DEVELOPMENT OF DIGITAL INSTRUMENTATION FOR BOND RUPTURE DETECTION

A thesis presented in partial fulfilment
of the requirements for the degree of

Doctor of Philosophy

in

Engineering

at Massey University, Palmerston North,
New Zealand

Matthew John van der Werff

2009

ABSTRACT

In the medical world the precise identification of a disease can take longer than it is safe to wait to start treatment so there is a need for faster and more precise biosensors. Bond Rupture is a new sensor technique that maybe able to improve disease detection. It does this by inducing bonds to rupture from the surface, and also measuring the point at which this rupture occurs this enables the forces to be measured on the surface. Specifically, this project has focused on the application of Bond Rupture to detecting antigens when bound to a surface using their specific antibodies, and the idea that the rupture force of these antigens can also be measured. The sensor that this project is based around is the Quartz Crystal Microbalance (QCM), which oscillates horizontally when a voltage is applied, and can also be used to measure mass change on its surface via change in resonant frequency.

The aim of this project was to investigate possible Bond Rupture detection methods and techniques and has involved the development of a high speed digital electronics system, for the purposes of inducing and detecting Bond Rupture. This has involved the development of a FPGA based high speed transceiver board which is controlled by a Digital Signal Processor (DSP), as well as the development of various graphical user interfaces for end user interaction. Bond rupture testing was carried out by rupturing beads from the surface of a QCM in an experiment taking as little as 20 seconds.

The Bond Rupture effect has been observed via the high accuracy measurement of the frequency change while inducing Bond Rupture on the sensor, proving that the Bond Rupture effect indeed exists. The research performed is believed to be a world first in terms of the method used and accuracy acquired.

ACKNOWLEDGEMENTS

Like many others before me, the thesis writing has taken much longer than originally expected. I am grateful for the patience of those around me, waiting for me to finish.

First I would like to thank my three supervisors. To Yong Yuan who initiated this project, I have appreciated working with many more ideas than I could ever hope to implement and for giving me a great appreciation of Chinese cuisine. To Peter Xu for inspiring me to do my PhD, his excellent management in ensuring that the project kept ticking along. To John Bronlund though turning up as an official supervisor in the second year, his advice, ideas, and dropping in for chats has been invaluable. I have appreciated the fun times we have as a group have had discussing ideas for the project, and not to mention the subjects from food and wine, to politics, to skiing. I would like to thank Harry Chen for laboratory experiments carried out and preparation and putting up with the initial bugs in the developed system. I would like to thank Evan Hirst (the other PhD working on this project) who has been in this project almost as long as me, for being there to bounce ideas off and pointing out some of the more obvious problems when I have become carried away with an idea.

I am grateful for the mentoring given to me by Robin Dykstra who has generously given of his time to explain the ins and outs of signal processing, among other topics. I also appreciate the hardware which Robin has donated to this project saving a significant amount of time, which may have been devoted to reinventing the wheel instead of focusing on the project on hand. I am also grateful to the boys of the NMR lab Rob Ward and Terry Southern, both of whom have given me many ideas and provided me somewhere to bounce ideas around. I should also thank them for the many hours of debates into issues as wide as, a fierce debate of plunger versus espresso coffee, to the origin of life and the universe. I would like to especially thank Terry for his input into the initial design specifications of the FPGA board and other help with circuit design and DSP programming problems.

I would like to thank Warwick Taylor for the many hours spent proof reading my thesis.

I would like to thank my lovely wife Larina who has lovingly supported me though this long time of financial deficit, and Caleb my little boy (now seven months old) who has managed to distract me from work many times though the final months of writing. I would like to thank my parents Kirsty and Dave for their support from afar (thanks Skype), and my little sister Jasmin who said if no mention was made of her that I would be in big trouble. I would like to thank my mother in-law Lyn for all the help provided over the years, and my brother in-law Max for a great source of questions and ideas. Also thank the rest of my extended family, other brothers and sisters and their in-law counter parts, for their input and support. I would also like to thank other friends around me that have supported me during this time. I will end with thanking my father in-law Keith Betteridge for not only being there to bounce ideas off, but for that constant dreaded question “So have you finished your thesis yet?” to which I can now finally answer yes!

RELATED PUBLICATIONS

van der Werff, M. J., Y. J. Yuan, W.L. Xu (2005). Quartz Crystal Microbalance for Medical Diagnostics. First International Conference on Sensing Technology. S. C. M. G.Sen Gupta, C. H. Messom. Palmerston North, New Zealand. ***1:*** *500-504.*

van der Werff, M. J., Y. J. Yuan, E.R. Hirst, J. Bronlund, W.L. Xu (2007). "Quartz crystal microbalance induced bond rupture sensing for medical diagnostics." Ieee Sensors Journal 7(5-6): 762-769.

Yuan, Y. J., M. J. van der Werff, H. Chen, E.R. Hirst, W.L. Xu (2007). "Bond rupture of biomolecular interactions by resonant Quartz crystal." Analytical Chemistry ***79****(23): 9039-9044.*

van der Werff, M. J., Y. J. Yuan, W.L. Xu (2009) "QCM Bond Rupture using a FPGA based transceiver". " IEEE Sensors Journal, to be submitted.

CONTENTS

1 INTRODUCTION

1.1 BACKGROUND

In the medical world the precise identification of a disease can take longer than it is safe to wait to start treatment which is why doctors are often forced to treat disease from initial symptoms. Diagnostic tests are also important in the pharmaceutical industry to test how effective a cure is on a disease. Traditional antigen identification has been performed using direct culture techniques that are very time consuming and may take hours or days to give a result. Many differing types of methods for the identification of antigens exist including: fluorescence labelling, radioactive labelling, optical scanning, and oligonucleotide based assays, to name a few all with their individual advantages and disadvantages, but most trade sensitivity for speed. Other methods include biosensors such as Surface Plasmon Resonance (SPR), potentiometric biosensors, and Quartz Crystal Microbalance (QCM). All of the above can detect surface binding of antibodies and antigens though they have problems related to unknown antigens bonding to antibodies with a much weaker bonding force (non-specific bonding). To get around the non-specific bonding problems a method was proposed by Dultsev et al. (2000) that measures the bond strengths of the attached antigen by applying an increasing voltage to a QCM while also measuring the loss of mass to the QCM so creating a spectrum of the bond forces. This method enables precise identification of antigens via their bond strengths.

This project is a collaborative project between Mechatronic engineers and chemical scientists. The Bond Rupture group is headed by Dr Yong Yuan from Industrial Research Limited, who supervised this project along with Prof Peter Xu from Massey University Albany, and Prof John Bronlund from Massey University Turitea. This author's task was to develop the electrical equipment for detecting antigens using a Quartz Crystal Microbalance as the sensor. This included developing methodologies and end user systems to induce Bond Rupture while effectively monitoring the resonant frequency and Bond Rupture 'noise'. The majority of the chemical experiments were carried out at IRL by lab technicians due to the sensitivity of the chemicals used. As the experiments were not always performed by the author, much work was involved to make the interfaces that control the equipment as intuitive as possible, while also enabling as much control of the equipment as possible to enable variability in the types of experiments and to determine the optimum settings in differing environments.

1.2 AIMS AND OBJECTIVES

The main mechanism that will be investigated in this thesis will be QCM based Bond Rupture. This has been proposed to be able to be detected via two methods, both using the QCM as a transducer to induce Bond Rupture. The first method monitors the frequency change and the second method monitors a signal that is generated at the QCM when Bond Rupture occurs. The goal of this thesis is to develop a QCM Bond Rupture system that can verify both Bond Rupture techniques and be used in the future for further research into more advanced Bond Rupture techniques. This should enable scientists to find future applications of this technology. The Bond Rupture equipment is developed to enable multiple different types of experiments

on one piece of equipment. This thesis describes the process that was taken in the development of this equipment.

1.3 OUTLINE

This thesis outlines the steps and background for the creation of a digital based Bond Rupture detection system. Chapter 2 contains literature pertaining to the background of Bond Rupture and relevant bio-sensor information. Chapter 3 includes the initial QCM experimentation and characterisation to determine the specifications for the Bond Rupture system. This included investigation into the effects of temperature and humidity on the QCM, the stability of the QCM over time, and the initial Bond Rupture testing and system specifications. Chapter 4 contains the design and development of the electronics system as well as the Bond Rupture experiments carried out. Frequency based Bond Rupture was verified within this chapter but it was found that further development was required to verify Bond Rupture. Chapter 5 outlines the development of an improved system (based on the experimental data from chapter 4) including the development of a Field Programmable Gate Array (FPGA) based system that implemented simultaneous monitoring of two narrow band frequencies – enabling the monitoring for Bond Rupture 'noise' while also monitoring and driving the resonant frequency of the QCM while using only one Analog to Digital Converter (ADC). This chapter also includes justification for the development of the equipment and the experiments that were carried out. Chapter 6 concludes the thesis with a summary of what has been achieved and a number of suggestions for future research.

1.4 MAJOR CONTRIBUTIONS

There are a number of contributions this thesis has made. This includes a comprehensive literature review into Bond Rupture techniques. Also, an initial Bond Rupture measurement system and software was developed to verify bond rupture. And after verification further development was carried out to develop the next generation Bond Rupture measurement system, which has enabled the development of new Bond Rupture measurement techniques. A number of experiments were also carried out to simulate an antibody-antigen Bond Rupture using Streptavidin coated beads, attached to a biotin coated QCM. These experiments confirmed the Bond Rupture effect, showing that bonds can be ruptured from the surface using a QCM and that this can be detected accurately. Though the Bond Rupture frequency method was confirmed unfortunately no conclusive results were found to confirm that Bond Rupture 'noise' is generated from bonds breaking during Bond Rupture, leading this author to believe that the Bond Rupture 'noise' does not exist in the way described in literature. However, the Bond Rupture frequency method shows significant promise with scan times performed in as little time as 20 seconds, and enabling Bonds rupturing to be identified to a significant amount of precision.

2 LITERATURE REVIEW

2.1 INTRODUCTION

There are many papers that underlie the biosensors and the Bond Rupture method. These will be covered in this literature review starting with introduction to immunological biosensors in which a combination of different sensors will be examined, as well as examining the antibody-antigen interactions. The QCM mass sensor is examined in more detail, as this is the main sensor that will be used, with focus being on modelling, electronic detection methods, and measurement limits. The next section is Bond Rupture in which the mechanism for this is examined in some detail and electronic detection methods are examined. The last section is the analysis of the literature in which conclusions are formed as to the direction the research will head.

2.2 IMMUNOLOGICAL BIOSENSORS

There have been many different types of biosensors developed for a variety of biological applications. The application of biological sensors that will be focused on is the detection of antigens. Antigens are defined as anything that is foreign to the body which stimulates the production of antibodies by the immune system. This can include foreign proteins, toxins, bacteria or viruses. It is possible to create certain antibodies that specifically bind to these substances and so most biosensors use antibodies to detect the antigens of interest. Immunoassay is the name given to the group of analytical tests that use antibodies in this way. Thus when an immunoassay is carried out with a biosensor the term immunological biosensor is formed. An immunological biosensor has many advantages over traditional immunoassays including: automated experiments, signal processing gain for detection, more sensitivity, faster and more specific antigen selection.

Biosensors are a combination between a biological detection method and an electrical sensor. Biological detection methods vary from an indicator that produces a colour change when chemicals react and can be seen with the naked eye, to a biological method that introduces glowing-particles to a solution that when bound with the desired particles they can be seen under a microscope - so identifying if particles exist and if so how many. Electrical sensors convert physical properties such as light, weight, or sound into an electrical signal representing an amount of the light, or weight as a current or voltage (see Figure 2.1). Biosensors combine both these methods resulting in a single sensor with considerable advantages over traditional biological detection methods.

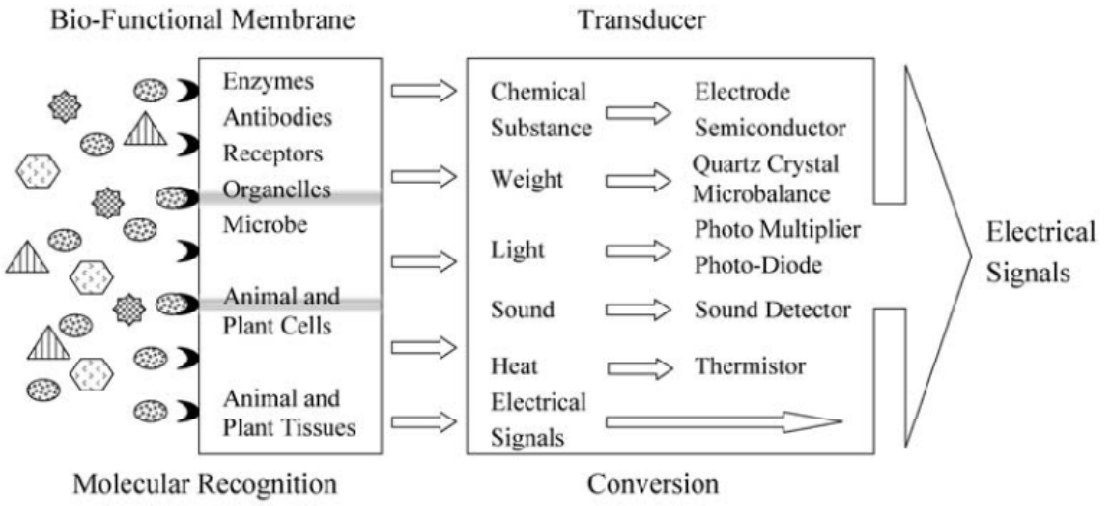


Figure 2.1: The Principles of the biosensor (Nakamura and Karube 2003).

Immunoassays are a subset of methods within biological detection using antibodies. So an immunological biosensor is the same as a standard biosensor except that it uses antibodies as its primary detection mechanism. The antibodies are used to select the antigens of interest which can then be tested to determine the amount of antigens present via a mass, light, and electrical conductivity via a sensor element.

2.2.1 TYPES OF BIOSENSORS

There are a large range of different devices that can detect minuscule mass changes. The majority of mass sensors use the piezoelectric effect. The piezoelectric effect occurs in various crystalline structures such as Quartz. When these structures are deformed a voltage is produced - this can be seen in modern electric lighters which use this effect to produce a spark. The opposite is also true as when a voltage is applied a deformation is observed. The mass detection methods mainly rely on measuring the resonant frequency of piezoelectric devices. This resonant frequency occurs due to the physical structure of the created devices with certain frequencies existing where the device can more easily vibrate at, or to put another way, the frequencies where the least mechanical resistance exists. These devices can be designed with very good frequency stability over a wide range of different frequencies. These devices can also be designed to be sensitive to mass changes on their surfaces, and so enabling detection of surface attachment.

Optical immunological methods are able to probe surfaces and films without destroying them. Optical methods can use many different properties of light including: fluorescence, reflectance, light absorbance, and refractive index. Though there are many methods for optical antibody-antigen detection two methods that are relevant (due to their use as verification for QCM methods) is Surface Plasmon Resonance (SPR) and the Atomic Force Microscope (AFM). AFM is not typically classed as a biosensor but due to the use of it in detecting antibody-antigen forces it has been included in this section. These techniques will be examined in detail in the following sections.

There have been a large number of reviews in this subject a few of the most relevant have been in referenced for further investigation (Collings and Caruso 1997; Ivnitski, Abdel-Hamid et al. 1999; Cooper 2003; D'Orazio 2003; Nakamura and Karube 2003).

2.2.1.1 QUARTZ CRYSTAL MICROBALANCE (QCM)

Quartz crystal resonators have been widely used for oscillators for radio transmitting and receiving and more recently for the timing of computer CPUs and microcontrollers. Quartz crystal resonators oscillate at a designed frequency using an inverter connected from one side of the crystal to the other. The crystal naturally operates at the frequency of least operating resistance, or resonant frequency.

Sauerbrey (Sauerbrey 1959) discovered that the mass change of the surface of a certain type of Quartz crystal (AT cut) when made into a resonator the mass added to the surface was proportional to the frequency change. So the Quartz crystal microbalance (QCM) was formed. This frequency based mass detection QCM typically consists of a Quartz crystal, electrodes on either side (usually gold but can be almost any conducting layer), and wire connects –as seen in Figure 2.2. A QCM is a shear mode oscillator meaning the surface moves horizontally along the surface with virtually no vertical motion. A QCM is used as a biosensor typically by attaching the particles of interest on the surface and monitoring the corresponding change in frequency.

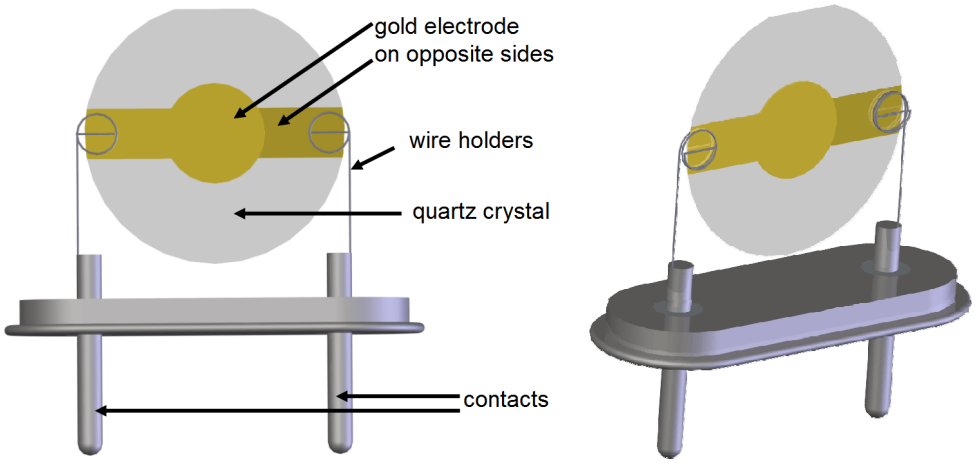


Figure 2.2: Typical Quartz crystal microbalance (QCM) - 3D CAD renderings developed in solid works.

Kanazawa (Kanazawa and Gordon 1985) discovered a formula that described the relationship between density and viscosity of a liquid on the surface of the QCM. This enabled further advancements for QCM operation in liquids and enabling a liquid based biosensor. The Quartz crystal microbalance as a sensor will be further investigated in the "Quartz Crystal Microbalance (QCM)" section on page 15.

There has been much research in the area of using QCM's to detect antigens. A recent review was carried out by Cooper (Cooper and Singleton 2007) investigating the different QCM techniques, which are used to detect bacteria (Table 2.1a) and viruses (Table 2.1b). The different methods vary a lot in their sensitivity and assay times. The sensitivities and times

using a biosensor are much better than traditional assay based techniques though most techniques seem to focus on one application rather than a generic technique.

(a)

Analyte	Matrix	Sensitivity (cells or CFU ml^{-1})	Assay time (min)	Reference
S. typhimurium	Not stated	10^7	40	Babacan *et al.* (2002)
S. typhimurium	Chicken exudate	10^3	Not stated	Olsen *et al.* (2003)
S. typhimurium	PBS	10^3	30	Kim *et al.* (2003)
S. typhimurium	BHY broth/chicken meat solution	10^2	60	Su and Li (2005b)
S. paratyphi	PBS	170	50	Fung and Wong (2001)
M. tuberculosis	Sputum	200	30	He *et al.* (2002)
E. coli	Food samples	1.7×10^5	20–30	Kim and Park (2003)
E. coli	BHY broth	10^3	30–50	Su and Li (2004)
E. coli	BHY broth	10^6	60	Su and Li (2005a)
P. aeruginosa	Nutrient broth	1.3×10^7	20	Kim *et al.* (2004)
F. tularensis	PBS	5×10^6	35	Pohanka and Skladal (2005)
S. enteritidis	Saline solution	10^5	35	Si *et al.* (2001)
L. monocytogenes	Tris buffer	10^7	30	Vaughan *et al.* (2001)
Salmonella spp.	PBS	10^4	10	Wong *et al.* (2002)

(b)

Analyte	Matrix	Sensitivity (virions or plaque forming units)	Assay time (min)	Reference
Herpes simplex virus 1	PBS	10 virions/μl	10	Cooper *et al.* (2001)
Herpes simplex virus 1	Serum	100 virions/μl	10	Cooper *et al.* (2001)
Flavivirus (dengue)	Serum	0.7 μg ml^{-1} (ca. 2.1×10^7 virions/ml)*	10	Wu *et al.* (2005)
Hepatitis B	PBS	0.02 μg ml^{-1} (ca. 3×10^6 virions/ml)*	55	Zhou *et al.* (2002)
M13 Bacteriophage	Water	800 virions/μl	16	Dultsev *et al.* (2001)
M13 Bacteriophage	PBS	5×10^9 pfu ml^{-1}	15	Uttenthaler *et al.* (2001)
Orchid viruses	PBS	0.2 ng μl^{-} (ca. 3×10^4 virions/μl)*		Eun *et al.* (2002b)
Bovine ephemeral fever virus	PBS	5 μg ml^{-1} (ca. 1.5×10^8 virions/ml)*	10	Lee and Chang (2005)

* Based on an average 10^8 Da/virion.

Table 2.1: Comparison of assays and detection limits (Cooper and Singleton 2007) for (a) bacteria (b) viruses.

The QCM technique that will be focused on will be the Bond Rupture technique that was introduced by Dultsev (Dultsev, Ostanin et al. 2000). This technique was continued on by Cooper detecting the Herpes Simplex Virus (Cooper, Dultsev et al. 2001), and presented as a generic technique termed rupture event scanning (REVS™) in Coopers later paper (Cooper 2003). This technique was also investigated by Edvardsson but with the opposite approach, which is to limit binding sites by oscillating at higher amplitudes rather than rupturing them. Edvardsson has published a series of three papers from 2005 to 2007 (Edvardsson, Rodahl et al. 2005; Edvardsson, Rodahl et al. 2006; Edvardsson, Zhdanov et al. 2007). There has also been further research by Heitmann (Heitmann and Wegener 2007) monitoring cell adhesion (via QCM impedance) during high driving amplitudes. Heitmann also monitors the impedance of the QCM while running at high amplitudes and found that a rupture occurred – though the results used did not have the precision needed to make a quantitative conclusion. The Bond rupture technique will be further analyzed and reviewed in the "Bond Rupture" section on page 29.

Some further papers related to the QCM background reading can be found in (Ward and Buttry 1990; Barnes, Dsilva et al. 1992; Janshoff, Galla et al. 2000; Janshoff and Steinem 2001; Lee, Kim et al. 2004).

2.2.1.2 SURFACE ACOUSTIC WAVE (SAW)

SAW devices are another example of a mass sensitive device. These devices are an input and output device and much can be gathered about what is bound on the surface between the input and output by the response of the SAW device. There are many different methods of analysis, for determining what is bound on the surface of a SAW device by monitoring properties such as: resonant frequency changes; signal delay; phase change; and returned amplitude of the output after exciting the input. SAW devices tend to work at high frequencies (30 to 1000 MHz) which has the advantage of greater frequency change which in turn enables better mass change monitoring, but it also adds to the complexity of the electronics in comparison to traditional QCM methods. SAW devices also usually have much lower quality factor than a QCM, but with increased precision in the semiconductor industry and advanced SAW modelling this could be improved. A SAW device input frequency response is dependent on the input fingers shape, number of fingers, width and height. A big advantage of SAW devices it that the output frequency response can be different than the input by using different finger structure on the output enabling SAW devices to be used as filters. There is also the option with SAW devices to develop them with multiple output finger sets enabling specific signals to be targeted, which could be advantages with the Bond Rupture 'Noise' method.

Although SAW devices are common in the telecommunications industry, there has not been the same amount of research into them for immunosensor devices. This is due to most devices being custom made and there does not seem to be any standard SAW device immunosensor. So while SAW devices are very configurable this also seems to be their short coming as there are too many options. These devices would be ideal if a large amount of time and research could be set aside to create the perfect sensor.

2.2.1.3 MICRO-ELECTRO MECHANICAL SYSTEMS (MEMS)

There has been a large amount of research lately in micro-electro mechanical systems (MEMS) with their most well known applications being on chip accelerometers and gyroscopes which have redefined the way user interfaces can be developed. MEMS also have huge potential applications for lab-on-a-chip type biosensors as they can be developed in bulk relatively cheaply and incorporate all the electronics needed on a single chip. The development is limited by the initial research cost due to the expensive equipment required. MEMs devices are made in a similar way to modern semiconductor devices but use piezoelectric materials which can then be etched into mechanical devices to make anything from gears (see Figure 2.3) to on chip mechanical oscillators. Other devices such as resonant cantilevers and membranes have been created by Baborowski (Baborowski 2004). Devices such as this can be used as biosensors with the small mass changing significantly when particles attach, making them much more accurate measurements than traditional methods – though increasing effects of thermal noise and other fundamental limits are present.

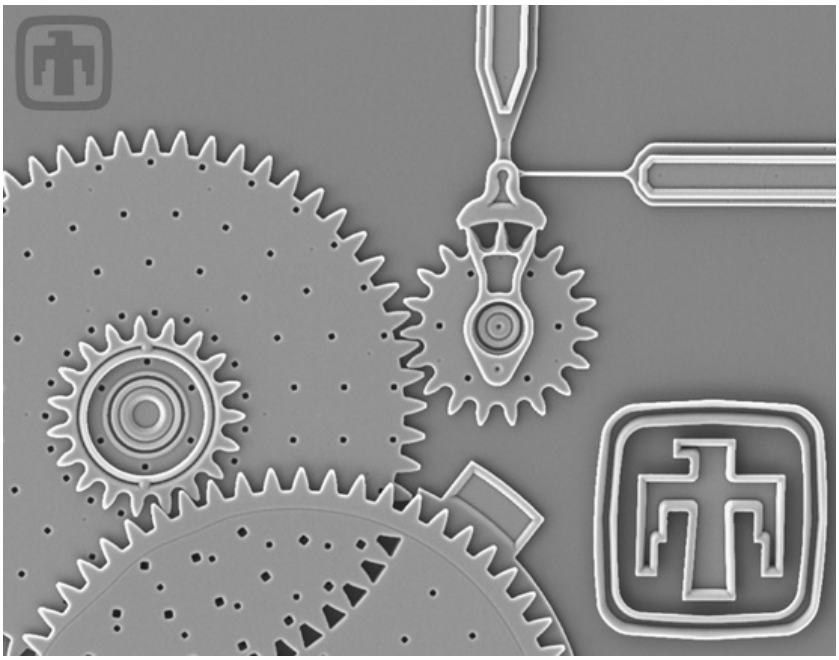


Figure 2.3: MEMS multiple gear reduction unit "Courtesy of Sandia National Laboratories, SUMMiTTM Technologies, www.mems.sandia.gov".

Decreasing the cost of the sensor is an important factor and is even more so when the sensor is not indefinitely reusable. Cost can be reduced by using mass fabrication and material minimisation techniques such as piezoelectric semiconductor fabrication like MEMS devices, Quartz etching and fabrication, and piezoelectric composites.

The first technique that will be discussed is the MEMS fabrication to make QCM like sensors. Devices such as the piezoelectric nano-mechanical micro cantilever developed by Lee et al (Lee, Hwang et al. 2005) have been used in the detection of prostate-specific antigen and could therefore be used in the detection of most antigen binding events although this would depend on the mass sensitivity of this sensor.

Micro cantilever sensors can work in a vacuum, gas, or liquid and have many advantages over normal sensors such as small size, high precision, and increased reliability (Moulin, O'Shea et al. 2000; Lavrik, Sepaniak et al. 2004). The main problem with them is that they may not be suitable for Bond Rupture due to the large change in sensitivity from the base to the tip and also the limited amount of force that could be applied to the particle.

There has been a large amount of development in MEMS over the last few years and commercially available devices include accelerometers, gyros and piezoelectric micro-mirrors. There is also a trend for a lab on a chip device to be made with DNA sensor chips, on chip mixers, and other chip related fabricated devices.

Another method is to etch many sensors on the same piece of Quartz crystal allowing multiple sensors on the same chip, also allowing the size and shape to be optimized for the application specified. An example of this is performed by Li et al (Li, Abe et al. 2003) with a created 1mm diameter miniaturized QCM using an etching technique that decreases the frequency noise and frequency drift in comparison to standard QCM's. Rabe et al (Rabe, Buttgenbach et al.

2000) also found that using etching methods they could make a miniaturized QCM of a frequencies up to 75Mhz which is much higher than the normal 30MHz QCM's that can be made with standard processes. It was found that the sensitivity was greatly increased with the higher frequencies and so higher precision experiments for mass addition could be investigated. Rabe et al (Rabe, Seidemann et al. 2003) also created an array of fabricated QCM's and interfaced them to the analysis circuitry using coils around each QCM allowing wireless analysis and transfer of power. In regard to Bond Rupture, experiments would have to be carried out to determine the maximum shear mode movement (to induce Bond Rupture), but the movement needed would be also reduced by using these higher frequency crystals.

2.2.1.4 SURFACE PLASMON RESONANCE (SPR)

SPR is a quantum optical-electrical phenomenon that arises from the interaction of light with a suitable metallic surface (usually silver or gold) causing the absorption of light at certain angles - see Figure 2.4. This is due to the fact that under certain conditions light energy couples with electrons on the metal surface, which sets up resonant oscillations within the cloud of electrons at the metal surface. The angle at which the metal surface absorbs the light is a function of the wave length of light, the thickness and dielectric constant of the metal surface, and the refractive indices of the media above and below the metal surface. Therefore if all the other parameters are kept constant, the change in refractive constant on the surface of the metal layer is relative to the amount of bonding at the surface (Newton 1997; Wild 2001). Newton (Newton 1997) found experimentally that the reactive constant of buffer solutions is about 1.33 whereas a protein attached to the same surface causes the refractive index to be about 1.46 so from this difference proteins can be detected binding to the surface. SPR is an important biosensor method that is significantly different than the piezoelectric mass sensing methods.

SPR as a biosensor has been demonstrated in the use of detecting antigen attachment (Oh, Kim et al. 2004). SPR angle was measured when the bacteria Salmonella typhimurium was bonded to the antibody. This was found to be four times more accurate than traditional techniques such as QCM. Although this method has good results, this method still suffers from the problem of non-specific particle absorption.

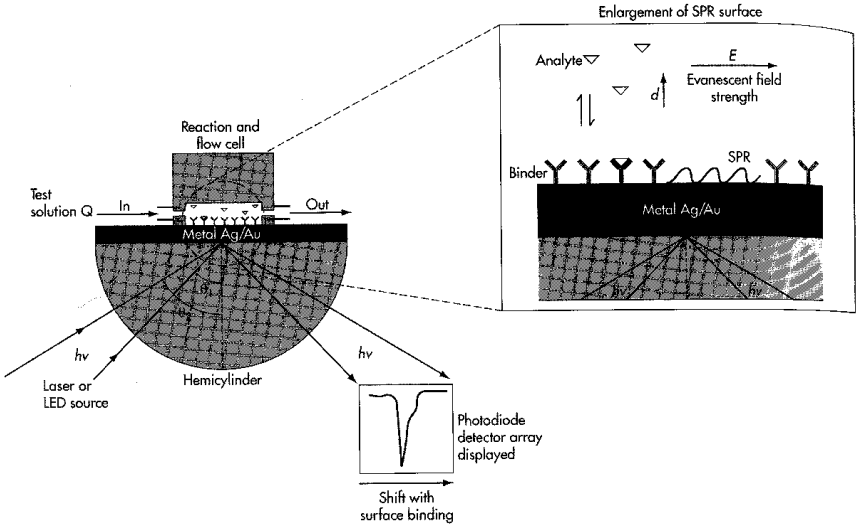


Figure 2.4 Surface Plasmon resonance biosensor based on the Kretschmann arrangement (Wild 2001).

The SPR method could be used to verify the antibody-antigen bond forming and possibly verify Bond Rupture but would not be able to actively force the Bond Rupture, as it does not introduce significant energy into the system and so is effectively a passive sensor. The main uses of SPR would be for the verification of bonding or in conjunction with a QCM as a transducer for verification or higher accuracy sensor without the problems such as drift that QCM's suffer from.

2.2.1.5 OTHER BIOSENSORS

The Atomic Force Microscope (AFM) is a laser based micro-cantilever system that detects the angle of the cantilever via reflecting a laser off the cantilever and on to a photo diode system enabling the measurement of the deflection of the cantilever. AFM also uses piezoelectric transducers to move the sample of interest around it and with this combination enabling 3D surface images to be acquired down nano-meter ranges. AFM can also be used to measure very small forces, and has been used to measure the forces for a range of different antibody-antigen bonds. For more information refer to the "Bond Strengths and Loading Rate" section. A technology which is similar to AFM called Scanning Tunnelling Microscopy (STM) which uses the same piezoelectric base as AFM but uses the tunnelling current to measure the distance from the surface instead of physical movement of the cantilever. STM was used by Borovsky (Borovsky, Mason et al. 2000) to measure the amplitude of movement of the surface of the QCM with change in voltage. It may be possible to perform a similar measurement with AFM for the purposes of detecting the movement of the surface of a Bond Rupture sensor.

Another application similar to AFM is using the resonant frequency of a micro cantilever to detect mass addition. Affordable micro-cantilevers can now be purchased off the shelf with the advent of AFM. These usually use laser technology to measure the offset of the cantilever, but can also incorporate piezoelectric layers on the surface for measuring or excitation. These cantilevers have been investigated for use as a biosensor as the resonant frequency is very dependent on mass of the cantilever. Lavrik (Lavrik, Sepaniak et al. 2004) has performed an excellent review for cantilever based biosensors. Lavrik (Lavrik and Datskos 2003) has also

published a cantilever detection system that could detect down mass change down to femto-grams using a laser excited cantilever such as in Figure 2.5. Cantilevers have also been made out of piezoelectric materials enabling them to be used in the same way as a QCM. It would be difficult to induce Bond Rupture with this method due to the small scale of this experiment and the amount of variation to the forces applied to the bonds over the surface of the cantilever.

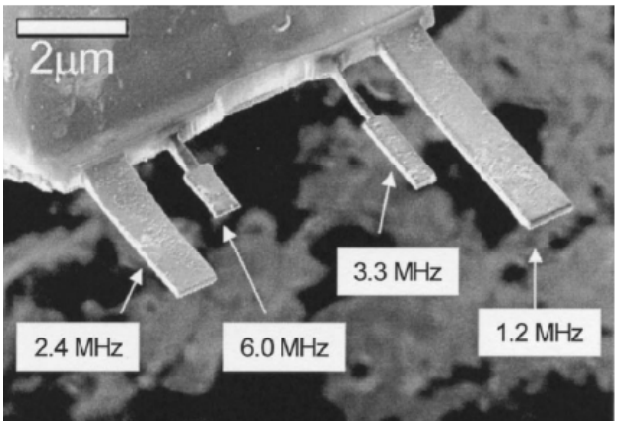


Figure 2.5: Silicon based micro-cantilevers with approximate resonant frequency (Lavrik and Datskos 2003).

2.2.2 IMMUNOLOGY BASICS

Though this project is based on a bio-sensor this author must admit his ignorance when it comes to all things biological as his background is essentially mechanical and electrical. Also as this thesis is primarily focused on the sensor and measurement system only some basic immunological background is needed. This section covers the antibody and antigen interaction, including the bond strengths, and a basic explanation of the Biotin-Streptavidin interaction – used to simulate the antibody-antigen interaction. For a more in depth look at immunology see a book such as Wild (Wild 2001).

2.2.2.1 ANTIBODIES AND ANTIGEN

Antibodies are basically proteins which form strong bonds to the antigen. There are over a million different bonds that antigens can form. Antibodies come in five main different forms including IgG (see Figure 2.6), IgA, IgM, IgD, IgE. An antigen is a substance foreign to the body which stimulates the production of antibodies by the immune system. Antigens include foreign proteins, toxins, bacteria or viruses. The antibody – antigen bond is one of the strongest non-covalent bonds that can be found in nature. This bond consists of a mixture of hydrogen bonds, electrostatic interactions, and van der Waals interactions. The bond point takes place at the antigen-binding fragment of the antibody.

