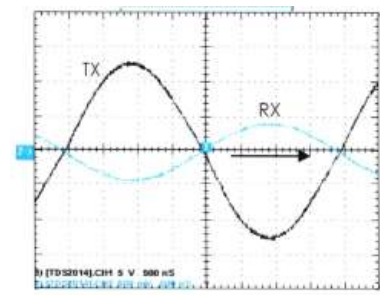


Figure 8 – Non-ferrous Signal [3]

It can be seen in figure 6 where the coil is in its balanced state that the RX signal is not quite zero, this is due to imperfections of the mechanical assembly and electrical balancing of the system but is the typical signal that is found in practicality. It is already known that ferrous metals have a good conductivity and are highly penetrable so they generate a signal with high amplitude and a clear phase shift. This is shown in figure 7 where the signal has been captured from a ferrous sphere. Non-ferrous metals have a fairly low penetrability but good conductivity so create a signal with small amplitude but with a large phase shift as shown in figure 8.

Alloys such as stainless steel has both poor penetrability and conductivity and therefore gives very small changes in both amplitude and phase which in turn means it is the hardest type of metal to detect and causes discrimination against this metal more difficult to achieve.

3.2 Market Research

Market research is to be carried out to see what is currently available on the market. As industrial metal detectors are quite specialised they are usually designed around customers' requirements. Although Eriez have customers who specify their requirements, finding out how other products operate can be a good way to find out what other customers' requirements are. Ideas from other products can be implemented and also possibly improved. Also, finding out what technologies are used to discriminate may help to design a solution.

3.2.1 Tectron 8050

The 8050 Metal Detector from Tectron Engineering was found using Google search and is the only industrial metal detector capable of discriminating that could be found. This metal detector is able to discriminate from non-magnetic and magnetic tramp metal. Tectron claims it was originally designed from the request of ore industries that couldn't use industrial magnets because the magnets also removed their good magnetic ore. Sometimes metal detectors are installed after a magnetic separator as a final check however, this metal detector is designed to be installed before an electromagnet.



Figure 9 – Tectron 8050 Discriminator [4]

When magnetic metal is detected by the 8050 a timed relay is used to power up an electromagnet to remove the magnetic metal contamination. When there is no metal present the magnet can also run at low power and only power-up when any non-magnetic tramp metal is found. The 8050 also has an "oversize" adjustment that if it detects metal that is too large for the magnet to remove a separate relay is activated that stops the conveyor belt so it can be removed manually. [4]

Although on the Tectron Engineering website they do not mention what type of metal detector is used, the same product is also sold by Process Automation who claim the 8050 'is the only pulsed eddy current metal detector that is able to differentiate between magnetic and non-magnetic metals' [5] Also, by looking at the design of the metal detector it looks as though it is a pulse induction type rather than a balanced coil type.

3.2.2 Velleman CS150

The Velleman CS150 is a typical ‘treasure hunter’ metal detector with a 3-tone discriminator that can distinguish between ferrous and non-ferrous metals. When the metal detector senses a metal object, the meter reading changes. The non-ferrous or ferrous indicators turn on or off and the detector will sound one of three tones. The reaction depends on which metal is detected. The higher the tone pitch, the stronger the detection. The CS150 has various control settings than can be tuned for specific applications. For example, the CS150 has two main modes “TR1” and “TR2”. TR1 is used to detect extreme differences in metals, such as between iron (ferrous) and gold (non-ferrous). TR2 is used to detect finer distinctions between metals, for example between aluminium and gold. The detector also has a potentiometer which is used to adjust the sensitivity of the discriminator [6]. The image below shows the tuning controls;



Figure 10 – Velleman CS150 Metal Discriminator [14]

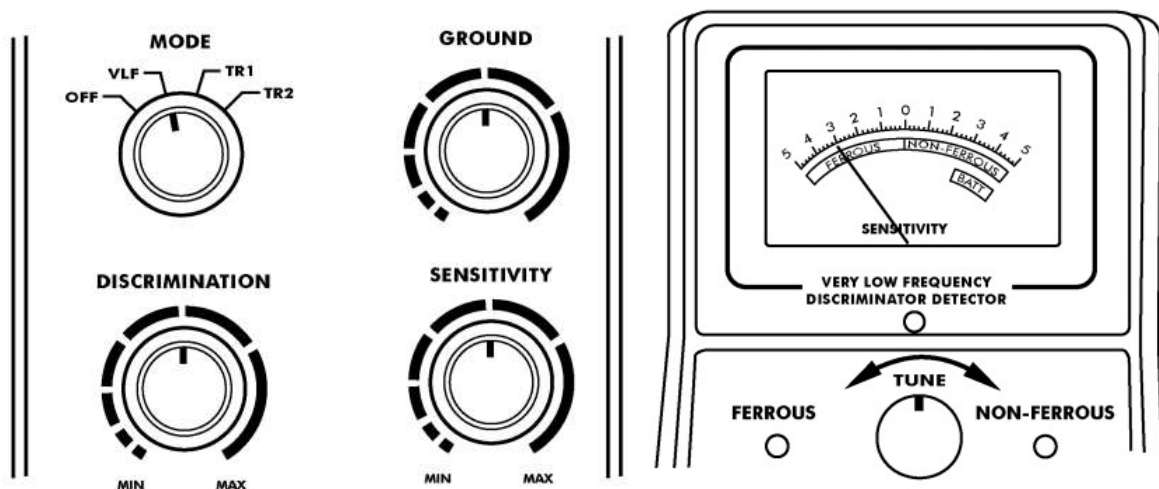


Figure 12 – Velleman CS150 Tuning Controls [6]

Figure 11 - Velleman CS150 Display [6]

If the detector is in mode TR2 then as the discrimination control is set higher the detector becomes more sensitive to the differences between large aluminium and gold pieces but some small valuable pieces such as coins or small rings might be overlooked. As the discriminator is set to high levels, the detector first does not detect small pieces of silver paper, then thick foil, and finally metal objects like pull tabs from aluminium cans. Each time the detector is used in a different area, the discrimination must be readjusted [6].

3.3 Technology Selection

Using the pulse induction technology to discriminate appears to be harder to achieve than the balanced coil technology because there is not much variation in the signal that is received. Research had found that ferrous metals cause the decay curve to be delayed more than non-ferrous metals however, it was also found that large metal also causes the decay curve to be delayed. Therefore if a large piece of non-ferrous metal was passed over the coil it may cause the decay curve to be significantly delayed causing it to be discriminated as ferrous when it is not. The ultimate test would be to build an amplifier circuit to amplify the signals as close to the decay curve as practically possible. These signals could be digitized by using an ADC on a microcontroller and the results from using different metals could be analysed.

Balanced coil technology uses two variables: phase and amplitude, discriminating even when different size metals are used is much more likely to be achievable. Using the same example as above when a large piece of non-ferrous metal passes over the coil it may give an amplitude the same as a ferrous but the phase shift should not match that of a ferrous metal signal or vice versa. Therefore, the technology that will be used to discriminate in this project will be balanced coils.

3.4 Coil Selection

As the project that is being proposed is to design and build a circuit which is capable of discriminating between metal types using balanced coil metal detection systems, the design and build of the metal detector itself will not be included and a balanced coil metal detector supplied by Eriez Magnetics will be used. The circuit to be designed will work in conjunction with one of Eriez Magnetics E-Z TEC 9000 series systems. There are many forms of coil designs for use with a 9000 series circuit board however a 'C' coil will be used which is a circular coil used for high pressure pipe or gravity fed systems. The reason for selecting this coil is due to its physical size being small and transportable so it can easily be taken to labs for testing;

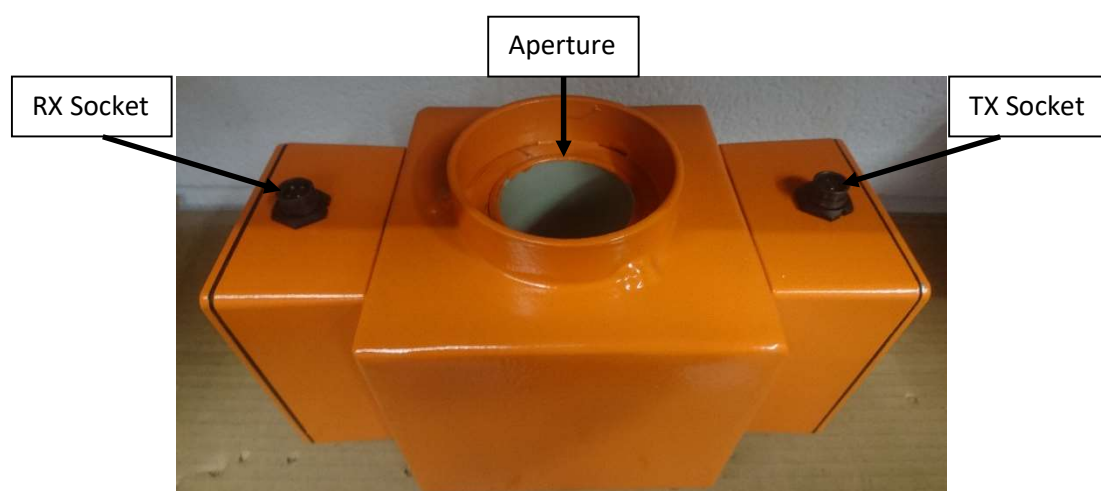


Figure 13 - Eriez Magnetics E-Z TEC 9000 C Coil

3.5 Solution

To find a solution for metal detection discrimination using an E-Z Tec 9000 the circuit schematics were analysed. It was found the receiver circuit incorporates two LM1496 modulator/demodulator ICs with their outputs fed to two TL082 $\pm 15\text{V}$ op-amps. This generates two signals on the board which are labelled as 'A Sig' and 'P Sig'. A Sig and P Sig are analog signals which represent the amplitude and phase receiver signal respectively. These two signals have their own veropin test points, TP8 is the amplitude signal and TP6 is the phase signal. The schematic is shown below;

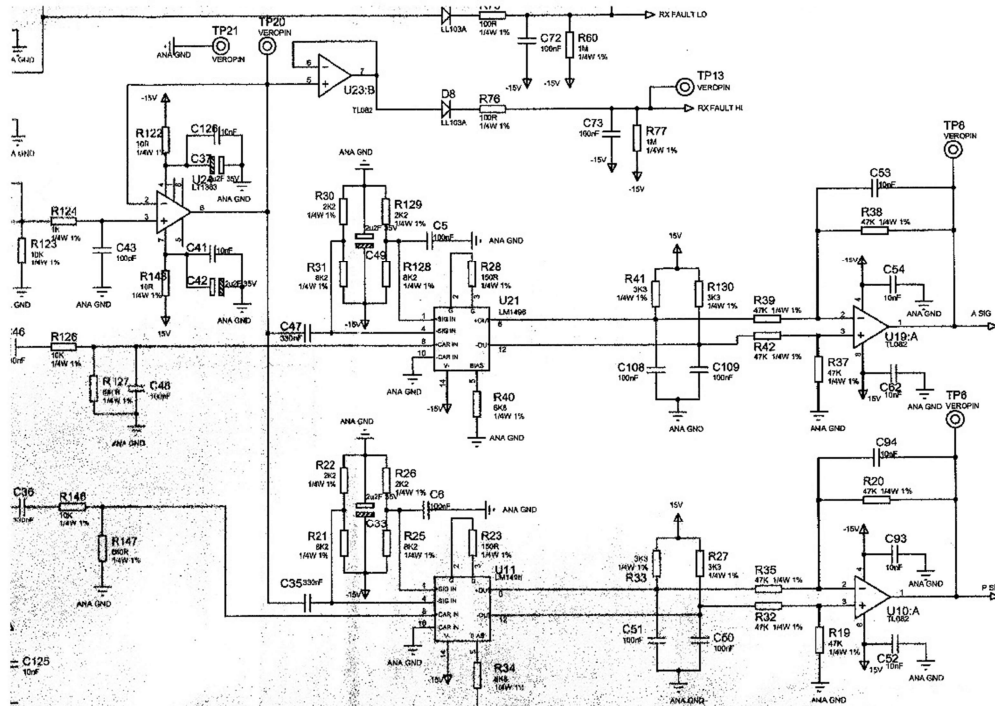


Figure 14 - E-Z Tec 9100 Receiver Mod/Demodulator Schematic [13]

An oscilloscope was connected to the two veropin test points to measure the signals when passing through different metal spheres through the aperture with a screen shot of the oscilloscope below;

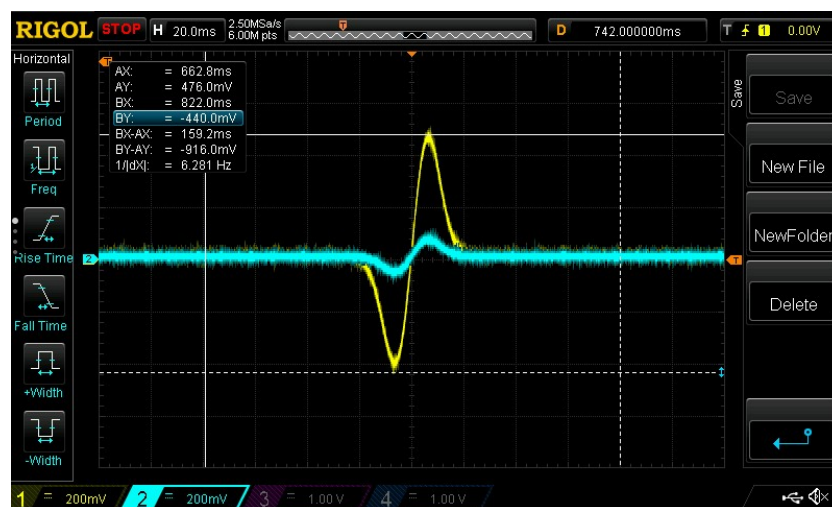


Figure 15 – Oscilloscope Screenshot TP6/TP8

The peak to peak voltage of both TP6 and TP8 was recorded for a range of test pieces with the results shown in the table below;

Test Piece	TP6	TP8
1.5mm Fe	60mVpp	42mVpp
1.5mm NFe	< 20mVpp	35mVpp
1.5mm ST/ST	< 20mVpp	< 20mVpp
3.0mm Fe	180mVpp	100mVpp
3.0mm NFE	70mVpp	135mVpp
3.0mm ST/ST	60mVpp	72mVpp
5.5mm Fe	780mVpp	110mVpp
5.5mm NFe	388mVpp	800mVpp
5.5mm ST/ST	48mVpp	630mVpp

Table 1 – TP6/TP8 Test Piece Signals

Metal spheres are the most reliable way of testing a metal detector as they give the same signal regardless of its orientation. Certified spheres were used to ensure the test pieces were definitely the correct material. To ensure a fair test the spheres were all dropped from 500mm height and as close to the centre of the aperture as possible. The reason for dropping the pieces from 500mm height is because the frequency of the RX signal is determined by the speed of the metal passing through the aperture. Also the centre of the aperture has the weakest sensitivity meaning it gives the smallest signal output. It was decided to use 316 stainless steel (ST/ST) as well as Fe/NFe pieces to see if ST/ST could also be discriminated. It can be seen that some of the results are recorded as <20mVpp, this means that the signal was so small that it couldn't be seen amongst the noise on the scope. When these signals are filtered and amplified they should give a measurable amplitude. A bar graph was created in excel to visualise the results;

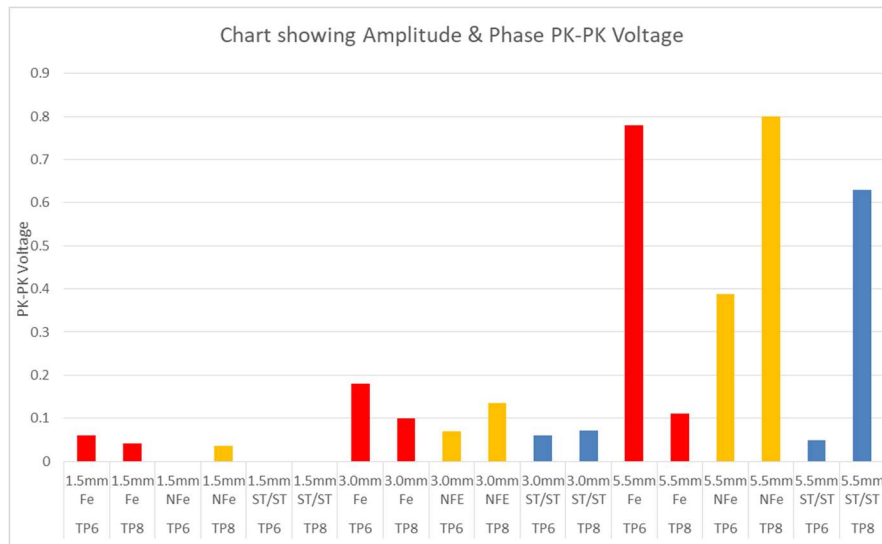


Figure 16 - Bar Graph showing Amplitude & Phase Voltage

It can be seen that Fe test pieces have a greater amplitude signal than phase and NFe have a greater phase signal than amplitude. The ratio between the amplitude and phase signals for both Fe and NFe are also quite symmetrical. ST/ST is less symmetrical giving a slightly greater phase signal with 3.0mm sphere but a much greater phase signal with a 5.5mm sphere. It was also noticed that although the phase signal for 5.5mm ST/ST is much greater than 3.0mm ST/ST the amplitude signal

for 5.5mm ST/ST was smaller than the 3.0mm amplitude signal. Due to this it has been decided to not discriminate against ST/ST as it proves to be too difficult, instead 316 ST/ST pieces could be discriminated as NFe as they give similar results which makes sense as 316 ST/ST is mostly non-magnetic.