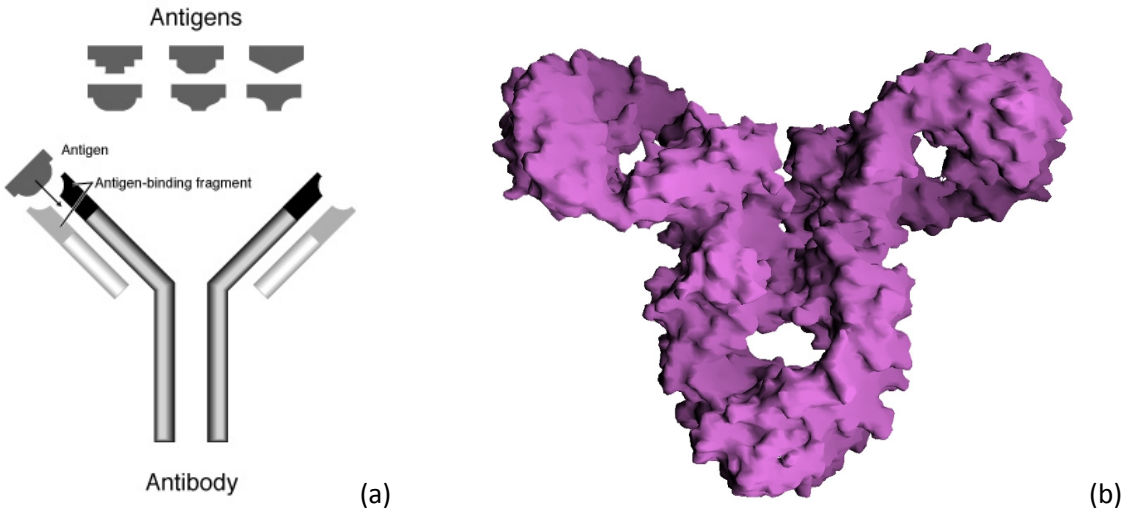


Figure 2.6: (a) Antibody binding with antigens (b) Molecular surface of an antibody Immunoglobulin (IgG) (Nicholls, Sharp et al. 1991).

The Bond Rupture method ruptures the bond between the antibody which would be attached to the surface of a QCM and the antigen (which would be introduced via some form of solution). It is found that binding can occur between antibodies and non specific antigens but with less binding strength. By shaking the antibody and antigen a force is introduced between them causing the bond to break when the shaking force is greater than the binding force. So the non specific bonds should break at, a low shaking force whereas the specific bonds should break at a higher force. This should enable more precise measurements with less false positives.

2.2.2.2 KINETICS OF ANTIBODY-ANTIGEN INTERACTIONS

The reaction rate of the antibody-antigen bonds can be seen in the equilibrium association/dissociation rate in Equation (2.1). Antibody and antigen reactions in liquid take from seconds to hours to reach equilibrium and their reaction rate depends on a number of different factors. Among them are pH, ionic strength, temperature, and average distance between antibody and antigen.

$$[A_g] + [A_b] \overset{K_a K_d}{\Longleftrightarrow} [A_g - A_b] \tag{2.1}$$

Where [Ag] is the antigen concentration, [Ab] the antibody concentration, [Ag-Ab] the antigen-antibody complex, K_a the association rate constant, and K_d the dissociation rate constant (Wild 2001).

The average distance between antibody and antigen is one of the important factors that can affect the design of the detection method. One way to increase the reacting speed would be to perform an in liquid diagnostic as this reduces the average distance between reagents, assuming there is adequate mixing. Reaction speed must be considered when designing a biosensor. Unfortunately, with the QCM we are limited to a surface that is covered in antibodies and that reacts with a solution of antigen, the average distance between the reagents is high and so largely decided by the liquid height, and concentration. There are other

methods to increase reaction speed including using higher flow rate, advanced mixing techniques (Campbell and Grzybowski 2004) increased concentration, optimum pH, and optimum temperature.

2.2.2.3 BOND STRENGTHS

The force needed to induce the Bond Rupture of the antibody-antigen bond is an interesting topic with much research having been performed. Most the experiments have been performed using an AFM to measure the force on single bonds required to break the bond. Included in this research has been not only the bond strength, but also the effect of the speed at which the bonds are attempted to be broken at. AFM has been used for a long time to image surfaces precisely; it has also been used to measure the bond association and dissociation strength of antibody and antigen. Bond experiments are quite hard to perform due to the small scale and magnitude of the forces involved. Though AFM can accurately generate and measure Bond Rupture it has as yet only been implemented on single bond experiments and due to the complexity of the experiment and equipment it is not practical for large scale experiments. This data however is useful to verify the forces needed for QCM Bond Rupture and provides an order of magnitude measurement of the force required.

As discussed the Immunoassay Basis section, the antibody - antigen reaction has a dissociation speed which is dictated by an array of factors including pH, temperature, and concentration of the antigen. The antibody – antigen bond is a non-covalent bond which includes hydrogen bonds, electrostatic bonds, Van der Waals forces and hydrophobic bonds, this means that it is not a permanent bond so it can be broken if enough force is applied. It was also found by Merkel (Merkel, Nassoy et al. 1999) that the rupture force required to break the bond is also dependant on the rate that the force is applied to the bond as can be seen in Figure 2.7. The bond strength for Biotin-Streptavidin was found to be between 5 pN to 170 pN depending on how fast the bonds are loaded. Avci (Avci, Schweitzer et al. 2004) analysed static forces of the bond strength using AFM to analyse the antibody-antigen interactions and found the bond strength to be 90 ± 40 pN. Weisel et al (2003) found after reviewing many published results using AFM the forces needed to rupture the antigen-antibody bond are around 100pN +/- 50pN. Although these measurements are at a force loading rate much lower than what would be applied using the QCM making a direct comparison difficult to perform, these values should still provide an approximate order of magnitude force value required to perform bond rupture.

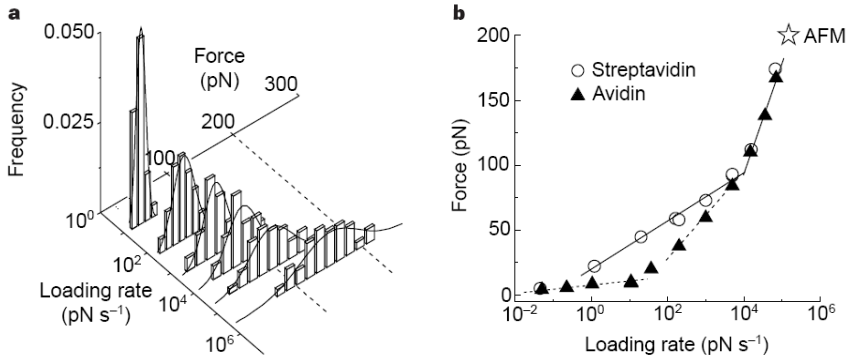


Figure 2.7: Force histograms from a. Force histograms from tests of single Biotin-Streptavidin bonds b. Dynamic strength spectra for Biotin-Streptavidin (circles) and Biotin-Avidin (triangles) bonds. (Merkel, Nassoy et al. 1999)

For further study into papers of interest related to bond force experiments performed with AFM Weisel (Weisel, Shuman et al. 2003) has correlated a collection of forces of adhesion and dissociation from a variety of sources. Other sources of further interest are as follows (Plant, Gray et al. 1993; Merkel, Nassoy et al. 1999; Lo, Zhu et al. 2001; Kulin, Kishore et al. 2002; Weisel, Shuman et al. 2003; Kaur, Singh et al. 2004).

2.2.2.4 BIOTIN-STREPTAVIDIN EXPLANATION

The Biotin-Streptavidin is a widely used bond in molecular, immunological, and cellular assays to simulate antibody-antigen bonds. Much research has been performed around this bond strength with it having one of the highest binding strengths for a non-covalent bond known. There is a multitude of commercial products that produce this in forms such as Streptavidin beads and pre-coated QCM's with Biotin coating. The Biotin-Streptavidin bond was used to simulate Bond Rupture by Dultsev (Dultsev, Ostanin et al. 2000). This bond when used in conjunction with latex beads enables the Bond Rupture mechanism that will be mentioned in later sections to be properly simulated without needing real antibodies and antigens – which could be potentially dangerous. The Biotin-Streptavidin bond is similar in strength to other antibody-antigen interactions so if the rupture of Biotin-Streptavidin can occur it can be assumed that the antibody-antigen bond will also rupture at around the same strength.

An example of how the Biotin-Streptavidin bond is used is shown from Dultsev's Bond Rupture experiment protocol. Firstly Streptavidin was covalently bonded to 5 micro-meter latex beads. Secondly the QCM surface was coated with Biotin via covalent bonding to the surface. Then the beads are introduced on to the surface in a liquid solution and left to bond. The Bond Rupture experiment was then carried out, with the Biotin-Streptavidin bond breaking simulating an antibody-antigen interaction. A bacterium is typically 0.5 to 5 micro-meters in length so the 5 micro-meter beads are a good test measure for large bacteria and to test the Bond Rupture method for other sizes a range of beads must be used.

2.3 QUARTZ CRYSTAL MICROBALANCE (QCM)

Piezoelectricity is the effect that certain crystals exhibit when the crystalline material is stressed as this causes a voltage to be generated, and vice versa when a voltage is applied the material moves. This effect was discovered by the Curie brothers in 1880.

There is a great range of piezoelectric resonators and sensors for many different applications. Some examples are ultrasound range and imaging, spark generators for lighters, oscillators for clock generation, micro-fabricated actuators and sensors, and sensitive mass sensors. A commonly known piezoelectric resonator is the AT cut timing crystal used in a variety of applications from the timing of most digital electronics to the QCM sensor which is used for the detection of nano size mass accumulation. AT cut crystals are used because of the high temperature stability they exhibit. AT cut crystals are also called thickness shear mode (TSM) resonators as they move horizontal to their surface with very little vertical movement. This is the primary crystal type used for Quartz Crystal Microbalance (QCM) sensors as their frequency change exhibited is mass dependant due to their horizontal (shear) operating mode.

The QCM can be used to not only monitor mass but can also be used to measure other environment variables such as humidity, temperature, flow rate, pressure, and density. Though this may seem like an advantage to be useful for many applications, it also means that the other variables need to be isolated from what is being measured. This can cause problems and inaccuracies, due to variations in these other variables while monitoring the mass.

This section describes the QCM in its use as a mass sensor and the related basic modelling as well as investigating the different measurement methods for determining the QCM's frequency. The measurement limitations shall also be examined.

2.3.1 BACKGROUND

The subject of QCM and other piezoelectric devices has been around for a long time. Because of this there have been many different reviews and summaries available, e.g., (Buttry and Ward 1992; Janshoff, Galla et al. 2000; Janshoff and Steinem 2001; Muramatsu, Kim et al. 2002; Buck, Lindner et al. 2004; Mecea 2005)

The basis of the QCM is characterized by Quartz crystal, a piezoelectric material that produces a voltage, when a force is applied to it. Resonators can be made out of piezoelectric material by taking into account that the Quartz crystal is a mechanical system similar to a bell, in that it resonates at a mechanically defined frequency. This resonant frequency of the QCM is very precise and when used in conjunction with an oscillating circuit, it can be used in a variety of applications which require an accurate time base, such as watches, or microprocessors. Using essentially a Quartz crystal oscillator, Sauerbrey (Sauerbrey 1959) demonstrated that the resonant frequency change was proportional to the mass change of the Quartz crystal. This was found to be accurate enough to measure in the order of magnitude of nano-gram changes of mass, or in some cases measure mass changes as small as a fraction of a monolayer or a single layer of atoms.

Sauerbrey (Sauerbrey 1959) demonstrated that the resonant frequency change is proportional to the deposited mass on the Quartz crystal and hence the Quartz Crystal Microbalance (QCM) was born. Sauerbrey's formula was proposed to stating that the frequency change of the QCM is proportional to the mass accumulation on the surface of the QCM - which is shown in Equation (2.2).

$$\Delta f = -\frac{2f_0^2}{A\sqrt{\bar{c}_{66}\rho_q}} \propto -S_f \Delta m \tag{2.2}$$

The variables in the formula are: f_0 is the harmonic frequency, A is the area over which the mass is spread, and $\bar{c}_{66}$ is the material properties of the crystal matrix and ρ_q is the density of the crystal. It was found that by substituting into the equation the constant S_f it can be found that the mass is proportional to the frequency change. Sauerbrey's formula (though not a perfect representation) does demonstrate the response of the QCM for small area constant mass additions. The Sauerbrey formula holds for predicting the resonant frequency for the QCM working in a vacuum or a gas. But when working in liquids a frequency shift associated with the liquid viscosity and density is observed. Kanazawa and Gordon (1985) proposed the Equation (2.3) for frequency change caused by liquid.

$$\Delta f = -\frac{f_0^{\frac{3}{2}}}{\sqrt{\pi\bar{c}_{66}\rho_q}}\sqrt{\rho_{liq}\eta_{liq}} \tag{2.3}$$

Where ρ_{liq} is the density of the liquid and η_{liq} the viscosity of the liquid. This formula is much better at predicting the frequency change in liquid but it still has a significant amount of variation from the real/measured frequency change.

2.3.2 MODELLING

To determine the response of the a resonant crystal, it can be modelled as a simple mechanical system, as shown in Figure 2.8, in which r is the resistance, C_m is energy storage and M the system mass. This mechanical system is a resonant system. If the system is given a push the system resonates backwards and forwards changing the force F, in a decaying sinusoidal fashion. This is a resonant system and if any factor in this system changes (such as mass) the resonant frequency will also change.

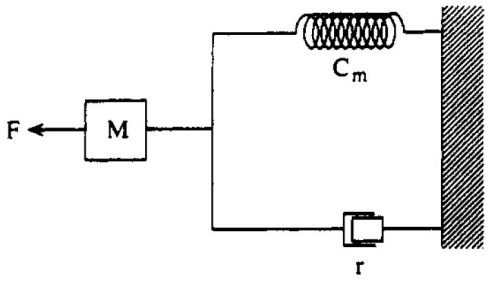


Figure 2.8 Mechanical model of a QCM (Buttry and Ward *1992*)

The differential equation of this mechanical system is described by Equation (2.4).

$$F = m.\frac{dx^2}{dt^2} + r.\frac{dx}{dt} + k.x \quad (2.4)$$

The variables in this formula are: m is the mass the r the damping constant or resistance and k the spring constant or $k = \frac{1}{C_m}$. This can be compared to an electronic resonator or RLC circuit which is described by Equation (2.5).

$$V = L\frac{dq^2}{dt^2} + R\frac{dq}{dt} + \frac{1}{Cq} \quad (2.5)$$

This equation contains L being inductance, R or the resistance and C which is the capacitance. As can be seen in the above formula this can also be represented by simple electronic components and such components can be used to make resonators – though they are usually very inaccurate due to component error and sensitivity to temperature. To enable this formula to be solved a Laplace transform was applied to this creating Equation (2.6).

$$Z_1 = sL_1 + R_1 + \frac{1}{sC_1} \quad (2.6)$$

The RLC circuit specified is missing a vital component which as QCM has that is a parallel capacitor – the QCM equivalent circuit is seen in Figure 2.9. The parallel capacitor is represented as Equation (2.7).

$$Z_2 = \frac{1}{sC_0} \quad (2.7)$$

To combine the two equations they must be taken as two parallel components as in the Equation (2.8).

$$Z = \frac{Z_1 Z_2}{Z_1 + Z_2} \quad (2.8)$$

Therefore, the crystal can be represented by a resistor (power lost), capacitor (elasticity of the material), and inductor (mass of resonator) in series. The parallel capacitor is formed because of the capacitance between the crystals top gold electrode and the bottom gold electrode with the crystal forming a dielectric in the centre. This model is called a Butterworth van dyke (BVD) which can be seen in Figure 2.9.

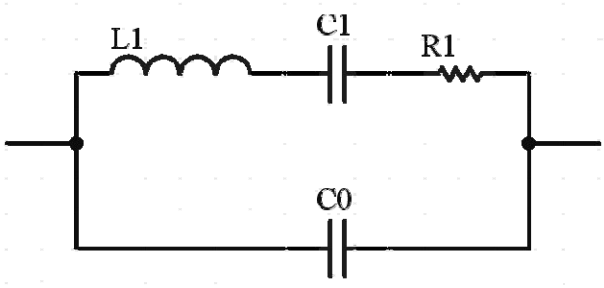


Figure 2.9: Butterworth van dyke (BVD) electrical equivalent of a QCM.

If equation 5 is simulated in Matlab, and the electronic components are given realistic values the system response can be viewed and using different electronic values the response can be changed. Figure 2.10 is a Bode diagram showing the simulated difference between operating

in air and a dampened environment such as water. The series resonant frequency is where the phase crosses zero first and the parallel frequency is where the phase crosses zero the second time.

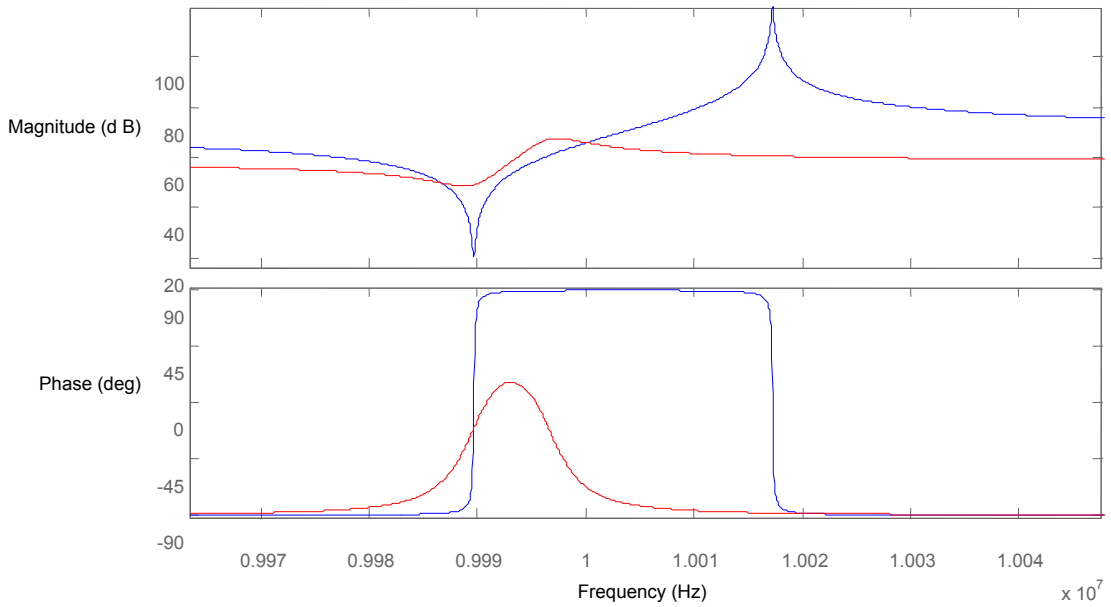


Figure 2.10: Bode Diagram showing the Magnitude and phase of a QCM equivalent circuit (blue) with low damping (red) high damping.

Experiments were performed by Ferrari (Ferrari, Marioli et al. 2001) looking at the admittance (inverse of the Figure 2.10) with air, and a number of different liquids of increasing viscosity. This is shown in Figure 2.11.

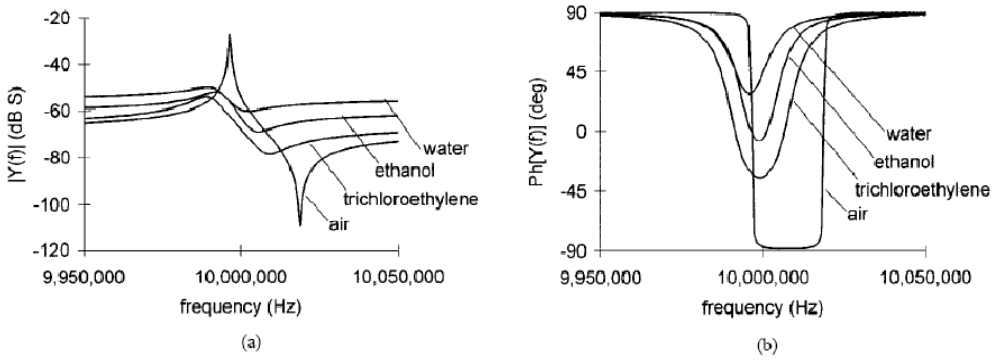


Figure 2.11: (a) Admittance and (b) phase of a 10MHz Quartz crystal in air and different liquids (Ferrari, Marioli et al. 2001).

The parameters for a Butterworth van dyke (BVD) equivalent circuit compared to the values of a AT-cut Quartz and the series and parallel resonant frequency can be seen in Table 2.2 which is from Janshoff's review paper (Janshoff, Galla et al. 2000). The variables from Janshoff's table are as follows: A is the electrode area; η_q is the viscosity; d_q is the thickness of the Quartz; ε_{22} the dielectric constant of the Quartz material; and e_{26} the piezoelectric constant dependent on the cutting angle. The equations that are of most interest are the series resonant frequency (f_s)

and the parallel resonant frequency (f_p). The series resonant frequency is the initial zero phase crossing and the parallel resonant frequency is where the phase crosses zero again. The parallel frequency value is in the order of magnitude of 20 kHz above the series frequency. The series resonant frequency is the frequency of most interest to us as this is the point of least resistance, so where the most movement of the QCM surface occurs. If the BVD values are calculated the resonant frequency can be also calculated. This is of some interest for simulating the QCM frequency change response.

Parameter	Expression
C_0	$\frac{\varepsilon_{22}A}{d_q}$
C_q	$\frac{8Ae_{26}^2}{\pi^2 d_q \bar{c}_{66}}$
L_q	$\frac{d_q^3 \rho_q}{8Ae_{26}^2}$
R_q	$\frac{d_q \eta_q \pi^2}{8Ae_{26}^2}$
f_s	$\frac{1}{2\pi}\sqrt{\frac{1}{L_q C_q}}\left(1+\frac{C_0 R_q^2}{2L_q}\right)$
f_p	$\frac{1}{2\pi}\sqrt{\frac{1}{L_q C_q}}\left(1+\frac{C_q}{2C_0}-\frac{C_0 R_q^2}{2L_q}\right)$
f_{Zmin}	$\frac{1}{2\pi}\sqrt{\frac{1}{L_q C_q}}\left(1-\frac{C_0 R_q^2}{2L_q}\right)$
f_{Zmax}	$\frac{1}{2\pi}\sqrt{\frac{1}{L_q C_q}}\left(1+\frac{C_q}{2C_0}+\frac{C_0 R_q^2}{2L_q}\right)$

Table 2.2: Parameters of the Butterworth-van dyke equivalent circuit for AT cut Quartz (Janshoff, Galla et al. 2000).

Much work has been done in using electronic components to model a QCM and this work has allowed for an increase in the understanding of QCM operation under different environments. Finite element models can approximate the real operation of the QCM. These are usually very complex and require much work to set up and require a long time to simulate but a significant amount of work in this area has been carried out including papers such as (Wu, Tsai et al. 2003; Lu, Lee et al. 2005). Though simulation provides an excellent place to start from it is of limited use when comparing real world examples due to the variation between the simulation and experimental results.

2.4 MEASUREMENT TYPES

Many different methods can be used to analyse QCM frequency change. They include oscillator circuits, impulse excitation, and impedance or phase analysis. These methods have their advantages and disadvantages for straight frequency analysis. A comparison between the

main methods is shown in Table 2.3. A review on the different detection methods has been carried out by Eichelbaum (Eichelbaum, Borngraber et al. 1999) - with the focus being on oscillation based methods. Impulse excitation has been mentioned by a few papers and is very common in the signal processing field as a method to define a system. But due to the high frequency response of the QCM sensor it is difficult to generate a small enough impulse with enough power to detect the frequency response making this a difficult method to implement. Network analysis is an interesting method which is usually quite expensive to implement but with the advent of higher speed and cheaper analogue to digital converters this method is becoming more practical. The Oscillator method is the traditional method for frequency detection as it is cheap and easy to implement for simple designs, but to get high stability oscillator circuits that work well in highly damped environments can be troublesome. Another method is the dissipation method. This method is about as complex as the network analysis circuitry but is very accurate. The dissipation method is similar to impulse excitation in the way that it excites the crystal and then looks at its unimpeded frequency response to obtain the frequency and also enabling the quality factor of the QCM to be measured.

Measurement Type	Advantages	Disadvantages
Oscillator	Inexpensive Simple electronics	Does not operate in dampened environments High stability circuit needed
Network analysis	System effects can be calibrated out quality can be measured via bandwidth	Hard to implement Expensive electronics
Impulse excitation	Resonance frequency accurate	Difficult to generate Expensive electronics
Dissipation	Resonance frequency accurate Dissipation and therefore quality is measured	Expensive electronics

Table 2.3: Advantages and disadvantages of different electronic frequency detection methods.

All these methods shall be examined in more detail in the sections to come.

2.4.1.1 OSCILLATOR CIRCUITS

To date oscillator circuits are the most widely used method for QCM frequency measurement, with many different types of oscillators having been created. Consider a basic oscillator that is used to generate a frequency for input to a micro controller which is basically an inverter connected from one side of the crystal to the other (see Figure 2.12). This is a very simple circuit and works for any high quality crystal unfortunately this circuit will not operate well under highly damped environments and stops operating where the phase of the crystal fails to cross zero – though with advanced analogue circuitry the response in dampened environments can be improved. Another problem with this type of oscillator is that when testing attached sensitive structures like Self Assembling Monolayer (SAM) detachment could occur if the voltage present is too large on the QCM side with the SAM attached. To prevent interaction from the oscillating voltage one side of the crystal would be required to be grounded which is not possible with this type of oscillator.

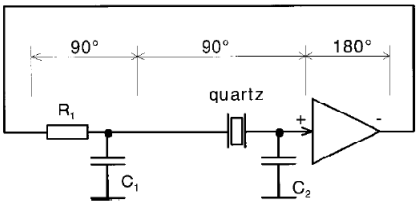


Figure 2.12: Oscillator circuit with inverting amplifier (Eichelbaum, Borngraber et al. 1999).

The liquid operation problems have created a need for new types of oscillators. There are now many different types of oscillators for use in liquids with some even providing magnitude for the amount of dampening in the system (giving valuable information about the liquid) as well as the standard frequency output. An example of a generic oscillator for operation in liquids is shown in Figure 2.13. This may look simple in the diagram but many components are required to create this, and the frequency can be changed quite easily by the component non-linear characteristics and other tolerance issues.

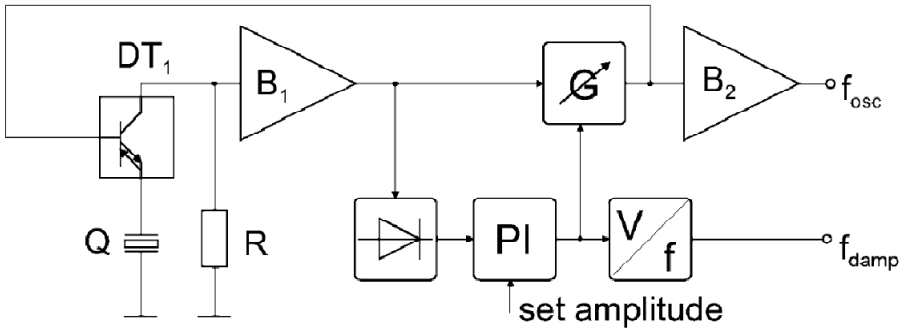


Figure 2.13: Basic structure of a gain controlled oscillator (Schröder, Borngraber et al. 2002) - for operation in liquid.

Eichelbaum (Eichelbaum, Borngraber et al. 1999) looks at the possible circuits for QCM in-liquid oscillators and finds that while the frequency can be measured well in liquid using oscillating circuits for more precision measurements the use of network analyser based methods is a better option.

For further oscillation based circuits refer to (Auge, Hauptmann et al. 1995; Eichelbaum, Borngraber et al. 1999; Ferrari, Marioli et al. 2001; Wessendorf 2001; Ehahoun, Gabrielli et al. 2002; Jakoby, Art et al. 2005).

2.4.1.2 NETWORK ANALYSIS

Network analysers are used to analyse Radio Frequency (RF) electrical circuits. They are used to analyse devices from antennas, to power grids, or in this case biosensors. The analysis that network analysers perform can be compared to how light reacts to a semi-transparent object. If wide spectrum light is transmitted into lens some wavelengths of light are reflected back, various wavelengths are absorbed in the lens, and an assortment of wavelengths are transmitted through the lens (top of Figure 2.14). The same concept is true with a Device Under Test (DUT) examined with a network analyser. When power is transmitted to a DUT

some of the power (depending on the frequency) is reflected back. An amount of power (also depending on the frequency) is absorbed in the DUT, and some makes it though the other side (bottom of Figure 2.14).

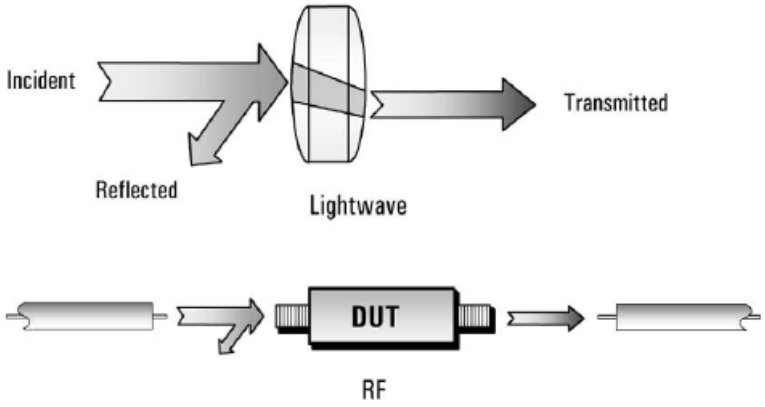


Figure 2.14: Network analyser analogy (Agilent 2004).

There are two types of network analysers. The first is Scalar network analyser (SNA) which measures only the amplitude of the output of the DUT. The second type is a Vector network analyser (VNA) which can measure the amplitude and phase signal reflected from the input of the DUT and transmitted from the output of the DUT. A SNA is the same as a Spectrum Analyser with a sweep spectrum signal generator. In analysing signals a VNA is the most commonly used equipment and is that will be referred to in chapter 3 when preliminary QCM tests are carried out.

A VNA can measure many variables that are useful when testing a biosensor. The most common variable to measure is magnitude, which provides information as to the gain or loss of signal verses frequency. The magnitude is defined by the transmitted voltage divided by the source voltage (incident in Figure 2.14). The magnitude is usually plotted as a log plot using equation (2.9). Phase can also be calculated by subtracting the source phase from the transmitted phase. The reflection data can be used to calculate other characteristics such as: return loss, impedance, and Voltage Standing Wave Ratio (VSWR). Another plot that can be generated is a Smith chart, which is a representation of the complex impedance. This is useful for impedance matching and examining the impedance response of circuits - Bowick's (Bowick 1997) book is an excellent reference for impedance matching with Smith charts. The most useful plots for QCM based biosensors would be the magnitude, phase, and impedance as from these variables the frequency of the QCM can be determined as well as the resistance (for impedance matching) and quality of the QCM. A VNA's frequency sweep time can range over almost any time period required, but most VNA's sweep time start at around 100ms and increase to around 60s.

$$Gain_{dB}(f) = 20log\left(\frac{V_t(f)}{V_s(f)}\right) \quad (2.9)$$

Schröder et al (Schröder, Borngraber et al. 2002) created a network analyser using a quadrature (in-phase and out of phase signal components) detection method that gives phase and impedance results. This was created using custom components for the specific application

of QCM sensing – see Figure 2.15. This has the advantages of a traditional network analyser system without the high costs or equipment size issues. This system is limited by how fast the impedance can be measured and the accuracy of the analogue components. Schröder uses a FPGA (Field-programmable gate array) to control the DDS (Direct Digital Synthesis - which is a frequency generator) chip as well as the phase detector and ADC grabbing. This data is then sent back to a computer to be recorded via an Ethernet or RS232 connection. A FPGA method enables quick development of a digital based system removing the need for high speed micro-processors and external control logic. It also enables quick development and the option of reconfiguring the logic to fix mistakes or to make improvements to the system, without requiring a board redesign. FPGA's are basically a big grid of reconfigurable digital logic blocks that can be configured in a number of different combinations to make anything that you would normally create with digital logic from an adder, to control logic, to a micro-processor, and this all within the same device. This is limited only by the number of logic blocks available within the FPGA. With FPGA devices becoming larger with every passing year the amount of signal processing that can be performed within them is now considerable, opening up significant improvements for Schröder's QCM network analyser.

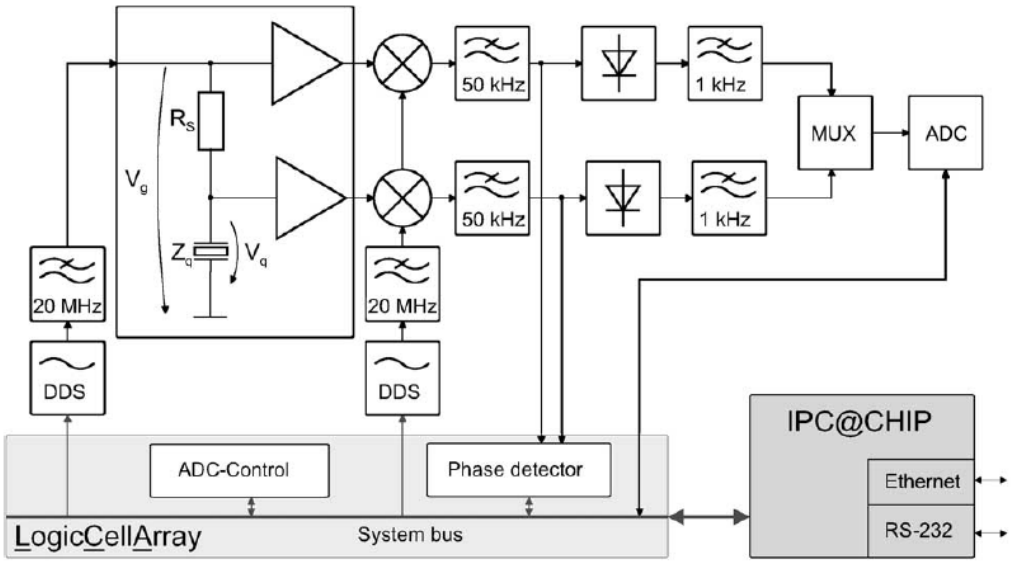


Figure 2.15: Principle of network analyser as a sensor interface (Schröder, Borngraber et al. 2002)

Network analysis is an excellent sensing method for the measurement of a QCM biosensor with a significant number of advantages over traditional oscillator methods. There is however some major disadvantages such as the large cost with a new network analyser costing upwards of NZD 30,000, and the limited amount of flexibility with these systems for experiments such as Bond Rupture. So though this is a good method for initial verification of QCM characteristics, this is not a good method for generic QCM sensing. There are however many network analyser methods that could be incorporated into a customised electronic QCM measuring device similar to what Schröder (Schröder, Borngraber et al. 2002) has achieved.

2.4.1.3 IMPULSE EXCITATION

Impulse excitation is a method frequently used in signal processing to define a black box system. This method could be described as a hammer hitting a bell. The hitting of the bell (input to the system) for a very short period of time excites all the harmonics of the bell producing a ringing sound (output of the system). The ringing sound gives information about the system under examination. Practically this involves generating a high amplitude, narrow as possible pulse (a lot narrower than the frequency of interest). In our case for a 10 MHz QCM the pulse would have to be at least 50ns at a reasonably high voltage so that the power back is large enough to be sensed. This is fed to the input of the QCM system and the output is then captured over time, and will define the QCM's fundamental frequency, quality, and any delay in the system will also be seen.

There is very little discussion in literature to the use of this method for QCM sensing, though it has been mentioned as a possible method by Eichelbaum (Eichelbaum, Borngraber et al. 1999) within a single paragraph. The reason for the lack of common use is probably due to the expensive components needed, as to generate a narrow enough pulse needs a very fast Digital to Analogue Converter (DAC) and to get a high enough voltage pulse (to be able to receive a response as the time is very small so very little power is introduced) a very high slew rate DAC and amplifier are required. If possible this would be an interesting method to investigate as this may provide information that is hard to attain from other sources.

2.4.1.4 DISSIPATION

Though it is important to measure frequency for mass accumulation, dissipation is also an important characteristic in order to monitor the damping effect of binding and also the operating environment of the QCM. This has been measured using a method as defined by Rodahl (Rodahl and Kasemo 1996). Rodahl uses a signal generator to excite the QCM then after a set amount of time turns off a relay connecting the signal generator to the QCM and uses an oscilloscope to measure the frequency response that is then transferred to, and post processed by a computer. The response of the QCM is a mixture of a decaying exponential and a sinusoid wave as (Rodahl and Kasemo 1996) as in equation (2.10).

$$\mathrm{V(t)} = \mathrm{A_0 e^{-t/\tau} \sin(2\pi ft + \emptyset)}, \quad \mathrm{t} \geq 0, \tag{2.10}$$

Where $\mathbf{A_0}$ is the amplitude at t=0, τ is the decay time constant, Ø is the phase, t is the current time, and $\mathbf{f}$ is oscillation frequency. The dissipation was also calculated by Rodahl (Rodahl and Kasemo 1996) depending on the operating frequency and time constant as shown in equation (2.11).

$$D = \frac{1}{Q} = \frac{1}{\pi f \tau} \tag{2.11}$$

Rodahl's experimental setup is shown Figure 2.16 showing the two separate configurations to measure QCM frequency and dissipation as shown in his papers. The setup "a" requires much more expensive equipment as a high speed oscilloscope is needed whereas setup "b" needs instead only a lower speed analogue-to-digital converter. Setup "a" has been arranged to

operate in series (normal oscillator mode) or parallel mode by opening or closing the “switch” whereas “b” has also only been arranged to operate in series resonant mode.

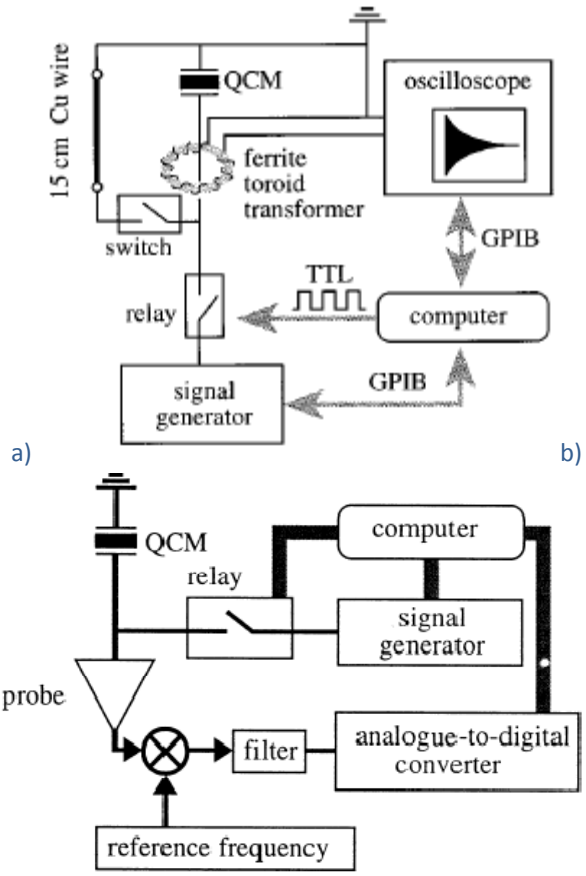


Figure 2.16: Schematic illustration of the QCM measurements a) (Rodahl and Kasemo 1996) b) (Rodahl, Hook et al. 1997).

Edvardsson et al (Edvardsson, Rodahl et al. 2005) also proposed a dissipation setup with dual frequency and elevated voltage oscillation amplitudes, which allows for better monitoring of the effects on voltage to particles binding to QCM. This method allows the running at elevated amplitudes for Bond Rupture frequency measurements while simultaneously driving at an elevated voltage.

To acquire a frequency measurement using the dissipation, a frequency estimation method must be used. Here are two such methods that would work. The first is taking the average samples between zero crossings from the captured exponential sinusoid (also called the zero crossing method). The frequency can then be calculated from this by multiplying the sample rate by the average samples and then taking the inverse of this creating a resonant frequency value. Another method would be to take a Fast Fourier Transform (FFT) of the data and measuring frequency that the highest point occurs at. This method can take some processing time depending on the sample size and is slow compared with the zero count method. There are many other frequency estimation algorithms and these can be found in any good signal processing book.

While these methods provide significant advances on traditional oscillator methods - with many advantages such as better operation in liquid, and more information gathered related to binding. There are some disadvantages such as the cost of equipment (though much less than the network analyser method) and a reliance on a computer for post-processing. Both these limitations could be removed by developing custom equipment to perform these measurements in a single product. From a development point of view, very little information on actual implementation of the dissipation and frequency algorithms are included in any of Rodahl's publications so to use these methods further research and development is required.

2.4.2 MEASUREMENT LIMITATIONS

There is a variety of different factors which affect the measurement accuracy of the QCM variables. Among these factors is the electrical noise which is introduced into the system which includes, amplitude and phase noise, noise introduced from operating in dampened environments, and the effects of different sensors on noise. Another factor that is addressed is the environmental effects that influence the frequency stability, such as, humidity, temperature, liquids, etc. The effects of impedance matching will be discussed. Lastly the effects on mass sensitivity will be discussed, for example how small a mass change can be measured.

2.4.2.1 NOISE

Frequency measurement noise is a limiting factor in the detection limits of a QCM and so must be addressed. This noise is the uncertainty in what the frequency really is and is caused by a number of different aspects depending on the detection method. In an oscillator based system the frequency noise is caused by phase noise and amplitude noise. This is represented by equation (2.12).

$$V(t) = \big(A_0 + e(t)\big)sin\big(2\pi f_0 t + \phi(t)\big) \tag{2.12}$$

With A_0 the amplitude, $e(t)$ amplitude noise function, f_0 fundamental frequency, and $\phi(t)$ the phase noise function found from IEEE standard (1999). The different types of noise in oscillator based systems can be seen in Figure 2.17.

Noise is caused by a combination of influences. There is electrical circuit noise and also environment effects changing the frequency of the QCM. It is impossible to isolate all the influences on frequency fluctuations so usually a large amount of averaging is used to obtain a stable value. With an oscillator based system it is normal to average for 1-10 seconds before the system outputs an answer. As a rule the longer the averaged time the more accurate a value that will be acquired – at expense of time resolution.

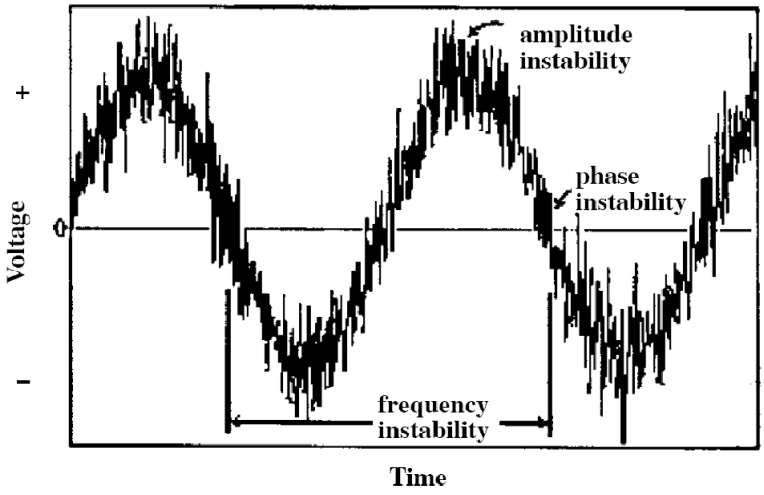


Figure 2.17: Instantaneous output voltage of an oscillator from IEEE standard (1999).

The noise at the zero crossing is relevant with a dissipation, impulse, or oscillator based measurement system as they rely on being able to count the zero crossing accurately. This does not have as much effect on a network analysis based method as network analysis filters out a lot of signal that is not needed (noise) and keeps on the signal of relevance at a chosen bandwidth (though at lower bandwidths a larger amount of time is taken to gather data).

When using Network Analysis methods it important to be able to measure the phase or the magnitude peak accurately as this accuracy effects the frequency variation measured. The noise in phase and magnitude depends on the band-width of the scanning frequency, the smaller the band-width the less the noise. Unfortunately the less band-width that is captured the longer the scan time, so a trade off must be made between scan time and noise.

A number of papers that have been found to be useful in the examining of noise are listed: (Vig, Vig et al. 2000; Bouzidi, Narine et al. 2003; Rodriguez-Pardo, Rodriguez-Pardo et al. 2005; Wakamatsu, Wakamatsu et al. 2007).

2.4.2.2 ENVIRONMENTAL EFFECTS

When measuring the attachment of mass in such small scales as on the QCM it is important to have a high frequency stability and low frequency noise to increase mass sensitivity. There are however, other important environmental factors which can lead to large frequency changes, so care should be taken to isolate any of these influential environment factors. These factors can include temperature, humidity, and for in-vitro experiments liquid viscosity and density, flow rate, and micro bubbles attaching to the surface of the QCM. These factors shall be addressed in the following paragraphs.

Bouzidi et al (2003) expressed the temperature change versus frequency change as a quadratic polynomial where the crystal is most stable at the minimum of the curve which is around 80°C as seen in Figure 2.18. This high temperature is not practical for the biological fluids but, a similar temperature graph would be useful for calibration of the QCM of choice.

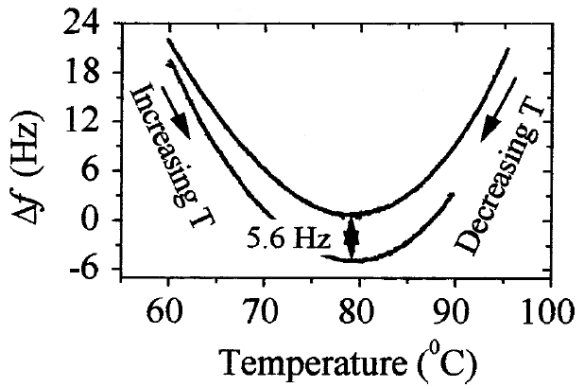


Figure 2.18. Resonance frequency versus Quartz temperature for warming and cooling cycles at 0.1 °C/min (Bouzidi, Narine et al. 2003).

Humidity causes a change in frequency on the QCM due to micro droplets forming on the air side of the crystal which in turn can cause a mass change. Stable temperature prevents the relative humidity from fluctuating and so stabilises the frequency change. Another option to stabilise frequency drift due to humidity would be to run nitrogen gas (which does not hold any moisture) over the surface.

Liquid viscosity and density are easy to overcome by using a liquid with a constant viscosity such as ethanol or a water buffer solution which is then in turn mixed with small amounts of the sample to be tested, but when testing biological fluids such as within blood samples this becomes a lot more complicated and no simple solution is apparent. Measuring from in air to liquid produces significant other problems, such as frequency and impedance change, and so care must be taken to use in one environment or the other and not to switch between them during measurements.

Flow rate is also a contributing factor that seems to increase the amount of frequency noise present in the system as well as adding to the total frequency drift. Sota (Sota, Yoshimine et al. 2002) found the frequency changes from noise of ±0.15Hz and drift of 12Hz/h for 0 µL/min to around noise of ±4Hz and drift of ± 60 Hz/h for 100 µL/min. Careful design of the flow cell and choice of QCM can decrease the frequency noise.

Micro bubbles can be introduced from a pump which forces bubbles though the system. These micro bubbles can attach to the surface of the QCM causing a significant frequency change. To prevent this care must be taken not to introduce bubbles into the system. One solution would be to use a vacuum pump rather than forcing the air into the system, so that the air will be removed first. Micro bubbles may also be introduced by the QCM when running at high voltages. Investigations into micro bubbles for electrochemical QCM have been made by Tsionsky et al (Tsionsky, Kaverin et al. 2005).

There are also other factors which can generally be associated with general degrading of the QCM such as frequency drift and the background noise. Another cause of this could be the inadequately cleaning of the QCM surface (which can lead to mass loss/addition) that adds to the drift and noise.

2.5 BOND RUPTURE

Bond rupture is defined as the breaking of the bond between two molecules. This has been performed on, single bond experiments using an AFM to induce the Bond Rupture, and with many bonds breaking simultaneously as performed using a QCM's surface movement (or shaking the bonds off) to induce rupture. By measuring at what point the bonds break information can be gathered about what type of bonds exist on that sensor surface. A major problem that this could solve with traditional biosensors is the problem of non-specific (or weak) bonds binding to the sensor, causing false positives. By using the Bond Rupture method the particles that are to be detected can be separated by their bond strengths.

This section discusses the background of Bond Rupture, and then moves on to the characterisation of it using a QCM as the actuator and sensor. It then continues with the electronic detection methods used, and then the experiments that have been carried out thus far.

2.5.1 BACKGROUND

Bond rupture is an experimental method that was first proposed by Dultsev (Dultsev, Ostanin et al. 2000) using a QCM in which the amplitude of movement is increased until such a time that bonds attached to the surface rupture. When this rupture occurs a frequency change can be measured due to mass change or a Bond Rupture 'noise' signal due to the bonds breaking. Bond rupture is an immunoassay based technique so uses antibodies as detection receptors to capture the antigen particles.

One problem that traditional immuno-biosensor methods face (such as standard QCM or SPR) is that in most cases more than just the antigen's specific to the antibodies receptors may bind – this is called non-specific binding. This non-specific binding is when other antigen's bind with the antibodies with a much lower force than the specific bonds that we are looking for. Non-specific bonding is going to occur more in complex solutions such as blood so if these bonds can be differentiated between, then measurements in complex liquids would be more reliable. The Bond Rupture method gets around the non-specific and specific binding problem by increasing the force applied between the antibody-antigen bonds until rupture occurs - by essentially shaking at a high frequency and increasing the shaking amplitude. A rupture force spectrum curve can be displayed showing the difference between the non-specific bonds (at the low force end) from the specific bonds (at the high force end).

There have been two main research groups investigating QCM based Bond Rupture technique. The first publications are associated with the Akubio group that started out of Cambridge University research, using the Bond Rupture noise technique (Dultsev, Ostanin et al. 2000; Cooper, Dultsev et al. 2001; Cooper 2003; Cooper 2003). It is interesting to note that no further publications have come out of this group in regards to Bond Rupture noise since (Cooper 2003), instead the group have focused on what is referred to as rapid acoustic profiling (RAP) which is monitoring association and dissociation rates of molecules for in liquid systems (Cooper and Whalen 2005; Cooper and Singleton 2007; Godber, Frogley et al. 2007).

Another group have monitored the frequency and dissipation (instead of the noise) of QCM while inducing Bond Rupture and also the effects of increased QCM amplitude on the surface bonding. This was carried out by Edvardsson and Rodahl (Edvardsson, Rodahl et al. 2005; Edvardsson, Rodahl et al. 2006; Edvardsson, Zhdanov et al. 2007). This work was based on Rodahl's initial frequency and dissipation measurement method (Rodahl and Kasemo 1996; Rodahl, Hook et al. 1997). Further experiments were carried out by Heitmann (Heitmann and Wegener 2007) in to the affect of driving voltage on cell adhesion onto the surface of the QCM to which the conclusion was that standard QCM operating voltages have little effect on the cell adhesion.

Other publications (though not directly related to Bond Rupture) provide important information regarding the amplitude of movement of a QCM such as Borovsky (Borovsky, Mason et al. 2000) and for the distribution of the amplitude over the QCM's surface Mecea (Mecea 1989; Mecea 1994). Bond rupture experiments on single bonds have also been measured using methods such as AFM (Merkel, Nassoy et al. 1999; Lo, Zhu et al. 2001; Avci, Schweitzer et al. 2004; Kaur, Singh et al. 2004).

2.5.2 CHARACTERISATION

The rupturing of the bonds is due to the inertia force of the antigen which is caused by the QCM's high-frequency movement (acceleration rate) and movement range. There are many factors that affect the force that is applied to the antibody-antigen bond. One of the largest effects would be the amplitude of movement that the QCM moves at and how this changes: at different places on the surface, at different driving voltages, at different operating frequencies, and with different types of QCM's. Other properties that effect the Bond Rupture is: the different size, shape, and mass of antigen particles; antibody-antigen bond strengths; change of strength with different loading rates; and possibly the type of bonds holding the antibodies on the surface of the QCM. These will be addressed in the following section.

The antibody-antigen attached to the QCM when modelled as a mechanical joint show a force being applied depending on the acceleration and mass of the antigen. The displacement of the crystal which is controlled by voltage applied to the crystal is significantly increased when operated at a harmonic. The operating frequency is the rate of change of the crystal's movement and so the acceleration can be calculated from the rate of change and displacement.

2.5.2.1 SURFACE AMPLITUDE

The voltage applied to the QCM is proportional to the surface movement amplitude as shown experimentally by Borovsky (Borovsky, Mason et al. 2000). Borovsky demonstrated this by looking at the surface of the QCM using a scanning tunnelling microscope (STM) while oscillating at different frequencies. When the QCM was oscillated it was found that the image gathered was blurred in a known direction, enabling the amplitude of movement to be measured. This experiment was repeated over a variety of different applied voltages and also with resonators of different quality enabling a formula to be developed to represent the movement of the QCM as can be seen in Equation (2.13).

$$A = C_{av}.Q.V_d \tag{2.13}$$

Where Q is the quality of the Quartz crystal, V_d is the driving voltage and C_{av} is a constant which was experimental derived to be $1.4 \pm 0.1\ pm/V$ (Borovsky, Mason et al. 2000). Consider an example where the resonator quality factor of 55,000, this was found to move around 50nm at 0.65V as can be seen in Figure 2.19.

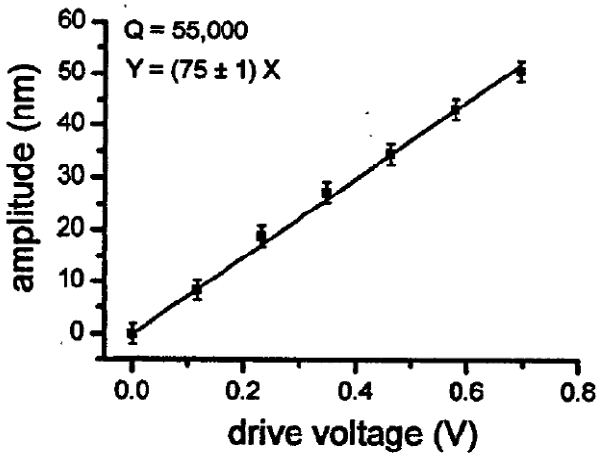


Figure 2.19: Amplitude at resonance versus peak drive voltage (Borovsky, Mason et al. 2000).

Another important factor is the change in amplitude when the driving frequency is offset from the resonant frequency. This is important to know as if there is an error in the measured resonant frequency, and then the amplitude can be significantly different. The amplitude versus driving frequency for a number of voltages was carried out by Borovsky and can be seen in Figure 2.20. Using Borovsky's results from Figure 2.20 it can be gathered that a change of around $\pm 25Hz$ would result in $\sim\%20$ loss of amplitude and $\pm 100Hz$ with a loss of $\sim\%70$. From this we can attain that it is important to control the drive frequency accurately to enable maximum amplitude of movement to be generated. It can be seen that the quality of the QCM significantly affects the maximum amplitude of movement. This is similar to the plots gathered in later chapters when scanning the voltage amplitude so without further experimental data it could be assumed that they are similar.

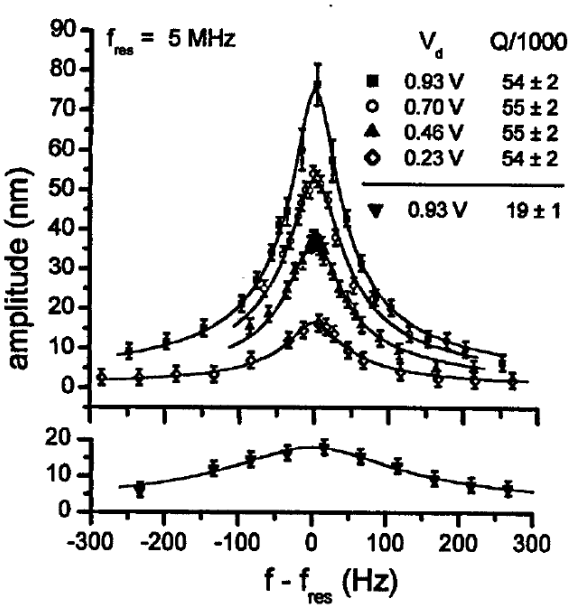


Figure 2.20: Amplitude of movement over a range of driving voltages versus difference from resonant frequency of a 5MHz AT cut quartz crystal (Borovsky, Mason et al. 2000).

So it can be concluded that the voltage, Quality factor, and resonant frequency are all very important factors to consider when looking at QCM Bond Rupture.

2.5.2.2 AMPLITUDE ACROSS THE SURFACE

There has been further work performed by Mecea (Mecea 1989; Mecea 1994) on the movement distribution of the QCM over the surface of the electrodes. This data can be seen in Figure 2.21 showing the relative amplitude over movement distributed over the surface radius, as well as the mass sensitivity distribution - for a 14mm AT cut Quartz crystal with 6mm silver electrodes. This amplitude data was created by placing a tip with a constant friction and downward force on the crystal and moving it across the surface and simultaneously measuring the energy dissipation for each point. Mecea concluded some relevant facts from his experiments. First it was found that the maximum mass sensitivity and amplitude of movement is near the centre of the electrodes. Also it was discovered that the mass sensitivity and amplitude movement extends outside the area of the electrodes.

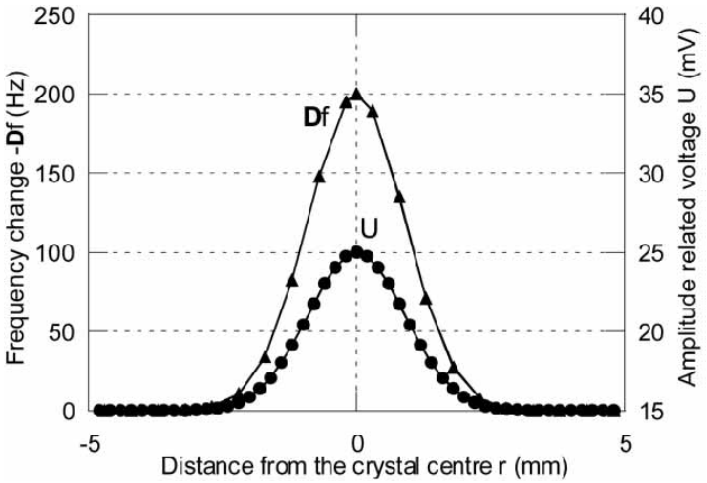


Figure 2.21: Mass sensitivity frequency change (Df) and vibration amplitude related distribution U versus distance from centre of resonator (Mecea 2005).

This has some significant effects on Bond Rupture experiments. For example, since the amplitude of movement varies over the surface, then the rupture force applied to the particles will also vary. A solution to this problem would be to mask the QCM so that only a small section of the surface is available for antibody-antigen bonding therefore limiting the surface on which the rupture could occur. Further experimentation would be required to determine the effects this has on Bond Rupture, but it could be assumed that it would cause the rupture to take place over a significant range of applied voltages.

Edvardsson carried out a series of experiments to analyse a variety of different effects related to the effects of increasing the voltage applied to the QCM at resonance. This included the effect of amplitude distribution over the QCM surface on the binding of particles; the effects of increased driving voltage on temperature and measured frequency change, and analysis on the effects of driving voltage on the inhibiting of particles binding (Edvardsson, Rodahl et al. 2005; Edvardsson, Rodahl et al. 2006; Edvardsson, Zhdanov et al. 2007).

Edvardsson (Edvardsson 2005) has investigated the relationship between driving voltage and displacement at different operating harmonic frequencies in air and water environments. Results from Edvardsson can be seen in Figure 2.22 using a 5 MHz QCM for measurements. Edvardsson found that a frequency change occurred due to the voltage that the QCM was operated at. A small amount of this frequency change was proposed to be due to temperature increase at the surface of the QCM – around 6Hz change. The rest of the change was theorized to be caused by the piezoelectric stiffening of the QCM as the amplitude of movement was increased. This is important in the frequency measurement of Bond Rupture occurring as this must be calibrated for or accounted for in some way, otherwise the frequency observed will be meaningless.

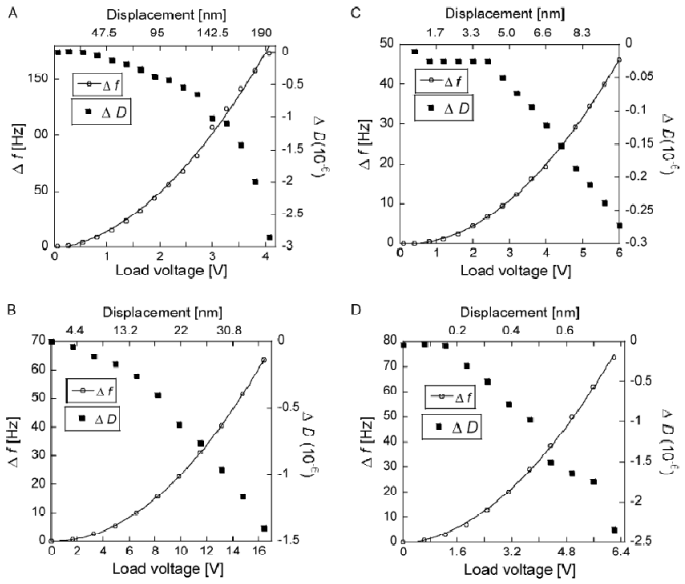


Figure 2.22: Changes in frequency and dissipation versus load voltage (and oscillation amplitude) in air (A,C) and pure water (B, D) and operated at fundamental harmonic 5MHz (A,B) and 7th harmonic 35MHz (C,D) (Edvardsson, Rodahl et al. 2005).

Edvardsson experiments did not find Bond Rupture due to QCM amplitude occurring but did find that the binding kinetics and distribution of binding is affected by oscillation amplitude. An example of this can be shown in Figure 2.23 showing the binding of 200nm beads at 0V and 10V. These show that there is a difference in the binding between the centre and the edge of the QCM. The amplitude of movement was found to be a Gaussian (see Equation (2.14)) distribution over the surface. This can be seen in Figure 2.24.

$$\frac{\Delta f}{f_0} = \alpha \left(\frac{V}{R}\right)^2 \tag{2.14}$$

Where f_0 is the natural frequency, α is the crystal dependant constant that determines the stiffening of the Quartz as the displacement increases, V is the load voltage, R is the motional resistance, and Δf is the change in frequency (Edvardsson, Rodahl et al. 2005).

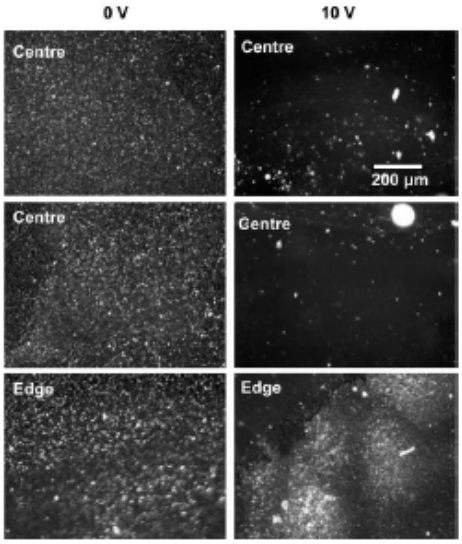


Figure 2.23: Microscope images showing the binding of ~200nm beads to the surface at 0V (left) and 10V (right) - imaged in wet state (Edvardsson, Rodahl et al. 2006).

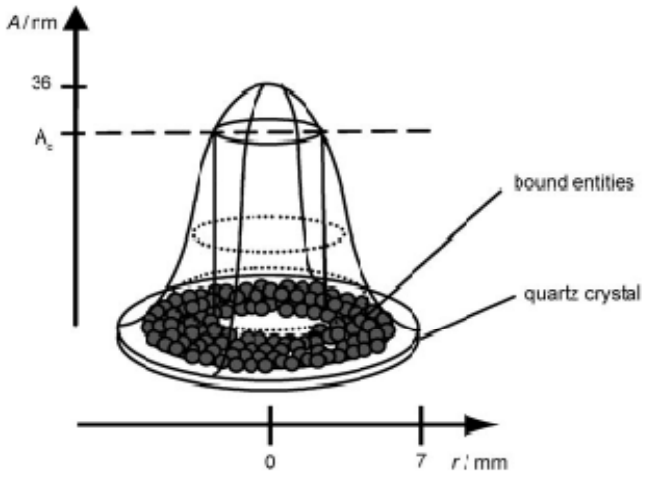


Figure 2.24: Diagram showing the Gaussian oscillation amplitude distribution. This also shows how the central region on the crystal maybe empty due to inhibition of binding (Edvardsson, Zhdanov et al. 2007).

It can be concluded from the literature mentioned that both the applied voltage and quality factor of the QCM is important in determining the displacement of the QCM. This requires a significant amount of consideration to be given to the measurement of the quality factor of the QCM. The voltage applied to the QCM also changes the measured resonant frequency, so creates a need to calibrate the voltage applied in some way when Bond Rupture is induced so that the resonant frequency can be measured accurately.

2.5.2.3 INDUCED FORCE

The force that will be applied between the antibody-antigen bonds is important as if this is known we can calculate the breaking force and from that characterise the bonds on the surface of the QCM. Cooper (Cooper 2003) proposed an equation to calculate the peak force that would be applied between particles in the equation below.

$$F = \frac{2}{7}.m.A.(2.\pi.f)^2 \quad (2.15)$$

Where m is the mass of the antigen, A is the amplitude of movement and f is the operating frequency. Using this formula it can be seen that the force is very dependent on the mass of the antigens so if the mass increases, more force will be applied between bonds, and so larger particles such as bacteria should be easier to rupture (according to this theory) than for example a virus. The assumption made is that the particles detected are spherical and attached at one pivot point to the surface. A figure depicting how this formula came about was created by Cooper and is shown in Figure 2.25, with the acceleration at the bottom the particle due to the movement of the QCM causing a rocking motion of the particle and therefore also a force produced – for further working of this formula and diagram see Coopers (Cooper 2003) paper. This model is very simple and may provide an initial idea of the forces involved. Though it does not take into account the multiple bonds that could be formed between the antigen-antibody or the differing shapes of antibodies, among a few of the other complex factors to take into account.

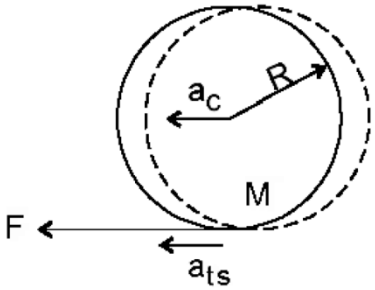


Figure 2.25: Inertia modelling for a particle (Cooper 2003)

To determine the force applied to the particle Coopers formula (Cooper 2003) as seen in Equation (2.16) can be used.

$$F = \frac{2}{5}.m.V.Q.(2.\pi.f)^2 \quad (2.16)$$

Where V is driving voltage and Q is the quality of the crystal. This gives us a force output if we know the mass, voltage, quality and frequency of the QCM. This is derived from the model in Figure 2.25 combined with Borovsky's (Borovsky, Mason et al. 2000) formula for amplitude of surface movement.

Using Coopers (Cooper 2003) formula it was determined that of a molecule with a mass of ~80 femto-grams and applied at a voltage of 7 and with a 14.3MHz QCM the acceleration of a molecule would be 3.4 x $10^7 ms^{-2}$. This would be a force of around ~3 nN applied to the bond. Typical measured biological bond forces require a force of around 10-200 pN. It is thought that due to multiple bonds forming per particle that the force needed to break the all bonds is therefore much larger due to having to break the multiple bonds at once.

There are a few factors that the author believes are not taken into account within Cooper's papers, such as the distribution of the movement amplitude across the surface, which would suggest when looking for a Bond Rupture effect that this would occur over a reasonable range

of voltages, and not all at once as suggested by Cooper. This will have to be confirmed experimentally to find the reason for this ambiguity. It is also unsure how Cooper ensures that the driving voltage is kept at the QCM's resonant frequency as this value varies as the voltage increases and the method shown seems to suggest an open loop frequency control is used. This would cause large variations on the amplitude of surface movement if operated off resonance. Further investigation and experimentation must be carried out to confirm this.

2.5.2.4 RUPTURE DETECTION METHOD

Inducing Bond Rupture is one thing but the important question is how to detect it? Dultsev's paper (Dultsev, Ostanin et al. 2000) demonstrates the detection of Bond Rupture using two methods the first and the simplest being the detection of the frequency change as the bonds rupture. The second method is listening for a noise given off at the third harmonic when driven at the first harmonic as the bonds break as demonstrated by Figure 2.26.

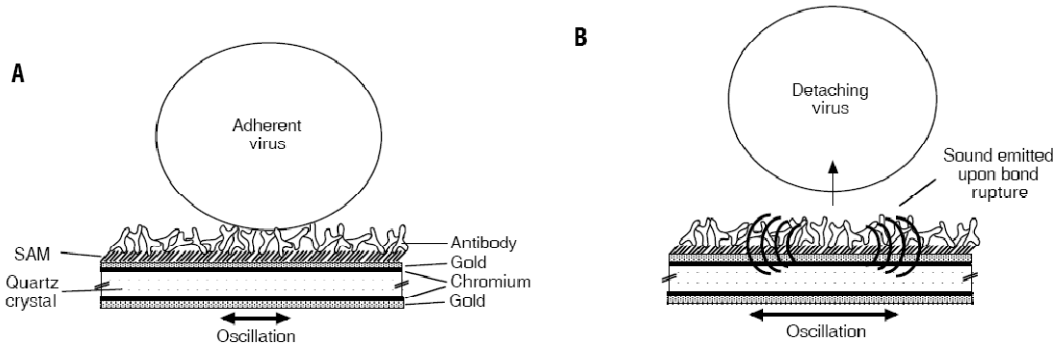


Figure 2.26: Idealized diagram showing Bond Rupture (a) oscillation before rupture (b) oscillation and rupture with emitted sound (Cooper 2003).

The frequency detection method has to this authors knowledge only been published by Dultsev (Dultsev, Ostanin et al. 2000) as seen in Figure 2.27. This figure shows the change in frequency as the voltage applied is changed for a variety of different bonds. It should be noted that there are very few points in the frequency curve plot in the Figure 2.27 (b), which limits the usefulness of this measurement. Also another interesting point is how narrow the Bond Rupture occurs over which seems conflicting with the Gaussian distribution of forces that from literature would appear to be applied across the surface. From the series of papers this group has written it is difficult to attain just how small the signal which is being detected is. Frequency detection seems to be a lot simpler but the Bond Rupture noise method has been claimed to be able to detect much smaller amounts of particles than traditional frequency change methods, but it shall remain to be tested as no independent body has verified the Bond Rupture via either of the two methods – though Edvardsson has shown that higher QCM operating voltages inhibit binding (Edvardsson, Rodahl et al. 2005).