It is now known there are two analogue signals that can be taken from the E-Z Tec 9000 board which will need to be amplified and filtered before inputting into a microcontroller in order to accurately measure the signals from the smaller test pieces. These signals are supplied from $\pm 15\text{v}$ op-amps. An ADC of a PIC microcontroller will be used to digitise these analogue signals. PICs are generally powered by $+5\text{v}$ and so the analogue input signals must be scaled accordingly. Any over voltage or negative voltage at one of the PIC pins will result in damage to the PIC so it must also be protected from this. Given the largest possible analogue signal is $\pm 15\text{v}$ this first needs to be attenuated. A voltage divider network can be used to attenuate the voltage from $\pm 15\text{v}$ to $\pm 2.5\text{v}$. This can then be scaled from $\pm 2.5\text{v}$ to $0\text{v} - 5\text{v}$ using an op-amp.

Three signal conditioning channels will be used with different amplification gains so that the voltage range from a few millivolts to 30v can be scaled between $0 - 5\text{v}$ while still getting enough resolution so that the signals can be digitised by an 8-bit microcontroller.

The 9000 control board uses various techniques to reduce false detections from noise induced signals such as 'twin peak detection' which will ignore any signals that do not give a sine wave like signal with two peaks. A 15v DC signal is generated when a metal detection takes place and therefore this DC detect signal will be an input into the microcontroller to utilise these techniques and so that the sensitivity controls are still operational. The microcontroller must give an output when this DC signal is pulsed high. Another 15v DC signal, labelled "Metal Sig", from the 9000 will be used which pulses when the analogue signals from the receiver circuit rise or fall below a set voltage threshold. This metal sig will be used to trigger the sampling code in the PIC so that it samples the correct signals. In the Oscilloscope screenshot, yellow trace is the A sig, light blue is P sig, pink is the metal sig and dark blue is the detect sig;

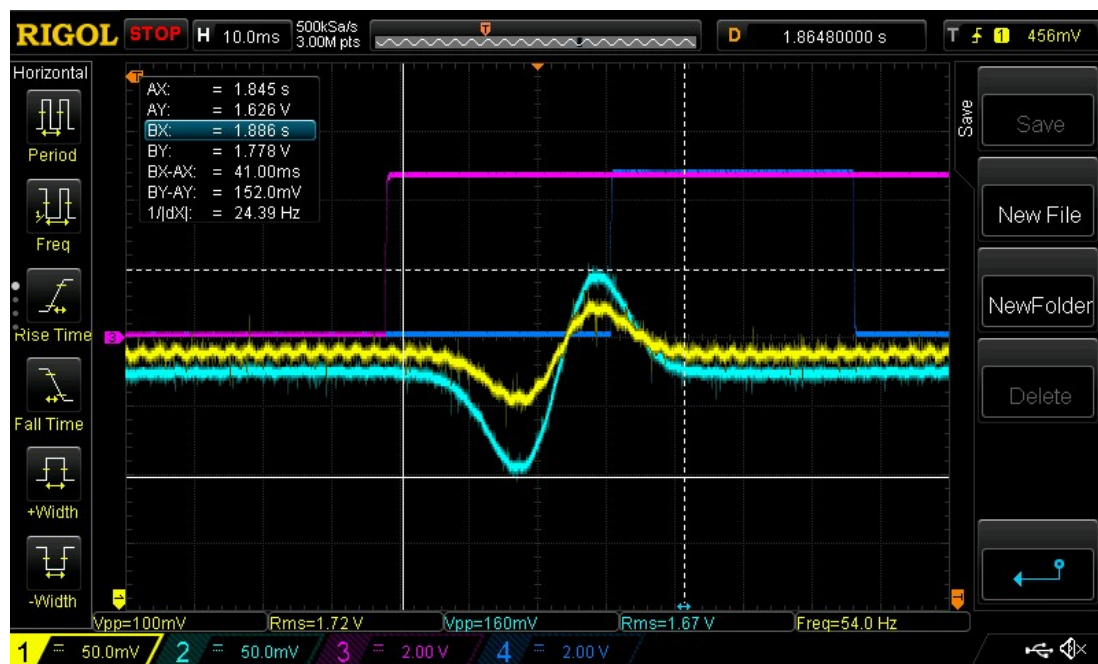


Figure 17 –TP6/TP8/MS/DS Oscilloscope Screenshot - 3mm Fe Sphere

It can be seen the pink trace rises high just before the analogue signals which will be used to tell the PIC to start sampling. The dark blue trace rises high just after the second peak which is the detection signal from the 9000 metal detector. The 9000 is in twin peak detection mode hence why the detect sig rises high after the second peak. If the metal detector is in single peak detection mode then this detect sig will rise high at the first peak. The detect sig could be used as an external interrupt to discriminate the previously sampled signals.

There will be two outputs which are Fe and NFe. These outputs will be amplified so that they can drive relays and panel indicators. Two potentiometers will be used to adjust an input voltage to control reject delay timers. The reject delay timers are used to delay the output to allow the detected metal to approach the reject mechanisms. Another two potentiometers will be used to control the active time of the outputs which sets the amount of time the outputs are active for.

A block diagram of the system is shown below;

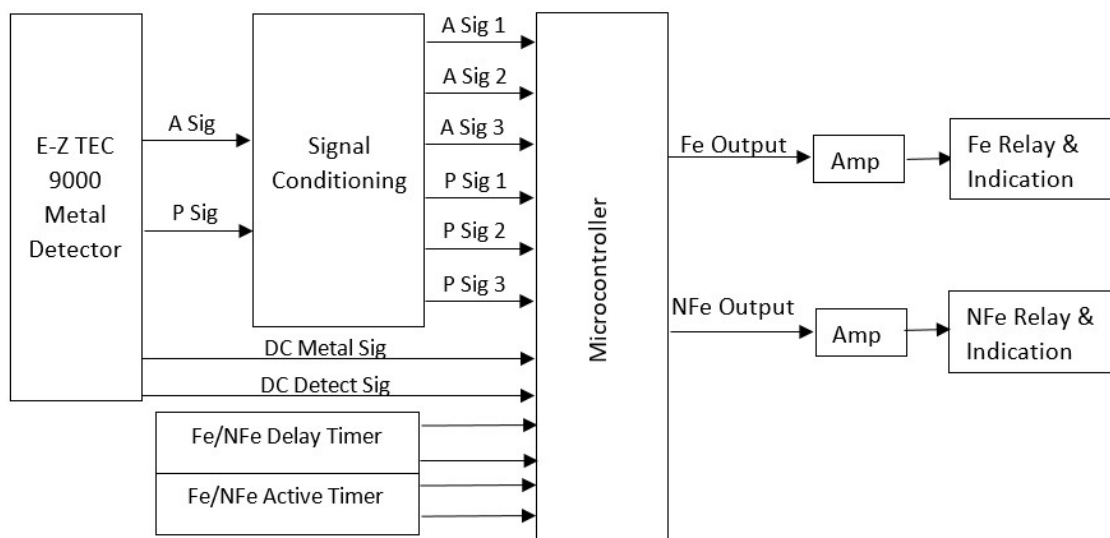


Figure 18 Circuit Block Diagram

A&P Sig 1 will be a high gain channel, A&P sig 2 will be a unity gain channel and A&P sig 3 will be an attenuated channel.

3.6 Summary

The purpose of this chapter was to investigate if industrial metal detection discrimination is possible. The basic principles of two different industrial metal detection technologies was included and discrimination using either of them appears possible. It was found that Fe and NFe metals do have an effect on metal detectors which could be used to discriminate. The balanced coil type seems the best option due to the two variables and using this technology seems more than plausible for this project to become successful.

The lack of industrial discrimination products on the market is quite surprising considering the amount of requests made from customers. The Tectron 8050 proves that this project could be accomplished and there is a place in the market for its purpose. It was however difficult to find much information on this discriminator in terms of its operation.

Chapter 4 – Design & Implementation

4.1 Circuit Design

The analogue signals to be taken from the 9000 control board can be very low voltage signals which are then amplified by a high gain amplifier. It is important that the discriminator circuit does not load these signals causing distortion that is then amplified giving unexpected results and thus affecting the performance of the original metal detector. In order to couple these signals into the discriminator circuit and to ensure the original signals are unaffected an op-amp buffer can be used. Op-amps have a high input impedance and low output impedance and so can be utilised as a voltage buffer by connecting the output directly to the inverting input. The op-amps output voltage will therefore follow the voltage at the non-inverting input without loading the source. The below screenshot shows the op-amp buffer for the A Sig;

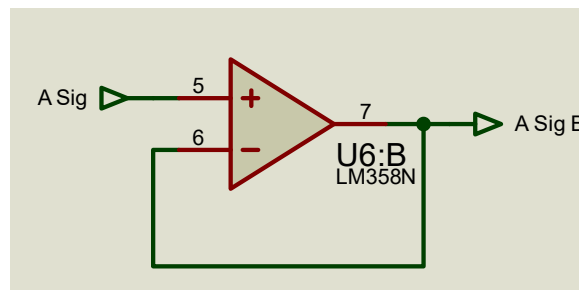


Figure 19 – Op-Amp buffer Schematic

The output of the buffer is labelled “A Sig B” with “B” referring to as “buffered”. It is done this way so that the buffered signal can then be split into 3 different signal conditioning channels: high gain channel, unity gain channel and attenuated channel. It should be noted here that everything that is designed for the A sig is exactly replicated for the P sig.

4.1.1 Channel 1 – High Gain Signal Conditioning

The circuit below shows the schematic for the high gain stage of the first channel;

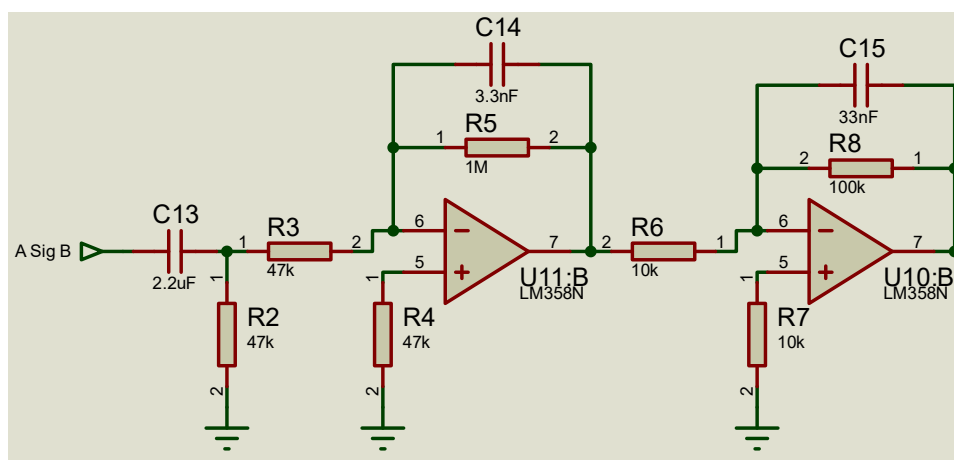


Figure 20 – Channel 1 Schematic

The 2.2uF and 47K resistor on the input is a high pass filter with a cutoff frequency of 1.54Hz. The cutoff frequency can be calculated by the equation below;

$$F_c = \frac{1}{2\pi * RC}$$

$$F_c = \frac{1}{2\pi(47 * 10^3) * (2.2 * 10^{-6})} = 1.54Hz$$

This high pass filter is used to block any DC bias voltage allowing only AC signals through. While measuring the peak to peak voltage of the analogue signals the frequency was noted which averaged around 20Hz. As drop tests were performed it was found that the larger test pieces had a higher frequency while smaller had lower frequency signals due to their mass and the speed at which they passed through the coils. It is decided to use a bandpass filter of approximately 2Hz - 40Hz throughout the design to allow signals through from all metal sizes.

The first op-amp has a gain of -21 which is calculated by the equation below;

$$Gain (A_v) = - \frac{R_f}{R_{in}}$$

$$Gain (A_v) = - \frac{1 * 10^6}{47 * 10^3} = -21.28$$

The 47K resistor from the non-inverting input to ground is to minimise DC offset by matching the input impedance. The 3.3nF capacitor across the feedback resistor sets the bandwidth limit of 46Hz which allows filtering and keeps the op-amp stable by preventing high frequency oscillations. The output is directly connected to another inverting op-amp with a gain of -10 and bandwidth of 44Hz. From input to output this gives a total gain of 212.8.

4.1.2 Channel 2 – Unity Gain Signal Conditioning

From the input to the output of the second signal conditioning channel there needs to be unity gain. As the voltage will be attenuated from $\pm 15V$ to $\pm 2.5V$ there will be a division of 6. Therefore, a gain of 6 is required before being attenuated so that the total overall gain of the channel is unity. Below is circuit schematic of channel 2;

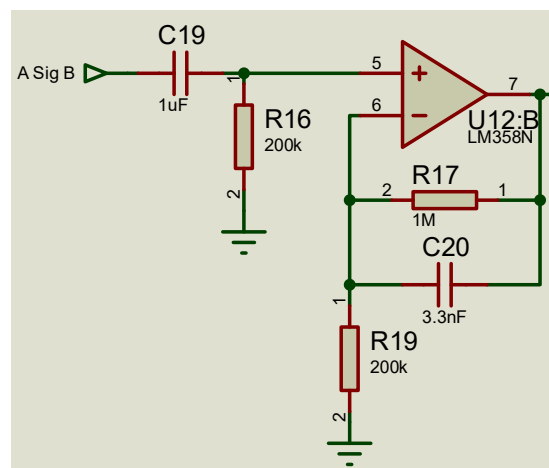


Figure 21 – Channel 2 Schematic

A 1uF blocking capacitor and 200k resistor to ground sets a high pass filter with a frequency cutoff of 0.8Hz. The 200k resistor also sets the input impedance which is matched to the 200k resistor to ground at the inverting input. This non-inverting op-amp gives a gain of 6 which can be calculated by the equation below;

$$Gain (Av) = 1 + \frac{Rf}{Rin}$$

$$Gain (Av) = 1 + \frac{1 * 10^6}{200 * 10^3} = 6$$

The 3.3nF capacitor sets the bandwidth limit of 40Hz.

4.1.3 Attenuation/Channel 3 & Voltage Scaling

The next stage is to attenuate the voltage from $\pm 15v$ to $\pm 2.5v$. Each amplifier stage uses the attenuation and voltage scaling circuit with the third channel uses this circuit without any initial amplification. The following equation is used to calculate the resistor values for a suitable voltage divider;

$$Vs \times \frac{R2}{R1 + R2} = Vout$$

Given that $Vs = 15v$, $Vout = 2.5v$, if $R2 = 20k$ then $R1 = 100k$.

The screenshot of the schematic below shows the voltage divider network connected to an inverting op-amp;

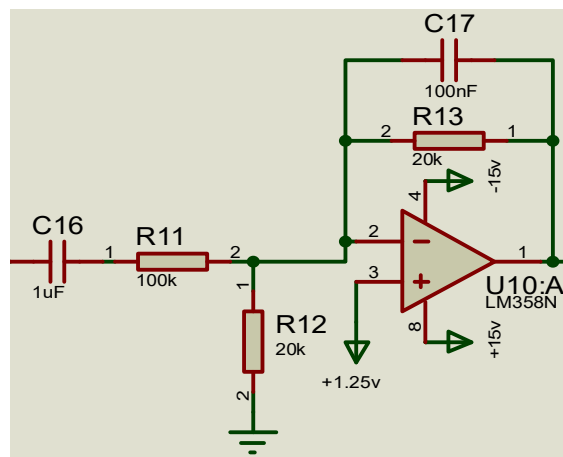


Figure 22 – Voltage Scaling Schematic

A 1uF blocking capacitor is used at the input with a cutoff of 1.33Hz. A reference voltage of 1.25v at the non-inverting input sets the baseline of the op-amps output at 2.5v. This means the voltage of $\pm 2.5v$ at the inverting input is scaled to 0 -5v which allows full resolution for the PIC. When there is no analogue signal at the inverting input the voltage output will be 2.5v which is a virtual "0v". The 100nF capacitor sets the bandwidth of 39.79Hz.

4.1.4 PIC Analogue Input protection & Anti-Aliasing

The last stage for the six analogue input channels is to include protection to prevent damage to the PIC from any overvoltage or negative voltage. The schematic below shows the protection circuit used;

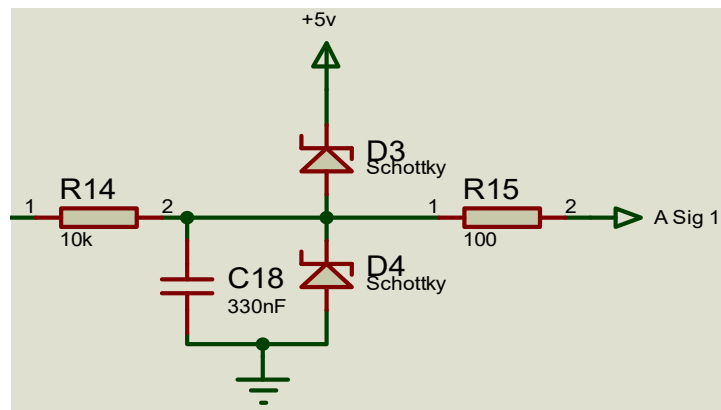


Figure 23 – PIC Analogue input Protection Schematic

Two Schottky diodes are connected to ground and to +5v supply. Schottky diodes have a low forward biased voltage drop and so any significant voltage over 5v will be shunted to the supply rail and any negative voltage will be pulled from ground. The 10k resistor and 330nF capacitor acts as an anti-aliasing filter to remove any unwanted high frequencies for the ADC. This provides a frequency cutoff of 48.23Hz. The 10k resistor also limits the current for when the Schottky diodes are forward biased. The 100R resistor is to limit the current into the ADC.