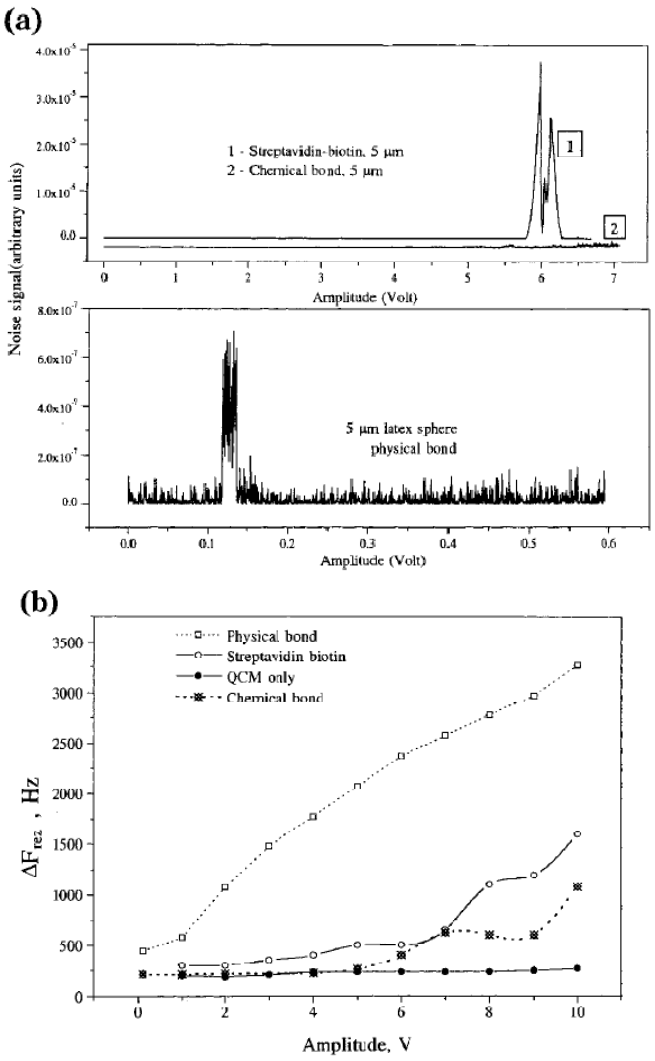


Figure 2.27: Bond rupture measurements from Dultsev (Dultsev, Ostanin et al. 2000) (a) Bond rupture noise measurement verses voltage (b) Frequency change verses voltage

2.5.3 ELECTRONIC EQUIPMENT

The inducing and detection of Bond Rupture is a problem that has thus far only been solved by the Cambridge/Akubio group. The electronic detection method that they used is a lock-in amplifier to detect the electronic Bond Rupture 'noise' signal and a frequency generator to produce the signal. There have also been multiple patents covering a assortment of different methods originating from this group. There has also been a method used by Edvardsson which though not for Bond Rupture as such, it is for the inhibiting of the bonds forming on the surface of the QCM and is still a relevant system to examine for electronic detection methods related to Bond Rupture.

Dultsev's method uses a signal generator model DS345 (Stanford Research Systems) driving the fundamental frequency, and a lock-in amplifier SR844 (Stanford Research Systems) operating at near the QCM's third harmonic as shown in Figure 2.28. During an experiment the Bond Rupture 'noise' data is captured using a sum of squares of the X (real) and Y (Imaginary) outputs from the lock-in amplifier which is then fed into the computer. Within the computer a

digital filter is used to remove false Bond Rupture event positives (Dultsev, Speight et al. 2001). Cooper's (2001) paper also presented the same experimental setup with the real and imaginary component of 2000 points from the lock-in being squared and added to make a power proportional signal as given in Equation (2.17).

$$P = X^2 + Y^2 \tag{2.17}$$

Where X is the real data squared and Y is the imaginary data squared or in-phase and out-of-phase data. This was then processed using median smoothing with a width of 25 in MathCAD using Equation (2.18) (Cooper, Dultsev et al. 2001).

$$S = P - medsmooth(P, 25) \tag{2.18}$$

Where P is the squared power from the last equation and medsmooth is a MathCAD function for taking the median of the data in vector P, at a width of 25. This function shows any outlyer points from the 25 point median so removes any problems from the 'noise' increasing slowly.

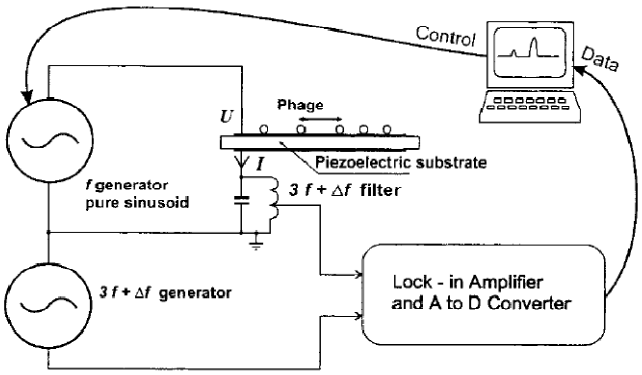


Figure 2.28: Schematic representation of experiment by (Dultsev, Speight et al. 2001).

The lock-in amplifier that the Akubio group used is of some interest as this should give the base specifications needed to detect the Bond Rupture signal. The diagram of the SR844 is shown in Figure 2.29. This uses RF amplifiers and attenuators to enable the optimum input level to be achieved. The signal is then multiplied by an analogue mixer with an in-phase and out-of-phase signal at the frequency of interest. This produces a low frequency signal around the frequency of interest and various high frequency signals of no interest which are filtered out using the low pass filter. Each channel is then sampled using a 16 bit ADC at a rate specified by the user. This data is then processed by a DSP and outputted in the format specified. The input frequency of interest has a separation of at least 80dB from any other interference signals. It also has a range of noise from 2 to 8 $\frac{nV}{\sqrt{Hz}}$. Though the exact configuration of the lock-in amplifier that the Akubio group used for the Bond Rupture 'noise' detection was not known, it can be assumed that the specifications of the lock-in amplifier are a good place to start for the designing of the system.

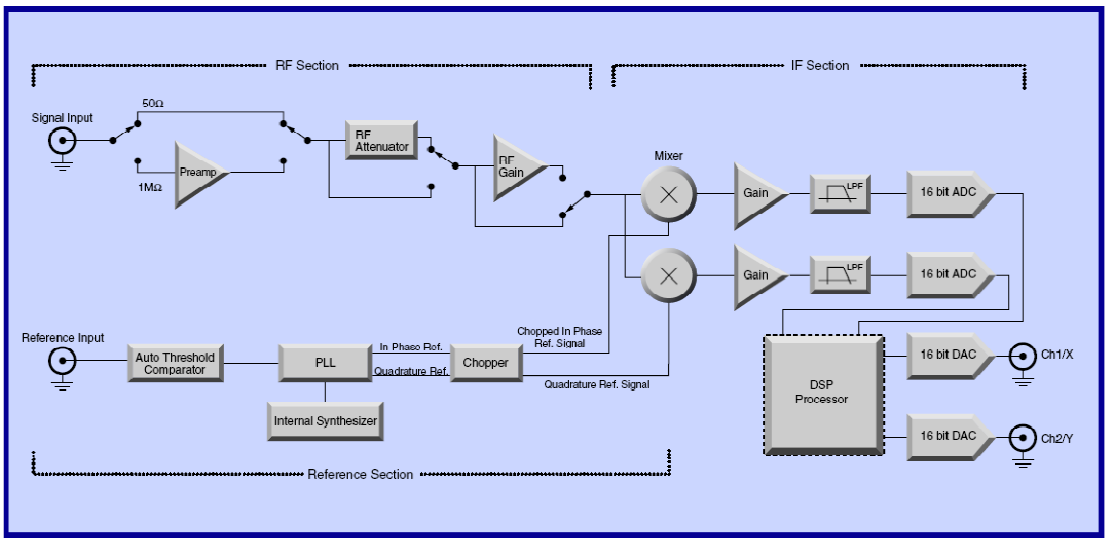


Figure 2.29: Lock-in amplifier (SR844 datasheet) used in Dultsev's experiments.

Edvardsson developed dual measurement equipment for frequency and dissipation monitoring of a QCM, while simultaneously generating a sinusoid at a specified voltage – see Figure 2.30. The use of the harmonics of the QCM enabled it to be stimulated at one frequency and measured at another. Edvardsson uses the dissipation method pioneered by Rodahl (also an author of this paper) to measure the resonant frequency change of the QCM. It is unclear however how Edvardsson ensures that the frequency at the secondary mode stays at the correct frequency as this would be sure to change depending on the driving voltage and other factors. Edvardsson used this method to inhibit the attaching of particles to the QCM. It was found that at higher driving voltages the beads did not attach on to the surface the same, with no attachment at the centre of the QCM while attachment still occurred further out on the QCM – depending on the amount of driving voltage. A similar effect would be expected to occur at the voltage increases for Bond Rupture.

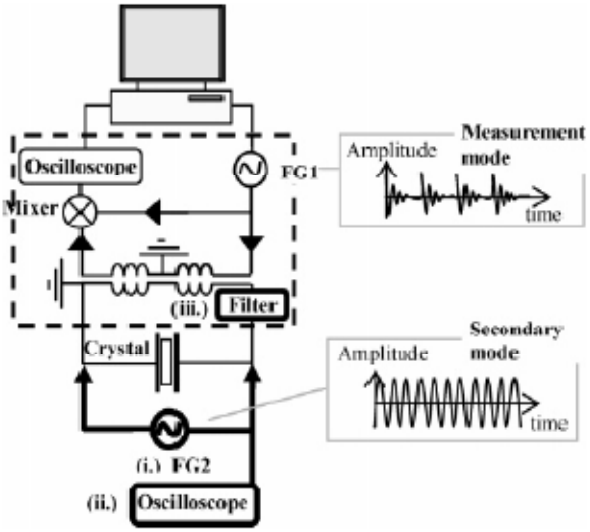


Figure 2.30: Dissipation measurement experimental improvements old setup in dotted line (Edvardsson, Rodahl et al. 2005)

There is much to investigate into the Bond Rupture electrical detection method. It would be believed that much could be taken from Edvardsson's paper for the investigation of the Bond Rupture electronics but method of primary interest is contained in Dultsev's and later Cooper's various papers, as it is the only in these papers that any information exists, pertaining to the Bond Rupture detection method. There is however a large hole in the Bond Rupture method literature and that is the monitoring of the frequency change, as no group has given this much attention. This should be the primary focus as this will enable the verification of Bond Rupture, which in turn should lead to the development of Bond Rupture 'noise' detection.

2.5.4 EXPERIMENTS AND RESULTS

To be able to detect the antigen its antibody must be attached on to the surface of the QCM. The attachment involves two steps (refer to Figure 2.31) with: firstly the Self Assembling Monolayer (SAM) being attached to the surface of the QCM forming a Sulphide-Gold bond, and secondly the antibody to the SAM via a Nitrogen-Carbon bond. The Sulphide-Gold bond and Nitrogen-Carbon are both covalent bonds so are much stronger than the antibody-antigen bond. Thus, the antigen will stay attached during Bond Rupture. For strengths of covalent bonds see Grandbois (Grandbois, Beyer et al. 1999). Now once the antigen is introduced in a solution onto the surface of the QCM bonding will occur between the antigen and the antibody. This bonding causes a mass change on the QCM which in turn causes the QCM to change its resonant frequency. Bond rupture (as mentioned previously) is caused by increasing the horizontal movement of the surface of the QCM (by changing the applied voltage at resonance) and monitoring the frequency change or the Bond Rupture 'noise' generated. This section outlines the Bond Rupture results and experimentation performed by the Cambridge, Akubio group.

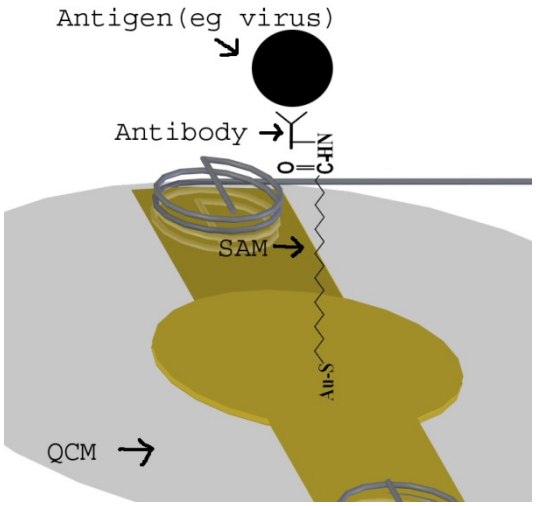


Figure 2.31: Rendered view of a QCM with rendition of self assembling monolayer, antibody and antigen from (van der Werff, Yuan et al. 2005).

Bond rupture is proposed to provide precise information of bond strengths present in a single scan. Dultsev et al (Dultsev, Ostanin et al. 2000) used a scan time of 500s and found that Bond Rupture occurred in a span from 1 to 10 seconds with around 10^4 to 10^5 balls attached on the surface. The detection was rated to be around 80 femto-grams in 1 µl of fluid which

corresponds to a mass sensitivity of $8x10^{-14}$ g mm^{-2}. Traditional methods are sensitive up to a QCM frequency change of, 10^{-9} g mm^{-2} and SPR sensor 10^{-12} g mm^{-2} (Cooper 2003). Dultsev found from a comparison of the results in air and water that the increase in viscous frictional forces as well as an increase of the effective mass of the antigen/particle causes the bonds to break with more ease in water that in air (Dultsev, Speight et al. 2001). It was also found that non-specific bonding mostly occurred at around 7.5 times less voltage than that of specific bonding sites (Dultsev, Speight et al. 2001). The results from Dultsev's 2001 paper are shown in Figure 2.32. This shows the non-specific bonds breaking at a reasonably low voltage and the specific bonds breaking at a higher voltage. Furthermore it was found that subsequent scans showed no significant noise presence.

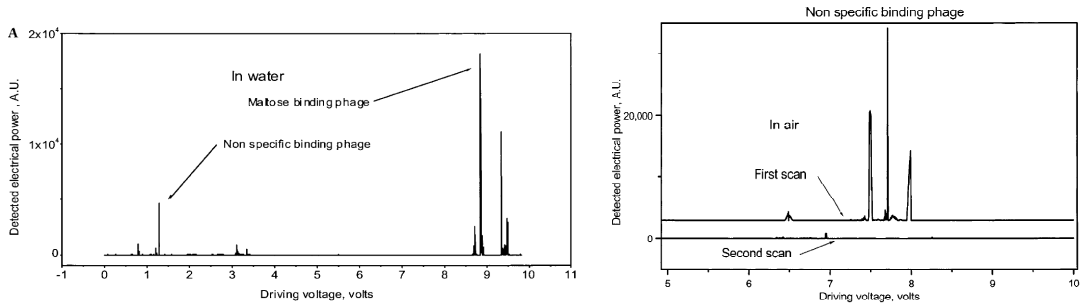


Figure 2.32: Experimental results from (Dultsev, Speight et al. 2001).

Dultsev et al (2001) also did a statistical analysis and fitted the integrated signal with the number of phage present using Figure 2.33. This created the following equation.

$$y = a + \left(\frac{x}{b}\right)^{C} \tag{2.19}$$

Where a, b and c are constants and x is the number of phage on the surface and y is the curve. If accurate this would suggest that the level of the Bond Rupture 'noise' signal could be used to determine the number of bonds that have broken at the specified voltage.

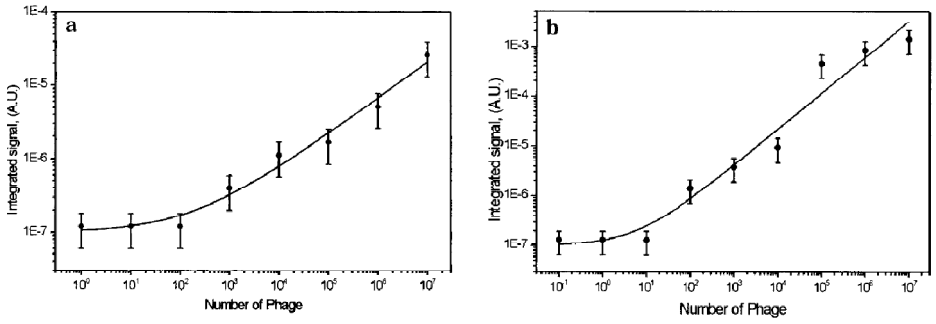


Figure 2.33: Curve fitting determined from experiment in determining Bond Rupture integrated signal versus number of phage rupturing (Dultsev, Speight et al. 2001).

Cooper et al (2001) found that after the first scan the antigen (HSV1) was removed, no signal or no Bond Rupture was found to occur during the five subsequent scans (Cooper, Dultsev et al. 2001). There is the need for further investigation to see if any small amount of rebinding or any loss of SAM's or antibodies occurs to determine whether this could be used to perform multiple experiments on the same QCM.

There are not many confirming sources of Bond Rupture from outside the work done by (Dultsev, Ostanin et al. 2000; Cooper, Dultsev et al. 2001; Dultsev, Speight et al. 2001; Cooper 2003). One of the few independent similar experiments that have been reported is by Edvardsson (Edvardsson, Rodahl et al. 2005) who used dual-frequency QCM for frequency and dissipation monitoring. Edvardsson (2005) concluded that although antibody-antigen binding was able to be prevented with increased oscillation amplitudes, no signs of Bond Rupture were found. Edvardsson concluded that this could be due to the different in crystals used, the shape of the virus particles, and that Streptavidin – Biotin bond that is used requires a larger force than Coopers and Dultsev's experiments.

2.6 ANALYSIS OF LITERATURE

The literature review covers a variety of topics all related to the development of the equipment of an antibody based specific antigen biosensor specialising in the Bond Rupture technique. It has been found that there is one main group working on Bond Rupture noise detection, and that there lacks independent experimental data pertaining to the Bond Rupture noise method. Further effort must be made to verify the Bond Rupture 'noise' method. The detection methods that have been published are also limited in that they are expensive, bulky (so not portable), and difficult to setup. To advance the Bond Rupture technique, further electrical detection methods need to be developed. It is also believed that Bond Rupture frequency monitoring is required to further verify the Bond Rupture.

Very little is known about the size of the electrical 'noise' signal emitted during Bond Rupture. There is therefore much work to be carried out to verify the size of this signal and the region that it occurs in or its band-width. It is not known if this signal is a wideband emitted signal or present only at the resonant frequencies of the QCM. The equipment developed therefore needs to be able to capture wideband signal information as well as narrow band signal information and be able to test the third harmonic frequencies to determine the optimum listening frequency. Another unknown factor is if the Bond Rupture occurs after a long driving time or after a very short time, as Cooper's group used times of 500s but states that only the voltage determines the rupture point. The effect on time on Bond Rupture remains to be investigated. The equipment would have to be able to capture a large amount of data for a significant length of time. Much experimentation is required to verify the Bond Rupture effect, but to do this a significant development is required in the base system from which to work from.

The resonant frequency of the QCM is required to be measured for Bond Rupture to ensure that the output voltage is operating within the optimum frequency range to induce Bond Rupture. This frequency change can also be used to verify when the Bond Rupture occurs by monitoring the frequency change as the bonds rupture from the surface. The measurement method for resonant frequency that was chosen was to use a network analyser based method. This was chosen as this enables much be learnt from the QCM which cannot necessarily be obtained with the oscillator based method. The dissipation method also shows considerable promise but would need custom equipment to be developed to measure in this way. The initial experiments were carried out in Chapter 3 using a network analyser and Chapters 4 and 5 by

using custom equipment which enables both the network analysis and the dissipation method to be implemented.

3 CHARACTERISATION OF QCM

3.1 INTRODUCTION

Now that the fundamentals in literature have been documented, further investigation is needed into what path should be followed for the equipment to be developed, with many possible ideas to be examined. The first aspects to be investigated was to find a method to accurately measure the frequency change of a QCM with the ability to modify the driving voltage while also being able to monitor the third harmonic Bond Rupture 'noise' found by Dultsev (Dultsev, Ostanin et al. 2000). The features specified are not found in any one QCM existing electronic driving and detection method, and so a new method must be found.

It was known that a network analyser could be used to characterise frequency of a QCM much better than a traditional oscillator based approach. This included many advantages such as the ability for in liquid operation and measurement of quality of the Crystal. It was also known that it was possible to create a custom network analyser system for QCM analysis such as Schröder (Schröder, Borngraber et al. 2002) had done. This led to the pursuing of a network analyser based QCM system.

Before designing the system, frequency monitoring methods were tested using an HP network analyser. This gave the advantage of having a calibrated system to which we could also verify our own equipment once created. To measure the QCM's resonant frequency with a network analyser, while simple, is very tedious so to enable the gathering of data over a long period of time a Matlab script was developed to accelerate the process. The Stability of the QCM was then able to be measured in terms of: Short term frequency stability, long term stability, and stability over a temperature change. The network analyser was also used to measure the impedance response of the QCM and so to determine how to enable the optimum power transfer in the interests of generating Bond Rupture.

It was found that the frequency versus temperature and humidity change was significant, so some equipment was developed to examine these effects in greater depth. This enabled the temperature to be stabilised as well as the temperature and humidity to be measured. This was then examined in conjunction with the network analyser equipment to investigate as much as possible about the QCM before deciding on the system specifications required to enable the detection of Bond Rupture via frequency and 'noise' measurements.

Majority of this chapter is based on the experimentation that is described in a paper presented by this author (van der Werff, Yuan et al. 2005).

3.2 QCM FREQUENCY MEASUREMENT

To measure the response of the QCM under differing environments a Network analyser was used. The Agilent Technologies RF network analyser (8712ES) was used due to the ability to analyse the frequency range required and also most importantly one was available for use. This network analyser can analyse any frequency between 300 kHz to 3 GHz. A network analyser is basically an electrical black box analyser – see Figure 3.1. The network analyser can be split into three main parts. Firstly, the source generator from which a signal of constant

power is produced which also constantly varies the frequency over the specified range of interest. Secondly the reflected power is measured from the source port – so measuring the power that is reflected back from the black box. Thirdly the power is measured from the destination port – measuring the power transmitted though the black box. When the information is combined several different measurements over frequency can be performed based on either the source or destination port data. Among these measurements are: phase, in-phase magnitude (real), out of phase magnitude (imaginary), power magnitude (gain/loss), impedance, group delay, and smith chart impedance plotting.

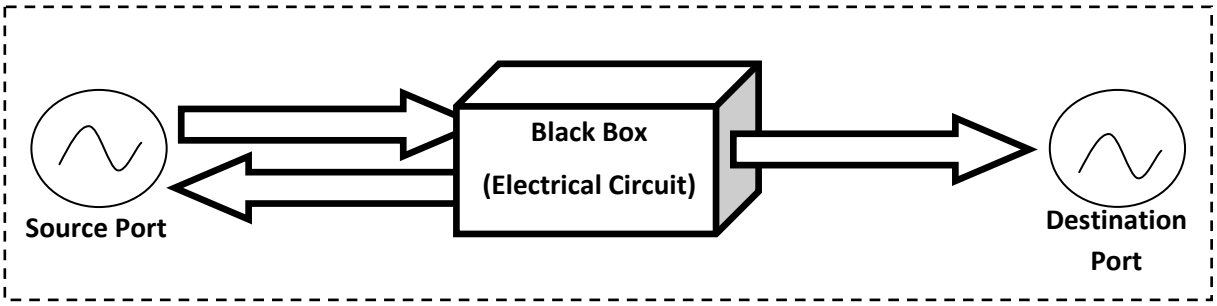


Figure 3.1: Response of a black box to a signal (Network Analyser DS)

The problem was to measure the frequency of a QCM. The main method used to measure frequency is to look for the frequency where the phase crosses zero - this is where oscillators operate as this is the frequency of least resistance to oscillate at (where the complex impedance of the QCM is zero as described in section 2.3.2). There are some draw backs with this method as in highly damped environments on the QCM's surface can cause the phase to drop below zero (due to complex impedance not being zero) making it difficult to detect the phase crossing with the network analyser. Another method was found which used the power magnitude and found the top point most point corresponded with the zero phase crossing. This method worked well in liquid because the plot becomes wider the peak can still be measured easily. It was also found that you can get information about the quality factor (Q) of the QCM by looking at the band-width of the magnitude curve. This was also the best method as it looks at the point of least resistance which is the real point where the maximum movement on the surface of the QCM exists.

3.2.1 EXPERIMENT SETUP

The experimental setup that was used to measure the QCM for in air and in liquid is comprised of: a computer running Matlab connected to the network analyser via the network communicating via TCP/IP SCPI commands and reading the scans via a FTP data download of the CSV (comma separated values) data file; and the network analyser's output connected to the input to the crystal and output of the crystal back to the input of the network analyser (Figure 3.2). This setup enables the automated measurement of the frequency of the QCM. To enable the measuring of frequency change in liquid, a setup comprised of a pump connected to the flow cell so that the liquid ran across the top of the QCM then out a liquid waste pipe was used (same as Figure 3.11 without Peltier setup).

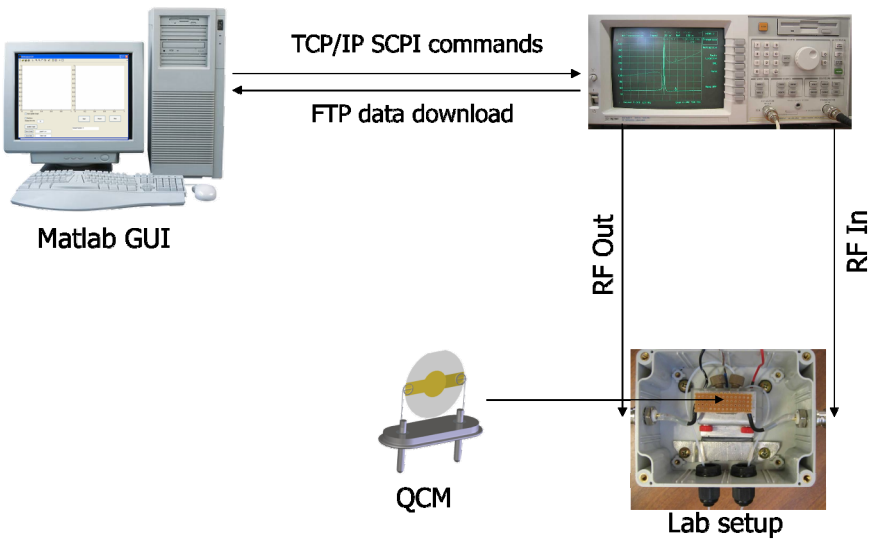


Figure 3.2: Experimental setup for driving the QCM with the network analyser.

3.2.2 SOFTWARE - MATLAB INTERFACE

One of the first objectives of this project was to stabilize the QCM's frequency noise, and determine that factors that affect its stability. To gather accurate measurements from the QCM an Agilent 8712ET network analyser was used to provide accurate resonant frequency measurements using the maximum impedance measurement in conjunction with Matlab to automate measurements. A Matlab GUI (Graphical User Interface) script (see Figure 3.3) was created to process and record the data from the network analyser, and send commands to the network analyser. The setup of the network analyser can be seen in Figure 3.2.

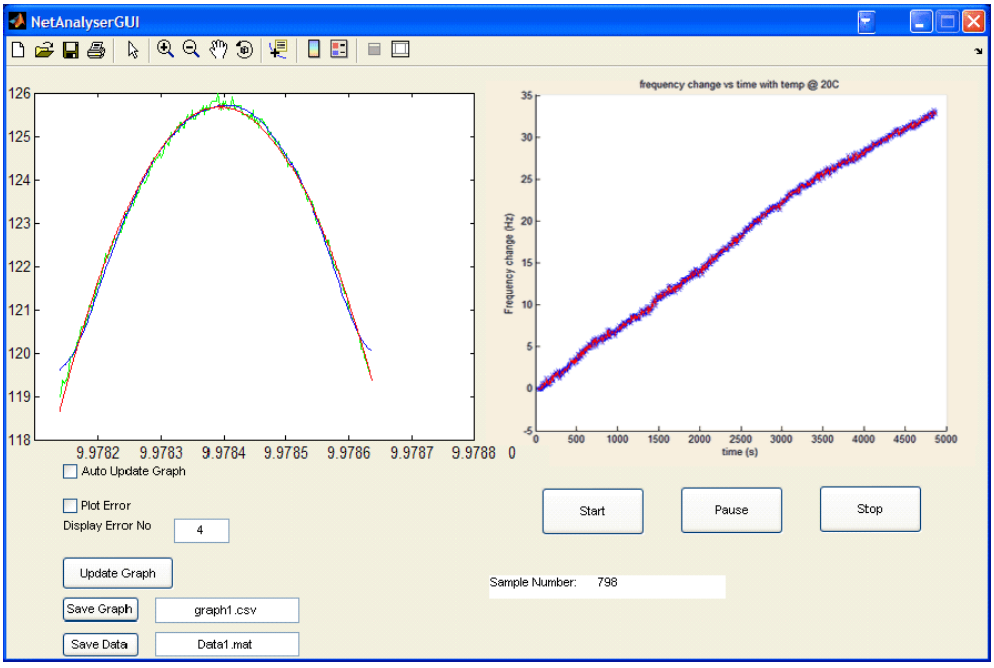


Figure 3.3: Matlab GUI for recording frequency change (van der Werff, Yuan et al. 2005).

Matlab GUI script as seen in Figure 3.3 consists of two output graphs, one of the curve fitting (left side), and one of the frequency change output (right side). It also consists of buttons to start, pause, and stop the frequency capture sequence for the control of the network analyser experiment. There are also some buttons to save the window data and output to a CSV file (coma separated values can be read by many different plotting programs including Microsoft Excel). When the 'Run' button is pressed on the GUI the following Matlab Script is carried out:

1) *A TCP/IP port connection is established to the network analyser to send the initial SCPI commands to setup the initial frequency range and set the measurement type to impedance.*
2) *Once finished waiting for Analyser to run a scan, the data is downloaded from the network analyser as a CSV file via ftp.*
3) *The resonant frequency value is calculated from the maximum impedance point, or if a narrow enough frequency band-width is used a 2nd order polynomial fit is used to find the maximum impedance point.*
4) *Change the frequency range on the network analyser via SCPI command unless at maximum frequency range.*
5) *Record maximum impedance frequency and draw graph.*
6) *Go back to step 2.*

The frequency can be found by finding the maximum point of the magnitude spectrum and then taking the corresponding frequency from this, unfortunately this is a very inaccurate method that depends on the precision of the frequency steps as well as the amount of noise on the magnitude spectrum. Another method to find the maximum magnitude point (and so the frequency) is by looking at a frequency width that approximates a second order polynomial (see equation 3.1). Then when a polynomial is fitted using the Least Mean Squares (LMS) fitting method, the polynomial will represent the best fit curve that will fit all the points. This takes advantage of all the points rather than just the maximum point and does not depend on frequency step width. Once the polynomial coefficients are gathered the maximum point can be quickly attained by firstly differentiating Equation 3.1 (which becomes an equation for a straight line) then solving for $y = 0$ to find the maximum point (Equation 3.2). This allows the frequency to be found and the new frequency is then used in the next run as the new centre frequency.

$$y = a.x + b.x^2 \tag{3.1}$$

$$x_{max} = -\frac{a}{2.b} \tag{3.2}$$

When performing the frequency measurement of the QCM using the network analyser noise figures of around 0.5Hz frequency resolution in air and about 2.5Hz resolution in liquid were gathered. But there are some drawbacks with this method. For instance if the frequency range of the curve is too wide it will not look like a second order polynomial, and so the frequency gained could be inaccurate, but if you zoom in too far large movements in frequency maybe missed and the tracking lost. This frequency range also needs to be optimised if the quality factor of the QCM changes, as the quality factor is directly related to the band-width of the

QCM (width of the frequency curve). The quality factor is slightly different between QCMs but is mainly affected by the difference of mediums it is operating in. For example the Q-factor of the QCM in air maybe about 10,000 whereas in water it is around 2000, which relates to a Frequency Band-width of around ~1000Hz and ~5000Hz respectively with a 10.7 MHz QCM. Therefore, the frequency range must be set wider in water to get the same polynomial curve. A result of a wider band-width is a reduction in frequency resolution – for example the frequency resolution in liquid would be about 5 times less than in air. Another drawback with this method is that though it will work with QCM's with large variations it does take time to stabilise. This time depends on the number of steps that it takes to be at the smallest frequency span, and as shown in Figure 3.4 with around 5 steps to be fully stable or about 30 seconds. Once stabilised, as seen in the right of Figure 3.4, the frequency remains very stable with the rising trend due to temperature effects or instability of the network analyser's base frequency.

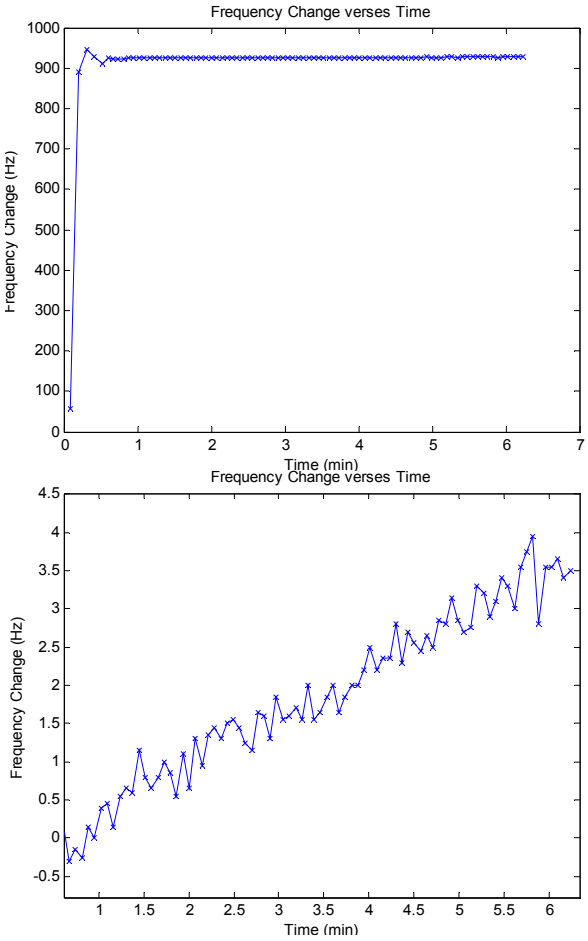


Figure 3.4: Frequency change of the QCM in air showing the frequency locking in air *(left)*. Magnified frequency change from left graph once stabilized *(right)*.

3.2.3 FREQUENCY STABILITY

The frequency stability was measured using the network analyser fitting method over 2hrs with a data point being produced at around every 30 seconds. This showed that the measurement technique was relatively accurate with frequency measurement accuracy to

around ±2.5Hz for a 10MHz crystal in air. It was also found that the QCM drift was around 2.5Hz over a time of 140min. While acceptable the cause of the drift is unknown and requires further investigation. This equates to be 0.25ppm variation in frequency measurement.

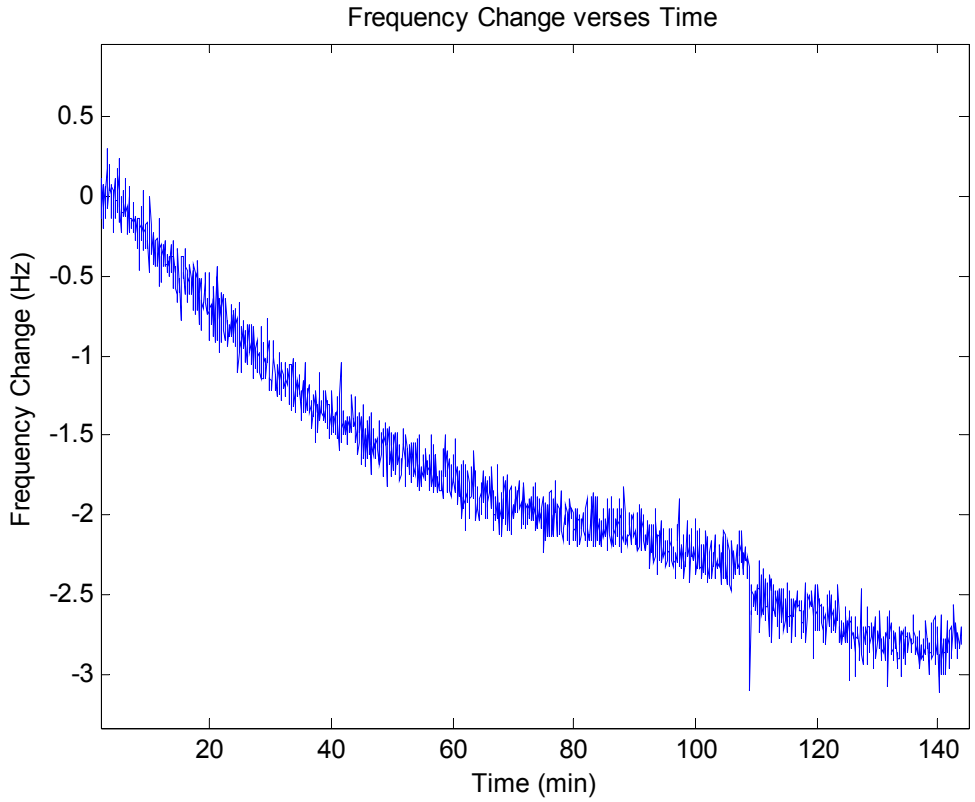


Figure 3.5: Frequency change versus time for the network analyzer peak measurement method for a period of 2 1/3 hours in a standard air environment.