4.1.5 Digital Inputs

The two digital inputs are both 15v and so need to be attenuated to below 5v with protection to protect the PIC. The schematic below shows the protection circuit used;

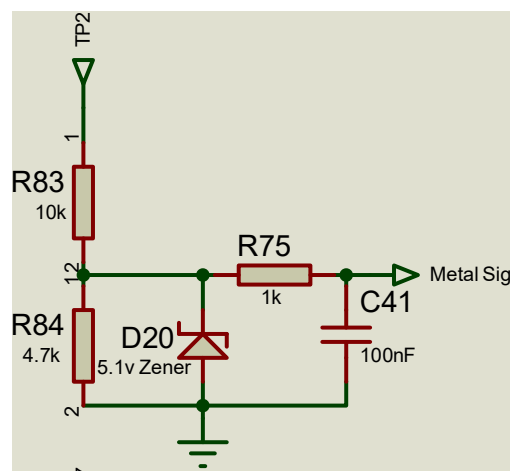


Figure 24 – Digital Input Schematic

The 10k and 4.7k resistors form a voltage divider with $V_{out} = 4.8v$. The 4.7k also acts as a pull-down resistor to ensure the PIC can distinguish between high and low and is not left floating. The 5.1v

Zener diode clamps the voltage in the unlikely event that the voltage exceeds this. The 1k resistor limits the current into the PIC while the 100nF adds some low pass filtering to minimise high frequency noise.

4.1.6 Digital outputs

The 5v outputs on the PIC need to switch a relay, an LED on the PCB and a panel mounted LED indicator. The amount of current a PIC can source is usually only around 20mA which is about the same amount of current to drive a single LED and so an amplifier circuit needs to be used. The schematic below shows the Fe output circuit which is identical for NFe output;

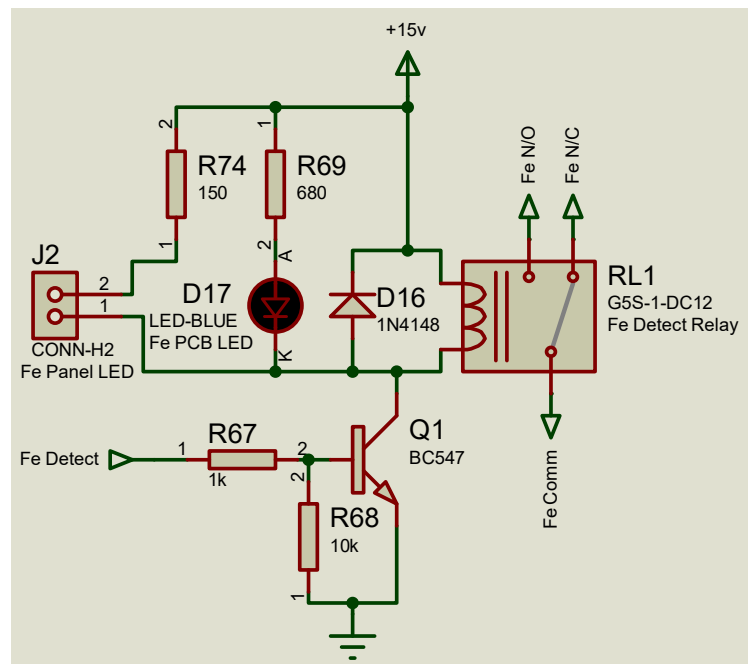


Figure 25 – PIC Output Schematic

The output from the PIC labelled “Fe Detect” is connected to the base of a transistor with a 1k resistor in series to limit the current. A 10k resistor is connected across the base and ground to ensure the transistor fully switches off when the PIC output goes low. When the transistor is switched on all the components connected to the collector are switched to ground turning them on. The relays chosen (Omron G5S-1) have a coil resistance of 360R and so at 15v consume 41.67mA. The two LED’s both consume around 20mA each. Therefore, the total amount of current to be switched is about 82mA. A BC547 transistor was chosen as it can switch up to 100mA.

4.1.7 1.25v Voltage Reference

As shown in the voltage scaling circuit a reference voltage of 1.25v is connected to the non-inverting input of the op-amps. A voltage divider circuit from +5v could be used to get 1.25v but its precision of the reference voltage would be limited to the precision of the two resistors in the voltage divider network. It was decided to use an ADC with 10-bit resolution which between 0 – 5v gives about 5mV per step. Therefore, it was decided to use a dedicated reference voltage IC with a precision better than $\pm 5\text{mV}$. It was found to be extremely difficult to find a high precision 1.25v through-hole

reference IC and so a surface mount IC has been chosen. The IC chosen is an MCP1501 by Microchip in an SOIC-8 package. A SOIC-8 to 8-DIP adapter can be used to keep the through-hole design style throughout the PCB design. The MCP1501 1.25v IC has an initial accuracy of 0.08% giving an accuracy of $\pm 1\text{mV}$. The MCP1501 is not included in the Proteus library and so a random 8-DIP component was chosen with the same pin out as a SOIC-8 to 8-DIP adapter. Below is the schematic for the voltage reference but as the IC shown is not the one that will be used the pin labels should be ignored.

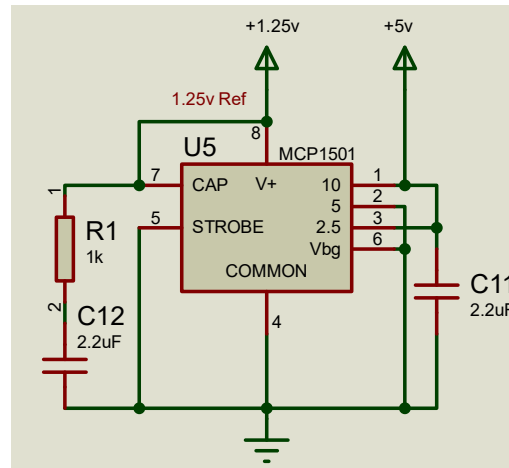


Figure 26 – 1.25v Reference Voltage Schematic

4.1.8 Delay/Active Timers

To set the timings for the delay and active timers, potentiometers are used which act as voltage dividers. The schematic is shown below;

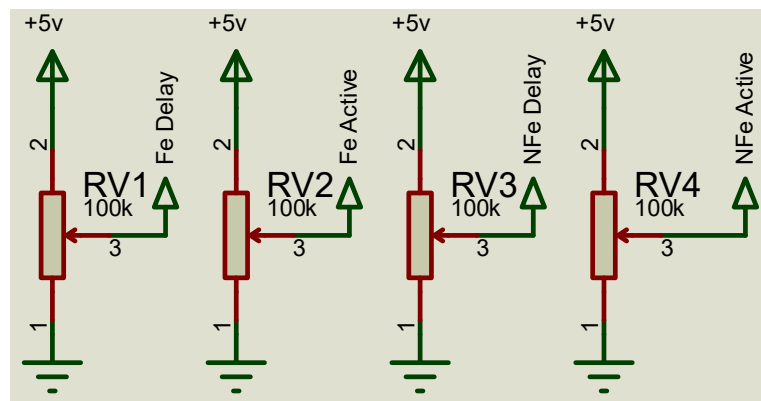


Figure 27 – Delay/Active Timer Schematic

By using 100K potentiometers the voltage at the PIC inputs can be adjusted from 0 – 5v. A 10-bit ADC will be used which gives 1024 steps and the timings will be adjustable between 0 - 30s for the delays and 0 – 10s for the actives. As $5\text{v}/1024$ gives about 5mV per step and $30\text{s}/1024$ gives about 30×10^{-3} . This means for roughly every 5mV the timing is changed by 30ms for the delays and about 10ms for the active timers.

4.1.9 Power Supply

Below is the schematic for the power supply;

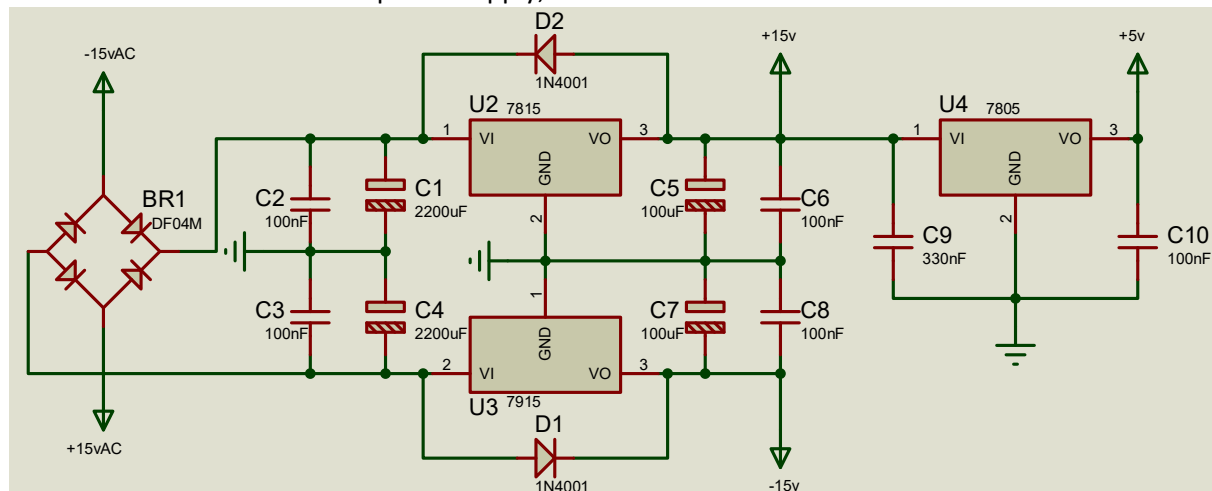


Figure 28 – Power Supply Schematic

The 9000 control board has a mains 230v input which is transformed down to $\pm 15\text{vAC}$. The $\pm 15\text{v}$ is then rectified and used to supply various voltage regulators. The power supply and voltage regulators on the 9000 have been carefully selected based on the overall power consumption of the 9000 metal detector. To ensure the discriminator does not overload the power supply on the 9000 it was decided for the discriminator to use its own power supply. The transformer on the 9000 has more than enough headroom to connect the discriminator to the secondary windings so that $\pm 15\text{vAC}$ is taken directly to the discriminator board. The $\pm 15\text{vAC}$ is connected to a bridge rectifier which converts the AC to DC and capacitors are used to smooth out the ripple.

To calculate the smoothing capacitors required the following equation was used;

$$\text{Capacitor Size} = \frac{I * t}{V_p}$$

Where: I = Maximum current output.

t = Discharge time

V_p = Peak volts

It was calculated the worst case power consumption of the discriminator would be no more than 500mA (assuming no faults).

To calculate the discharge time the mains supply used is 50Hz and because a bridge rectifier is used the negative wave is converted to a positive wave increasing the frequency to 100Hz. The time period is then given by the equation below;

$$t = \frac{1}{f} = \frac{1}{100} = 0.01\text{s} = 10\text{ms}$$

As the transformer used is 15-0-15v RMS. The peak voltage can be calculated by;

$$V_p = V_{rms} * 1.414 = 15 * 1.414 = 21.21\text{V}_p$$

This peak voltage at any given time is dropped across two diodes which have a voltage drop of around 0.7v. Therefore the actual peak voltage is given by;

$$21.21V - (0.7v * 2) = 19.81Vp$$

The dual supply $\pm 15v$ regulators used (7815 and 7915) have a dropout voltage of 2v. This means V_{in} must be 2v higher than V_{out} so V_{in} must be at least 17v. The remainder of the peak voltage for the above equation can therefore be calculated;

$$19.81Vp - 17V = 2.81Vp$$

Therefore;

$$Capacitor\ Size = \frac{500 * 10^{-3} * 10}{2.81V} = 1779\mu F$$

The closest common capacitor size is 1800 μF however, to allow for some overhead 2200 μF capacitors were selected.

Smaller 100nF capacitors are used to smooth out any remaining ripple. The capacitors on V_{out} are used to stabilise the output and to minimise noise. Diodes are connected across the regulators to protect the regulators from input short circuits. As there is a fair amount of capacitance on the output, when the supply is switched off the input may drop rapidly while the output may take longer. If the energy is sufficiently high the transistor in the regulator will be damaged. The external diodes therefore bypass the current from the IC to ground. [7] A 5v regulator (7805) is also used to power the PIC, 1.25v reference IC and the delay/active potentiometers.

4.2 PIC Microcontroller Selection

An appropriate PIC microcontroller needs to be selected for this project. The PIC must have at least 10 analogue inputs, 2 digital inputs, and 3 digital outputs. The resolution of the ADC must be 10-bit in order to accurately measure the analogue signals. 10-bit was chosen because the smallest measured signal into the discriminator is $\pm 30mV$. An 8-bit ADC gives 256 steps which from 0-5v gives about 20mV per step. It was decided that 20mV was too close to 30mV and so 10-bit was chosen which gives about 5mV per step. The table below shows the I/O required with the minimum amount of RAM required for each;

Pin N ^o	Pin Label	Input/output	Digital/Analogue	ADC Resolution	Memory (Byte)
1	A Sig 1	Input	Analogue	10-Bit	4
2	A Sig 2	Input	Analogue	10-Bit	4
3	A Sig 3	Input	Analogue	10-Bit	4
4	P Sig 1	Input	Analogue	10-Bit	4
5	P Sig 2	Input	Analogue	10-Bit	4
6	P Sig 3	Input	Analogue	10-Bit	4
7	Fe Delay	Input	Analogue	10-Bit	2
8	NFe Delay	Input	Analogue	10-Bit	2
9	Fe Active	Input	Analogue	10-Bit	2
10	NFe Active	Input	Analogue	10-Bit	2
11	Metal Sig	Input	Digital	1-Bit	1
12	Detect Sig	Input	Digital	1-Bit	1
13	PWR On	Output	Digital	1-Bit	1
14	Fe Detect	Output	Digital	1-Bit	1
15	NFe Detect	Output	Digital	1-Bit	1
Total	15	12xI , 3xO	10xA, 5xD	10-Bit	37

Table 2 – PIC Microcontroller Initial Specification

The total number of IO pins required is 15 however an additional 2 pins are required for power and ground equating to a total of 17 pins.

A PIC microcontroller can only sample one analogue signal at a time. The ADC will sample one analogue channel and then move to the next channel that requires sampling and so on. There is a time delay between setting-up each analogue channel for ADC conversion which is known as the acquisition time. The acquisition time is determined by the PIC selected however this is usually extremely fast (microseconds) and is excluded from the PIC selection criteria. The ADC will sample the analogue voltage and will only save the minimum and maximum measured values. As a 10-bit ADC is used 2 bytes per reading are required hence 4 bytes required for each channel. Other memory allocations are not included in the table such as internal flag bits which may be required in the software and so to allow for this the total memory of 37 can be multiplied by 4 to give an estimate RAM size of 148 Bytes.

The PIC also requires at least 4 internal timers for the delay and active timers.

The Microchip 'parts selector' tool [8] on the Microchip website was used which allows information to be entered that then searches for appropriate PICs based on the entered information. The information entered was the minimum requirements as shown in the table below;

Minimum PIC Requirements	
Voltage Range	0 - 5v
Total N ^o of Pins	17
Analogue Pins	10
Digital Pins	5
ADC Resolution	10-Bit
N ^o of Timers	4
RAM	148 Bytes

Table 3 – Minimum PIC Requirements

The PIC16F1829 was chosen as it exceeds all the minimum requirements in a 20-pin package. The PIC allows for future expansion of this project while maintaining a suitable package size for the current project. Below is a screenshot from the microchip website showing the PIC16F1829 specifications;

Parametrics	
Name	Value
Program Memory Type	Flash
Program Memory Size (KB)	14
CPU Speed (MIPS/DMIPS)	8
SRAM Bytes	1,024
Data EEPROM/HEF (bytes)	256
Digital Communication Peripherals	1-UART, 2-SPI, 2-I2C2-MSSP(SPI/I2C)
Capture/Compare/PWM Peripherals	4 Input Capture, 2 CCP, 2 ECCP,
Timers	4 x 8-bit, 1 x 16-bit
ADC Input	12 ch, 10-bit
Number of Comparators	2
Temperature Range (C)	-40 to 125
Operating Voltage Range (V)	1.8 to 5.5
Pin Count	20
Low Power	Yes

Figure 29 – PIC16F1829 Specifications

4.3 PCB Design

A 2-layer PCB was designed using Proteus Ares. The advantage of using Proteus Ares is that the circuit was built using Proteus ISIS which is a very powerful simulation program and the circuit was tested in simulation to ensure that it was correct. The circuit from Proteus ISIS is then carried over to Proteus Ares where the PCB can be designed but Proteus can carry out design checks to ensure all the correct connections are made. The below screenshot below shows the top copper of the PCB;

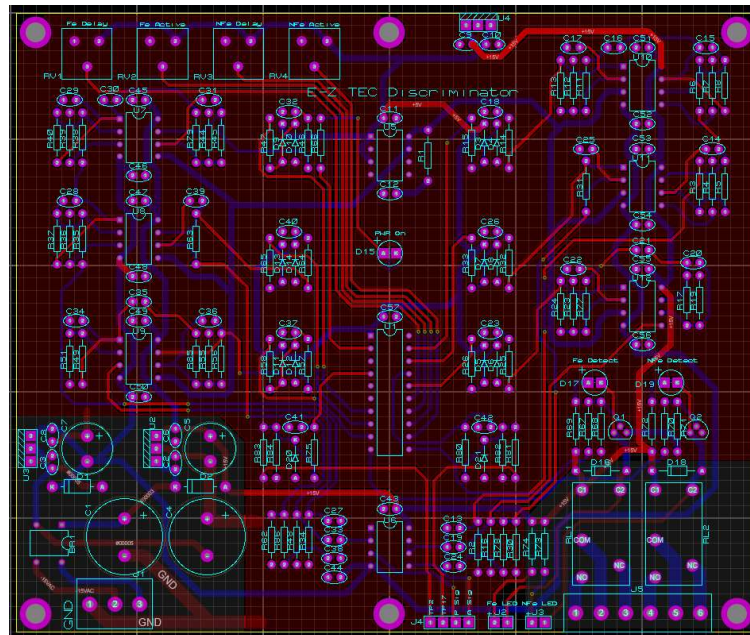


Figure 30 – PCB Design – Top Copper

The Screenshot below shows the bottom copper of the PCB;

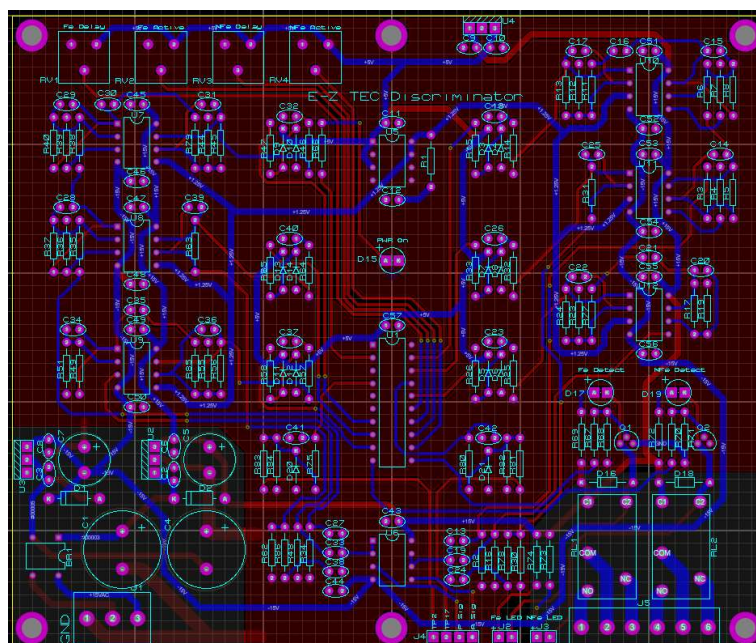


Figure 31 – PCB Design – Bottom Copper

A ground plane was created on the top copper to ensure a low ground resistance and to minimise noise. The ground plane was removed around the power supply and relays to minimise AC noise from interfering with the ground plane. The voltage connected to the relay contacts is unknown as it will be supplied by the customer so it was decided to use large PCB tracks (100thou) to reduce heat dissipation and to remove the ground plane from the area to reduce interference. 50thou tracks were used for all the power connections and 25thou for the signals. M4 holes are located in each corner of the board with 2 in the middle to minimise mechanical stress from the user adjusting the potentiometers or screwing down the terminals. Below is a 3D view created in Proteus Ares;

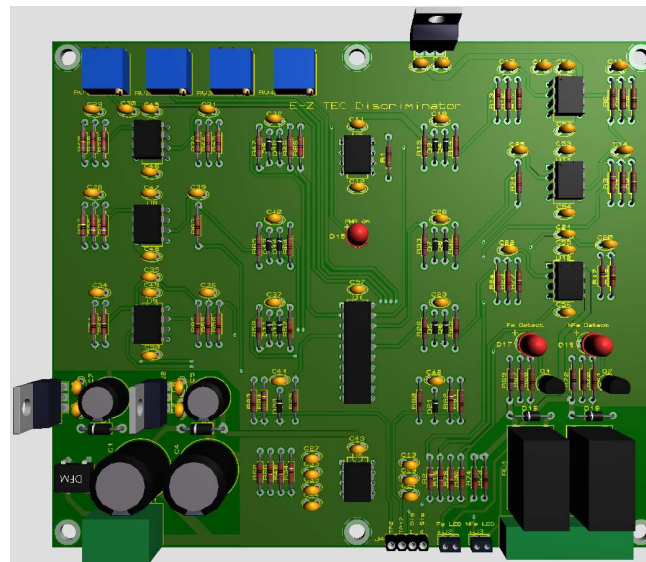


Figure 32 – PCB Design – Proteus 3D View

The 3D view was used to place the components so that they are relatively symmetrical and to ensure all the components will fit and are not too close together. A Gerber file was then created and sent to a PCB manufacturing website called OSHPark.com [9]. The PCB was received and populated as shown;

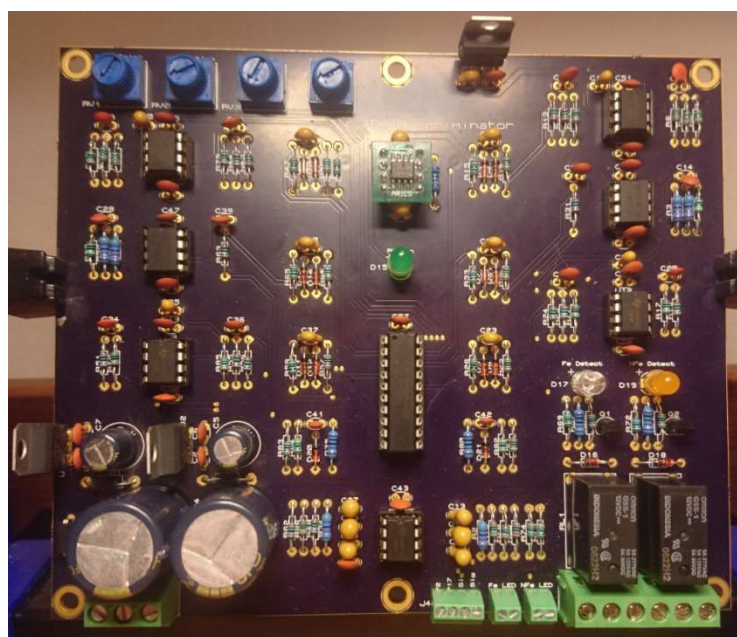


Figure 33 – PCB Design – Populated PCB

4.4 Software Design

The software was written using CCS and tested throughout using Proteus ISIS and downloaded to the PIC16F1829 using a PICKit3 and tested using both the prototype build and final build. To help explain how the program operates it has been split into sections with flowcharts in each. The program consists of one external interrupt input and 5 internal timers each having their own interrupt service routine (ISR). An interrupt service routine allows an event to initiate a control sequence which takes priority over the current activity and then return back once the ISR is complete [10]. The first section explains the main program and the other sections explain each ISR.

4.4.1 Main Program

The flowchart below shows the 'main' program loop;

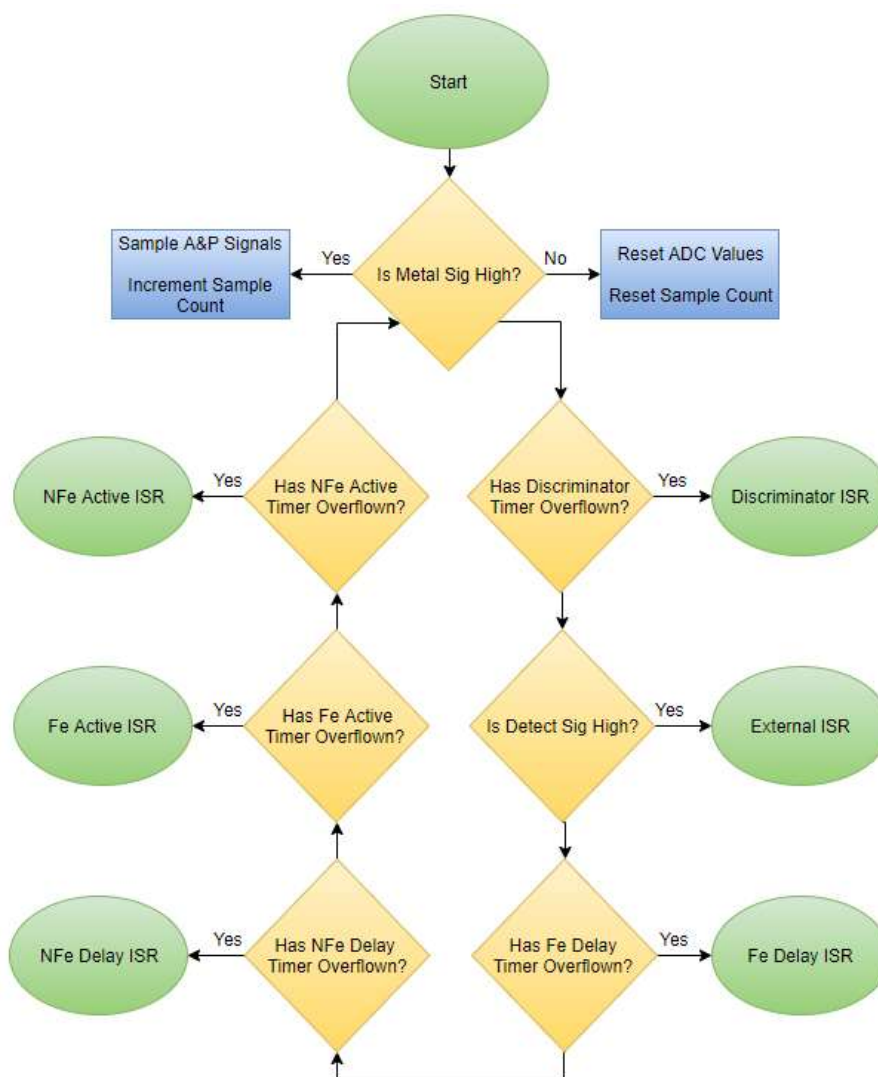


Figure 34 – Software Main Loop Flowchart

If the program is not inside an ISR then it will be in the main loop waiting for either the 'Metal Sig' input to rise high or for an interrupt to occur. The metal sig is a DC pulse which is generated by the 9000 metal detector when the signals it receives from metal passing through the coil is greater than a given threshold. This means the metal sig will rise high at the start of the analogue signals we wish to measure. Therefore, when the metal sig input is high the program begins sampling each analogue channel while saving the maximum and minimum peak values. The metal sig can however also rise when interference is induced and therefore when the metal sig falls low and a detect flag is not set the peak values are reset to 512. 512 is 2.5v which is a 'virtual 0v'. The code for this section of the program is shown below;

```

302 while(1) // Main loop
303 {
304
305 while(!input(Metal_Sig)) // While waiting for metal sig - reset values
306 {
307 if(Detect!=1){
308 AS1Max=512; AS2Max=512; AS3Max=512;
309 AS1Min=512; AS2Min=512; AS3Min=512;
310 PS1Max=512; PS2Max=512; PS3Max=512;
311 PS1Min=512; PS2Min=512; PS3Min=512;
312 St=0; AS1=0; AS2=0; AS3=0;
313 Fe=0; NFe=0;
314 LargeFe=0;}
315 else{break;}
316 }
317
318 while(input(Metal_Sig)) // During metal sig - Sample signals
319 {
320
321
322 set_adc_channel (A_Sig_1); // A Sig 1 read
323 delay_us(15); // Wait for ADC
324 value=read_adc(); // Write ADC value
325 if(value>AS1Max){AS1Max=value; // Write Max value
326 AS1=St;} //Write Phase Time of Max Value
327 else if(value<AS1Min)AS1Min=value; // Write Min value
328
329 set_adc_channel (P_Sig_1); // P Sig 1 read
330 delay_us(15);
331 value=read_adc();
332 if(value>PS1Max){PS1Max=value;
333 PS1=St;}
334 else if(value<PS1Min)PS1Min=value;
335
336 set_adc_channel (A_Sig_2); // A Sig 2 read
337 delay_us(15);
338 value=read_adc();
339 if(value>AS2Max)AS2Max=value;
340 else if(value<AS2Min)AS2Min=value;
341
342 set_adc_channel (P_Sig_2); // P Sig 2 read
343 delay_us(15);
344 value=read_adc();
345 if(value>PS2Max)PS2Max=value;
346 else if(value<PS2Min)PS2Min=value;
347
348 set_adc_channel (A_Sig_3); // A Sig 3 read
349 delay_us(15);
350 value=read_adc();
351 if(value>AS3Max)AS3Max=value;
352 else if(value<AS3Min)AS3Min=value;
353
354 set_adc_channel (P_Sig_3); // P Sig 3 read
355 delay_us(15);
356 value=read_adc();
357 if(value>PS3Max)PS3Max=value;
358 else if(value<PS3Min)PS3Min=value;
359
360 St++; //Increment Sample Count
361
362 }
363 }
364 }
365 }

```

Figure 35 – Software Code – Main While Loops

Inside the main program which is executed before the code shown in the previous screenshot are initial settings that are executed when the PIC is powered on. This is where the clock speed for the ADC and timers are initialised. The ADC ports are also setup so that the PIC knows which pins are analogue inputs. The ADC clock and timer clocks are divisions of the base system clock which is set at 8MHz. The interrupts are also enabled in this section so that the interrupts can operate. The PIC16F1829 has one external interrupt pin which is pin17. This pin is connected to 'detect sig' which is a DC pulse generated from the 9000 when a detection has occurred. The polarity of an external interrupt pin can be set which is shown in the screenshot below as "L_TO_H" which means low to high. Therefore when the detect signal pulses the program will enter the external ISR on the rising edge of the pulse.

Figure 36 – Software Code - Main

34 | Page

4.4.2 External ISR

The external ISR is executed when the detect sig rises high. If the 9000 metal detector is in twin peak detection mode then the detect signal rises high at the second peak of the analogue signals. Initially the external ISR was used to start discriminating however, it was found when the metal detector is in single peak detection and large metal is used then the detect signal comes in just before the first peak meaning the ADC hasn't finished finding the high and low peaks of the analogue signals and thus discriminating incorrectly. Large metal test pieces also created a double detect pulse. This caused the discriminator to attempt discriminating two different parts of the signals giving false results. Therefore, it was decided to add a 200ms delay to ensure that the whole analogue signals had been sampled before executing the discriminator code. This was achieved by using one of the five internal timers. The flowchart and code showing the external ISR is shown below;

```

////////////////////////////////////
////////////////////////////////////External Interrupt////////////////////////////////////
////////////////////////////////////
// Interrupt name
#INT_EXT
// External ISR
void EXT_isr(void)
{
    Detect=1; // Set Detect flag
    clear_interrupt(INT_Timer1); // Clear interrupt flag
    set_Timer1(15124); // Set discriminator timer - 200ms delay
    enable_interrupts(INT_Timer1); // Enable discriminator interrupt
    disable_interrupts(INT_EXT); // Prevents double detect from large metal
}

```

Figure 37 – Software Code – External ISR



Figure 38 - External ISR Flowchart

The external ISR firstly enables a detect flag bit which prevents the ADC values from resetting when returning back to the main program. It was found during testing that without this detect flag sometimes when the program returned from the external ISR the ADC values were being reset. The detect flag bit solved this issue. Timer 1 is then enabled which is a 16-bit timer meaning it counts from 0 to 65535. The clock speed is set at its slowest in the main program with the prescaler dividing the system clock by 8. By loading a value into the timer the timer will begin counting up from that value. It was calculated that by loading the timer with the value '15124' with the prescaler on 8 it would take ~200ms to overflow. Before ending the ISR the external Interrupt is disabled to prevent a double detection from large metal that was causing problems during testing. The external ISR is then re-enabled after timer1 has overflowed.

4.4.3 Discriminator ISR

When Timer1 has overflown the ISR is executed which is where the program will discriminate. Below is the flowchart showing the discriminator ISR;

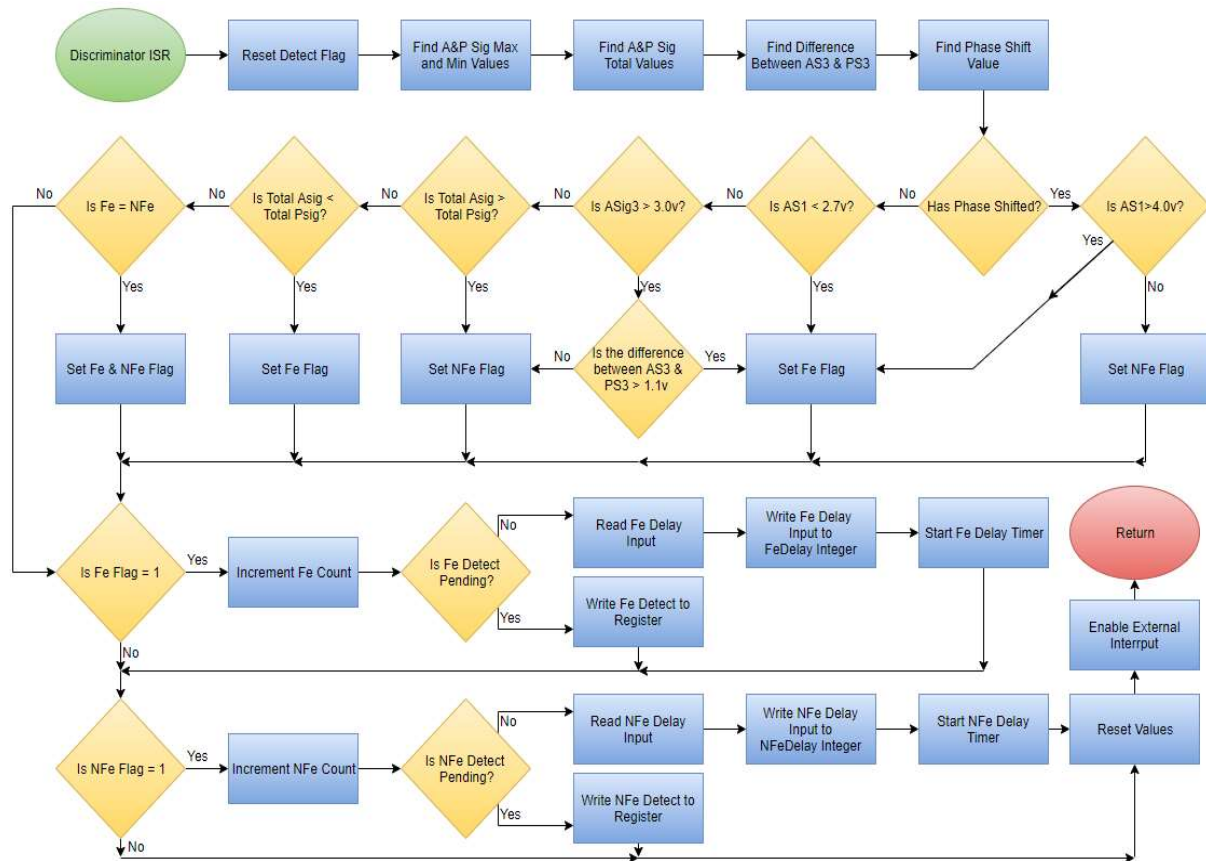


Figure 39 – Discriminator ISR Flowchart

During the first stage of the discriminator ISR calculations are made to find the following;