The next experiment was run overnight in order to see how the QCM's frequency is affected by the fluctuations in temperature. The results are shown in Figure 3.6. A maximum change of around 2.5ppm was measured.

It was decided that the short term fluctuations were of more interest than long term fluctuations as the experiments to be run should be around a minute long. Therefore, there should be a minimal amount of fluctuation due to QCM stability. The temperature effect however, will be further investigated because, if a significant amount of energy in applied to the QCM for Bond Rupture a temperature change will occur. This discussion is continued in the temperature stability section (section 3.3).

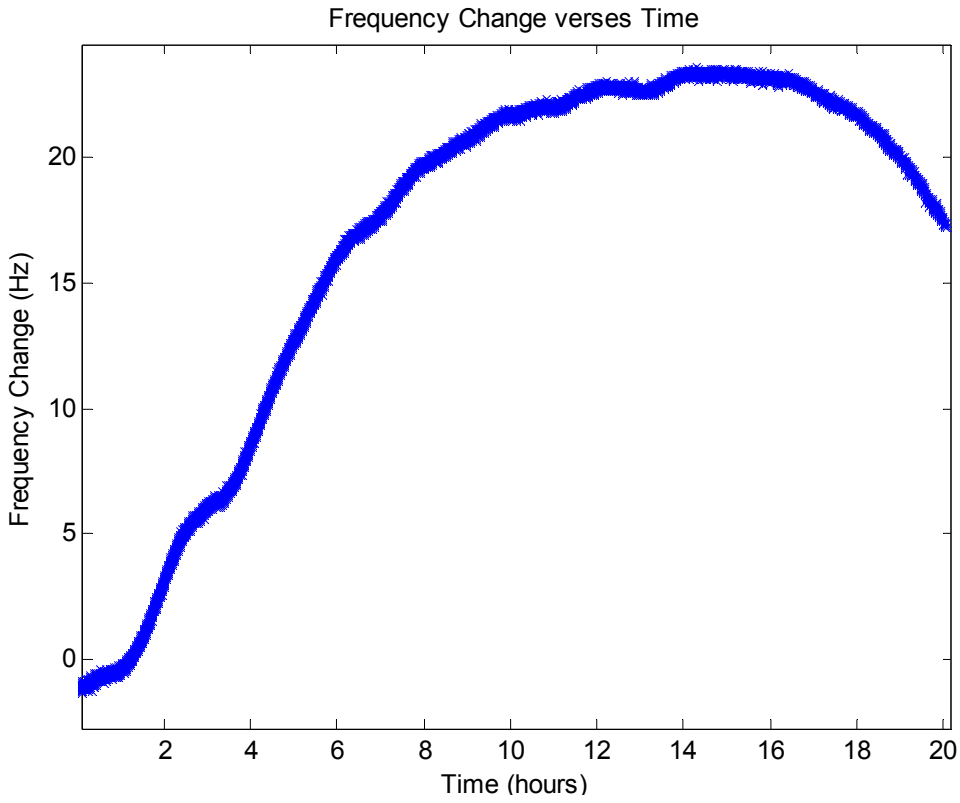


Figure 3.6: Overnight (20Hrs) frequency Change monitoring in air.

3.2.4 IMPEDANCE ANALYSIS

An important aspect regarding the transfer of the power to the QCM is it's impedance at resonance. Resonance is where the imaginary impedance components, capacitance and inductance, ideally cancel each other out leaving pure resistance. This enables the QCM to operate at a point of least mechanical resistance. The QCM impedance must be matched to the source amplifier otherwise a proportion of the signal will reflect back to the amplifier with the power not being converted to movement and therefore not inducing Bond Rupture on the surface. This creates a situation where some sort of impedance matching circuitry is necessary. Most systems use 50 Ohms as the base impedance, so any equipment made would use this to match to impedance. The impedance of a high quality QCM for operation in air is about ~10 Ohms (see Figure 3.7a) so some sort of matching is necessary. For liquid operation the same QCM may be around ~100 Ohms (see Figure 3.7b). This creates a problem when trying to match for both. The liquid causes an additional problem as the impedance is changed to not purely resistive (see literature review section 2.4.1.1) and so some effort must also be made to move the resonant frequency into the resistive region.

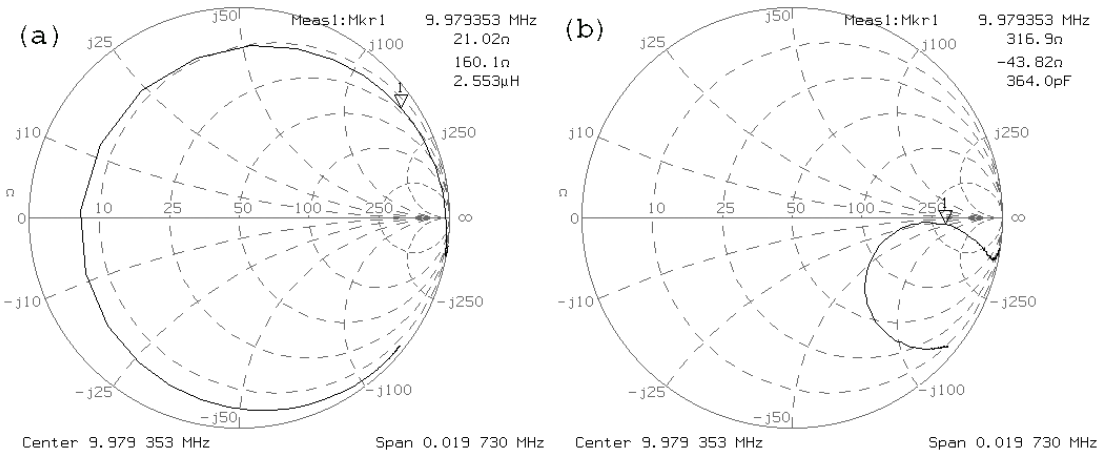


Figure 3.7: Smith Cart of the impedance in (a) air and (b) water.

There are two main types of impedance matching. The first and simplest method is to use a transformer. The transformer can scale the impedance directly depending on the winding ratio. An ideal transformer has no effect on the imaginary impedance although in reality all transformers will add a small amount depending on how well the transformer is made. The amount that the transformer shifts the impedance is the squared value of the ratio of the transformer.

The second impedance matching method is using a simple L network of a capacitor and an inductor (see Figure 3.8). This is solved by firstly finding X_p so that the resistance of the R_L equals the resistance of R_s. Then we are left with a complex resistance which is then canceled out with impedance opposite to the parallel impedance. So if we used a capacitor for X_p an inductor must be used for X_s and vice versa. Of course this only works if $R_L >= R_s$ in this configuration, so to match for $R_L < R_s$ the L network is turned around with X_p in parallel with R_s.

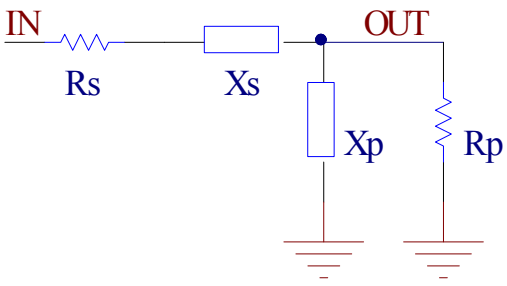


Figure 3.8: L Network for impedance matching Rs is the driving resistance, Rp is the load resistance, Xs is the series reactance, and Xp is the parallel reactance.

When using the L network impedance matching the Q (quality factor) of the match must also be considered – which is effectively the bandwidth of the impedance match or the frequency range over which the impedance is correctly matched. Q is important as if it is too wide you maybe including frequencies that are not wanted or too narrow and you may not be covering the frequency of interest to you especially if you are after a wide frequency range. Q is defined as in equation 3.3, where f_c is the center frequency and f_1 is the low frequency 3dB point and f_2 is the high side -3dB point.

$$Q = \frac{f_c}{f_2 - f_1} \tag{3.3}$$

There are a wide range of network topologies for impedance matching with the pi and T topologies giving a narrower Q response as well as wideband matching networks which are basically cascaded L networks see Figure 3.9. The topologies are all calculated from the base L network method and then cascaded using a virtual resistance.

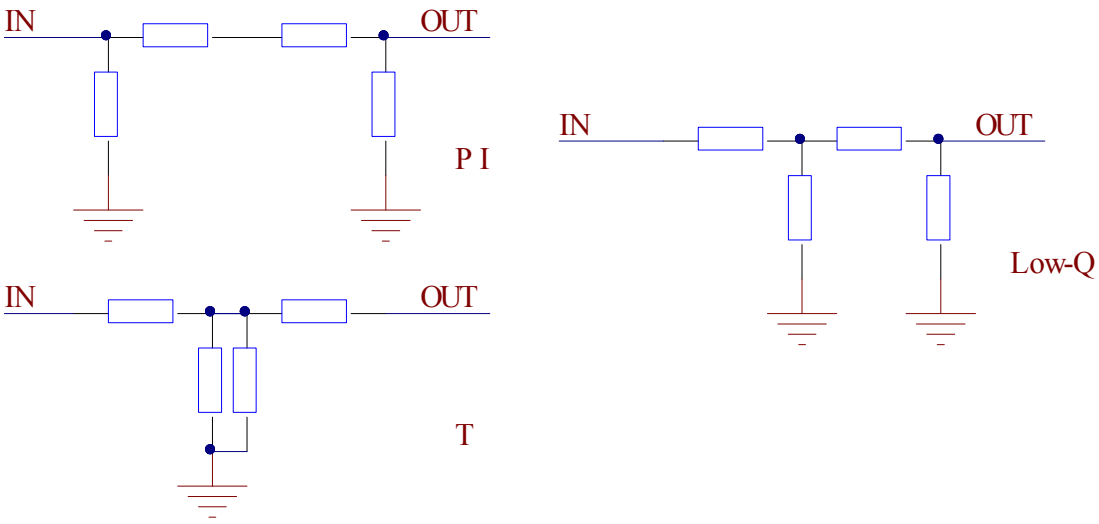


Figure 3.9: Network Topologies

To calculate the values for a simple L network impedance matching network a few simple formulas are used. The first is equation 3.4 which is used to calculate the Band-width (Q) of the network. If a higher Q is required a different network topology may need to be used. The next step is to use the Q value calculated and substitute this into equation 3.5 and equation 3.6 for the Series and parallel impedance values X_s and X_p. Now chose whether to use a capacitor or inductor for Xp and then use the opposite for Xs. After this calculate the values for C and L by substituting the correct impedance into equation 3.7 and equation 3.8. Where X_p is parallel impedance, X_s is the series impedance, R_p is the parallel resistance, and R_s is the series resistance.

$$Q_s = Q_p = \sqrt{\frac{R_p}{R_s} - 1} \tag{3.4}$$

$$Q_p = \frac{X_p}{R_p} \tag{3.5}$$

$$Q_s = \frac{X_s}{R_s} \tag{3.6}$$

$$C = \frac{1}{2\pi f_c X_c} \tag{3.7}$$

$$L = \frac{X_L}{2\pi f_c} \tag{3.8}$$

This network topology works well for pure resistive loads and the other stray reactance of the circuit can be compensated for as well by choosing the correct capacitor and inductor values.

This is performed by either absorption which is where the reactance is absorbed into the impedance matching circuit or resonance where an extra component is added in series to cancel out the reactance of the circuit. The reactance can be measured using a network analyzer plotting a smith chart see Figure 3.7. The network analyzer can also be used to find an accurate impedance match by changing the inductor and capacitor values while the device is under test.

The impedance matching must be investigated further as it is unsure how much affect the unmatched impedance will cause and what happens if the impedance changes in circuit. This shall be investigated in later chapters.

3.3 TEMPERATURE STABILITY MEASUREMENT

To keep the temperature constant a Peltier thermo-electric cooler/heater was used with a switched mode PID temperature control chip supplied by maxim (MAX1978) and a high accuracy calibrated thermistor to control our temperature within at least ±0.05°C. The controller can control accurately down to 0.001°C but noise in the potentiometer and thermistor make are much higher so to get higher temperature control better components must be used. The temperature at this stage is controlled by a high precision ten turn potentiometer which the MAX1978 compares with the thermistor value and is controlled by a PID controller that is implemented in hardware (resistor and capacitor choice). It was then changed to a using a digital microcontroller to measure and set the temperature and use an ADC to read the values to our system so that we can set and monitor the heat in software.

To transfer the heat to the liquid a primitive heat exchange system is used (see Figure 3.10). Because of our relatively low flow rate at this stage we can just run the pipes past the Peltier device using an aluminium plate with four holes just big enough to allow the biological grade pipe to be fed through also with a hole drilled halfway through for thermistor placement. As the pipe runs thru the aluminium plate the biological fluid will heat up to close to the Peltier temperature setting.

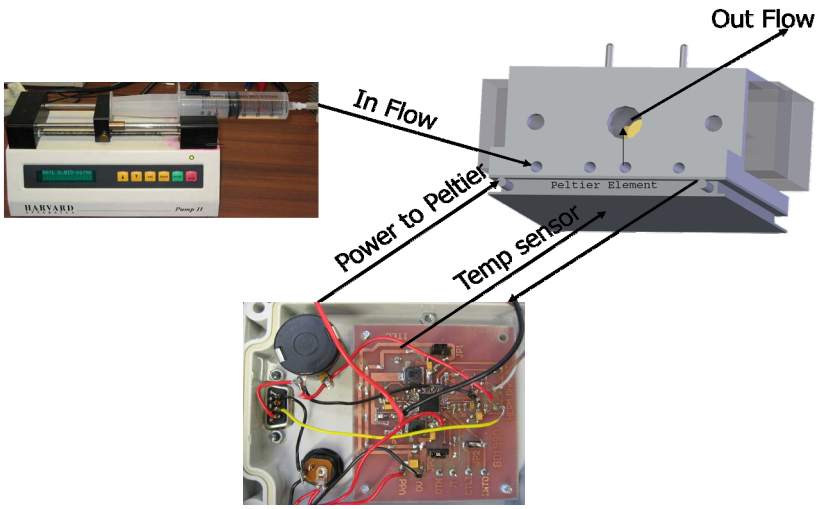


Figure 3.10: Experimental setup for the flow and temperature control.

This experimental setup has managed to stabilise the frequency fluctuations to a certain amount of accuracy, but did not enable the temperature or the humidity to be measured, which are both significant contributors to frequency change of the QCM. The frequency change using this system in liquid is shown in Figure 3.11. This figure shows the temperature initially set to the highest setting which is around ~45 °C and then changed to close to the lowest setting of ~3 °C, while all the time monitoring the frequency change of the QCM with the network analyser. This shows a change of greater than 1000 Hz which is significantly more than the tests performed in air.

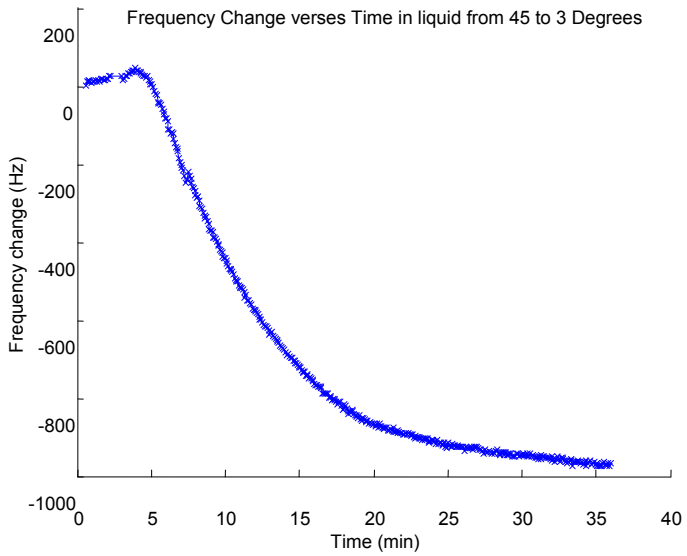
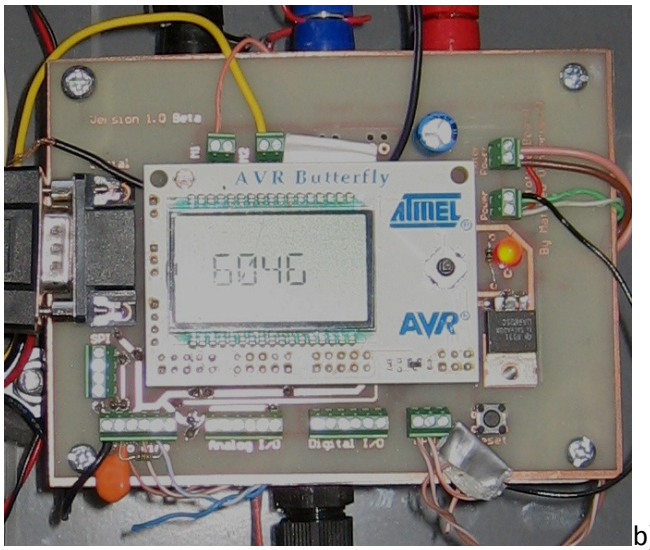


Figure 3.11: Frequency change on a QCM with liquid passing pass one side as the liquid temperature was changed from 45 to 3 Degrees.

The effect of humidity was also wanted to be known so a new design was created which used a microcontroller to control the temperature, and also all the required equipment was placed within a metal box which was then kept stable at a temperature set point.

A Peltier element when current is applied moves heat energy from one side of the device to the other and allows a temperature differential of around 20 °C from the side to side. Depending on the current direction applied to the Peltier one side will warm while another side will cool or vice versa. If one side is kept at ambient temperature this enables the temperature to be controlled to anywhere between ±20 °C ambient temperature. For example if the temperature was 25 °C then the Peltier could control between 5 and 45 °C. The temperature control was performed using a MAX1978 chip from Maxim. This chip has a built in switch mode power controller which can output up to ±3 Amps at ±4 Volts and also has a PID loop which is configured via resistors and capacitors with the input to the loop being the thermistor sensor that was on the metal. The PID loop is tuned using the surface mount resistors and capacitors on this board. This is also the problem of how to measure the temperature easily and so it was decided not to use the built in PID circuit and instead control the circuit via a microcontroller with feedback to a PC.

a) 

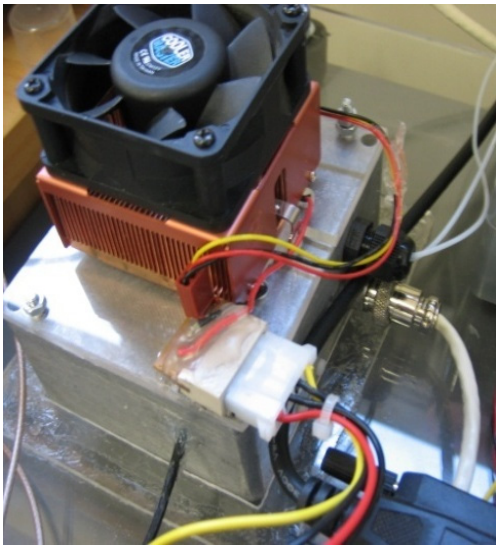
b) 

Figure 3.12: a) Temperature Control Board b) Temperature Controlled box

The microcontroller chosen was the AVR169 which features, an ADC that can sample on eight different ports, a DAC output, two PWM controllers, SPI port, and serial port. The AVR169 has a development board called AVR Butterfly. This was used as it has a LCD screen and several other useful onboard peripherals. Another board was made to interface the AVR Butterfly, which had a serial plug, regulator, dual H-Bridge (for fan motor control), and plenty of wire connectors to enable ease of wiring. The microcontroller's DAC is connected to the MAX1978 Peltier controller and set in control set point input with the control algorithm inside the microcontroller. The microcontroller also interfaced to a SPI precision Temperature and Humidity sensor (manufactured by Sensirion, SHT71). The temperature and humidity sensor was placed inside the temperature controller metal box beside the fan (see Figure 3.14 for a picture of inside the box). The Peltier is attached to one side of the metal box with a heat sink on the other so the metal box is a closed system with the air inside being kept very stable. The temperature was controlled using a digital PID loop within the microcontroller using the temperature as the feedback and digital set point with the controller outputting to the MAX1978 which then in turn controls the Peltier device. This was tuned by inputting a set point into the controller and measuring the set response. To enable the plotting of

temperature and humidity data a program was developed in C# (see Figure 3.13) to interface to the microcontroller. This program reads the XML data sent from the microcontroller and also can be used to set the PID control values and also the temperature set point or even turn the controller on or off. It also saves any runs including the PID and temperature set point.

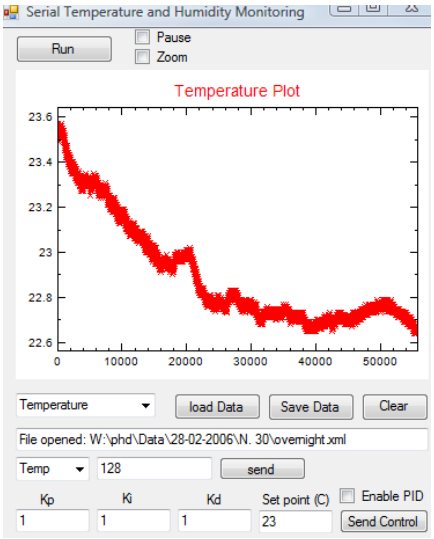


Figure 3.13: Temperature and Humidity monitoring software with control turned off.

To stabilise the incoming liquid temperature for in-liquid operation, a coil of stainless piping is used. The humidity on the air side of the QCM is taken to be the same as the air inside the box and is stabilised by the fan mixing the air within the box. Silica gel crystals have also been used to reduce the humidity in the box and to bring the humidity down to very close to zero. Since this is a closed system it will not be affected by outside influences as long at the outside temperature stays within 20 degrees of the temperature set point.

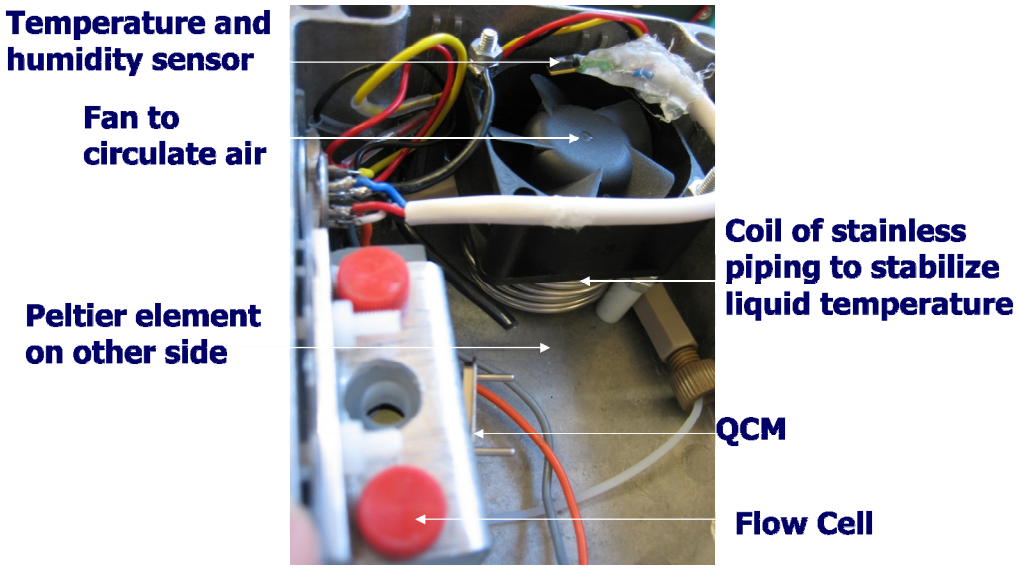


Figure 3.14: Inside temperature controlled box.

The temperature controller can set the temperature and monitor as it brings the temperature to steady state. The controller has a stability of 0.05°C (see Figure 3.15b) which is very close to

the limit of the temperature sensor. It takes around 30min to reach steady state as is shown in Figure 3.15a.

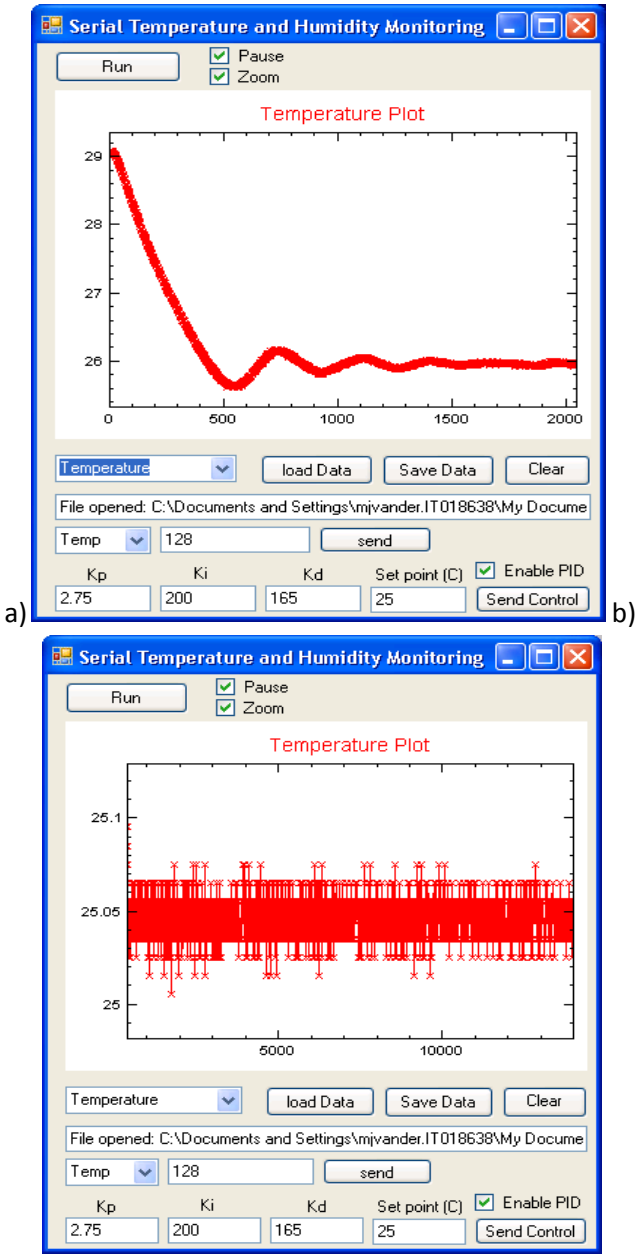


Figure 3.15: Temperature control and monitoring software. (a) PID loop operating on start-up. (b) Temperature stabilised.

3.4 INITIAL BOND RUPTURE EXPERIMENTS

To verify the Bond Rupture technique before the equipment was developed the frequency was measured at a low voltage using the network analyser then driven for 10s at a voltage using a frequency generator. The process was then repeated after waiting around one minute for the QCM to stabilise with the frequency being measured at the low voltage and then the driven voltage changed to the next step up and driven for 10s. Figure 3.16 shows the results from performing this experiment after putting Streptavidin beads onto the surface of a biotin coated QCM for 5hrs and then drying the QCM. Figure 3.17 is the same experiment but the

change measured at the third harmonic frequency. The labels on Figure 3.17 and Figure 3.18 N48 to N52 are labels used to identify the QCM that was used. The results found that Bond Rupture seems to be occurring at between 3 and 10 Volts. It can also be seen that the frequency changes significantly more at the third harmonic, and this may suggest that it is a better place to measure the frequency change.

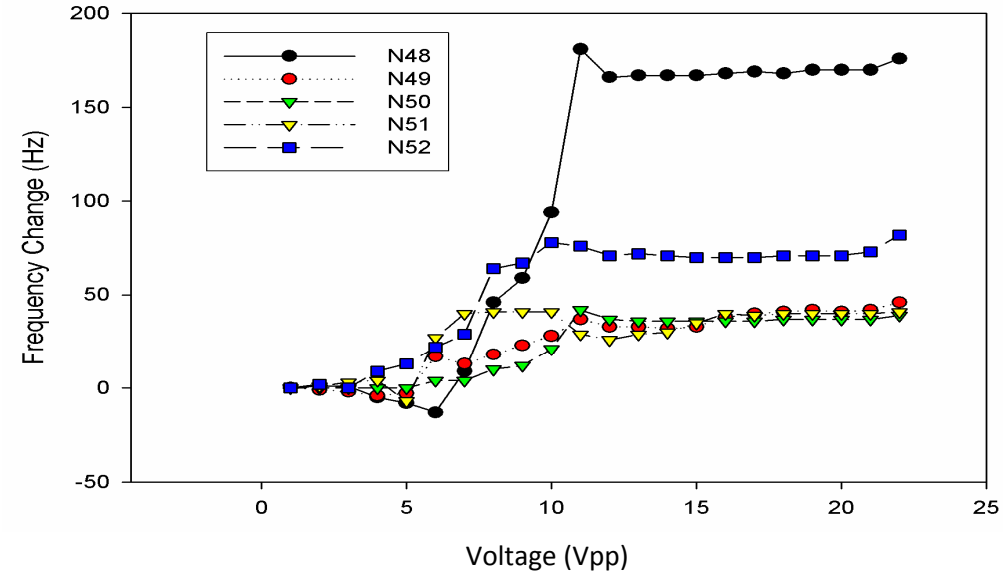


Figure 3.16: Plot of the resonant frequency (~10MHz) change versus voltage applied after the voltage is applied for 10s each sample using Streptavidin coated beads placed introduced onto a biotin coated QCM for 5hrs.

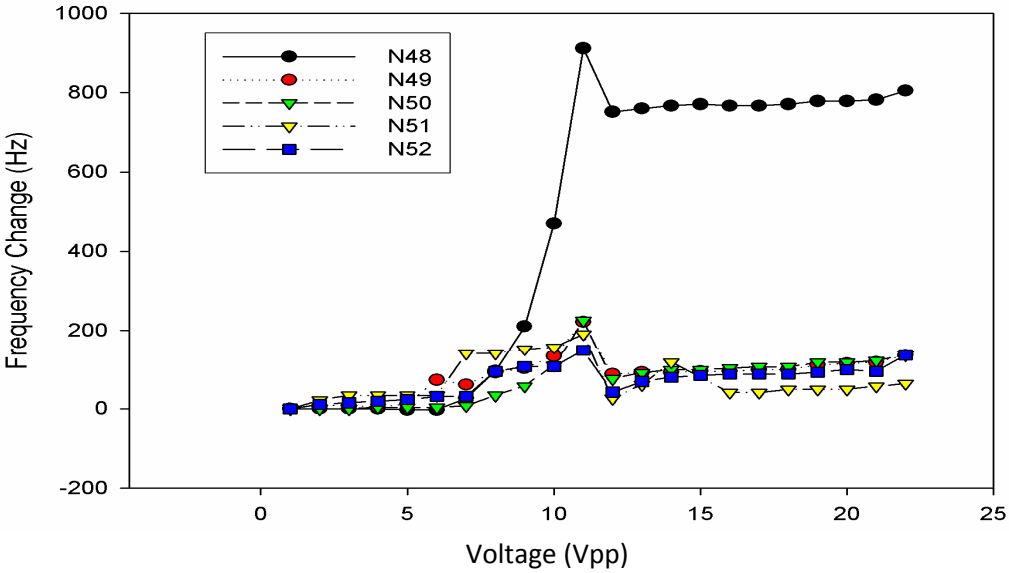


Figure 3.17: Plot of the third harmonic resonant frequency (at ~30MHz) change versus voltage applied after a voltage is applied to the QCM for 10s each sample using Streptavidin coated beads placed introduced onto a biotin coated QCM for 5hrs.

These results suggest that Bond Rupture can be induced and that frequency change can be used to measure Bond Rupture. There are a number of disadvantages with this method with the most obvious being it is very slow! This method required waiting for the frequency to

stabilise after driving with the high voltage – around one minute was chosen to ensure that the frequency has returned to normal. The resonant frequency measurement accuracy of this method is limited and requires significant improvement. Also the time resolution is limited so as not to take too long. From these results it can be seen that significant improvement in the measurement equipment and method is required for this to be the diagnostic equipment required. The next section outlines the requirements and ideas for a Bond Rupture system.

3.5 IMPLEMENTING BOND RUPTURE

There are a number of different electronic methods that could be used to detect Bond Rupture using a QCM. Bond rupture requires the QCM to be driven with an increasing voltage at its resonant frequency. At higher amplitudes the bonds will rupture off the surface and cause a frequency change to occur. Since the frequency will change as the voltage changes the frequency must be followed with some accuracy to ensure the amplitude also continues to increase. If the frequency tracking is not accurate the output movement of the surface of the QCM will decrease causing a reduction in Bond Rupture efficiency. It has been decided that the network analyser in its present form is not fast enough to track the frequency, and so to detect the changes due to Bond Rupture. It would also be required to monitor the third harmonic for noise, and thus this would be another requirement of the equipment which cannot be met by the network analyser. No off the shelf equipment has been found as yet that can perform the frequency following, as well as generating the amplitude range required for Bond Rupture. It would also be useful to experiment with further new ideas and methods with our equipment as little is known about the Bond Rupture mechanism and there are many new ideas that it could be tried. It was decided from these specifications that some sort of custom system would be required to perform the Bond Rupture experiments.

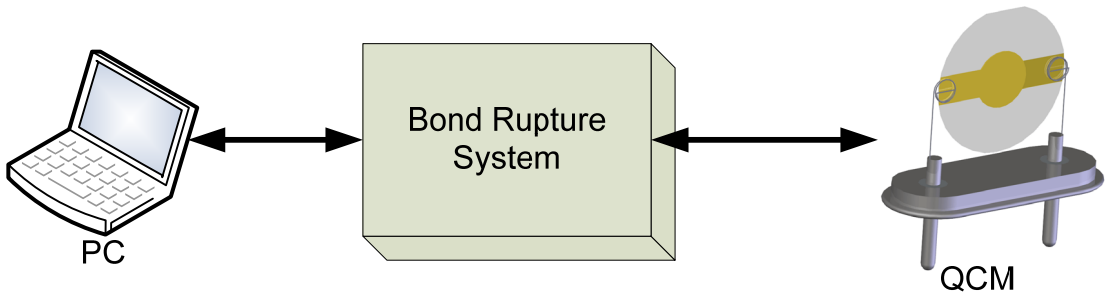


Figure 3.18: Black box QCM Bond Rupture system.

Among the requirements of this system is the need to be able to perform multiple experiments within the same system with little or no hardware changes between experiments. This would include experiments such as performing a frequency scan (like a network analyser), to the dissipation type measurement, and also a full Bond Rupture scan. Another requirement is the ability to be able to control the system with a PC and have a user friendly Graphical User Interface (GUI) with controls and graphing ability. This would need to be like the black box Bond Rupture system seen in Figure 3.18, simple to setup and use with only a PC and the Bond Rupture system required to carry out the experiments on the QCM. A further specification would be the requirement to output a voltage at a high enough level to induce Bond Rupture at the frequency required. Also to be able to measure the third harmonic ‘noise’ using the

same device, so operation up to around 32 MHz would be required. This system would also be required to support QCMs of different frequencies with little or no hardware changes.

Due to the relatively high operating frequency and need for this system to be able to work at multiple frequencies, the options are limited significantly. There is the option of using an analogue multiplier with a synthesised frequency to bring the frequency down to a more suitable operation range for use with a low speed ADC. This method can be fairly complex and depends on an accurate synthesised signal that can be controlled accurately and adequate filtering. Another method that was investigated was using a high speed ADC and high speed DAC and performing the down sampling in the digital domain. Both these methods have their advantages and disadvantages with the digital method usually more expensive due to the high specification components required, but also having a much more predicable response, as all the processing can be done in the digital domain. The digital method has the advantage of being able to operate over a large frequency range and different bandwidths without having to change the analogue frequency generator so enabling multiple frequencies to be measured. This enables this system to work with different frequency QCMs. In the end both methods could have worked but a path was required to be chosen and the digital path was a much more attractive from a features and future development point of view.

3.6 DESIGN SPECIFICATIONS

The initial design specifications for the digital method described in the previous section is as follows. The first specification required was the ADC and DAC requirements. The ADC would be required to receive signals from 1-32MHz and so the sample rate of the ADC would need to be around ~65Msps. The ADC would also have to be high precision to enable the capture of the small signal Bond Rupture ‘noise’. To process the signal from the ADC a Digital Receive Processor (DRP) would be required. The DRP would have to be also able to output a quadrature signal. The DRP would convert the ADC digital signal to the frequency and band width required. This would need to be able to take the ADC acquisition rate and convert it down to at least 100ksps. The DAC would need to be able to generate at least the 32 MHz but higher would be better to remove any aliasing affects and decrease the filter requirements – so around 100Msps. To enable the generation of a high frequency waveform, a Direct Digital Synthesiser (DDS) integrated circuit would be required, which comes with an integrated DAC.

Further requirements are required to ensure that the output is at the correct amplitude and that the input is scaled to a range that maximises the ADC’s dynamic range. This would be achieved with output amplifiers taken from the DAC and input amplifiers fed to the ADC. It was decided that the output amplifiers would be required to output at around 1 Watt of power at the resonant frequency of the QCM or a voltage range of around 0.1 to 20 Volts peak to peak, while also maximising the dynamic range of the DAC. These amplifiers would have to be able to operate at the specified frequency range required by the QCM. The amplifiers would also have to be reasonably low noise to reduce output noise fed back to the ADC. The input amplifier is also required to have a low noise characteristic while also ensuring the voltage range fed to the ADC is high enough not to have an adverse affect to the dynamic range and also ensuring that the signal will not cause an overflow condition on the ADC.

It is also important to match the impedance into the QCM so that the optimum power transfer occurs with minimal loss from impedance mismatch. Also the method for measuring the QCM signal needs to be further examined and experimented with to find the optimal method.

Another important requirement is the embedded processor and its communication to a PC. A processor is required to enable real time processing and storage of the gathered data for later transfer back to the PC. It is not known what exact specifications would be required for the processor but it would want to be powerful enough to handle any advanced future implementation ideas. Some sort of high speed Digital Signal Processor (DSP) would be adequate to ensure that this device meets any future requirements. The communication with the PC is important so that the data gathered can be transferred to the PC for further processing and plotted. The PC communication should enable the development of Graphical User Interfaces (GUI) for the end user to operate the system with.

4 DIGITAL TRANSCEIVER SYSTEM

4.1 INTRODUCTION

To create a system with all the requirements mentioned in the previous chapter requires a significant amount of Research and Development. Fortunately a head start was given in the form of a similarly developed portable system for the measuring of Nuclear Magnetic Resonance (NMR) developed by Robin Dykstra in conjunction with a New Zealand company Magritek. Two main components were donated for use in this project including a DSP board, and Digital transceiver board. A PC program called prospa developed by Magritek was also used which enabled the simple communication with the DSP and creation of DSP software. It also enabled the development of GUI's and processing and plotting of the captured data. Using this as the base equipment enabled this author to concentrate on the development of the Bond Rupture equipment. NMR is similar to this application as it also looks at very small signals and requires the measurement of resonant frequencies impedance matching of the NMR probes so this makes it an ideal hardware platform to start our Bond Rupture measurements with. This Chapter contains the work carried out adapting the developed system to Bond Rupture application, developing further improvements to the system, and the development of the software and GUI controls that drive the system.

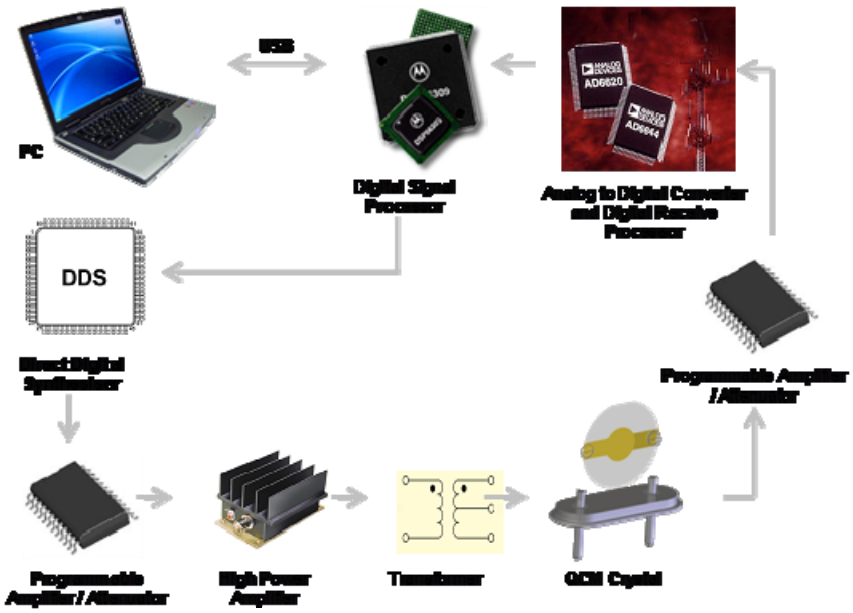


Figure 4.1: Layout of the QCM equipment.

The QCM (Quartz Crystal Microbalance) equipment developed enables the monitoring of the frequency change of a QCM while also varying the output voltage for the purpose of Bond Rupture (BR) excitation. The equipment can also monitor the QCM for BR noise around the 3rd harmonic frequency. The QCM can be driven at a maximum of one Watt using a combination of an Analog Devices Variable Gain Amplifier (VGA) and a Mini-Circuits external amplifier. The measurement of the QCM is performed using an Analog Devices 62.5MHz Analog to Digital Converter (ADC) of which the input is scaled via a VGA enabling a wide Voltage input range. A Digital Receive Processor (DRP) is used to process the ADC data enabling the selection of the

frequency and band-width of interest. See the Figure 4.1 for the graphical layout of the components and see Figure 4.2 for the flow diagram layout of the hardware with the digital and analogue sections separated.

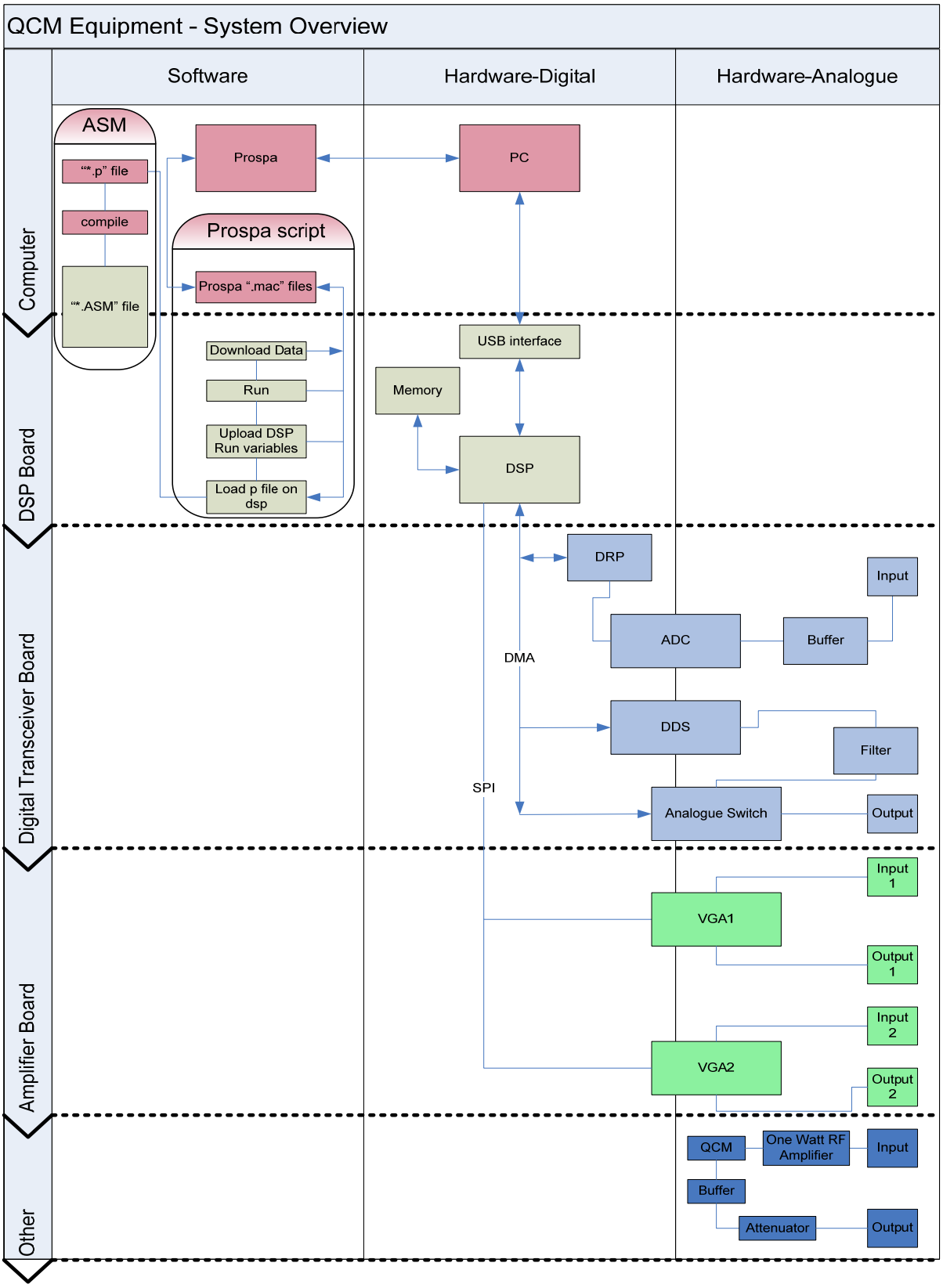


Figure 4.2: System software and hardware component interconnection diagram.

To run an experiment on the QCM equipment the computer runs software called prospa (product website http://www.magritek.com/prospa.html), which loads the DSP (Digital Signal Processor) with compiled DSP file, then runs the DSP program, and acquires the stored data points from the DSP (see the software column in Figure 4.2. This ability to change the DSP software enables multiple different experiments to be run on the same DSP within quick succession of each other by using a different DSP program. For a flow diagram showing the flow of the software and hardware interconnection see Figure 4.2.

Figure 4.2 describes the system as a whole. As can be seen in this figure there are three main categories that make up this system – as labelled at the top of Figure 4.2. First is the software which is made up of: the GUI, the scripting for processing of the data and plotting, and the DSP Assembly language ASM file which is run on the DSP. The next category is the digital hardware which includes the DSP and memory, and also includes the DRP, ADC, DDS, analogue switch, and variable gain amplifiers (VGA). The last category also includes the ADC, DDS, analogue switch as these devices are half digital, half analogue as well as the input buffer for the ADC, filter for the DDS, the analogue parts of the VGA's and the external amplifiers, matching, and buffer electronics.

There are five main components of boards that this system is broken up into. The PC is one, which communicates to the DSP via USB. The DSP board communicates to the digital transceiver board and the amplifier board. The transceiver board is connected to the amplifiers on the amplifier board via smb RF connectors. The amplifier board then connects to the QCM board and external amplifiers via smb connectors. For a detailed diagram see Figure 4.3.

4.2 HARDWARE

There are two main methods used to detect the resonant frequency of the Quartz crystal an oscillator circuit (OC), or network analysis. Both these solutions are options for frequency determination but they do not meet the requirements for Bond Rupture and is not the smart point-of-the-care sensing that is pursued in this device. An integrated digital processing solution or smart sensor system is proposed, as shown in Figure 4.3. It consists mainly of following parts: a digital transmitter that excites the crystal; a digital receiver that measures QCM response; amplification and impedance matching for maximum power transfer to the QCM; and a DSP that communicates with the transmitter and receiver; and is reprogrammable via a PC. A printed circuit board (PCB) is built to accommodate all semiconductor devices and connected to the PC and the QCM sensor. The system is based on the DSP and transceiver board that was created by Robin Dykstra for his PhD thesis (Dykstra 2006) for the application of compact NMR sensing.

There are many advantages of using the proposed method over any previously discussed methods. One advantage is that the DSP software can be replaced in between experiments enabling operation as a network analyser in one setup and the next as a crystal quality measurement device. The ability to reprogram also enables debugging of hardware problems. This also has the advantage of being able to off-load any processing the DSP cannot handle to the PC with an example being a Fast Fourier Transform (FFT) for advanced spectral analysis

with the data captured by the DSP and transferred to the PC for post processing. Another advantage is that with a custom calibration program we can take care of any factors introduced by the analogue components. This is something that can be done in a network analyser system that is impossible to do in an analogue system such as an oscillator. All of the experiments can be done with the same PCB's with only the software needing to be changed.

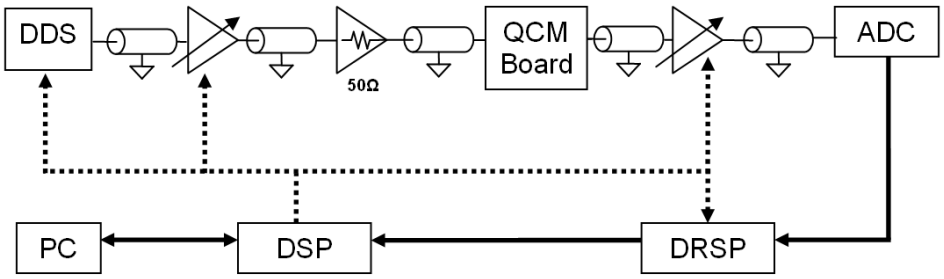


Figure 4.3: QCM smart sensor system. Top left to right: high frequency DDS; output variable gain amplifier (VGA); fixed high power amplifier; QCM impedance matching and buffer board; input VGA; and analogue to digital converter (ADC). Bottom left to right: Personal computer (PC); DSP which controls DDS, and output and input VGA's; and the Digital Receiver Signal Processor (DRP). The Digital Transceiver board contains the DDS, ADC, and DRP.

Though there are many advantages there are also some disadvantages. One would be that the system is quite complex and a significant level of expertise and background knowledge is needed to reprogram it. Also at times software bugs can be very hard to find. The parts are reasonably costly though well under the cost of a network analyser. After considering all pros and cons from literature and recent technology advancements (fast ADC's and DRP's) it was determined that the network analyser method was the best method to follow.

There are multiple boards that make up the QCM equipment as can be seen Figure 4.2. The main boards are; DSP board; digital transceiver board; amplifier board. The DSP board is the main real time processor board, and controller for setting up and gathering data from the other boards. The DSP stores the data from the digital receiver in on-board memory and then sends its data back to the computer. The memory size impacts the number of data points that can be stored per DSP run.

4.2.1 DIGITAL SIGNAL PROCESSOR (DSP)

The most important part of the system is the DSP. The DSP handles controlling all the chips involved including; Direct Digital Synthesis (DDS), Digital Receive Signal Processor (DRP), and both VGA's. It also handles the data capture, processing, and storage of the down-sampled data from the DRP. The DSP board has an external 16k x 24bit memory for storage of data from the DSP. This means that the amount of data points per program run is limited to the amount of memory that is available and if too many points are stored an overflow will occur, causing errors. The DSP is interfaced to the computer via a USB 1.1 connection enabling data transfer speeds of up to 12 Mb/s. This USB interface enables the uploading of programs and data variables to the DSP as well as the ability to download the data captured from the DSP. This DSP board seen in Figure 4.4 was designed by Robin Dykstra for his PhD thesis (Dykstra

2006) and is now sold by the company Magritek (http://www.magritek.com/) which specialise in NMR software and hardware.

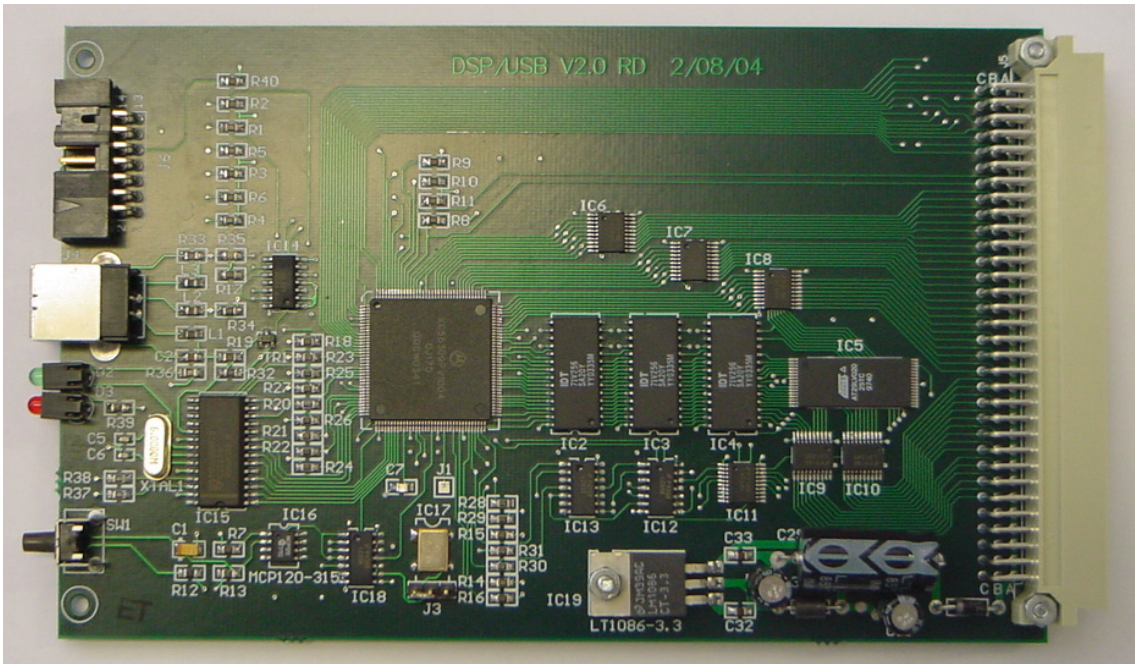


Figure 4.4: Picture of the DSP board developed and sold by Magritek – Dykstra (2006).

The DSP used is the 24bit processor created by Motorola named DSP56309 as seen in Figure 4.5. This uses a high performance pipelined one clock cycle per instruction engine enabling fast processing for digital signal processing applications. The DSP56309 can process up to 100MIPS at 3.3V and also contains a large amount of on-chip memory. The DSP56309 has direct memory access (DMA) and so enables fast communication to external memory. It also has an enhanced synchronous serial interface (ESSI) enabling fast communication to serial peripherals such as amplifiers. This DSP is sufficient for what is needed in this system.

The DSP runs an operating system (OS) created by Dykstra (Dykstra 2006). This OS handles the communication between the DSP and the PC via USB. It enables a compiled pulse program (usually written in assembler) to be uploaded to the DSP. It can then start the program running and sends a signal back to the PC when the program has finished running. It also has another function which enables data to be written or read directly to or from the addresses within the DSPs internal or external memory. The normal process for running a program on the DSP would be to upload the program, write any data needed for the program, run the experiment and wait until the end, and then download the data from the experiment.

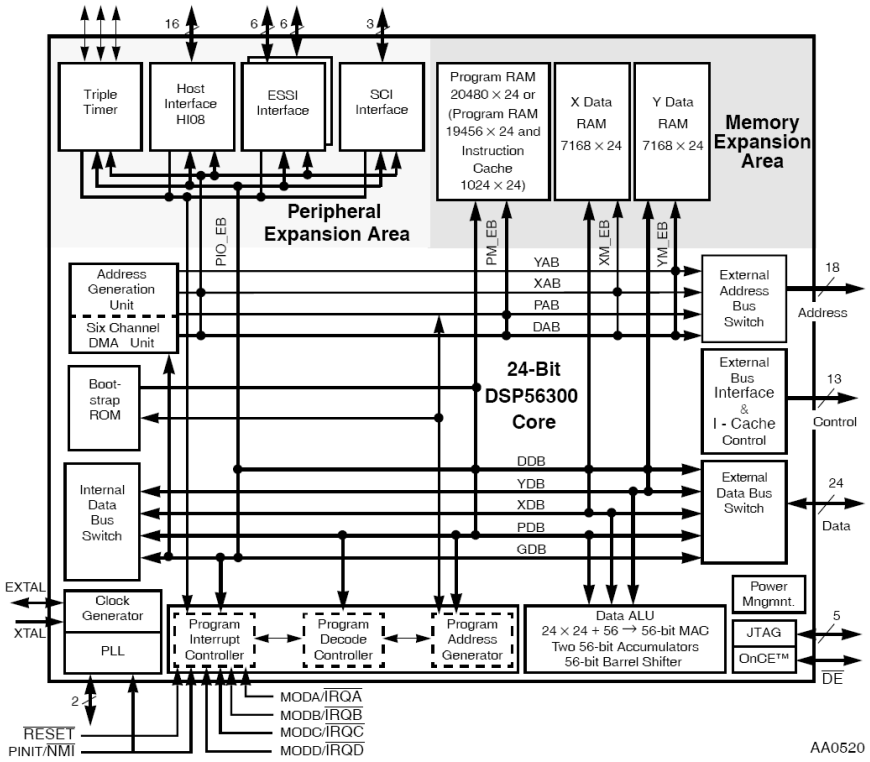


Figure 4.5: DSP56309 Block Diagram (Motorola 1998).

Although the DSP to PC USB connection has been introduced one vital component that has not yet been discussed in detail is prospa. Prospa is a mathematical software package that is specifically tailored by Magritek for scientists working in the field of NMR – basically a simpler, faster, customised version of Matlab. Prospa has multiple features including: 1D, 2D, and 3D plotting; simple GUI development; huge amount of built in post processing macros such as FFT; most common mathematical functions; and support for most matrix operations. Prospa also has specialty drivers developed to communicate to the DSP board via USB. The DSP is controlled using a DLL plug-in in prospa which enables control of the commands mentioned in the previous paragraph and a list of the functions used from prospa as can be seen in Table 4.1.

Macro Name	Description
dspsrec	read an s-record and store in a vector
dspwrite	write data to the DSP
dspwritepar	write pulse program parameters to the DSP
dsprunpp	run a pulse program in the DSP
dspabort	send abort signal to DSP
dspread	read data from the DSP
dspreadpp	read pulse program from file and store in a vector

Table 4.1: DSP DLL module listed from prospa.

There are a series of commands needed to run a pulse program from prospa – these are shown in Table 4.2. The first step is to read the pulse program from the hard drive and save it as a variable – variable pp. This variable is then written to the program memory or p-space at the memory position where programs are run from. Next any parameters that are needed for

the running of the program (such as the frequency range to scan over) are written to a place in memory (x or y memory) predetermined by the pulse program. Now that all the setup has been performed the DSP pulse program can be run, and once run the prospa macro waits until the function returns a value before continuing. The only thing to do once an experiment has been run is to read the data (from a predetermined memory location) and perform any post processing necessary. Once the pulse program has been loaded into memory the program can be run multiple times without having to reload it. So the input variables can be updated and the pulse program can be run sequential in a loop involving multiple experiments.

```
pp = dspsrec("FFT_Cap.p") # read pulse program
dspwrite("p",0x1000,pp) # read data points
dspwritepar("x",0x00,p) # Send parameters to DSP
r = dsprunpp() # Run pulse program
D = dspread("y",0x10000,nrDataPnts) # read data points
```

Table 4.2: Typical sequence of commands to run a program and download the data in a prospa script.

Before a prospa script to control the DSP can be run a pulse program must be written and compiled. This pulse program is usually written in Motorola assembler code (as opposed to a high level language) to enable an accurate idea of how long different processing tasks are taking to complete. Though writing in assembler enables a significant amount of speed and code optimisation, errors can occur at a much higher rate than using a high level language. One of the deciding factors to using assembler was the problems with working around the operating system running on the DSP, to enable C-complier definitions to define areas not used. This is much simpler in assembler at the expense of some of the more advanced memory and stack features on the DSP. For further information see Dykstra (Dykstra 2006).

To write in assembler the ConTEXT editor was used with a 56300 assembler highlighter created by Rob Ward as colour highlighting makes writing much easier. The complier used was the Motorola “ASM56300.exe” to create the linking “*.cld” file and then the “srec.exe” to create a “*.p” file for running on the DSP. This can be seen in Table 4.3, with “Filename.asm” as the input and “Filename.p” as the output. For further information on this compiler see the “Assembler reference manual” from Motorola.

```
asm56300.exe -a -b -l -z Filename.asm
srec.exe Filename.cld
```

Table 4.3: Typical sequence of complier options to compile an ‘asm’ file into a ‘p’ file.

The DSP Board plugs into a 96pin back plane and though this the DSP can communicate to the other boards in the system. To communicate with the digital transceiver the DSPs DMA bus is used which enables addresses to be mapped to x or y memory within the DSP. This is done using address attributes AA2, and AA3 in conjunction with the WR (write) and RD (read) control lines. When the corresponding address attribute line is low and the WR or RD is low then a write or read will occur. The DSP uses DMA to setup and read data to and from the digital transceiver board. Therefore all that is required for the DSP to setup chips is a memory write or read to get data from the bus.

For the DSP to communicate to the amplifiers, the enhanced synchronous serial interface (ESSI) is used. ESSI when used for write-only operation is made up of only 3 outputs signals. The first is a clock signal which is generated internally in the DSP and this clock is fed to the device to be controlled (in this case an amplifier). The next is the data line which sends the data to the device. Finally is the chip select signal which enables only the chip that is required so that multiple devices can use the same clock and data lines.

This DSP is a highly versatile device which is will enable support for the all the functions that maybe required including most complex.

4.2.2 DIGITAL TRANSCEIVER

Though the DSP is a powerful base to the system it still cannot interact with the real world. This is where the digital transceiver comes in – see Figure 4.6. The digital transceiver was created by Dykstra (Dykstra 2006) and is marketed by Magritek as part of their portable NMR systems. This is well suited for the needs of QCM sensing as it operates around the same frequency ranges that are of interest for QCM. The digital transceiver enables the generation of high frequency digital waveforms for the use of exciting the QCM and inducing Bond Rupture. Also the digital transceiver enables the capture of high speed waveforms using a high speed ADC processed in real time using a DRP - which enables the capture of narrow band frequency from a wide band ADC input.

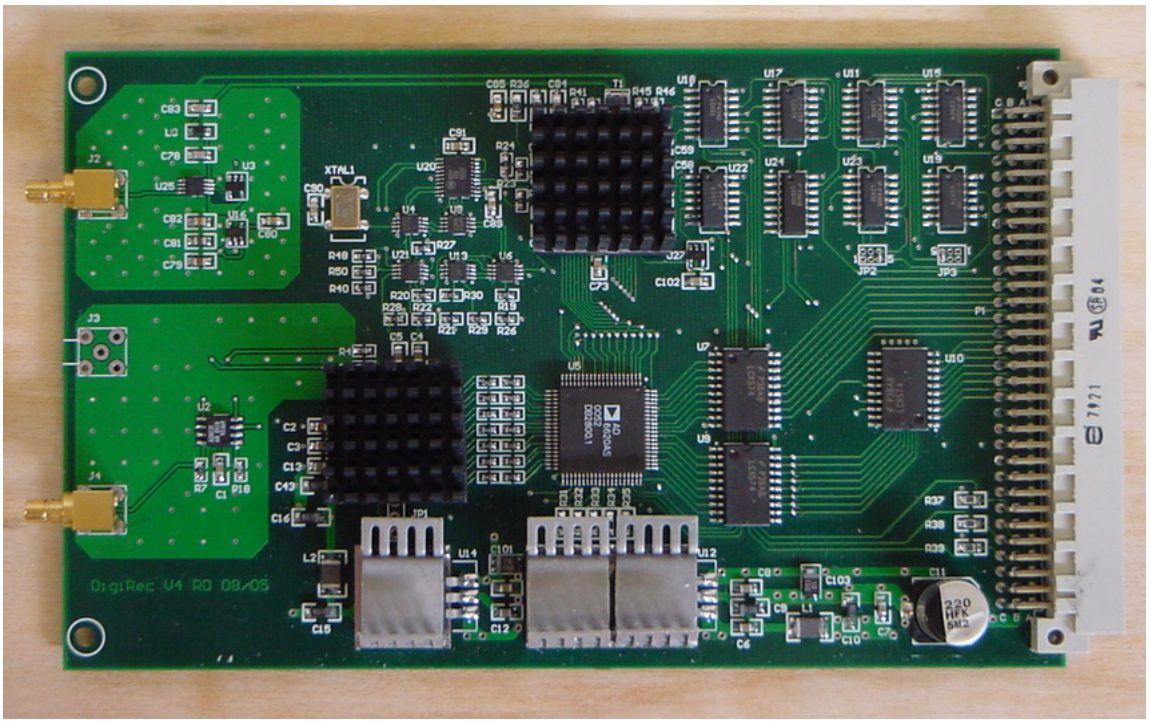


Figure 4.6: Picture of the digital transceiver – Dykstra (2006).

The simplified hardware diagram of the digital transceiver system is shown in Figure 4.7. Though some audio based systems can connect the ADC directly to a DSP and process the input signal as well as generate the output signal with high speed systems, such as a DSP, they would not have time to process the data in real time as well as perform other required tasks.

This is why the DDS and the DRP are required to off load the processing from the DSP. The DDS can generate a single tone output at high frequencies which is specified by simply updating a register from the DDS. The DRP is used to down sample the high speed ADC signal to a band width that is required at the frequency of interest. The data generated is then fed to the DSP via the communication bus.

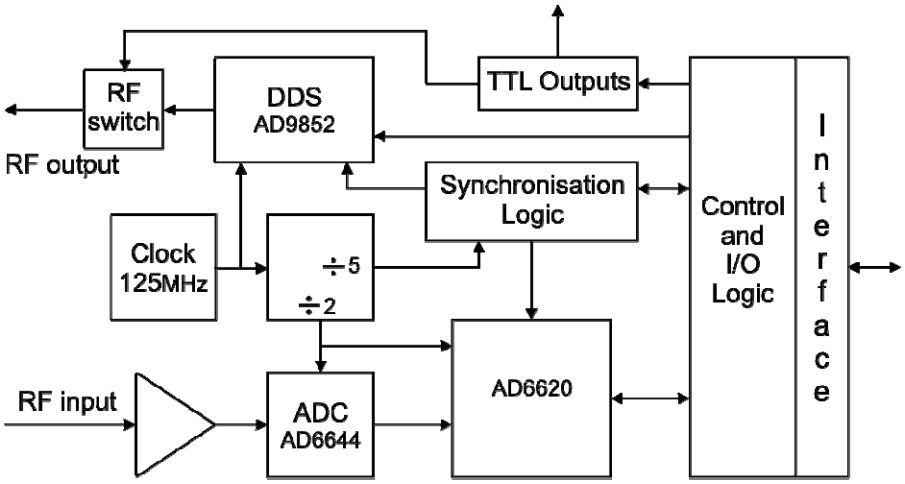


Figure 4.7: The hardware overview of the transceiver board – Dykstra (2006).

The Digital transceiver device requires a number of components to synchronise the output frequency generation, turn the output on and off with an RF switch and provide the correct clock frequency to each of the components. There is also a significant amount of latching circuitry required to provide the interface between the DSP and each of the controllable components on the digital transceiver board.

4.2.2.1 TRANSMITTER

The transmitter side of the transceiver board consists of a DDS for waveform generation and scaling, an analogue output switch, and an analogue filter – see Figure 4.8. This enables the excitation of the QCM with the ability to independently switch the output on and off enabling impulse excitation using narrow pulse output and also enabling the DDS to run independently from the output. This is useful between runs to keep the phase constant. The analogue filter is necessary to filter out any high order harmonics generated by the DDS output DAC.

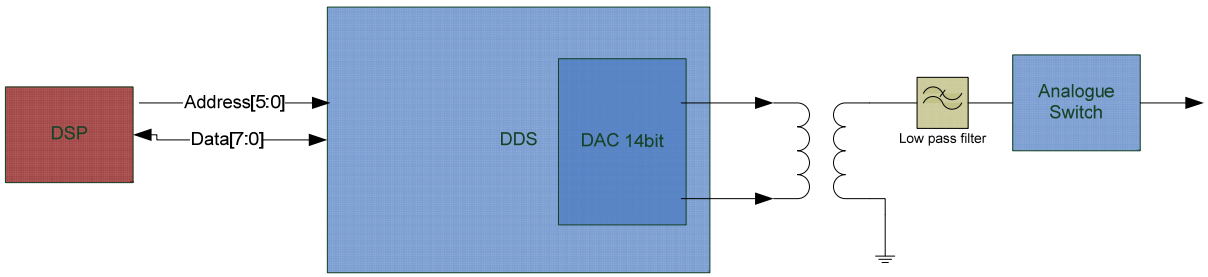


Figure 4.8: Diagram of transmitter.

The DDS used was an Analog Devices AD9852 DDS – see Figure 4.9 for the functional diagram. This DDS has many features useful for QCM excitation as well as many redundant modes. One

of the useful features is the high operating frequency sinusoidal wave that can be generated. The DDS used can be clocked at up to 300 MHz, but on the board created it is clocked at 125 MHz enabling the generation of output signals of any frequency up to 62.5 MHz. The phase and amplitude can also be changed while outputting the signal. The DDS can also run in ramping amplitude mode which can be useful when inducing Bond Rupture. The DDS also has other modes which were not relevant for the QCM Bond rupture as this is a multipurpose chip.

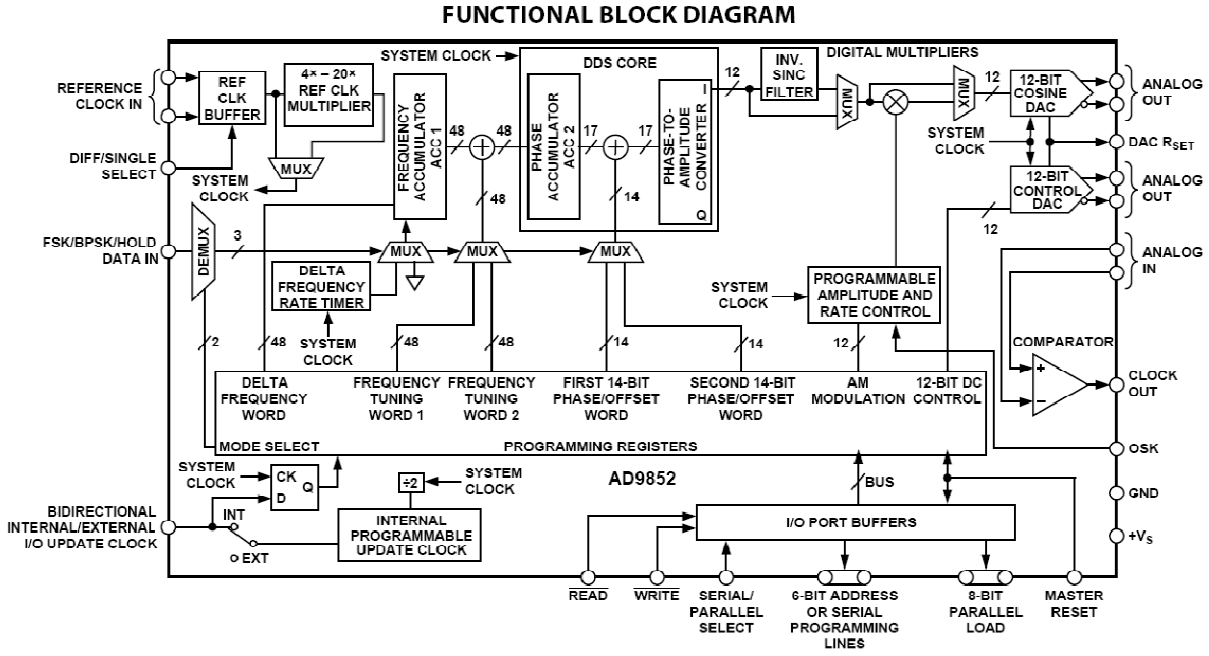


Figure 4.9: Functional block diagram for the 9852 Analog Devices chip from datasheet.