- A & P Sig Max Values
- A & P Sig Min Values
- A Sig total peak to peak value
- P Sig total peak to peak value
- Difference between A Sig 3 and P Sig 3
- Phase shift

To find the A and P Sig Max values, each max value from each analogue channel is added together giving the total max value. This is also true for the min value which are also all added together. The total peak values are then calculated by subtracting the total max value with the total min value giving the overall peak to peak value of all 3 channels. The difference between A Sig 3 and P Sig 3 is found by subtracting them from one another then this value is stored to the integer labelled 'LargeFe'. This 'LargeFe' value was added to solve a problem encountered during testing with large ferrous metal. Large ferrous metal causes strange results where A and P sigs are in phase but their amplitudes are much different. Therefore, if the signals are in phase but their amplitudes are

significantly different then it is known that ferrous was detected. The phase shift is calculated by subtracting the A sig sample time (AS_t) from the P sig sample time (PS_t). The code for this section is shown below;

```

////////////////////////////////////
////////////////////////////////////Discriminator////////////////////////////////////
////////////////////////////////////
// Interrupt name
// Timer1 ISR
void isr_Timer1(void)
{
    Detect=0; // Reset detect flag
    Fe=0, NFe=0; // Debug
    ASMax=AS1Max+AS2Max+AS3Max; // Find total A sig max
    ASMin=AS1Min+AS2Min+AS3Min; // Find total A sig min
    PSMax=PS1Max+PS2Max+PS3Max; // Find total P sig max
    PSMin=PS1Min+PS2Min+PS3Min; // Find total P sig min

    ASValue=ASMax-ASMin; //A Sig Total Value
    PSValue=PSMax-PSMin; //P Sig Total Value

    LargeFe=AS3Max-PS3Max; // Find difference between AS3 & PS3

    Pt=ASt-PSt; //Time between A&P sig positive peaks (Phase Shift)
}

```

Figure 40 – Software Code – Discriminator ISR

A series of if statements is then made to determine if the detected metal was ferrous or non-ferrous. A Fe or NFe flag is set depending on which if statement is true. If none of the statements are true for example if both the A sig and P sig calculate to the same value then both Fe and NFe flag bits get set. This is to ensure that even if the discriminator cannot discriminate it will always give an output when the 9000 sends it a detect signal. The metal may end up in the wrong reject bin but it is better than not being rejected at all. The code for this section is shown below;

```

if(Pt>40){ //If Phase has shifted
    if(AS1Max>819)Fe=1; //Phase shift with ASig>4.0v = Fe
    else if(AS1Max<819)NFe=1;} //Phase shift with ASig<4.0 = NFe

else if(Pt<-40){ //If Phase has shifted
    if(AS1Max>819)Fe=1; //Phase shift with ASig>4.0v = Fe
    else if(AS1Max<819)NFe=1;} //Phase shift with ASig<4.0 = NFe

else if(AS1Max<553)Fe=1; //If ASig1 < 2.7v and no phase shift = Fe

else if(AS3Max>615){ //If ASig3 > 3.0v
    if(LargeFe>225)Fe=1; //If difference between AS3 & PS3 > 1.1v
    else if(LargeFe<225)NFe=1;}

else if(ASValue>PSValue)NFe=1; //NFe Detect

else if(ASValue<PSValue)Fe=1; //Fe Detect

/////ERROR/////
if(Fe==NFe){
    Fe=1;
    NFe=1;
}
}

```

Figure 41 – Software Code – Discriminator ISR 2

The program then checks if the Fe and NFe flag bits are set. If they are set then the following code gets executed;

```

//////////////////////////////////Fe Detect//////////////////////////////////
if(Fe==1){
    FeDRA++;
    if(FeDelay==0)                //If no detection pending
    {
        set_adc_channel (Fe_Delay);    // Fe Delay read
        delay_us(15);
        FeDelayADC=read_adc()*28;
        FeDelay=FeDelayADC;
        enable_interrupts(INT_Timer0); // Enable Fe Delay Timer interrupt
        set_Timer0(0);
    }
    else if(FeDRA==2)FeDelay1=FeDelayADC-FeDelay;
    else if(FeDRA==3)FeDelay2=FeDelayADC-FeDelay-FeDelay1;
    else if(FeDRA==4)FeDelay3=FeDelayADC-FeDelay-FeDelay1-FeDelay2;
    else if(FeDRA==5)FeDelay4=FeDelayADC-FeDelay-FeDelay1-FeDelay2-FeDelay3;
    else if(FeDRA==6)FeDelay5=FeDelayADC-FeDelay-FeDelay1-FeDelay2-FeDelay3-FeDelay4;
    else if(FeDRA==7)FeDelay6=FeDelayADC-FeDelay-FeDelay1-FeDelay2-FeDelay3-FeDelay4-FeDelay5;
    else if(FeDRA==8)FeDelay7=FeDelayADC-FeDelay-FeDelay1-FeDelay2-FeDelay3-FeDelay4-FeDelay5-FeDelay6;
    else if(FeDRA==9)FeDelay8=FeDelayADC-FeDelay-FeDelay1-FeDelay2-FeDelay3-FeDelay4-FeDelay5-FeDelay6-FeDelay7;
    else if(FeDRA==10)FeDelay9=FeDelayADC-FeDelay-FeDelay1-FeDelay2-FeDelay3-FeDelay4-FeDelay5-FeDelay6-FeDelay7-FeDelay8;
    else if(FeDRA>10){FeDRA--;}
}

```

Figure 42 – Software Code – Discriminator ISR 3

The above code shows the program which gets executed when the Fe flag bit is set. The code for NFe is identical except that instead of activating timer0 it activates timer4 and the registers are labelled as NFe instead of Fe. See appendix1 for the full code. “FeDRA” gets incremented to keep a count on how many times Fe metal is detected. “FeDRA” was named after ‘Fe detect register allocation’. It allows the software to know where to store the current detect delay time. “if(FeDelay==0)” checks to see if there has already been a Fe metal detected that is using the delay timer. If there hasn’t then the FeDelay value will be 0 and so the program will continue and read the Fe Delay input voltage to determine the delay time required. The Fe Delay input voltage is controlled by using the Fe Delay potentiometer. This value is then stored in “FeDelay” and Timer0 is enabled. Each time Timer0 overflows it will decrement the value of “FeDelay” until it reaches 0 again. If however, the “FeDelay” is not equal to 0 then the program will use the “FeDRA” to determine the delay time required and store this value in the appropriate register. The delay time in this instance is calculated by subtracting the previous detect times. If an eleventh detect takes place before the first detect has finished the delay timer then the eleventh detect gets ignored. This is not an ideal situation.

Before ending the discriminator ISR the ADC values are reset and the external interrupt is re-enabled as shown;

```

ASMax=0; ASMin=0; PSMax=0; PSMin=0; //Reset variables
AS1Max=512; AS2Max=512; AS3Max=512;
AS1Min=512; AS2Min=512; AS3Min=512;
PS1Max=512; PS2Max=512; PS3Max=512;
PS1Min=512; PS2Min=512; PS3Min=512;
ASValue=0; PSValue=0;
st=0; pt=0, Fe=0; NFe=0;
LargeFe=0;
clear_interrupt(INT_Timer1);    //Clear Interrupt
clear_interrupt(INT_EXT);      //Clear Interrupt
enable_interrupts(INT_EXT);    //Enable External Interrupt
disable_interrupts(INT_Timer1); //Disable Discriminator Interrupt
}

```

Figure 43 – Software Code – Discriminator ISR 4

4.4.4 Delay Timer

The flowchart below shows the operation of the FeDelay timer;

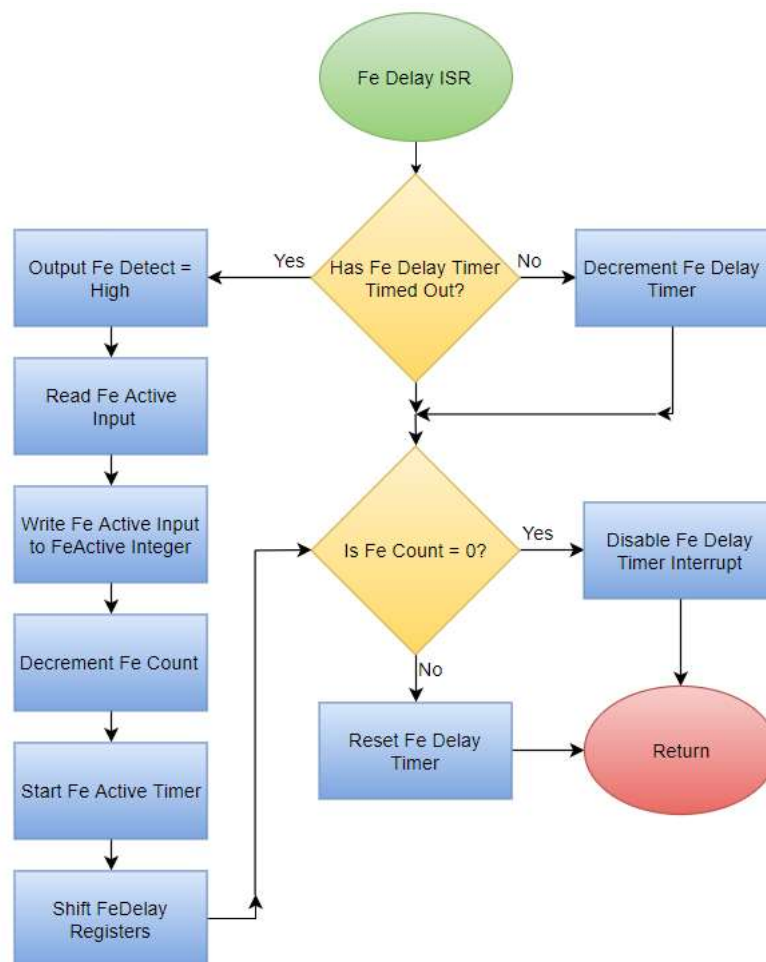


Figure 44 – Fe Delay ISR Flowchart

Fe Delay uses Timer0 which is enabled in the discriminator ISR if Fe metal was detected. It should be noted again that the code is the same for the NFe delay except it uses Timer4 and the variables are labelled differently. See appendix1 for full code. If Fe is detected then the Fe Timer (Timer0) will begin counting from 0 to 255. When it reaches 255 the ISR is initiated. If “FeDelay” is greater than 0 then “FeDelay” will decrement by one. “FeDelay” value is determined by the delay time required, see Discriminator ISR for details. When “FeDelay” reaches 0 the ferrous detect output is set high and the active timer is initialised. The active timer is initialised similar to that of the delay timer by reading the Fe active input which is controlled by a potentiometer. “FeDRA” is then decremented so that the discriminator ISR knows that a detect register has been freed up. The registers are then shifted so that multiple detection times are possible. “FeDelay” is then checked again to see if it is still a 0. If it is then the ISR is ended. Otherwise, the timer is reset with the value found in the register

```

////////////////////////////////////
////////////////////////////////////Fe Delay////////////////////////////////////
////////////////////////////////////
#INT_TIMER0 // Interrupt name
void isr_Timer0(void) // Timer0 ISR
{
    if(FeDelay!=0)
    {
        FeDelay--;
    }

    else{
        disable_interrupts(INT_Timer0);
        output_high(Fe_Detect); //Fe Detect Output
        set_adc_channel (Fe_Active); // Fe Active read
        delay_us(15);
        FeDRA--;
        FeActive=read_adc()*13;
        enable_interrupts(INT_Timer2); // Enable Fe Active Timer interrupt
        set_Timer2(0);

        FeDelay=FeDelay1; //Register shift
        FeDelay1=FeDelay2;
        FeDelay2=FeDelay3;
        FeDelay3=FeDelay4;
        FeDelay4=FeDelay5;
        FeDelay5=FeDelay6;
        FeDelay6=FeDelay7;
        FeDelay7=FeDelay8;
        FeDelay8=FeDelay9;
        FeDelay9=0;
    }

    if(FeDRA!=0){ //Restart Timer
        clear_interrupt(INT_Timer0);
        Enable_interrupts(INT_Timer0);
        set_Timer0(0);}

    else disable_interrupts(INT_Timer0); //Stop Timer
}

```

Figure 45 – Software Code – Delay Timer ISR

4.4.5 Active Timer

The flowchart for the Fe active timer is shown below;

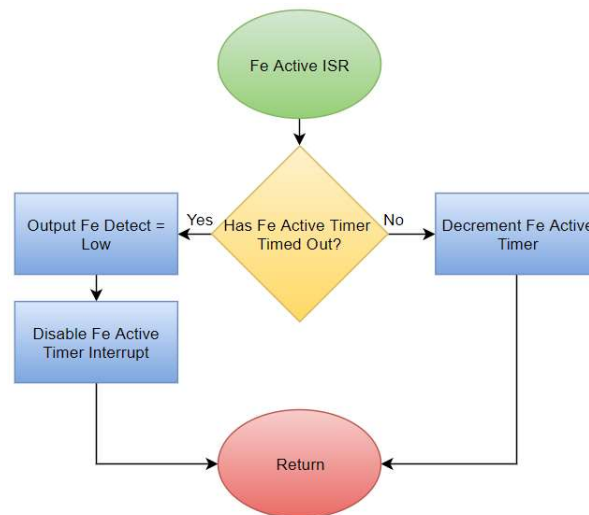


Figure 46 – Fe Active ISR Flowchart

The Fe active timer uses timer2 whereas the NFe active timer uses timer6. The active timers are initialised in the corresponding delay ISR. “FeActive” is decremented each time the timer overflows until it reaches 0. Once “FeActive” has reached 0 the Fe detect output is set low and the active ISR is disabled. See the code for the Fe active timer below;

```
//////////////////////////////////////  
//////////////////////////////////////Fe Active//////////////////////////////////////  
//////////////////////////////////////  
#INT_TIMER2 // Interrupt name  
void isr_Timer2(void) // Timer2 ISR  
{  
    if(FeActive>1)  
    {  
        FeActive=FeActive-1;  
    }  
    else{  
        output_low(Fe_Detect);  
        FeActive=0;  
        disable_interrupts(INT_Timer2); //Clear Fe Active Interrupt  
    }  
}
```

Figure 47 – Software Code – Fe Active ISR

4.5 Prototype Build

The circuit has been designed and simulated using Proteus however, the true test is to build the circuit and test it using the signals from the 9000 metal detector rather than ideal signals generated in Proteus. It was decided to build the prototype using breadboard as it is quick and easy to insert components and as there is no soldering required it is easy to correct any problems. Below is the prototype build connected to a 9000 metal detector;

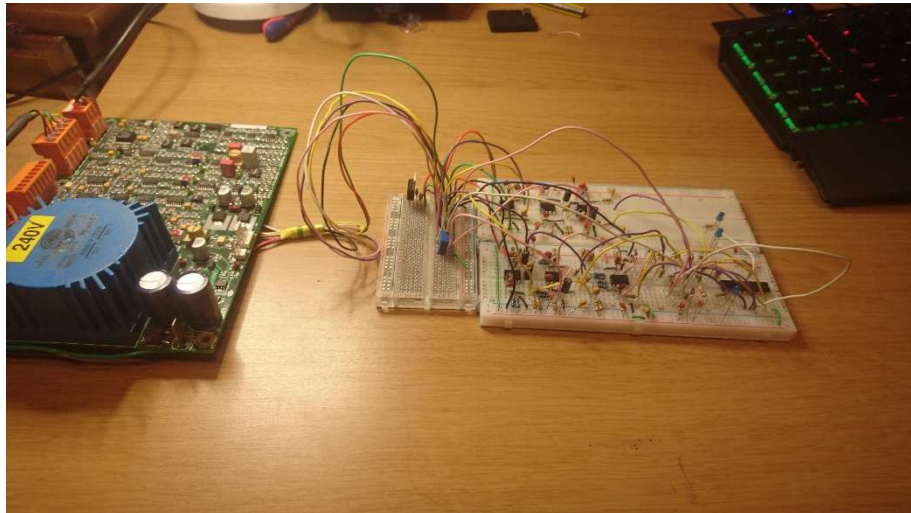


Figure 48 – Prototype Build

The above prototype was used to help design the circuit for example finding the appropriate gain for the smallest size metal to have a large enough amplitude to discriminate was mostly trial and error using the prototype. It also helped with the PCB design as it gave a better perspective of how the components can connect to the PIC. For example it was realised during the prototype build that it is better to keep all A sig channels on one side of the PIC and all the P sig on the other side. There are a few components missing from the prototype build such as the relays and 1.25v reference IC however, LED's were used to indicate the PIC output was operating correctly and a potentiometer was tuned with a 1.25v output. The prototype was also used to help develop the software while the PCB was being fabricated. Below shows the system connected to a test coil;



Figure 49 – Prototype Build 2

4.6 Final Build

Unfortunately due to time constraints, at the time of submitting this report the final build was not completed. An old customer returned control panel where the 9000 metal detector and discriminator PCB were to be mounted was selected and was sent to production for refurbishment. The control panel was bead blasted and then placed in the queue at the paint shop. Due to large scale projects taking priority the control panel was not painted in time for the submission of this report. The control panel selected for refurbishment is shown below;

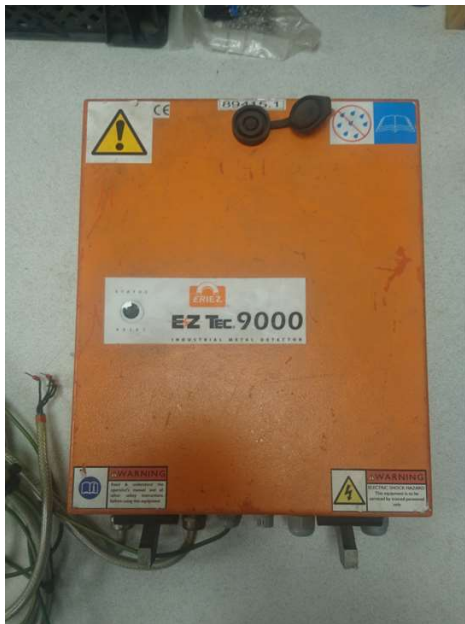


Figure 50 – Control Panel 1



Figure 51 – Control Panel 2

Additional holes will be drilled in the front of the control unit which panel lights can be installed into. These panel lights will give a visual indication of the metal type that has been discriminated as illustrated in the image below.