The DDS basically consists of a 48 bit frequency accumulator to which a Frequency Tuning Word (FTW) is added to each clock cycle. This is then offset by adding the phase word ($\emptyset_x$). This output is then fed to a Cosine lookup table which determines the output value. This is then multiplied by a 12 bit amplitude variable and fed to the DAC. To calculate the frequency tuning word for a desired output frequency Equation (4.1) is used.

$$FTW = \frac{Desired\ Output\ Frequency \times 2^{48}}{SYSCLK} \tag{4.1}$$

SYSCLK is the operating clock frequency (125 MHz), and the desired output frequency is anywhere from DC to SYSCLK/2. Another factor to consider is the frequency resolution. The frequency resolution can be calculated using Equation (4.2).

$$F_{res} = \frac{SYSCLK}{\frac{2^{48}}{2}} = \frac{125MHz}{2^{47}} = \sim 0.9\mu Hz \tag{4.2}$$

With this formula $2^{48}$is divided by 2 as only half of the values are used to satisfy Nyquist frequency (the sampling frequency must be 2 times the output frequency for DAC). This gives very high accuracy frequency determination.

To update the DDS via the DSP the data is written to 6 bits of the address and 8 bits from the data ports. This is using the parallel I/O operation mode on the DDS. To perform a write to the

48 bit variable for frequency tuning word the write must be performed over six successive 8 bit writes, at the end of the last write the whole word is transferred from temporary memory to the active frequency tuning word therefore updating the DDS frequency. The procedure to update the phase and amplitude is same except for the number of writes needed for the phase adjust, and amplitude multiplier, and any other control bytes. More information can be found in the AD9852 datasheet from Analog Devices.

The procedure to control the digital output switch is basically the same as setting up the DDS. All that is needed is a write to the correct location. An 8 bit write to this location also updates the reset switch and sync switch on the DDS and DRP.

The filter design is not overly critical as the distance in frequency between the operating and clock frequency is a long distance. The working frequency is at around 10 MHz to drive the QCM and the DDS is running at 125 MHz.

4.2.2.2 RECEIVER

The receiver receives a digital input from the ADC and shifts the frequency band of interest to close to DC - by multiplying the input signal by a sinusoidal wave at the frequency of interest. The signal is also digitally filtered and then down-sampled to reduce the amount of data that needs to be processed via the DSP. This allows the band of the signal being observed to be switched from the fundamental frequency of the QCM to the third harmonic or anywhere in-between by simply reprogramming the DRP via the DSP. Limiting the band-width of the frequency reduces the effects of signal noise and reduces the amount of data that the DSP is required to process, with the added advantage of a digital gain from the effective averaging. Depending on the amount of down sampling the 14bit ADC could be effectively equivalent to 24 bits of data.

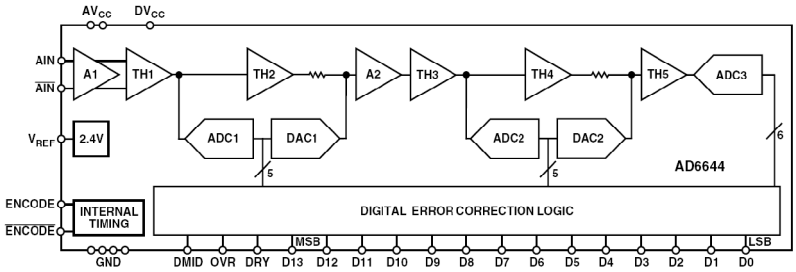


Figure 4.10: Analog Devices ADC AD6644.

The receiver is built using a 14bit ADC created by Analog Devices designated the AD6644 (see Figure 4.10) and the DRP, or also called DRSP (Digital Receive Signal Processor), created by Analog Devices designated the AD6620 (see Figure 4.11). The DRP enables a small bandwidth segment to be selected from a high bandwidth signal with the added advantage of the noise floor of the signal being analysed being reduced. This technology is used in many applications such as communications.

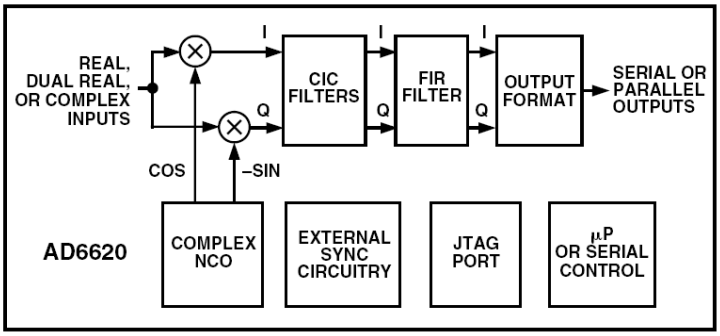


Figure 4.11: Functional diagram of the Analog Devices AD6620 Digital Receive Processor.

The DRP, works by firstly sampling the signal of interest with a high speed ADC and then once in the digital domain multiplying the signal by a DDS generated cosine wave at the frequency of interest for in-phase (real) component and negative sine for the quadrature (imaginary) component. The real and imaginary signals contain a modulated signal with a low frequency component (or, DDS frequency less the signal frequency) as well as a high frequency component (or, DDS frequency plus the signal frequency). Since the low frequency signal of interest is around DC an internal digital low pass filter is used to get the signal of interest, as shown in Figure 4.12. The signal is then digitally down-sampled - via the DRP - to get the bandwidth/sample rate required. The real (in-phase) and imaginary (quadrature) components of the signal are used to generate the magnitude and phase of the signal around DC with a bandwidth specified by the digital filtering and down-sample amount.

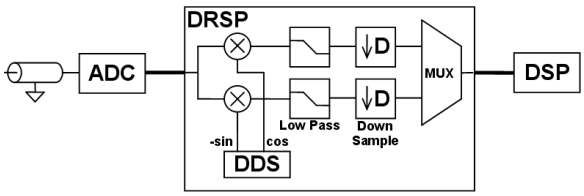


Figure 4.12: Simplified Receiver diagram.

The ADC used can run at a maximum of 65 MHz but is clocked at 62.5 MHz on the transceiver board. The data coming in is therefore at 62.5 MSPS which at 14 bit of data is 875 Mbps of data coming in to process. This is a huge amount of data and this is why a DRP is needed as the data rate coming out of the DRP can be reduced to as little as 122kHz at 16bit with 2 output channels or 3.9 Mbps of data – much more manageable.

The DRP uses cascade-integrate-comb filters or CIC for short – see Figure 4.13. CIC filters are filters that make very efficient use of hardware as they require no multiplications such as normal filters but are only made up of adders. The DRP contains two CIC decimation filters in series with the first being made up of two CIC elements with a decimation rate of between 2 of 16. The second one is made up of five elements with a decimation rate of between 2 and 32. This gives a minimum and maximum decimation rate as 4 and 512 respectively. This effectively means that at a sample rate of 62.5 the bandwidth can be varied between 15.625 MHz and 122 kHz. When the decimation rates are varied the amount of aliasing (as a result from the filters not filtering out all the elements higher than the frequency of interest when down sampling) is also changed with the aliasing increasing as the band width is larger.

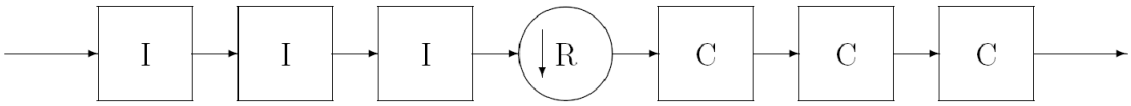


Figure 4.13: Three stage decimating CIC filter taken from (Donadio 2000).

To read and write to the DRP the address and data output lines from the DSP are used. The DRP is addressed using the address line bits 6 and 7. When the correct value is selected on the address line this selects the chip select line enabling it for read and write operations. To write to the DRP chip the lower 8 bits of the data line are used as well as the 3 lowest bits on the address line. Because only three bits are available for addressing the DRP an address offset mode is used – see Table 4.4 for registers. Address offset mode works by writing the lower 8 bits of the address to the low address register and then the next high 2 bits of the register get written to the address mode register as well as setting the mode to write increment. The block sizes are at 36 bits so a series of 5 writes are needed to perform a full write. To perform a write the procedure is to start at the high bytes and when the lowest byte is written the full 36 bit word will be transferred to the address specified by the address registers.

Table XI. External Interface Registers

A[2:0]	Name	Comment
000	Data Register 0 (DR0)	D[7:0]
001	Data Register 1 (DR1)	D[15:8]
010	Data Register 2 (DR2)	D[23:16]
011	Data Register 3 (DR3)	D[31:24]
100	Data Register 4 (DR4)	D[35:32]
101	Reserved	Reserved
110	Low Address Register (LAR)	A[7:0]
111	Address Mode Register (AMR)	1-0: A[9:8] 5-2: Reserved 6: Read Increment 7: Write Increment

Table 4.4: External interface registers for writing to the DRP.

To read the processed data back from the DRP 16 bits from the data bus are connected to the DSP. An interrupt is also outputted from the DRP to the DSP to signal when the data is ready. When the interrupt signals the DSP that the data is ready the DSP must perform two reads to read the real and imaginary set of data.

4.2.3 AMPLIFIERS

To induce Bond Rupture we need to be able to provide the QCM with enough power to rupture the bonds. This power needs to be at the frequency of the QCM operation. The DDS power is very limited with the standard output being around 0 dBm (which is 0.220 V_{rms} into 50 Ohms). To increase the power provided to the QCM an amplifier is used. An ideal amplifier increases the output power by a gain amount linearly over the specified frequency range. Unfortunately amplifiers usually introduce a harmonic distortion so their output needs some sort of filtering to minimise the harmonic interference.

The amount of power that needs to be supplied to the QCM to induce Bond Rupture is not known exactly but we know that Cooper drove the QCM's at around 7 Volts peak to peak to induce Bond Rupture. A QCM's resistance can vary from 10 ohms to 150 ohms giving a range in driving power from 0.04 Watts to 0.6 Watts. A conservative estimate of around 1 Watt or 30 dBm of power was set as the power needed for Bond Rupture.

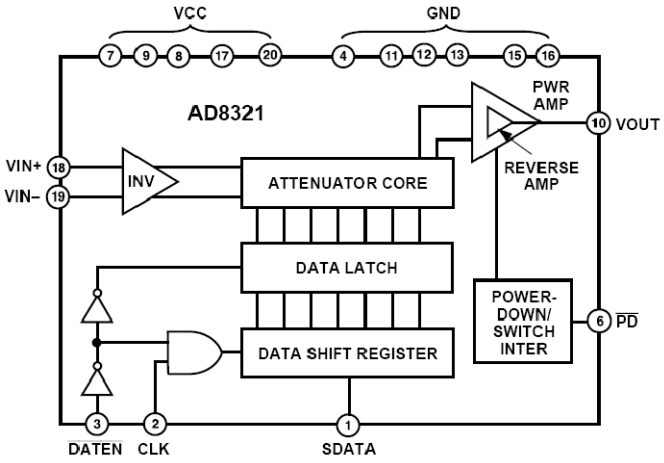


Figure 4.14: Variable gain amplifier AD8321 functional diagram from Analog Devices.

To perform Bond Rupture it was needed to ramp up from very small power ranges to very large amounts of driving power. To enable this range a variable gain amplifier or VGA was used. The VGA used was the Analog Devices AD8321 as seen in Figure 4.14. The VGA used is a programmable gain line driver with a 53 dB gain range and has 72 linear in dB gain increments over this range.

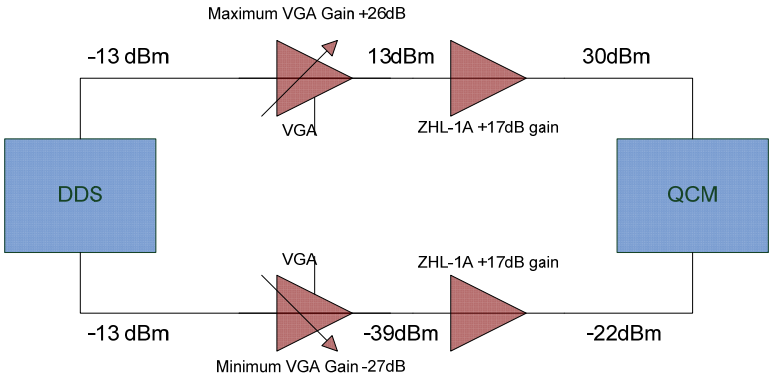


Figure 4.15: Diagram showing the maximum (top) and minimum (bottom) gain paths of the amplifiers output.

The VGA has a maximum power output of 19.5 dBm and so is not powerful enough by its own to generate Bond Rupture and another amplifier is required. To enable close to the 30dBm output the Mini-Circuits ZHL-1A amplifier was used. This amplifier can output a maximum of 30dBm at 10 MHz and has a fixed gain of 17 dB. This means the maximum output of the VGA needs to be 13dBm so as not to overload the amplifier. The output range on the VGA is between -27 dB and 26 dB must be attenuated by 13 dB to enable the amplifier not to overload at full gain output of 0dBm on the DDS. To protect the amplifier from overload the

DDS output is attenuated by 13 dB. This means the highest output with the VGA on maximum is 30dBm and the minimum output with the VGA on lowest is -22dBm giving a large range to use. A diagram showing the gain paths is shown in Figure 4.15.

The output of the VGA is limited to 72 different gain values which is a gain change of 0.71 dB per step. This is not precise enough to enable linear voltage stepping for accurate Bond Rupture ramping. To enable more accurate gain control the DDS scaling unit is used. The DDS scalier multiplies the generated wave with a fixed scale value, producing a scaled output to the internal 12 bit DAC. This scale value can be changed at the expense of the signal to noise value at the DAC. It was decided to use a minimum of a 10 bit value as a scale value, and so leaving 2 bits to use for scaling. This enables the output value to be scaled down by 4 and so give a range of scale values between 1024 and 4096 which is 3072 scale points. This converted to dB provides a range of 12 dB and an output resolution of 0.004 dB. So when the DDS scaling and the VGA range is combined the output gain can be controlled to a very high accuracy.

Unfortunately the VGA has quite a high error in the output gain values, so to get a high level of accuracy, calibration of the output is required. This will be covered in more detail in the software section.

Due to the large gain range that the QCM can be driven over, a VGA amplifier is required as a pre-amplifier before the signal is sent to the ADC. The same VGA is used as the one for the output stage as a similar gain range is required on the input to the ADC. The ADC can handle a maximum input of 1 Vpp, which is around 4.8 dBm. This needs to be kept as close to this voltage as possible to enable best possible use of the dynamic range. We are also limited to the maximum input power into the VGA, which is limited also to 1 Vpp and so an attenuator needs to be placed in front of the VGA to limit the input power. To decrease the input power enough a 26dB attenuator was needed. Unfortunately this means the full gain range of the VGA cannot be taken advantage of. But there is still an amplification of 26 dB which still increases the dynamic gain significantly. A diagram showing the flow of the input to the ADC via the amplifiers is seen in Figure 4.16.

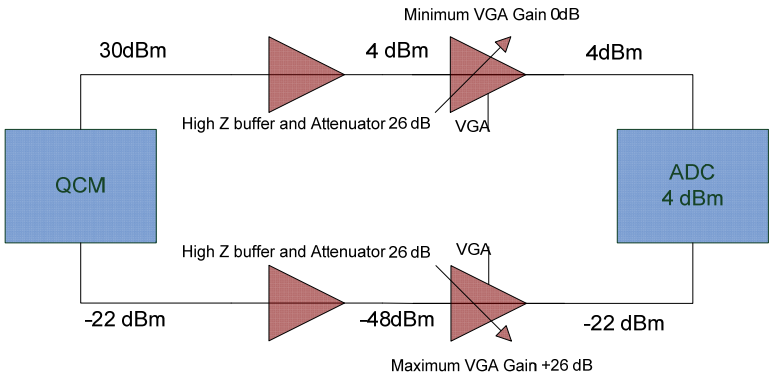


Figure 4.16: Input diagram showing the gain for the maximum output (top) and for the minimum output (bottom).

The two amplifiers were designed to be on a plug in board that was compatible with a connecting backplane that is used to connect to the DSP and transceiver board. This plug-in design enables the single boards to be replaced without having to redesign the whole

hardware setup. The DSP connects to the amplifier board via the ESSI port which is routed via the backplane. This serial port enables the writing of new gain values to each of the amplifier chips independently via one 8 bit write at speeds of 1 MHz or less. This means the chips are updated with new gain values fairly fast – within 1 µs of writing.

4.2.4 DRIVING BOARD

The assumption behind the amplifiers in the previous chapter is that the connections are all matched to 50 ohms impedance. This impedance matching is necessary to maximise the power transferred and minimise reflections and distortion from bad matching. To ensure that the connections between the amplifiers are correct 50 ohm impedance cable is used with high quality connectors. Unfortunately the QCM's impedance can vary between 8 ohms and 30 ohms at resonance in air and from 100 to 300 ohms in water. This creates a requirement to match the amplifier output impedance to the QCM.

To maximize the power that is absorbed by the QCM from the RF amplifier the impedance of the connection must be as close as possible to the amplifiers impedance at resonance (50 Ohms). This is achieved using a transformer to match the impedance to the crystal as shown in Figure 4.17. Impedance was found using the impedance function of a Hewlett Packard network analyser's (HP3589A), to match the QCM impedance at resonance to 50 Ohms. It was found that there is a large difference between the impedance of the QCM when measured in gas and liquid.

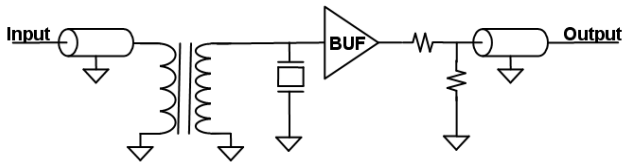


Figure 4.17: QCM board impedance matching using transformer.

To match the impedance between the amplifier and the QCM a Mini-Circuits wideband transformer was used. The impedance of a transformer is set by using the ratio of the windings to change the impedance. The relationship between impedance ratio and impedance changing is a N^2 factor. For example to match a 50 ohm transmission line to a 10 ohm QCM, would require a ratio of 2.24:1 (input winding : output winding). One transformer does not solve the problem of change of impedance in different mediums. The solution that is proposed is to create separate boards with different transformers optimised for the different QCM operating environments.

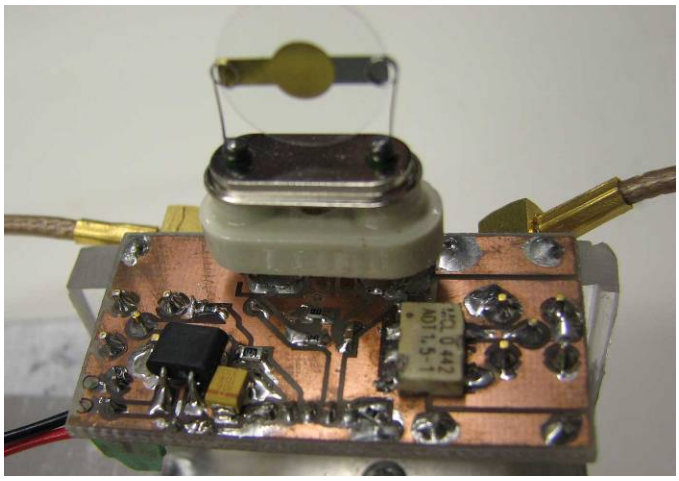


Figure 4.18: QCM board.

To measure the response of the QCM without loading it a high speed buffer used for its relatively flat gain up to 100 MHz, its high voltage input, and also the high current output (250mA for line driving). The buffer output is also matched to 50 Ohms with 26dB attenuation to ensure that the output does not overload input to the VGA board. The board that was created in the end is shown in Figure 4.18.

4.2.5 HARDWARE INTEGRATION

The board designs are based around the Eurocard style using a 96 pin plug connector to plug into a back plane. The back plane is designed for the first slot to be occupied by a DSP board, and the next two by a transceiver with full access to the data and address port. The next slots after this do not have access to the parallel data and address lines and so rely on the ESSI ports for and other general ports for general electronic communication. There is also room on this back plane for a power supply to be plugged in to the board.

Because this device is designed to be a desktop device is required to have its own internal power supplies and so two mains supplies are incorporated into the box enabling the device to rely directly on AC mains supplies. Three separate DC power supplies are needed with a 9 V supply needed for the DSP and transceiver board, a 12V supply needed to run the VGA amplifiers, and a 24 V supply for the external one watt amplifier. To implement these power supplies a mains 12 V supply which also supplies a 9V linear regulator is also integrated into the box. Another mains supply runs at 24 V. A mains switch is mounted onto the outside of the case controlling both supplies.

To enable enough air circulation two fans are mounted inside the case running off the 12 V power supply.

The RF connections between boards are smb connectors and using a RG174 shielded 50 ohm coaxial cable. The cable and connectors are designed to work from DC into the GHz frequency range and so are well within our operating range. They have many advantages over other plugs such as bnc, such as their small size and are made to be connected and disconnected easily and often.

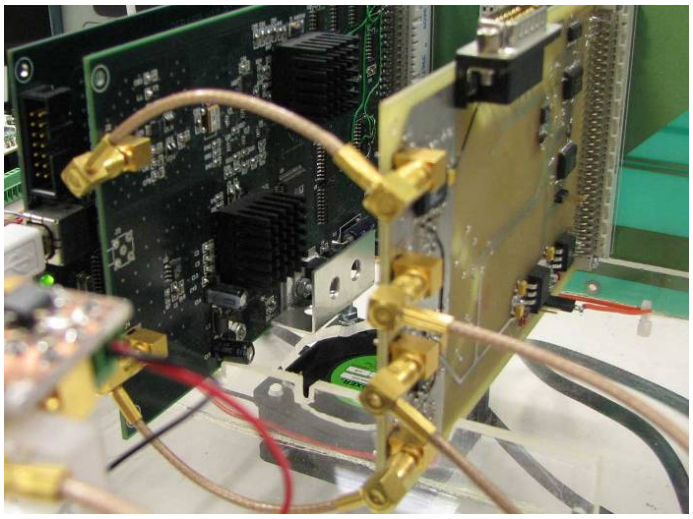


Figure 4.19: Completed digital sensing system with from left to right: DSP, transceiver, and VGA.

The completed sensing system is depicted in Figure 4.19 and Figure 4.20. Figure 4.19 is a photograph of the completed variable amplifier board connected via the backplane to the DSP and transmitter/receiver boards.

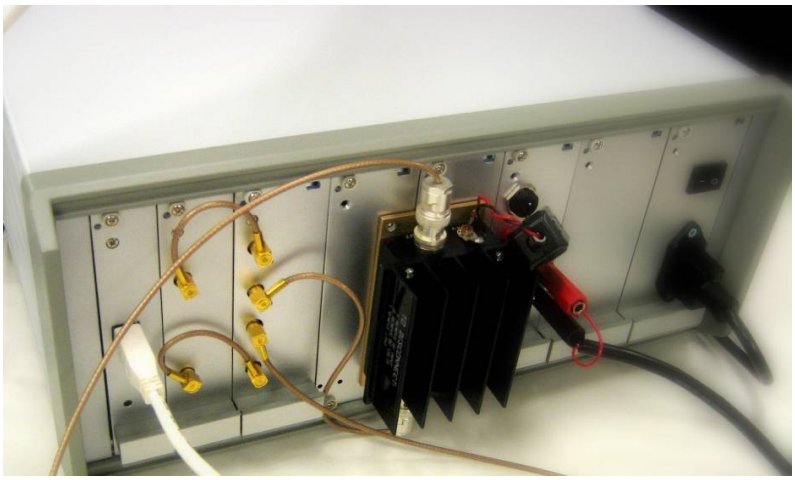


Figure 4.20: Bond Rupture System in a self contained box powered from a mains supply.

4.3 SOFTWARE

As the hardware has been introduced we will now focus on the other major elements of the setup, the software. The software is the reason for the hardware though they have a symbiotic relationship with both requiring each other and one not working if the other is not there. The difference with software is that the concepts (and some code) can be taken and applied to similar hardware designs. The software for this equipment make it a multipurpose piece of equipment enabling it to perform a frequency scan and then change modes and perform a data capture and FFT to test the equipment.

There are two main parts to this software firstly there is prospa and secondly there is the DSP code. Prospa is scientific software similar to Matlab but a lot smaller and faster. Prospa enables the easy creation of GUIs and scripts to control all the front end, calculations,

converting, and post processing that is not needed to be performed by the DSP. Prospa loads the DSP program, runs, and transfers data to and from the DSP. The DSP program on the other hand is compiled from an assembler file. The DSP performs the excitation and data capture from the QCM sensor. The DSP also performs any real time processing needed to reduce the size of the data being transferred to prospa software. The DSP program uses variables transferred from prospa to control actions such as the frequency to run at, or how many data points to capture.

The first stage to creating an experiment is to produce the assembler file – see Figure 4.21 on the left. The language used is the Motorola assembly language. Assembly language is the direct writing of the DSP's hardware instructions such as ADD, MOVE, JMP, DO and many more which act on the registers and memory locations. The Motorola assembly used allows for commands for 'loops' and 'if' statements and macros making it faster than writing straight assembly. When writing a program memory is set aside for prospa to write the data relevant to the run. The program will communicate with the transceiver board and the amplifiers performing the task it was designed for writing any data to a specified location for retrieval by prospa later. Once this assembler file is written it is compiled using the Motorola compiler into a "*.p" file. The compiled file can be then used by prospa to be uploaded to the DSP.

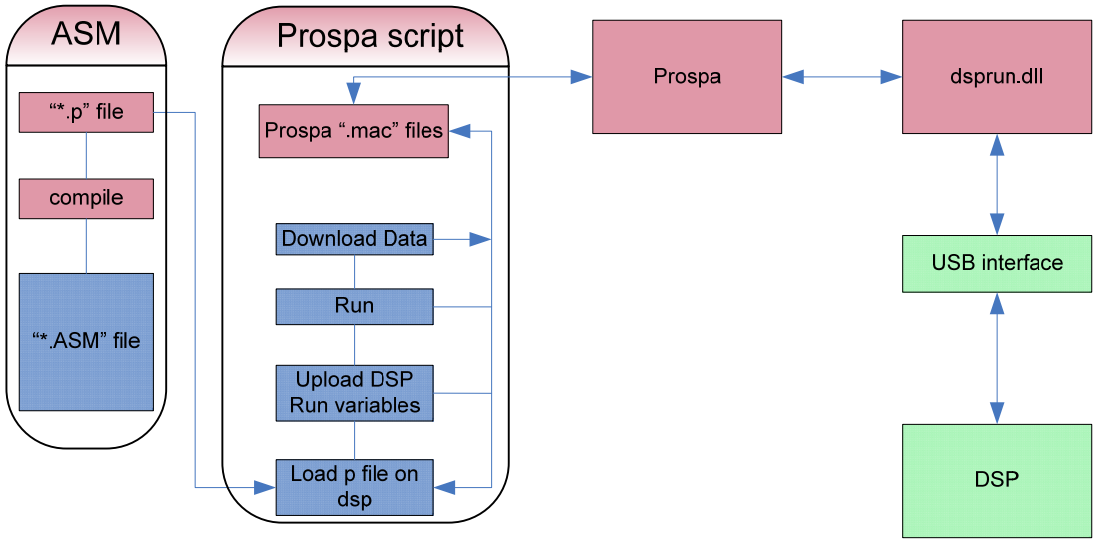


Figure 4.21: Software flow diagram.

The second stage to an experiment is to produce the prospa script – see Figure 4.21 under the 'Prospa script' box. This consists of code to create a GUI, and a macro to carry out the running of the experiment. The GUI is very simple to make with single lines to create an interface element such as a user box or button. Once the GUI is developed with all the GUI elements for any user input required for the experiment, the experiment macro can be created. This macro is usually set up to be carried out when a RUN button is pressed. This macro first reads all the GUI variables needed. Next the compiled program is loaded onto the DSP and the run variables are uploaded to the DSP. After this the experiment is run and when finished the data is read back from the DSP. The data is and post processed then outputted on a plot. Prospa communicates to the DSP with a DLL created to use the USB interface on the DSP.

What really makes the equipment suitable for QCM Bond Rupture is the number of different types of experiments that it has been designed to carry out.

The first experiment is called 'Wobble Centre'. This experiment provides a fixed output voltage to the QCM sweeping the frequency over a specific span around a given frequency centre and then records the voltage magnitude from the sweep. This also has the option of fitting a polynomial to the curve for accurate resonant frequency measurement.

The second experiment is called 'Bond Rupture Experiment using Magnitude'. This monitors the resonant frequency on the QCM in real time by finding the minimum magnitude and calculating from this the current frequency, while simultaneously ramping the voltage output and scaling the input voltage to maximise the ADC's dynamic range.

The third experiment is called 'Bond Rupture Experiment using Zero Crossings'. This monitors the frequency, the Q (quality factor), and the third harmonic noise of the QCM. This is implemented by driving the QCM at its resonant frequency for a specified amount of time then turning off the output and monitoring the ring down frequency and exponential time constant, and from these values determining the new frequency to drive at. This program also monitors the third harmonic noise while driving the QCM when not measuring resonant frequency.

The fourth program is called 'Calibrate', which handles all the calibration of the output and input power of the equipment. This is needed to enable the correct inducing of Bond Rupture and also to keep the dynamic range at the optimal value.

4.3.1 WOBBLE CENTRE

The method used to determine the QCM's resonant frequency is similar to that of Network Analysis. It works by measuring the magnitude and phase response of the QCM over a frequency range. This can be used to detect the resonant frequency of the QCM as when the QCM is excited near its resonant frequency it oscillates in a resonant manner altering the operating phase, magnitude and impedance. Both phase and magnitude of the signal in relation to what it was being operated at can be measured. When being driven by a power amplifier into a 50 Ohm source, the model can be further simplified to a voltage divider. This means when the impedance of the QCM changes so will the voltage potential across the divider, so creating an output voltage proportional to impedance which is measured by the ADC.

The software to measure the magnitude and phase is run on the DSP. The DSP algorithm does the following: firstly it modifies the gain of the output VGA and input VGA to achieve the desired output voltage and make best use of the available ADC resolution; secondly it successively updates the transmitter DDS and the DDS inside the Digital Receiver Processor with the current frequency values, and with each frequency step the real and imaginary values are stored in memory, lastly the real and imaginary signals are sent back to the computer. This is demonstrated in the flow chart in Figure 4.22.

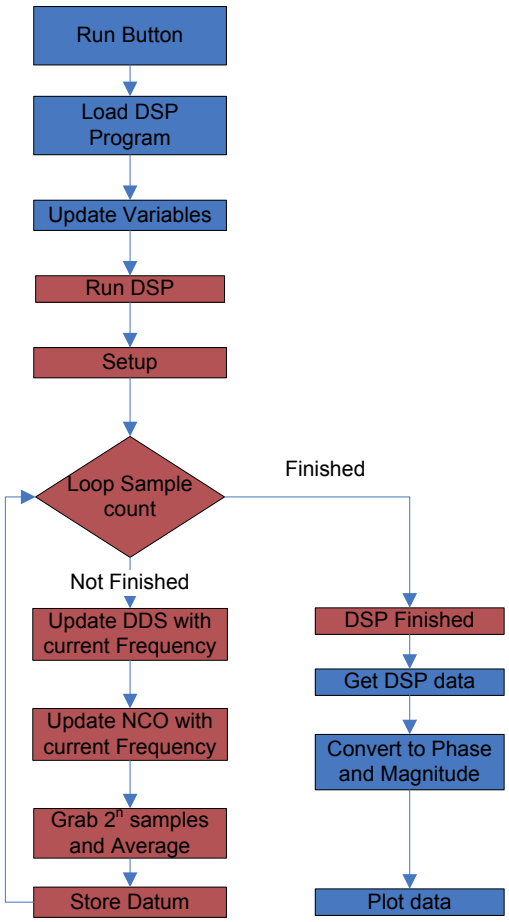


Figure 4.22: Wobble Centre software flow chart (blue is computer and red is DSP subroutines).

Once in the PC the magnitude and phase are calculated. This now presents a problem of how to get an accurate representation of the resonant frequency from the magnitude or phase. A plot of magnitude versus frequency is shown in Figure 4.23. A minimum polynomial fit method was created using the magnitude data as it is a method that works well in air and in liquid. This is implemented by first calculating the minimum point of the magnitude plot. Once the point is found the corresponding frequency value can be found.

There are a number of compromises to be made in operating the algorithm outlined above, the first being the width of the frequency band examined, if it is too large then the accuracy of the frequency will depend on the step size (so needing a large number of points to accurately find the resonant frequency). However, if the step size is too small the resonant frequency will be very hard to find. Also if the noise at around the minimum is large it is difficult to resolve what the frequency is.

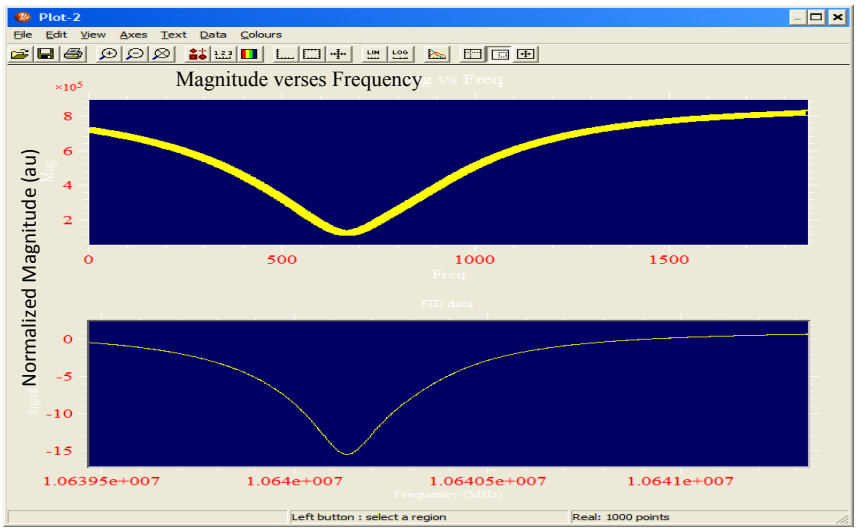


Figure 4.23: A magnitude versus frequency plot of a QCM using Wobble Centre.

To increase the accuracy of the resonant frequency measurement a polynomial fitting algorithm was used. The polynomial algorithm works by fitting a polynomial to the magnitude plot over a region where the plot approximates a polynomial. Once a polynomial has been fitted a minimum can be easily calculated numerically from the polynomial coefficient. This forgoes the noisy minimum problem essentially averaging the points gathered. This also enables better accuracy than the frequency steps method used. There are some set problems with this method including the initial frequency must be known to know where to start the scan. Also the frequency span cannot be too large as then the algorithm will cease to approximate a polynomial therefore reducing the accuracy and producing an offset from the real resonant frequency. Also the width of the magnitude curve changes depending on the quality factor of the QCM so this width must be optimised for different QCM's and operating mediums as there is a different Q between air and water.

Another method to increase the accuracy of the measurement is to average the data points coming in for each point using a power of 2 average. This reduces the amount of time necessary to perform the divide as if it is a divide by a power of 2 the answer can always be calculated by shifting right by that power. For example a divide by 16 in binary twos complement format can be performed using a simple shift right by 4 or another example to divide by 32 the binary values need to be shifted right by 5. Equation (4.3) shows the average where, R_x is the current datum, 2^n is the number of points to average, x is the sample number, $1/2^n$ is a shift right by n, and $R_{average}$ is the averaged value which is stored as a datum. This increases the resolution off the datum by averaging effectively narrowing the bandwidth.

$$R_{average} = \frac{1}{2^n}\sum_{x=1}^{2^n} R_x \qquad (4.3)$$

The wobble centre prospa GUI that was created is shown in Figure 4.24. This shows that the output voltage can be set. When the run button on this program is pushed the macro runs then returns a plot that looks similar to Figure 4.23.