Figure 52 - Eriez Magnetec E-Z TEC 9000 Control Panel & Discrimination Indicators

4.7 Project Testing

Initially during the design phase the project was tested using Proteus simulation. The project was designed in stages and simulated with corrections made before designing the next section. It was found that as the project grew bigger the circuit and software became extremely demanding on the computer hardware and was unable to simulate the whole project with successful results. Therefore, the different blocks of the project were simulated separately and then all the blocks were put together and tested using both the prototype breadboard design and the PCB design. The first block to design was the signal conditioning channels. It is explained in the literature review that the signals from the metal detector that are to be signal conditioned and then sampled are sine-wave-like signals. Therefore, to simulate this in Proteus a sine wave generator is connected to the input of the signal conditioning channels and an oscilloscope on the output;

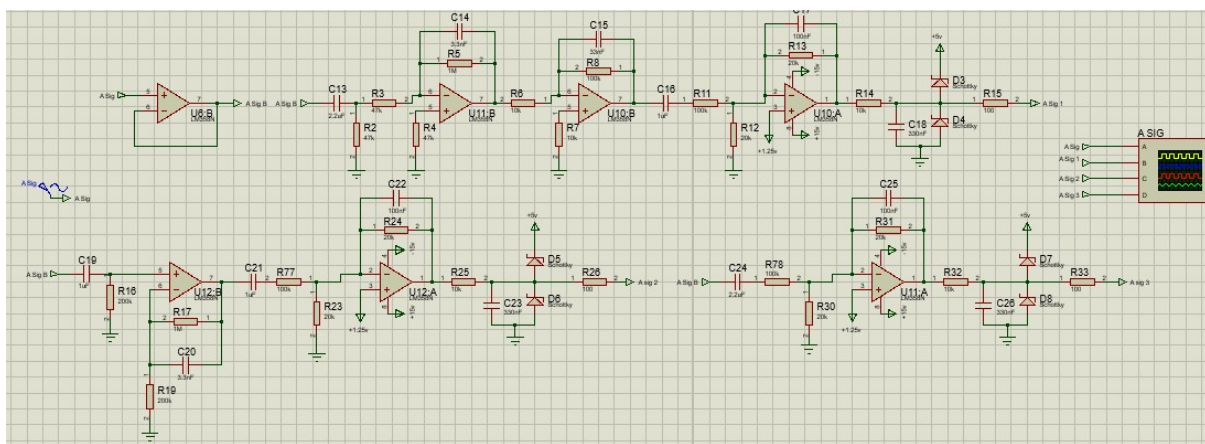


Figure 53 – Simulation Testing – A Sig

The screen shot above shows the simulation circuit for the 3 signal conditioning channels of the A sig which is identical to the 3 channels for the P sig. A 4-channel oscilloscope is used with channel 1 connected directly to the sine wave generator and the other 3 channels connected to the output of each signal conditioning channel. The sine wave generator is set with an amplitude of 100mV peak with a frequency of 20Hz. A screenshot of the oscilloscope during simulation is show;

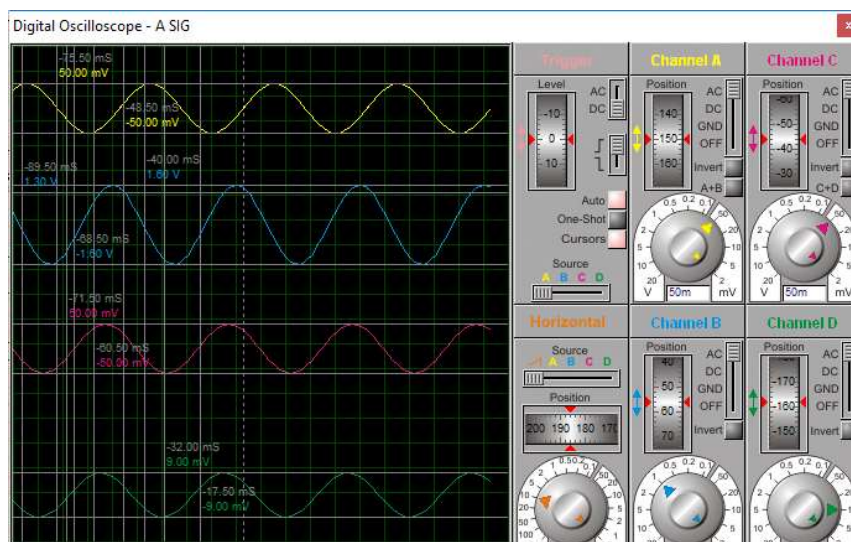


Figure 54 – Simulation Testing – A Sig Oscilloscope Result

While developing the software a separate Proteus simulation was created which excludes the signal conditioning channels due to overloading the CPU of the computer causing errors. The schematic was kept simple using sine wave generators as the output of the signal conditioning channels. Switches were used to simulate the DC pulse from the metal sig and detect sig while LED's were used to visually indicate which outputs were activating;



The signal conditioning channels for both the prototype build and PCB were tested by using a square wave at the input of all the channels. The amplitude and the frequency of the square wave was adjusted to ensure the circuit is operating correctly. As the frequency is increased above 40hz the amplitude at the output of the channels begins to attenuate as expected due to the filtering. The oscilloscope screenshot shows the results during this test;

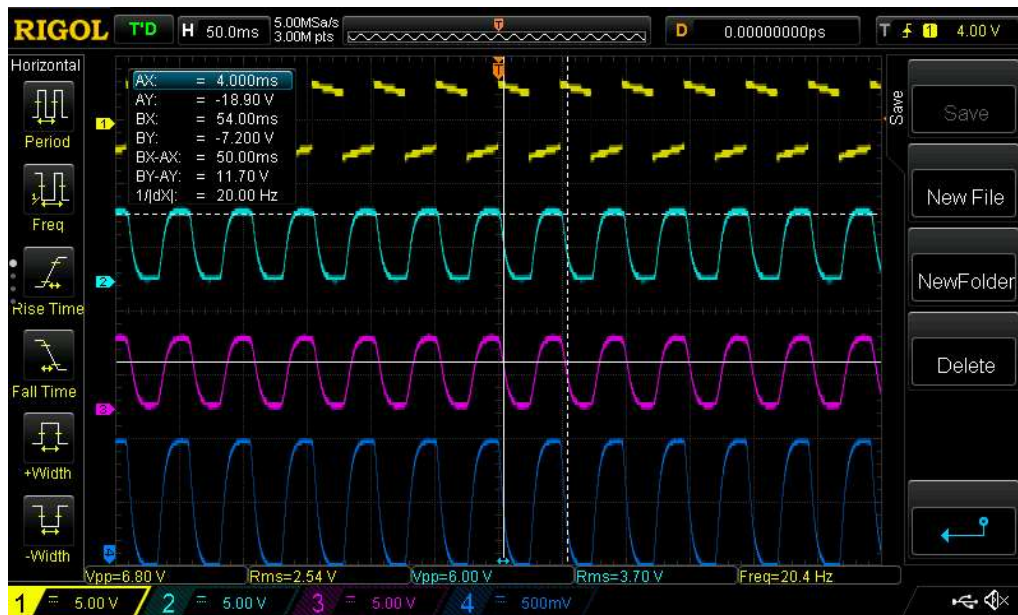


Figure 56 – Asig Testing

The yellow trace is the input with a peak voltage of about 6.8 volts and a frequency of 20Hz. The light blue, pink and dark blue traces are Asig1, Asig2 and Asig3. It can be seen from this screenshot that the signal conditioned signals are scaled correctly with 2.5v as the base line. It can be seen in Asig1 and Asig2 that the signals are clipped and proves the over voltage and negative voltage protection is operating correctly. The signals were then compared to the Psig channels and the results were identical as expected.

While testing the discriminator using the signals from the metal detector some unexpected results were discovered. The software for the most part was working correctly for a range of test pieces between about 2.0mm to 8.0mm Fe. Any metal sizes outside of this were causing the metal to be discriminated incorrectly or unreliably. It was discovered that as ferrous metal increases in size the Asig and Psig begin to shift and the amplitudes swap positions. For example, the below screenshot shows the signals from a 5.5mm Fe sphere;

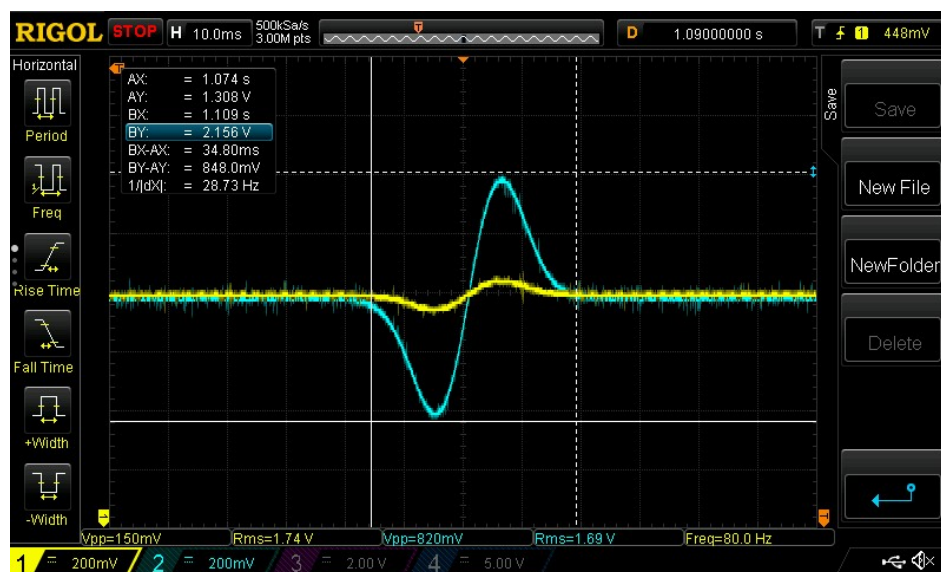


Figure 57 – A&P Sig 5.5mm Fe Sphere

The yellow trace is Asig and the blue trace is Psig. It can be seen clearly that the signals are in phase and the amplitude of Psig is allot bigger than Asig. The next screenshot shows the signals from a 10mmFe sphere;

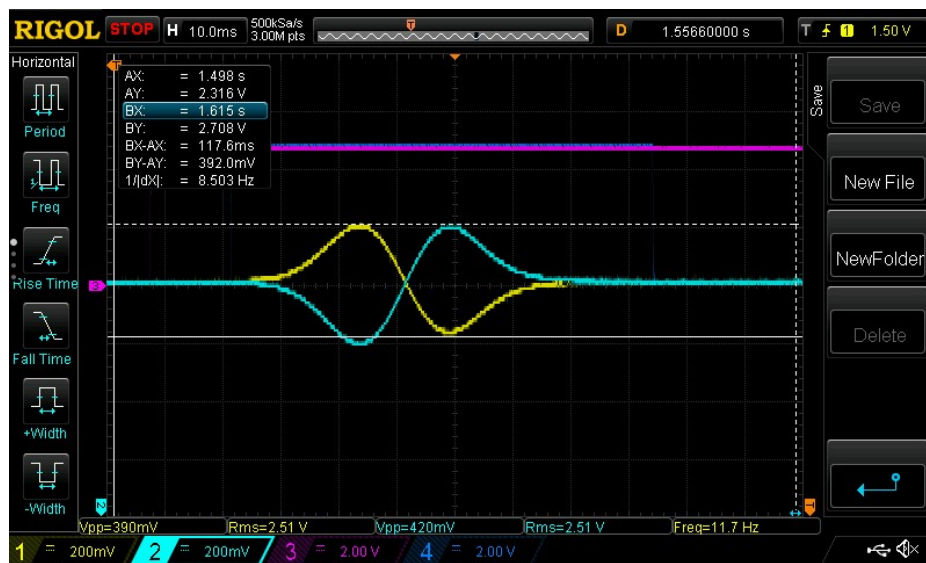


Figure 58 – A&P Sig 10mm Fe Sphere

It can be seen that the amplitude of Psig is still greater than Asig however they are almost the same amplitude and the phase has shifted by 180° . The next screenshot shows the signals from a 12mm Fe sphere;

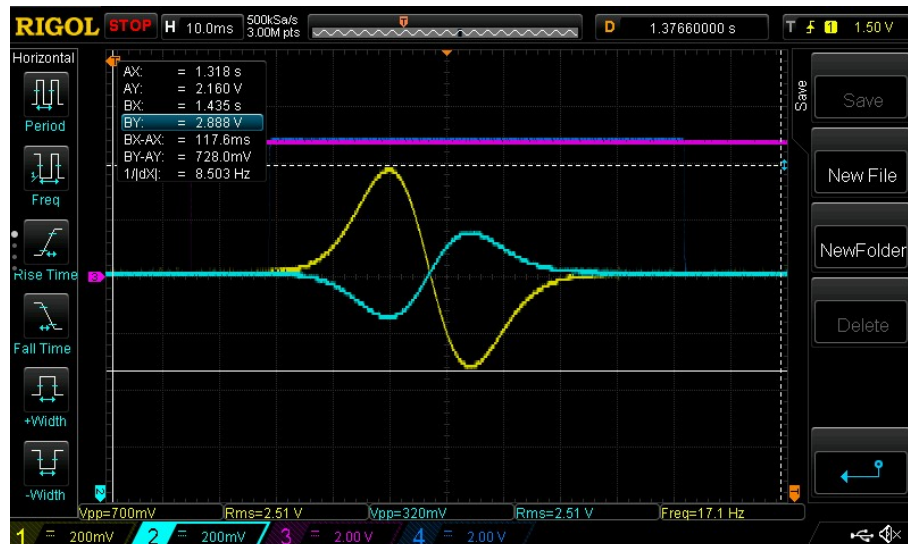


Figure 59 – A&P Sig 12mm Fe Sphere

It can be seen now that the amplitude of Asig is greater than Psig. Due to this when the software was just looking at the peak to peak of the signals large ferrous was being discriminated as non-ferrous. Various large NFe test pieces were tested using the oscilloscope and a phase shift between A and P sig was not found. The next screenshot shows the A and P sig from a 40mm NFe cube;

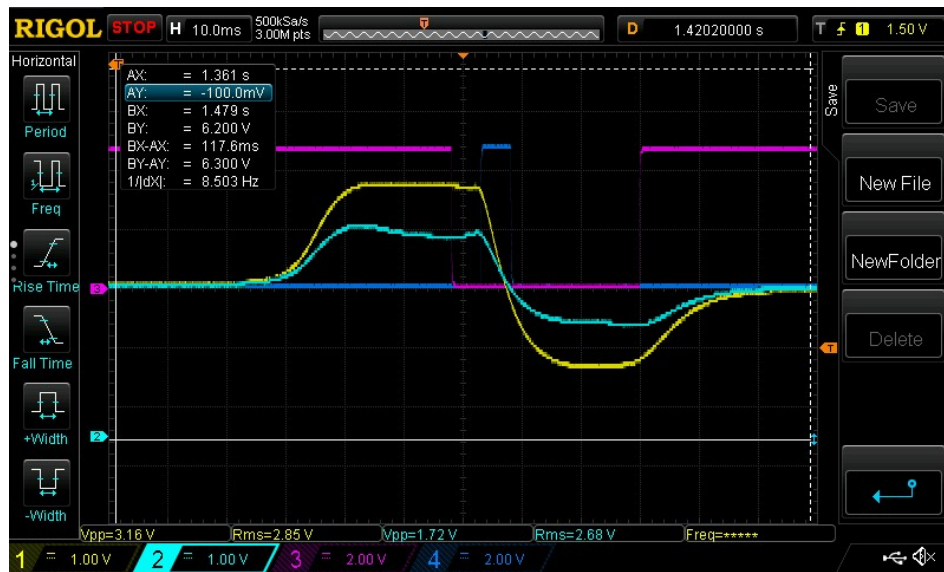


Figure 60 – A&P Sig 40mm NFe cube

Although the signals from a large NFe cube looks distorted it can be seen that the A sig is greater than P sig and they are in phase as expected. Therefore the software was updated to check for a phase shift. If there was a phase shift then the metal was discriminated as Fe. Otherwise if there was no phase shift then the discrimination was dependent on the peak to peak values of the signals which worked for the majority of test pieces.

As testing continued and even larger metal was used, Fe started discriminating as NFe once again. When passing through a 30mm Fe cube the discriminator kept outputting NFe every time. When checking the signals on the oscilloscope an even stranger discovery was found than the previous. Below is a screen shot of A and P sig for a 30mm Fe cube;

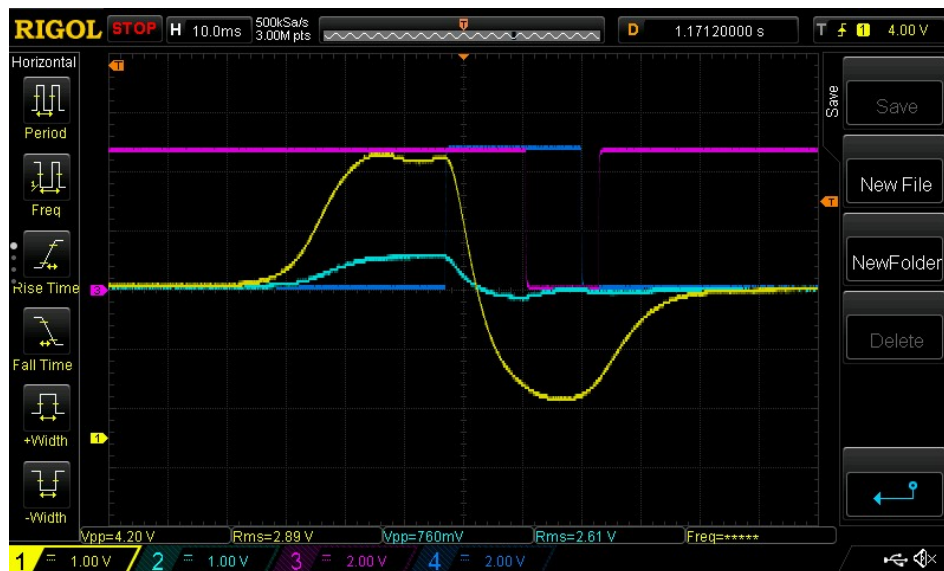


Figure 61 – A&P Sig 30mm Fe cube

It can be seen that the phase has now shifted back and the P sig is much smaller than the A sig. This result initially looks like the signals from a NFe metal however on closer observation it was noticed that the difference between the A and P sig amplitude for large Fe metal as shown in the screen shot

above is much greater than any size NFe metal. For example the difference between the peak values of the two signals for a 40mm NFe cube and a 30mm Fe cube are as follows;

40mm NFe cube

A sig peak to peak = 3.16v

P sig peak to peak = 1.72v

Difference = $3.16 - 1.72 = 1.44\text{v}$

30mm Fe cube

A sig peak to peak = 4.20v

P sig peak to peak = 760mV

Difference = $4.2 - 0.76 = 3.44\text{v}$

The difference for Fe metal is about 2 volts greater than the difference for NFe metal. Therefore the software was updated again so that if there was no phase shift detected then the software will calculate the difference in amplitude between the A sig and P sig. If the difference is greater than 3 volts then the discriminator will output Fe.

The software at this stage was much improved however problems were then discovered when using the smallest size metal test pieces. The smallest size metal the metal detector can detect is 0.8mm Fe and 1.0mm NFe. When testing the discriminator with these size test pieces they were being discriminated incorrectly. The following screenshot shows the Asig1 and Psig1 signals from 1.0mm NFe sphere;

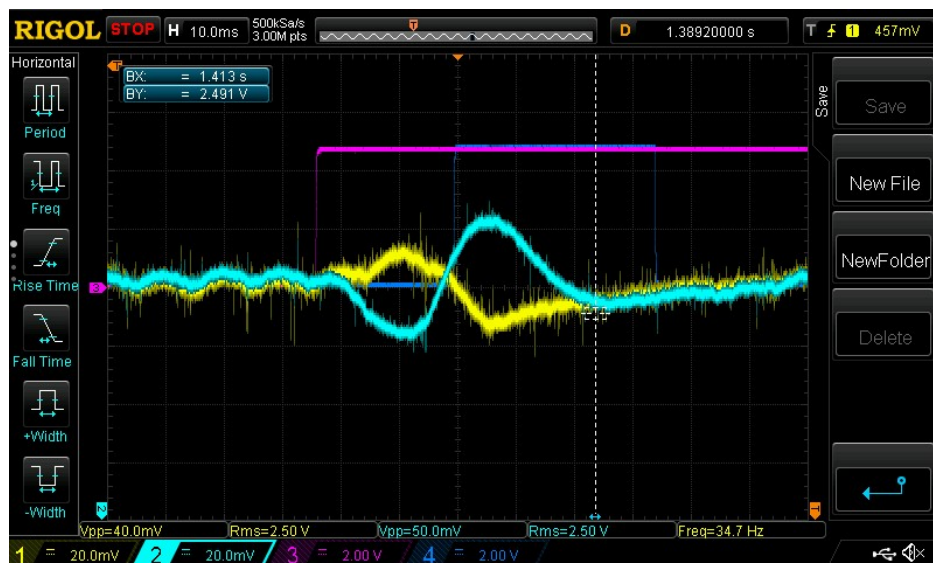


Figure 62 A&P Sig 1.0mm NFe sphere

The result shown here is similar to what was found with large Fe metal. Small NFe metal appears to create a phase shift and the amplitudes change just like that shown from large Fe metal. The phase does not shift for 0.8mm or 1mm Fe metal. To overcome this, in the program where it checks for a phase shift it then checks the amplitude of Asig1. If Asig1 is greater than 4v then Fe is detected whereas if it is less than 4v it discriminates as NFe. The final result is a discriminator which discriminates correctly using any size metal test piece.

4.8 Build Cost

Item Description	Part No	Cost Each	QTY	Total Cost
PCB Components from uk.farnell.com [11]				
Microcontroller	PIC16F1829-I/P	£1.66	1	£1.66
LM358 OP-Amp	LM358P	£0.301	7	£2.11
+15v Regulator	L7815ABV	£0.292	1	£0.29
-15v Regulator	L7915ACV	£0.215	1	£0.22
+5v Regulator	L7805ABV	£0.281	1	£0.28
1.25v Reference IC	MCP1501	£0.643	1	£0.64
8-SOIC to 8-DIP Adapter	LCQT-SOIC8-8	£3.73	1	£3.73
Bridge Rectifier	DF04M	£0.301	1	£0.30
20pin DIP Socket	110-93-320-41-001000	£1.22	1	£1.22
8pin DIP Socket	SPC15494	£0.121	7	£0.85
100R resistor 1% 1/4w	LR0204F100R	£0.046	6	£0.28
220R resistor 1% 1/4w	LR0204F220R	£0.046	3	£0.14
680R resistor 1% 1/4w	LR0204F680R	£0.047	2	£0.09
1K resistor 1% 1/4w	MF25 1K	£0.0671	5	£0.34
4.7K resistor 1% 1/4w	LR0204F4K7	£0.047	2	£0.09
10K resistor 1% 1/4w	LR0204F47K	£0.047	14	£0.66
20k resistor 1% 1/4w	LR0204F20K	£0.047	12	£0.56
47K Resistor 1% 1/4w	LR0204F47K	£0.047	6	£0.282
100K Resistor 1% 1/4w	LR0204F100K	£0.048	8	£0.38
200k Resistor 1% 1/4w	LR0204F200K	£0.0307	4	£0.13
1M Resistor 1% 1/4w	LR0204F1M0	£0.046	4	£0.18
100k Potentiometer	M63M104KB30T607	£2.68	4	£10.72
3.3nF Capacitor	MCBU7332K6	£0.106	4	£0.42
10nF Capacitor	D103K43Y5PH63L2R	£0.355	2	£0.71
33nF Capacitor	30LVD33	£0.432	2	£0.864
100nF Capacitor	SR215E104ZARTR	£0.088	26	2.29
330nF Capacitor	MCCB1V334M2ACB	£0.544	7	£3.81
1uF Capacitor	MCMLR50V105MY5V	£0.412	6	£2.47
2.2uF Capacitor	MCMLR50V225MY5V	£0.796	6	£4.78
100uF Capacitor	EEUFC1V101B	£0.276	2	£0.55
2200uF Capacitor	MAL213650222E3	£2.63	2	£5.26
1n4001 Diode	1N4001	£0.075	2	£0.15
1N4148 Diode	1N4148TR	£0.0825	2	£0.17
Schottky Diode	BAT85S-TR	£0.269	14	£3.77
5.1v Zener	BZX55C5V1-TAP	£0.147	2	£0.29
BC547 Transistor	BC547B	£0.213	2	£0.43
5mm Green LED	L02R5000Q1	£0.0869	1	£0.09
5mm Blue LED	C503B-BCS-CV0Z0461	£0.167	1	£0.17
5mm Amber LED	MCL053BD	£0.198	1	£0.20
2way terminal block	1725656	£0.494	2	£0.99
3way terminal block	1715734	£1.10	1	£1.10
4way terminal block	1725672	£1.03	1	£1.03
6way terminal block	MC000030	£0.886	1	£0.89
			Total	£55.59

Item Description	Part No	Cost Each	QTY	Total Cost
Components from eBay.co.uk [12]				
Relay	Omron G5S-1 12Vdc	£0.60	2	£1.20
15mm 12v 15mA Blue Pilot LED	N/A	£2.99	1	£2.99
15mm 12v 15mA Amber Pilot LED	N/A	£2.89	1	£2.89
Total				£7.08

Item Description	Part No	Cost Each	QTY	Total Cost
PCB from OSHPark.com [9]				
PCB	E-Z Tec Discriminator	\$54.25	1	£38.73
Total				£38.73

Total Project Cost	£101.40
---------------------------	----------------

Chapter 5 – Evaluation

Overall the project was a success. The main objective of the project was to design and build a solution to industrial metal detection discrimination using an existing Eriez metal detector. That is exactly what was done and the results were successfully achieved.

There are a few things that could be improved if the project were to be carried out again. In the aims and objectives it was specified that all the settings on the original metal detector must still be useable. While this has mostly been achieved, one function which has been left out for no particular reason is the option to have automatic reset or manual reset. The discriminator always automatically resets regardless of what mode the original metal detector is in. If this project were to be put into real world use then this function would seriously be considered and is very easy to implement. The discriminator could have a DIP switch or similar that pulls another PIC pin high or low that tells the PIC to either latch on the outputs until an additional reset pin is toggled (from a reset push button) or automatically reset as it is doing currently.

Before the project was started it was thought that the project would be doable to some extent but there was doubt in the repeatability and reliability for the design to be used in industry. As very small test pieces and very large test pieces had such big effects and opposite results it was thought from the stage of initial testing that the project would only work for a range of test pieces and that the project would not be suitable for industry reliability due to this reason. However, after facing and overcoming these problems by using some simple software code the end result proved that the circuit is very much capable of being sold and used as a solution to industrial metal discrimination.

While comparing the results with the research and literature review there is some correlation to the two but also some differences. It was explained that ferrous metals have a high penetrability factor which increases the coupling factor between the TX and RX. This results in a signal with high amplitude and small phase shift. Whereas non-ferrous metals are the opposite with a low amplitude and high phase shift. On the 9000 receiver circuit there are two demodulator IC which output two signals A sig and P sig. These two signals which are inputted into the discriminator are to represent the amplitude and phase shift of the signal. Therefore it was expected for ferrous metals to give a large peak to peak A sig and small P sig and vice versa for non-ferrous. This was found only to be true with Fe metal spheres greater than 10mm and NFe metal spheres smaller than 2.0mm. Otherwise the results are inverted and the opposite of what was expected. This may be due to how the demodulators are setup and perhaps in the future the demodulators could be modified with discrimination in mind.

The delay and active timers work very well. There was some doubt about whether the timings would be accurate as there are so many interrupts routines and some of them are quite lengthy. A number of test pieces were passed through the coil and the time it took for the detections to take place was very accurate within a few milliseconds even when mixing Fe and NFe detections in between. The system was extensively tested using the delay and active timers and various different metal test pieces. The tester deliberately tried to find bugs with the program such as passing through NFe when the discriminator just detected Fe to see if it would miss the detection or change the delay/active time but no bugs such as this could be found. If this were to be improved however, there could be more detect registers implemented so that the discriminator can store more than 10 detects at a time.

Chapter 6 – Conclusion

Due to increasing requests from industries for a solution to industrial metal detection discrimination, this project was set out to research and develop a circuit capable of discriminating ferrous metals from non-ferrous metals. The project has successfully demonstrated what the project was set out to achieve.

Looking at the aims and objectives of the project in more detail the project has mostly achieved these objectives. The only objective missing is the ability for the discriminator circuit to have a manual reset mode where the outputs remain on until reset by the user. Although this wasn't specifically stated as an objective it was required that all the settings on the original metal detector were still operational with manual reset being one of them. This manual reset option could however still be used on the original metal detector despite the discriminator circuit automatically resetting its outputs. This option could easily be implemented in a future revision.

Research was carried out on two different types of metal detection technologies with the best technology chosen. It is clear that the technology chosen was a good decision due to the project being a success.

Research was carried out to find similar products which already exist on the market. The lack of industrial discrimination products on the market is quite surprising considering the amount of requests made from customers. Metal discriminators are however very popular with treasure hunting detectors such as the Velleman CS150. Although these types of discriminators have a similar goal to that of this project they are also quite different. These types of metal detectors give a signal based on the strength of the metal detected. The user then decides based on experience if the tone sounded is worth the time to dig up. The Velleman CS150 is fairly similar to this project however as it does have ferrous and non-ferrous indicators to tell the user if it was mostly ferrous or non-ferrous. These types of metal detectors also have a lot of tuning functionality. The tuning of metal detectors was omitted from this report as it is not important for the circuit which was designed. However, in a future revision of the project it would be nice to see more options in terms of tuning capabilities which the operator can control. This could allow certain objects to be ignored and others to be detected. For example some products cause what is known as 'product effect' which creates a signal in the metal detector and if this is known then is tuned out during manufacture or commissioning. A discriminator with tuning controls could allow the operator to dampen out certain signals without the need to call an engineer.

The discriminator can have many benefits in industry and also on the environment from reducing the amount of product removed from industrial process lines. Some metals can stay in the product as it doesn't affect the process whereas other types of metals need to be removed. The current solution is to reject all metals which results in product being put in landfill sites which is perfectly fine to use in the process. The discriminator can help to eliminate this wasted material by only rejecting product containing metals which are damaging to the process line.

Overall the project has proved that industrial metal detection discrimination works successfully and this project has opened up new ideas and possibilities for future designs.

Chapter 7 – Reflection

From starting my degree I had not long transitioned into a new job. Originally working as an industrial maintenance electrician then transitioning into a service engineer role with a new company which heavily involves knowledge of electronics was a fairly big step. The BEng has allowed me to improve my knowledge and understanding of electronics and with the aim of progressing into a career developing electronics this project has been very helpful with understanding what is required and how to plan a project from start to finish.

This project has also allowed me to kick start my career as a metal detection engineer by researching into how metal detectors function. There was a lot of outside work which involved analysing circuit schematics from the different metal detectors in order to help achieve a solution for this project. This has greatly improved my knowledge of metal detectors particularly the ones supplied by Eriez magnetics. As a service engineer I am required to travel globally when customers have problems with their metal detectors. The knowledge I have gained from this project greatly helps me to understand the faults with circuits that I am faced with and gives me confidence during difficult times that I am able to repair the equipment on time.

Unfortunately, due to my career requiring I travel globally at a short notice, it felt almost impossible to plan the project as I wouldn't know if I was in the country to be able to work on the project during any given date. I did however eventually take this up with my company who understood this problem and we decided on set dates where I would be available for travel or when I would need time to stay in the country to work on my project. This helped me to better plan the progress of the project. I did slip behind the plan and was unable to get everything completed and I feel if I had spoken to my employer about it sooner then perhaps I could have gotten everything completed. I have now learnt to always discuss problems that I am facing rather than trying to fight them alone and has given me confidence to discuss my needs and requirements.

Overall I am happy with the project. The main objective was accomplished which gives a very rewarding feeling. My company are also happy with the project and the senior engineers are willing to someday work with me to develop a new product using the information I have required from completing this project.

Chapter 8 - References

- [1] Eriez Magnetics Europe Limited, "Pulse Induction - Basic Principles".
- [2] L. Bies, "Pulse induction metal detector with DSP," August 2017. [Online]. Available: https://www.lammertbies.nl/electronics/PI_metal_detector.html. [Accessed 07 October 2017].
- [3] Eriez Magnetics Europe Limited, "E-Z TEC 9000 Series Training Notes".
- [4] Engineering, Tectron, "Model 8050," Tectron Engineering, [Online]. Available: <https://www.tectron.net/specialty-metal-detection-equipment/model-8050/>. [Accessed 07 October 2017].
- [5] Process Automation Ltd, "Model 8050 Metal Detector - "The Discriminator"," Process Automation Ltd, [Online]. Available: <https://www.process-auto.com/product/model-8050-metal-detector-the-discriminator>. [Accessed 07 October 2017].
- [6] jr-international, "CS150 MEtal Detector Owner's Manual," [Online]. Available: <https://jr-international.fr/kiosque/notices/DM6%20AN%20NOT.pdf>. [Accessed 15 January 2018].
- [7] J. ELECTRONICS, "L7800 Series Voltage Regulators," [Online]. Available: <https://www.jameco.com/Jameco/Products/ProdDS/889348.pdf>. [Accessed 02 April 2018].
- [8] Microchip, "Microchip Advanced Parts Selector Tool," [Online]. Available: <https://www.microchip.com/maps/Microcontroller.aspx>. [Accessed 16 April 2018].
- [9] O. Park, "Prototype PCB Manufacturing," [Online]. Available: <https://oshpark.com/>. [Accessed 12 March 2018].
- [10] M. P. Bates, Programming 8-bit PIC Microcontrollers in C, Newnespress, 2010.
- [11] F. U. Limited, "Electronics Supplier," [Online]. Available: <http://uk.farnell.com/>. [Accessed 2018].
- [12] eBay. [Online]. Available: <https://www.ebay.co.uk/>. [Accessed 2018].
- [13] Eriez Magnetics Europe Ltd, *E-Z Tec 9100 Schematic*, 2003.
- [14] jr-international, "Velleman CS150 Metal Detector," [Online]. Available: https://jr-international.fr/detector-professional-metal-detector_DM6_itm_english.html. [Accessed 15 January 2018].

Chapter 9 – Appendix 1 – Software Code

```

1  #include <Discriminator.h>
2  ///////////////////////////////////////////////////
3  ///////////////////////////////////////////////////Variables//////////////////////////////////////
4  ///////////////////////////////////////////////////
5  int16 value; // ADC Value
6  int16 AS1Max=512; int16 AS2Max=512; int16 AS3Max=512; // A Sig Max Values
7  int16 PS1Max=512; int16 PS2Max=512; int16 PS3Max=512; // P Sig Max Values
8  int16 AS1Min=512; int16 AS2Min=512; int16 AS3Min=512; // A Sig Min Values
9  int16 PS1Min=512; int16 PS2Min=512; int16 PS3Min=512; // P Sig Min Values
10 int16 ASMax, PSMax, ASMin, PSMin, ASValue, PSValue; // Absolute Values
11 int16 FeDelayADC; int16 NFeDelayADC; // Fe/NFe Delay Time
12 int16 FeDelay, FeActive, NFeDelay, NFeActive; // Delay/Active Values
13 int16 FeDelay1, FeDelay2, FeDelay3, FeDelay4, FeDelay5, FeDelay6, FeDelay7, FeDelay8, FeDelay9; //Fe Delay registers
14 int16 NFeDelay1, NFeDelay2, NFeDelay3, NFeDelay4, NFeDelay5, NFeDelay6, NFeDelay7, NFeDelay8, NFeDelay9; //NFe Delay registers
15 int8 FeDRA=0; int8 NFeDRA=0; // Fe/NFe Delay Register Allocation
16 int16 St, AST, PSt;
17 int16 LargeFe;
18 signed int16 Pt;
19 int1 Detect, Fe, NFe;
20 ///////////////////////////////////////////////////
21 ///////////////////////////////////////////////////External Interrupt//////////////////////////////////////
22 ///////////////////////////////////////////////////
23 #INT_EXT // Interrupt name
24 void EXT_isr(void) // External ISR
25 {
26 Detect=1; // Set Detect flag
27 clear_interrupt(INT_Timer1); // Clear interrupt flag
28 set_Timer1(15124); // Set discriminator timer - 200ms delay
29 enable_interrupts(INT_Timer1); // Enable discriminator interrupt
30 disable_interrupts(INT_EXT); // Prevents double detect from large metal
31 }
32
33 ///////////////////////////////////////////////////
34 ///////////////////////////////////////////////////Discriminator//////////////////////////////////////
35 ///////////////////////////////////////////////////
36 ///////////////////////////////////////////////////
37 #INT_TIMER1 // Interrupt name
38 void isr_Timer1(void) // Timer1 ISR
39 {
40 Detect=0; // Reset detect flag
41 Fe=0, NFe=0; // Debug
42 ASMax=AS1Max+AS2Max+AS3Max; // Find total A sig max
43 ASMin=AS1Min+AS2Min+AS3Min; // Find total A sig min
44 PSMax=PS1Max+PS2Max+PS3Max; // Find total P sig max
45 PSMin=PS1Min+PS2Min+PS3Min; // Find total P sig min
46
47 ASValue=ASMax-ASMin; //A Sig Total Value
48 PSValue=PSMax-PSMin; //P Sig Total Value
49
50 LargeFe=AS3Max-PS3Max; // Find difference between AS3 & PS3
51
52 Pt=AST-PSt; //Time between A&P sig positive peaks (Phase Shift)
53
54 if(Pt>40){ //If Phase has shifted
55     if(AS1Max>819)Fe=1; //Phase shift with ASig>4.0v = Fe
56     else if(AS1Max<819)NFe=1; //Phase shift with ASig<4.0 = NFe
57
58 } else if(Pt<-40){ //If Phase has shifted
59     if(AS1Max>819)Fe=1; //Phase shift with ASig>4.0v = Fe
60     else if(AS1Max<819)NFe=1; //Phase shift with ASig<4.0 = NFe
61
62 } else if(AS1Max<553)Fe=1; //If ASig1 < 2.7v and no phase shift = Fe
63
64 } else if(AS3Max>615){ //If ASig3 > 3.0v
65     if(LargeFe>225)Fe=1; //If difference between AS3 & PS3 > 1.1v
66     else if(LargeFe<225)NFe=1;
67
68 } else if(ASValue>PSValue)NFe=1; //NFe Detect
69
70 } else if(ASValue<PSValue)Fe=1; //Fe Detect
71
72
73
74 //ERROR//
75 if(Fe==NFe){
76     Fe=1;
77     NFe=1;
78 }
79

```

```

80 ///////////////////////////////////////////////////Fe Detect//////////////////////////////////////
81 if(Fe==1){
82     FeDRA++;
83     if(FeDelay==0) //If no detection pending
84     {
85         set_adc_channel (Fe_Delay); // Fe Delay read
86         delay_us(15);
87         FeDelayADC=read_adc()*28;
88         FeDelay=FeDelayADC;
89         enable_interrupts(INT_Timer0); // Enable Fe Delay Timer interrupt
90         set_Timer0(0);
91     }
92     else if(FeDRA==2)FeDelay1=FeDelayADC-FeDelay;
93     else if(FeDRA==3)FeDelay2=FeDelayADC-FeDelay-FeDelay1;
94     else if(FeDRA==4)FeDelay3=FeDelayADC-FeDelay-FeDelay1-FeDelay2;
95     else if(FeDRA==5)FeDelay4=FeDelayADC-FeDelay-FeDelay1-FeDelay2-FeDelay3;
96     else if(FeDRA==6)FeDelay5=FeDelayADC-FeDelay-FeDelay1-FeDelay2-FeDelay3-FeDelay4;
97     else if(FeDRA==7)FeDelay6=FeDelayADC-FeDelay-FeDelay1-FeDelay2-FeDelay3-FeDelay4-FeDelay5;
98     else if(FeDRA==8)FeDelay7=FeDelayADC-FeDelay-FeDelay1-FeDelay2-FeDelay3-FeDelay4-FeDelay5-FeDelay6;
99     else if(FeDRA==9)FeDelay8=FeDelayADC-FeDelay-FeDelay1-FeDelay2-FeDelay3-FeDelay4-FeDelay5-FeDelay6-FeDelay7;
100    else if(FeDRA==10)FeDelay9=FeDelayADC-FeDelay-FeDelay1-FeDelay2-FeDelay3-FeDelay4-FeDelay5-FeDelay6-FeDelay7-FeDelay8;
101    else if(FeDRA>10){FeDRA--;}
102 }
103
104 ///////////////////////////////////////////////////NFe Detect//////////////////////////////////////
105
106 if(NFe==1){ //NFe Detect
107     NFeDRA++;
108     if(NFeDelay==0)
109     {
110         set_adc_channel (NFe_Delay); // NFe Delay read
111         delay_us(15);
112         NFeDelayADC=read_adc()*36;
113         NFeDelay=NFeDelayADC;
114         enable_interrupts(INT_Timer4); // Enable NFe Active Timer interrupt
115         set_Timer4(0);
116     }
117     else if(NFeDRA==2)NFeDelay1=NFeDelayADC-NFeDelay;
118     else if(NFeDRA==3)NFeDelay2=NFeDelayADC-NFeDelay-NFeDelay1;
119     else if(NFeDRA==4)NFeDelay3=NFeDelayADC-NFeDelay-NFeDelay1-NFeDelay2;
120     else if(NFeDRA==5)NFeDelay4=NFeDelayADC-NFeDelay-NFeDelay1-NFeDelay2-NFeDelay3;
121     else if(NFeDRA==6)NFeDelay5=NFeDelayADC-NFeDelay-NFeDelay1-NFeDelay2-NFeDelay3-NFeDelay4;
122     else if(NFeDRA==7)NFeDelay6=NFeDelayADC-NFeDelay-NFeDelay1-NFeDelay2-NFeDelay3-NFeDelay4-NFeDelay5;
123     else if(NFeDRA==8)NFeDelay7=NFeDelayADC-NFeDelay-NFeDelay1-NFeDelay2-NFeDelay3-NFeDelay4-NFeDelay5-NFeDelay6;
124     else if(NFeDRA==9)NFeDelay8=NFeDelayADC-NFeDelay-NFeDelay1-NFeDelay2-NFeDelay3-NFeDelay4-NFeDelay5-NFeDelay6-NFeDelay7;
125     else if(NFeDRA==10)NFeDelay9=NFeDelayADC-NFeDelay-NFeDelay1-NFeDelay2-NFeDelay3-NFeDelay4-NFeDelay5-NFeDelay6-NFeDelay7-NFeDelay8;
126     else if(NFeDRA>10){NFeDRA--;}
127 }
128
129 ///////////////////////////////////////////////////
130
131
132 ASMax=0; ASMin=0; PSMax=0; PSMin=0; //Reset variables
133 AS1Max=512; AS2Max=512; AS3Max=512;
134 AS1Min=512; AS2Min=512; AS3Min=512;
135 PS1Max=512; PS2Max=512; PS3Max=512;
136 PS1Min=512; PS2Min=512; PS3Min=512;
137 ASValue=0; PSValue=0;
138 st=0; pt=0, Fe=0; NFe=0;
139 LargeFe=0;
140 clear_interrupt(INT_Timer1); //Clear Interrupt
141 clear_interrupt(INT_EXT); //Clear Interrupt
142 enable_interrupts(INT_EXT); //Enable External Interrupt
143 disable_interrupts(INT_Timer1); //Disable Discriminator Interrupt
144 }
145
146
147 ///////////////////////////////////////////////////
148 ///////////////////////////////////////////////////Fe Delay//////////////////////////////////////
149 ///////////////////////////////////////////////////

```

```

147 ///////////////////////////////////////////////////////////////////
148 ///////////////////////////////////////////////////////////////////Fe Delay/////////////////////////////////////////////////////////////////
149 ///////////////////////////////////////////////////////////////////
150 #INT_TIMER0 // Interrupt name
151 void isr_Timer0(void) // Timer0 ISR
152 {
153     if(FeDelay!=0)
154     {
155         FeDelay--;
156     }
157 }
158
159 else{
160     disable_interrupts(INT_Timer0);
161     output_high(Fe_Detect); //Fe Detect Output
162     set_adc_channel (Fe_Active); // Fe Active read
163     delay_us(15);
164     FeDRA--;
165     FeActive=read_adc()*13;
166     enable_interrupts(INT_Timer2); // Enable Fe Active Timer interrupt
167     set_Timer2(0);
168
169     FeDelay=FeDelay1; //Register shift
170     FeDelay1=FeDelay2;
171     FeDelay2=FeDelay3;
172     FeDelay3=FeDelay4;
173     FeDelay4=FeDelay5;
174     FeDelay5=FeDelay6;
175     FeDelay6=FeDelay7;
176     FeDelay7=FeDelay8;
177     FeDelay8=FeDelay9;
178     FeDelay9=0;
179 }
180
181 if(FeDRA!=0){ //Restart Timer
182     clear_interrupt(INT_Timer0);
183     Enable_interrupts(INT_Timer0);
184     set_Timer0(0);}
185
186 else disable_interrupts(INT_Timer0); //Stop Timer
187
188
189 }
190
191
192 ///////////////////////////////////////////////////////////////////
193 ///////////////////////////////////////////////////////////////////Fe Active/////////////////////////////////////////////////////////////////
194 ///////////////////////////////////////////////////////////////////
195 #INT_TIMER2 // Interrupt name
196 void isr_Timer2(void) // Timer2 ISR
197 {
198 }
199
200 if(FeActive>1)
201 {
202     FeActive=FeActive-1;
203 }
204
205 else{
206     output_low(Fe_Detect);
207     FeActive=0;
208     disable_interrupts(INT_Timer2); //Clear Fe Active Interrupt
209 }
210
211 }

```

```

210 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
211 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
212 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
213 #INT_TIMER4 // Interrupt name
214 void isr_Timer4(void) // Timer4 ISR
215 {
216     if(NFeDelay!=0)
217     {
218         NFeDelay--;
219     }
220 }
221 else{
222     disable_interrupts(INT_Timer4);
223     output_high(NFe_Detect);
224     set_adc_channel (NFe_Active); // NFe Active read
225     delay_us(15);
226     NFeDRA--;
227     NFeActive=read_adc()*13;
228     enable_interrupts(INT_Timer6); // Enable NFe Active Timer interrupt
229     set_Timer6(0);
230 }
231 NFeDelay=NFeDelay1; //Register shift
232 NFeDelay1=NFeDelay2;
233 NFeDelay2=NFeDelay3;
234 NFeDelay3=NFeDelay4;
235 NFeDelay4=NFeDelay5;
236 NFeDelay5=NFeDelay6;
237 NFeDelay6=NFeDelay7;
238 NFeDelay7=NFeDelay8;
239 NFeDelay8=NFeDelay9;
240 NFeDelay9=0;
241 }
242 }
243 if(NFeDRA!=0){ //Restart Timer
244     clear_interrupt(INT_Timer4);
245     Enable_interrupts(INT_Timer4);
246     set_Timer4(0);}
247 }
248 else disable_interrupts(INT_Timer4); //Stop Timer
249 }
250 }
251 }
252 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
253 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
254 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
255 #INT_TIMER6 // Interrupt name
256 void isr_Timer6(void) // Timer6 ISR
257 {
258 }
259 if(NFeActive>1)
260 {
261     NFeActive=NFeActive-1;
262 }
263 }
264 else{
265     output_low(NFe_Detect);
266     NFeActive=0;
267     disable_interrupts(INT_Timer6); //Clear NFe Active Interrupt
268 }
269 }
270 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
271 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
272 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
273 void main()
274 {
275     delay_us(500000); //Debug - Without this program hangs
276     setup_adc_ports(sAN0|sAN1|sAN4|sAN5|sAN6|sAN8|sAN9|sAN10|sAN11); //Setup ADC ports
277     setup_adc(ADC_CLOCK_DIV_8); //Setup ADC clock
278     setup_timer_0(RTCC_INTERNAL|RTCC_DIV_8|RTCC_8_BIT); //Setup Timer0 Clock
279     setup_timer_1(T1_INTERNAL|T1_DIV_BY_8); //Setup Timer1 Clock
280     setup_timer_2(T2_DIV_BY_4,0xc0,2); //Setup Timer2 Clock
281     setup_timer_4(T4_DIV_BY_4,0xc0,2); //Setup Timer2 Clock
282     setup_timer_6(T6_DIV_BY_4,0xc0,2); //Setup Timer2 Clock
283 }
284 }
285 enable_interrupts(INT_EXT); // Enable External interrupt
286 enable_interrupts(GLOBAL); // Enable all interrupts
287 ext_int_edge(L_TO_H); // External interrupt signal polarity
288 }
289 FeDelay=0, NFeDelay=0;
290 FeDelay1=0, NFeDelay1=0;
291 FeDelay2=0, NFeDelay2=0;
292 FeDelay3=0, NFeDelay3=0;
293 FeDelay4=0, NFeDelay4=0;
294 FeDelay5=0, NFeDelay5=0;
295 FeDelay6=0, NFeDelay6=0;
296 FeDelay7=0, NFeDelay7=0;
297 FeDelay8=0, NFeDelay8=0;
298 FeDelay9=0, NFeDelay9=0;
299 }
300 output_high(PWR_ON);
301 }

```

```

302 while(1) // Main loop.
303 {
304
305 while(!input(Metal_Sig)) // While waiting for metal sig - reset values.
306 {
307 if(Detect!=1){
308 AS1Max=512; AS2Max=512; AS3Max=512;
309 AS1Min=512; AS2Min=512; AS3Min=512;
310 PS1Max=512; PS2Max=512; PS3Max=512;
311 PS1Min=512; PS2Min=512; PS3Min=512;
312 St=0; AS1=0; PS1=0;
313 Fe=0; NFe=0;
314 LargeFe=0;}
315 else{break;}
316
317 }
318
319 while(input(Metal_Sig)) // During metal sig - Sample signals
320 {
321
322 set_adc_channel (A_Sig_1); // A Sig 1 read
323 delay_us(15); // Wait for ADC
324 value=read_adc(); // Write ADC value
325 if(value>AS1Max){AS1Max=value; // Write Max value
326 AS1=St;} //Write Phase Time of Max Value
327 else if(value<AS1Min)AS1Min=value; // Write Min value
328
329
330 set_adc_channel (P_Sig_1); // P Sig 1 read
331 delay_us(15);
332 value=read_adc();
333 if(value>PS1Max){PS1Max=value;
334 PS1=St;}
335 else if(value<PS1Min)PS1Min=value;
336
337 set_adc_channel (A_Sig_2); // A Sig 2 read
338 delay_us(15);
339 value=read_adc();
340 if(value>AS2Max)AS2Max=value;
341 else if(value<AS2Min)AS2Min=value;
342
343 set_adc_channel (P_Sig_2); // P Sig 2 read
344 delay_us(15);
345 value=read_adc();
346 if(value>PS2Max)PS2Max=value;
347 else if(value<PS2Min)PS2Min=value;
348
349 set_adc_channel (A_Sig_3); // A Sig 3 read
350 delay_us(15);
351 value=read_adc();
352 if(value>AS3Max)AS3Max=value;
353 else if(value<AS3Min)AS3Min=value;
354
355 set_adc_channel (P_Sig_3); // P Sig 3 read
356 delay_us(15);
357 value=read_adc();
358 if(value>PS3Max)PS3Max=value;
359 else if(value<PS3Min)PS3Min=value;
360
361 St++; //Increment Sample Count
362
363 }
364 }
365 }
366

```

Discriminator.h;

```
1  #include <16F1829.h>
2  #device ADC=10
3
4  #FUSES NOWDT           //No Watch Dog Timer
5  #FUSES NOMCLR          //Master Clear pin used for I/O
6  #FUSES NOBROWNOUT     //No brownout reset
7  #FUSES NOLVP           //No low voltage prgming, B3(PIC16) or B5(PIC18) used for I/O
8
9  #use delay(internal=8000000)
10 #use FIXED_IO( A_outputs=PIN_A5 )
11 #use FIXED_IO( B_outputs=PIN_B7,PIN_B6 )
12
13 #define A_Sig_1  0      // A Sig Inputs
14 #define A_Sig_2  4
15 #define A_sig_3  1
16
17 #define P_Sig_1  7      // P Sig Inputs
18 #define P_Sig_2  9
19 #define P_Sig_3  8
20
21 #define Fe_Delay  5      // On/Off Delay Inputs
22 #define Fe_Active 6
23 #define NFe_Delay 10
24 #define NFe_Active 11
25
26 #define Metal_Sig PIN_A3 // Digital Inputs
27 #define Detect_Sig PIN_A2
28
29 #define PWR_On    PIN_A5 // Digital Outputs
30 #define Fe_Detect PIN_B6
31 #define NFe_Detect PIN_B7
32
```