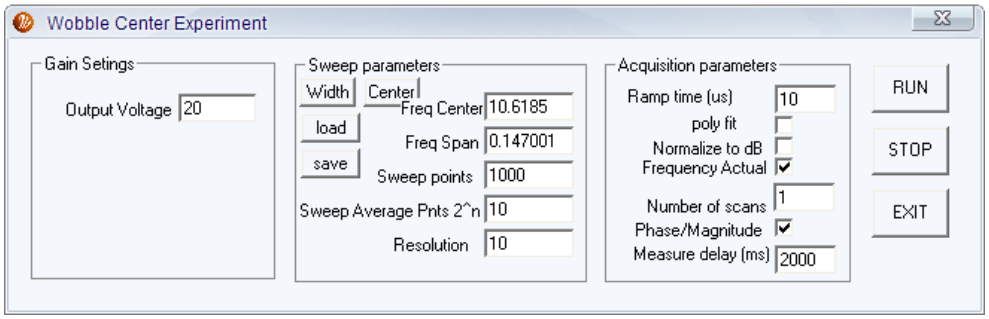


Figure 4.24: Wobble Centre experiment prospa GUI.

There are a range of settings for the Wobble GUI which need to be inputted. The range of frequency that the magnitude plot runs over is set by the frequency centre and frequency span variables. This range can be set anywhere from DC to 62.5 MHz - using sub sampling above 31.25 to measure from the ADC. The upper limit is set by the DDS maximum output frequency. The number of sweep points is the total points the DSP captures and returns to prospa. The average points 2^n is the number of samples to average for each frequency point if the value was 10 then 1024 points would be averaged for each outputted datum. The output voltage variable uses the calibration data to select a gain setting based on voltage into a 50 ohms resistor.

The rest of the descriptions for the Wobble Centre GUI variables are listed below;

- Output voltage: This is calibrated for when there is no load so the actual outputted voltage at the QCM will change significantly when the QCM is introduced and will therefore change whenever the impedance of the QCM changes. An example of this would be when a solution is placed on the QCM.
- Frequency Centre: This is the frequency that the scan is centred around.
- Frequency Span: the width of the scan is centred on the Frequency Centre.
- Sweep Points: the number of points to capture in the scan and to display.
- Sweep Average Points 2^n: This is the number of points to take for each frequency measured this is averaged.
- Resolution: Not used
- Ramp time (us): Not used
- Poly fit: when checked this fits a polynomial to the figure displayed and calculates the minimum point from the polynomial and outputs the corresponding frequency.
- Normalize to dB: this scales the output on the figure to log units.
- Frequency Actual: if ticked the figure will contain units of actual frequency in MHz, if it is not ticked then the units will be displayed in as the span without the frequency centre information.
- Number of scans: Not used.
- Phase/Magnitude: if ticked then the display will output the data as magnitude if not ticked then the phase data will be outputted.
- Delay (us) : this is the delay between each measurement point in us.
- Width: this button enables the user to select a width from the prospa plot to perform another scan on To use click once then click the lower part of the width, then click the higher part of the width.
- Centre: this when clicked enables the user to select a new centre. To use click centre then click your new centre on the prospa plot.
- Save: Enables the user to save the GUI variables when clicked.

- Load: when the user clicks enables the user to load a previously saved GUI setup.

4.3.2 BOND RUPTURE EXPERIMENT USING MAGNITUDE

This Bond Rupture experiment concentrates on the frequency method for measuring Bond Rupture. That means constantly monitoring the current resonant frequency and simultaneously increasing the voltage applied to the QCM at a predetermined rate. When rupture occurs a frequency change also occurs.

The Bond Rupture software consists of prospa and a DSP program. The prospa program sets up the initial values and passes them the DSP. The DSP has the job of monitoring the frequency (using a magnitude based method) and also controlling the ramp of the driving voltage. When the DSP has finished the data gathered is returned to prospa where it is plotted on screen. Because there are a number of factors that affect the frequency change other than Bond Rupture, which can vary from QCM to QCM, two or more successive runs is used as a comparison. The difference between the runs is then plotted showing the actual rupture frequency change with the factors not relating to rupture cancelled out.

The Bond Rupture driving voltage is needed to be changed by the DSP in this program. To enable the voltage ramp using a simple and easy method it is ramped using the 12 bit scale factor variable within the DDS. So the VGA gains are set using the highest voltage needed in a ramp and then the scale factor is used to start the ramp at the lowest voltage and work up to the highest output voltage. This means that the DSP only needs to change one variable. This method is simple as all the complex VGA gain calibrations can be performed in prospa rather than have to be calculated by the DSP. The disadvantage with this is that if a large voltage (gain) range is needed the resolution of the DAC output is reduced at lower voltages. For example a ramp between 0.1V and 10V is 40dB range so a loss of 40dB of dynamic range occurs at the low output voltage on the DAC leaving only 32dB of dynamic range to use at the output converting what would be a 12 bit output to around 5 bits of usable data. Though a range of 1V to 10V would only lose 20dB of resolution which works out to be around 8.7 bits of resolution. As can be seen this is not ideal and this will be addressed in later sections.

The magnitude frequency tracking method was developed for this application. This method needed to be fast and accurate and able to be implemented easily on the DSP. This method works by sampling the magnitude at three different frequencies and using a basic fitting algorithm to find the lowest point from this frequency. This method works well in low signal environments but suffers when the initial frequency is not precise enough, and the tracking can be lost when large frequency changes occur over a small amount of time.

The Magnitude method works well for accurate tracking of the frequency with a high resolution. But it needs a starting centre frequency that is within the frequency magnitude curve, otherwise the frequency tracking may be lost. To find the starting frequency the wobble software is used. Once the starting frequency is found the Magnitude program can run. This starts by measuring the magnitude at three points – as seen in Figure 4.25. The first point is X_{p1} which is run at the centre frequency minus FStep (FStep a variable defined by the user usually set around 20 Hz). The second point is X_{p2} which is run at current centre frequency. The third point is X_{p3} which is run at current centre frequency plus FStep. These variables are then used

to calculate the new frequency. The magnitude frequency detection is based on the QCM's frequency magnitude looking similar to an inverse bell curve. If we measure three points on a curve such as this at known interval the centre can be found after a few iterations using a tracking algorithm.

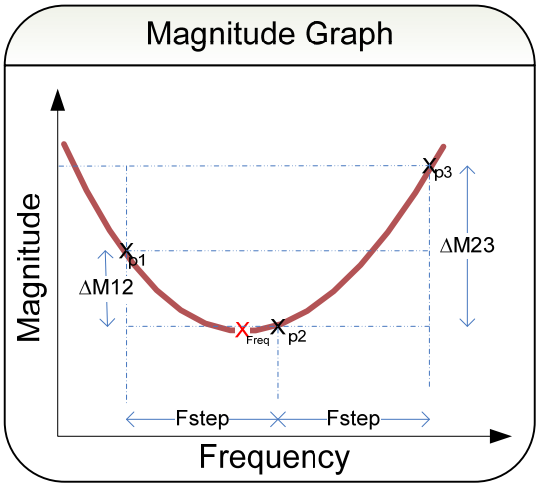


Figure 4.25: Stylized picture showing the points gathered to find the centre point.

The tracking algorithm works by looking at the three variables gathered and from those calculating two more variables ΔM_{12} and ΔM_{32} – see Figure 4.25. These variables are calculated by subtracting X_{p2} from both X_{p1} and X_{p3} to calculate ΔM_{12} and ΔM_{32} respectively - see Equation (4.4) and Equation (4.5). So now we have two variables that are both relative in magnitude to the centre point (X_{p2}).

$$\Delta M_{12} = X_{p1} - X_{p2} \quad (4.4)$$

$$\Delta M_{32} = X_{p3} - X_{p2} \quad (4.5)$$

Once the variables are calculated a qualifying algorithm is then carried out to select the operating mode. This starts by firstly checking if ΔM_{12} is negative if it is then it is assumed that the centre frequency must be at or lower than X_{p1} so the new frequency is set to point X_{p1}'s frequency. If that was not true and ΔM_{32} is negative then it is assumed that the centre frequency must be at or lower than X_{p3} so the new frequency is set to point X_{p3}'s frequency. If none of those were true and both ΔM_{12} and ΔM_{32} are negative then it is assumed that the locking has been lost and so the frequency is left unchanged then lastly if both ΔM_{12} and ΔM_{32} are positive then the centre finding program is run.

ΔM_{12}	ΔM_{32}	New Frequency
Negative	Positive	X_{p1}'s frequency
Positive	Negative	X_{p3}'s frequency
Negative	Negative	X_{p2}'s frequency
Positive	Positive	Centre Finding Algorithm

Table 4.5: Frequency qualifying algorithm.

This qualifying algorithm (shown in Table 4.5) basically assumes that we are tracking a bell shaped curve. So if we are looking at upward slope then the new frequency is set to the lowest frequency point X_{p1}. If we are looking at a downward slope the new frequency is set to the highest frequency point X_{p3}. If the points are on either side of a bell curve with the middle near the centre then the centre finding algorithm is used to find a more accurate centre value.

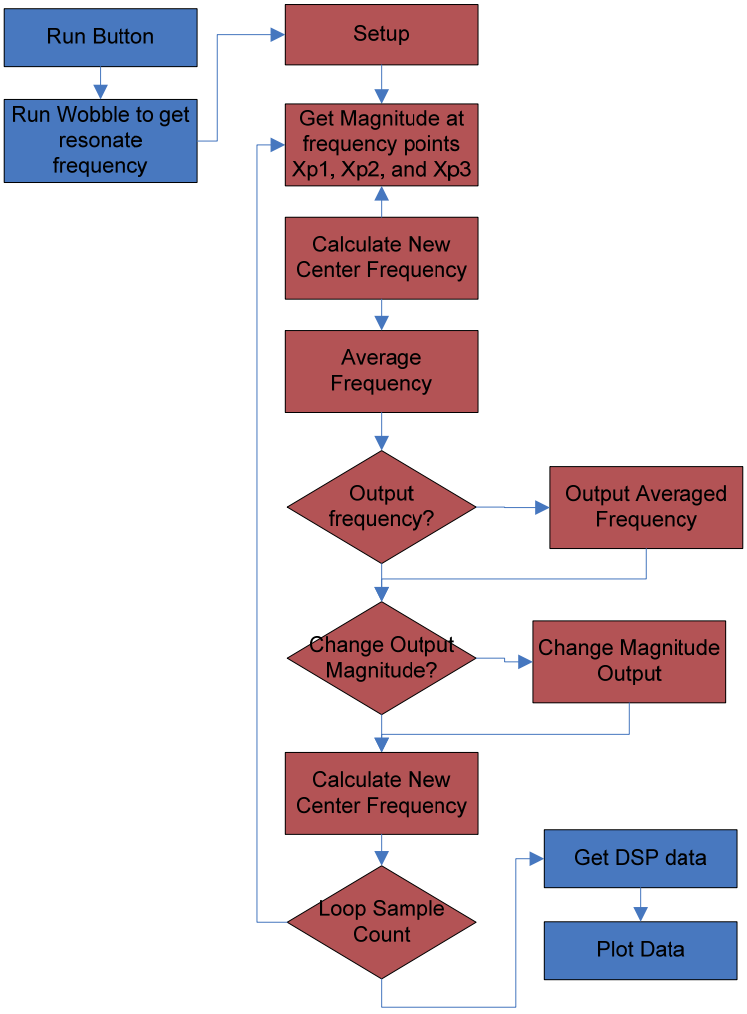


Figure 4.26: Software flow diagram for the Magnitude tracking method, with prospa (blue) and DSP (red).

The centre finding program starts finding the largest value out of ΔM_{12} and ΔM_{23} then calculate the negative or positive ratio as performed in Equation 4.6. This ratio (R) is then put into Equation (4.7) which multiplies the ratio and the frequency step (F_{step}) and also another variable called Proportional Control ($P_{control}$). This equation limits the total change in frequency

to decrease frequency jumps caused by noise – but also limits the response of the frequency control.

$$R = \left(1 - \frac{\Delta M_{32}}{\Delta M_{12}}\right).(\Delta M_{12} > \Delta M_{32}) + \left(\frac{\Delta M_{12}}{\Delta M_{32}} - 1\right).(\Delta M_{12} \leq \Delta M_{32}) \quad (4.6)$$

$$\Delta F = R.F_{step}.P_{control} \quad (4.7)$$

The magnitude method is very good at monitoring very accurate frequency measurements but runs into problems if the frequency changes too fast – such as if you put water straight on to the QCM from air inducing a frequency change of around 10 kHz. Another limitation of this method is that the quality factor of the QCM cannot be measured. Another limitation is that the Bond Rupture (BR) noise cannot be measured as the digital receive processor is being used to monitor frequency change and as there is only one digital receive processor only one frequency can be measured at a time.

The overall flow of this software can be seen in Figure 4.26. This shows the interaction between prospa and the DSP. This prospa GUI first runs the "wobble centre" program to calculate the starting frequency with some degree of accuracy. Once this is attained the resonant frequency is passed to the DSP which then handles the capturing and calculating of the new frequency points and increasing output driving voltage. Once the DSP has finished running the experiment the data is then returned to prospa which then processes the data and outputs a plot.

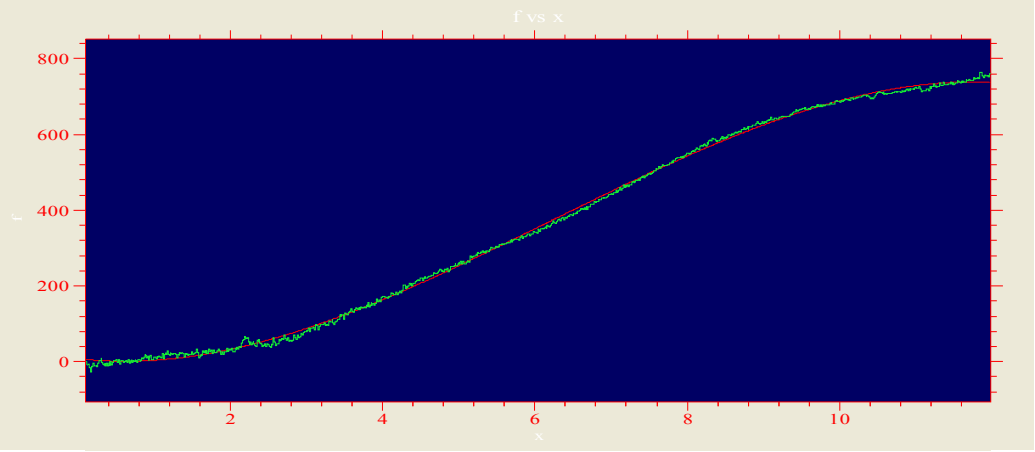


Figure 4.27: Bond Rupture Magnitude tracking software frequency change versus time. (green) actual frequency change. (red) fitted polynomial.

Once the program has run a plot is outputted as in Figure 4.27. As can be seen the plot shows frequency change versus time or voltage. To properly analyse this graph for Bond Rupture, it needs to be compared with a plot of the same QCM without Bond Rupture occurring. This cancels out any effects such as temperature and voltage dependence factors that are specific to each QCM. This GUI was a development version and the need for comparison was not known until further experimentation occurred. Initially the feature to enable the comparison of two plots was implemented using Excel - this comparison feature is implemented in later software methods.

The prospa GUI for the Magnitude Tracking software is shown in Figure 4.28. There are quite a few different parameters to configure but once setup and running it should produce a graph similar to Figure 4.27 on completion. The most important parameters for a scan would be the Frequency Centre, the start voltage, the end voltage and the ramp time. These are the essential variables with most of the other variables there to fine tune the running of the DSP software.

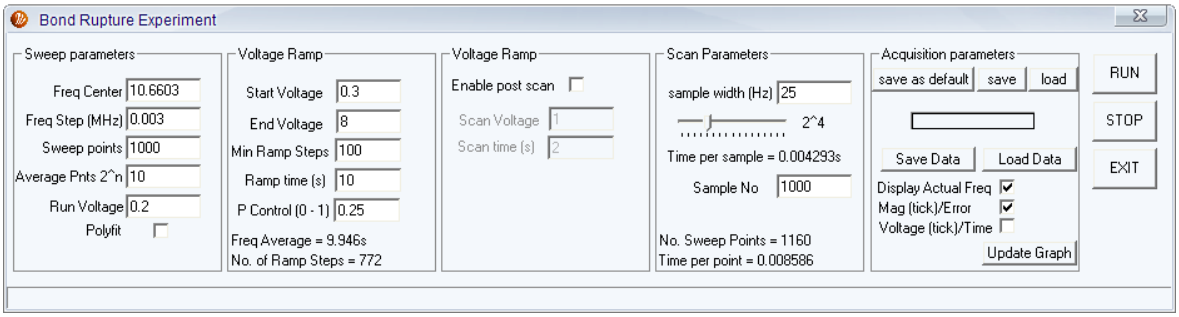


Figure 4.28: Magnitude Frequency Track prospa GUI.

The Sweep Parameters box is a cut down version of the parameters inputted within the wobble centre macro. This has all the parameters needed to do an initial first scan to find the resonant frequency of the QCM. This is ideally run at a voltage far under the initial start voltage, to prevent unintended Bond Rupture.

The Voltage Ramp box contains the parameters needed for the voltage ramp including the start and end voltage, the number of ramp steps to use, ramp time and the proportional control variable for frequency tracking. There is also an output at the bottom showing the actual time that that the run will last for and the actual number of ramp steps which relates to the voltage ramp values as well as the values set in the Scan Parameters box. The third box across from the left is for enabling a post scan at a set voltage and time. This is to try and understand what the relationship between the frequency and drive voltage, over time.

The Scan Parameters box is for setting the bandwidth and number of frequency samples to gather and how much to average. The sample width is the FStep variable mentioned in the algorithm explanation, which simply is the frequency width to sample at left and right of the centre sample point. The next point is the number of magnitude samples to average together to get one magnitude point. This has to be a power of 2 to simplify calculations within the DSP. The time per sample output is amount of time spent after averaging per frequency sample. The number of samples is the next box down which specifies a start point to calculate the number of samples from. The actual number of samples is calculated in software taking into account the ramp steps, time per sample, while trying to get as close as possible to the ramp time specified. The software can use frequency samples and also average successive frequency samples (this time is shown in time per point which is the actual time per frequency point saved) to get within 50% of the sample number specified – assuming there is enough time.

The Acquisition Parameters box is used to save or load parameters, or to save or load plot data, or change the data that is being displayed on the plot. There are three main plot settings. The first one is a tick box to set whether or not to display the actual frequency or the frequency offset from the frequency centre value. The reason this is needed is that it can be

difficult to read the frequency change when you are looking for 100Hz change in offset with a scale of around 10 MHz. The next tick box is whether to display magnitude or error in the second graph window. Magnitude is the centre point of the magnitude value measured and outputted as on the auxiliary plot, which can help to provide some feedback to what the QCM voltage change is as it is being driven. Error is a variable that can be set in the DSP for feeding back data for diagnostic purposes. The voltage time tick box changes between displaying voltage or time on the x-axes on the plots. This enables both time and voltage to be analysed and for a fixed voltage output enables the data to still be plotted versus time.

The variables that are specified in Figure 4.28 are defined as follows:

- Sweep parameters: this set of variables are similar to the Wobble Centre program and are for the first frequency run - for further information on setting up these variables see the wobble centre prospa GUI operation section.
- Voltage Ramp: this set of parameters is for setting how the voltage ramp is performed.
- Start Voltage: the voltage you want to start your BR scan from.
- End Voltage: the voltage you want to stop you BR scan at.
- Min Ramp Steps: the minimum number of steps that will be allowed in the ramp.
- Ramp time: the amount of time that the ramp/experiment will occur over.
- P Control: the value for controlling how fast the frequency can change.
- Freq Average: displays the actual time that will pass over the experiment.
- No. of Ramp Steps: this is the number of steps or times the voltage will change over the experiment.
- Enable Post Scan: is a checkbox that enables you to select if you want to do a post scan at a fixed voltage.
- Scan Voltage: the fixed voltage for the second scan.
- Scan Time: is the amount of time to perform this scan over.
- Scan Parameters: this set of parameters is for setting the frequency control values.
- Sample width: is the F_{offset} value that is described in the previous section, it defines the offset each sample is taken at.
- The slider: is the amount of averaging that is performed for each magnitude variable gathered.
- Time Per Sample: displays the amount time to spend gathering each frequency point.
- Sample No: around about how many samples/frequency points you want gathered.
- No. Sweep Points: displays the actual number of samples that will be gathered.
- Time per point: displays the amount of time spent for each sample value gathered.
- Save As Default: saves all the GUI parameters for next time the GUI is opened.
- Save Data: Saves the experimental data to be opened later using Load Data.
- Load Data: loads the data from a previous save.
- Display Actual Freq: this checkbox when checked displays the absolute frequency, and when not checked it just displays the offset frequency from 'Freq Centre' box.
- Mag. (tick)/Error: when checked this checkbox displays the averaged magnitude each frequency run performed, and when not checked it displays the error from the frequency tracking algorithm.
- Voltage (tick)/Time: when checked this displays voltage on the x scale of the plot and when not checked it displays time.
- Update Graph: if the plot is changed this will restore it to what it was before.

4.3.3 BOND RUPTURE EXPERIMENT USING ZERO CROSSINGS

The second method used to measure Bond Rupture is also a frequency based method, but it has the added feature of being able to monitor for Bond Rupture noise when it is not monitoring frequency, as the frequency measurement only occurs for half the cycle. This method involves measuring the QCM oscillation frequency by looking at the number of zero crossings and the quality factor after driving the QCM at near to it's resonant frequency. During the driving cycle an average of 3rd harmonic magnitude is also recorded to monitor for Bond Rupture. The wave form produced is approximated by Figure 4.29 - with the first half of the figure the driving of the QCM (which is used for 3rd harmonic measuring) and the second half of the figure where the driving is stopped (which is used for measuring the frequency and the quality of the QCM. This method was first mentioned in literature by Auge (Auge, Hauptmann et al. 1995) though it only covers monitoring frequency and quality and did not include inducing Bond Rupture or monitoring for Bond Rupture noise. This method calculates the exponential decay value and measures the frequency of the free running crystal.

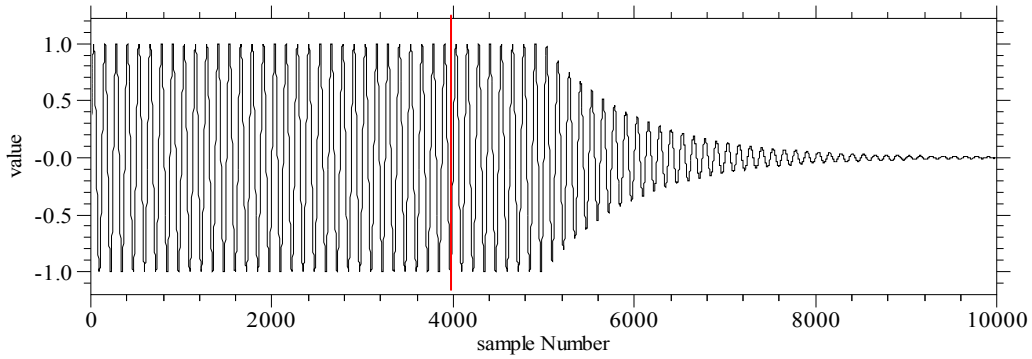


Figure 4.29: Driving (to the left of the centre line) then decay (to the right of the centre line) as performed by the zero count algorithm.

The QCM oscillations decay at an exponential rate and from this rate the quality factor of the QCM can be calculated. Because the data is converted to real and imaginary by the DRP we can calculate the magnitude from Equation (4.8). The magnitude once calculated is a decaying exponential. A simple method of determining the time constant τ is to start by measuring the first magnitude value captured then measure the time taken to get to 36.8% of that maximum value and once measured this time can be substituted in for τ. To enable accurate determination of the time constant a simple linear interpolation algorithm is using the magnitude and time before and after the magnitude crosses 36.8% enabling a value to be calculated that has a greater accuracy than using the sample resolution alone.

$$Magnitude = \sqrt{R^2 + I^2} \tag{4.8}$$

The time constant once calculated can then be converted to seconds and used to calculate the quality factor of the QCM. This is calculated by using Equation (4.9). The quality factor is the inverse measure of the amount of dampening of the QCM. The quality factor can be measured by examining the width of frequency where the magnitude is at 3 dB on either side of the frequency of interest. The quality factor also determines the efficiency driving of the Bond

Rupture – as seen by the amplitude of movement formula proposed by Dultsev (Dultsev, Speight et al. 2001) which can be seen in the literature review.

$$Q = \pi f(Hz)\tau(s) \tag{4.9}$$

The QCM resonant frequency is measured by looking at the oscillation frequency when the output stops driving the QCM. The QCM keeps moving for a time after this much like a swing keeps swinging after you stop pushing it. During the driving cycle the QCM is being driven by a voltage but after the output is disabled the QCM generates a voltage dependent on the amount of energy stored in the QCM as mechanical motion - at the QCM's true resonant frequency. This voltage generated decays at a rate which depends on the quality factor of the QCM and the load the QCM is driving.

The resonant frequency of the QCM's oscillation is determined (over a specified number of samples) by counting the number of zero crossings (n_{count}), and the number of samples contained within that number of zero crossings (t_{total}) – as demonstrated by Figure 4.30. These values are then put into Equation (4.10), producing the average period per sinusoid wave (t_{av}) with the units of samples. The time is converted from samples to time in seconds by multiplying by the time per sample, which is the ADC sampling time (1/62.5MHz = 16ns) multiplied by the down-sampling factor Mcic2 and Mcic5 of the DRP as shown in Equation (4.11).

$$t_{av} = \frac{2.t_{total}}{n_{count}-1} \tag{4.10}$$

$$f_{av} = \frac{1}{16.e^{-9}.M_{cic2}.M_{cic5}.t_{av}} \tag{4.11}$$

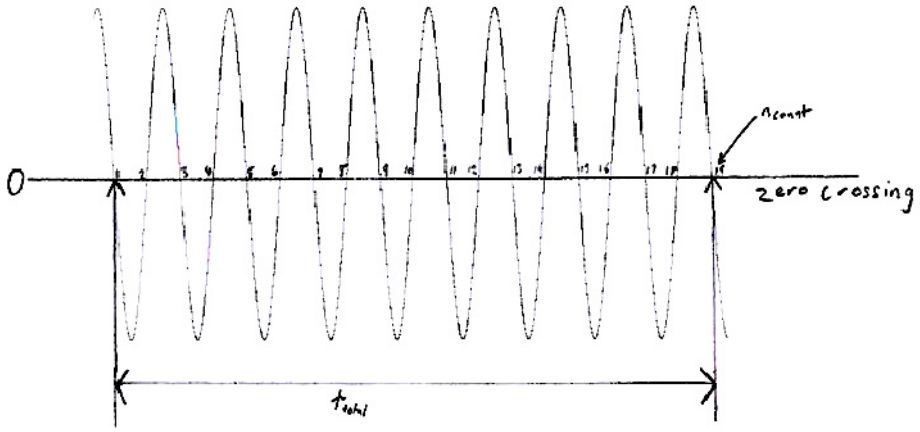


Figure 4.30: Zero crossing count example.

The zero count algorithm relies on a DDS/NCO within the DRP to shift the frequency down from high frequency (1 – 62.5 MHz) to around DC (0Hz) by multiplying the input waveform by the amount the input is required to be shifted. An example is if you want to shift 10.7MHz down to DC you would multiply the input by 10.7MHz and the resultant wave would be centred on DC - for further information see section under hardware, digital transceiver, and receiver. To get a more manageable bandwidth the resultant waveform is now filtered and down-sampled around DC. Now to be able to count the zero crossings we need to have a

waveform with enough zero crossings to count - since a DC waveform is essentially a flat line with no sinusoid component. To do this a frequency offset (f_{offset}) is added to the frequency multiplied by the input as demonstrated in Equation (4.12). So if the frequency of the QCM is 10.7MHz then the shift amount would be calculated as 10.7 + f_{offset}. The frequency offset chosen must not be to large as to be outside the bandwidth specified, or filtered to much by the CIC filters (as CIC filters have a large amplitude fall off outside around 10% of the Bandwidth), or the signal maybe to small and no zero crossing will able to be detected over the specified period.

$$f_{NCO} = f_{qcm\_last} + f_{offset} \tag{4.12}$$

The number of zero crossing that will occur and that can be measured will depend on the quality of the crystal how long it resonates for, on the bandwidth that is captured, the frequency offset from resonance, and the amount of time captured for. All of these factors need to be optimised to get the best zero crossing and quality factor measurement.

The flow of the zero crossing software is seen in Figure 4.31. The zero crossing software starts by running the GUI in prospa. When the run button is pressed in prospa the DSP is setup with variables needed and then a Zero_count function within the DSP software runs to measure the resonant frequency and quality factor as well as measuring an average 3rd harmonic noise factor. Once this function has run it returns the new frequency value and dissipation value and average magnitude. These variables are then averaged with subsequent runs depending on the amount of averaging set in the prospa GUI. The next step is to change the output voltage which is set mainly in the GUI and only an initial value and a step amount is provided which is then added onto the current DDS output scale value to change the output voltage. The next step outputs the averaged values to memory and resets the average value. The last step checks whether the run has finished by looking at the sample count. If the software has not finished then the loop jumps back to running the Zero_count function and if it has finished it returns to prospa.

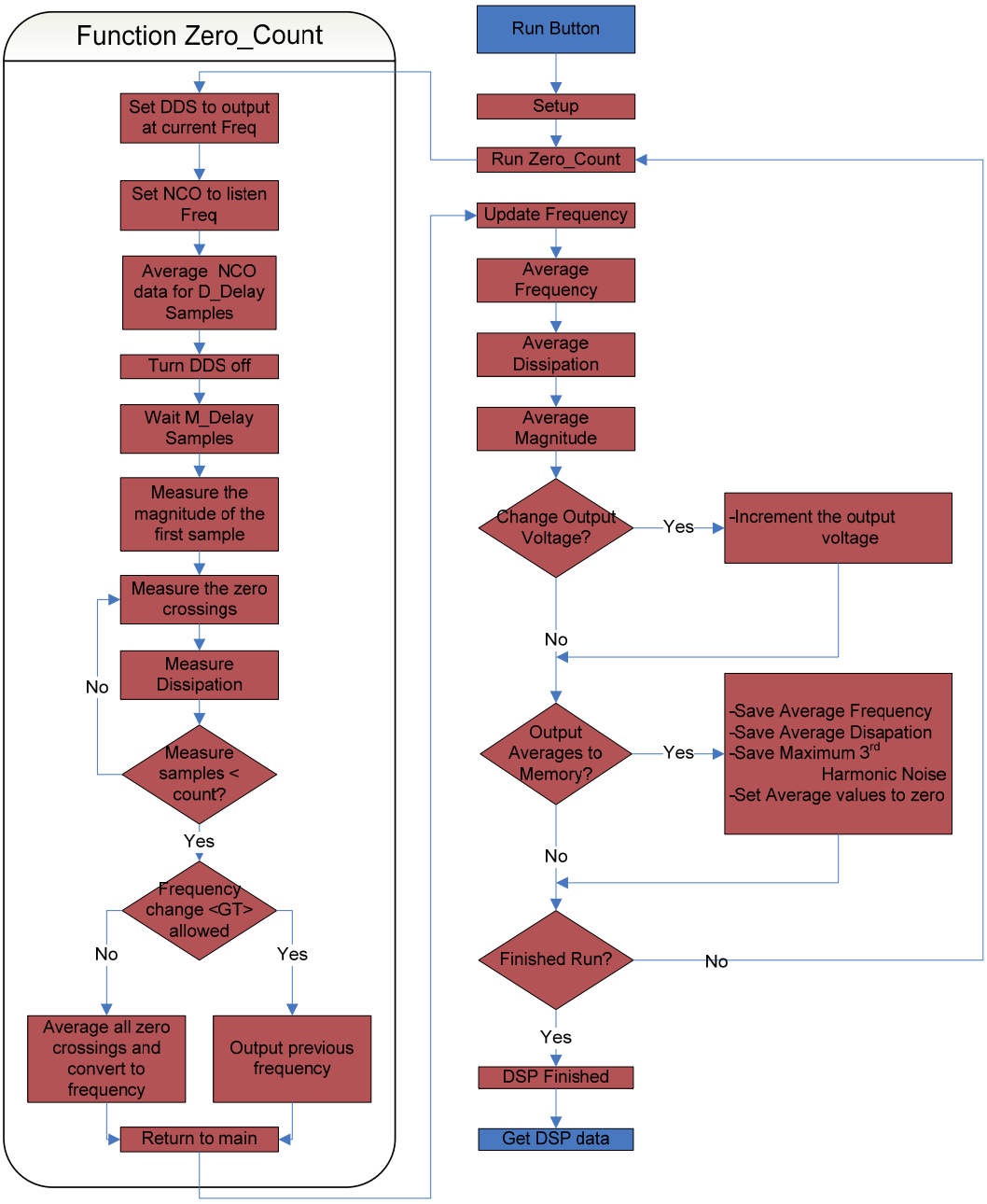


Figure 4.31: Flow diagram for Bond Rupture using zero crossings.

The DSP sets the DDS to output and turns on the output RF switch. The DRP is then set to listen to the 3rd harmonic frequency and averages the frequency noise at a specified band-width. This is done for a certain number of samples specified by the D_Delay variable. After the drive time has finished the output RF switch is turned off and a small amount of time is waited specified by M_Delay for the samples to propagate though the ADC and CIC filters. Next the magnitude of the first sample is measured for a quality factor reference. A loop is then created which will run for capturing and processing M_Samples. The loop measures the number of zero crossings that occur as well as the looking for the point where the magnitude crosses the 38.7% mark. The waveform when measured at the QCM with an oscilloscope looks similar to Figure 4.32. Once the loop has finished the program compares the frequency calculated to the previous frequency and if the frequency change is outside a realistic value the change is not

allowed and the previous frequency value is kept. This prevents single glitches destroying the frequency tracking. This process can be seen in on the left side of Figure 4.31.

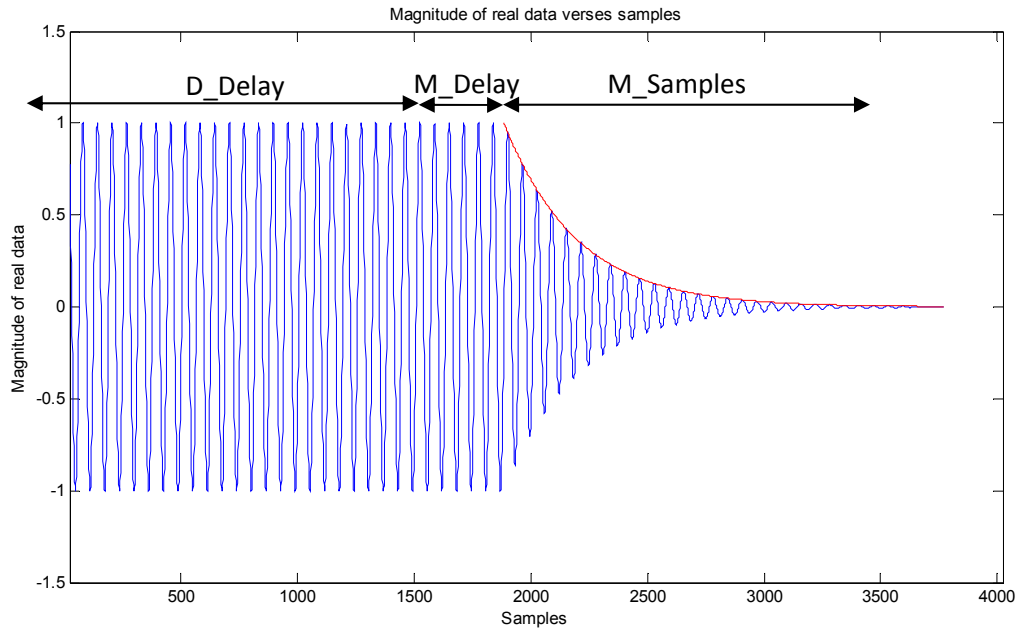


Figure 4.32: Simulated waveform at QCM showing the three stages of the Zero_count algorithm.

After the zero crossing macro has been run the main software performs averaging on the frequency, quality value, and the magnitude to minimise the amount of storage points while smoothing the frequency value. Then the DSP software scales the output using the amplitude multiplier value on the DDS. The disadvantage of using the DDS amplitude multiplier is that when the output is low DAC resolution is lost. The input VGA (which is supplies the signal to the ADC) constantly decreases the gain when the signal reaches over a certain threshold set in prospa to prevent overflow in the ADC. The full software flow diagram can be seen in Figure 4.31.

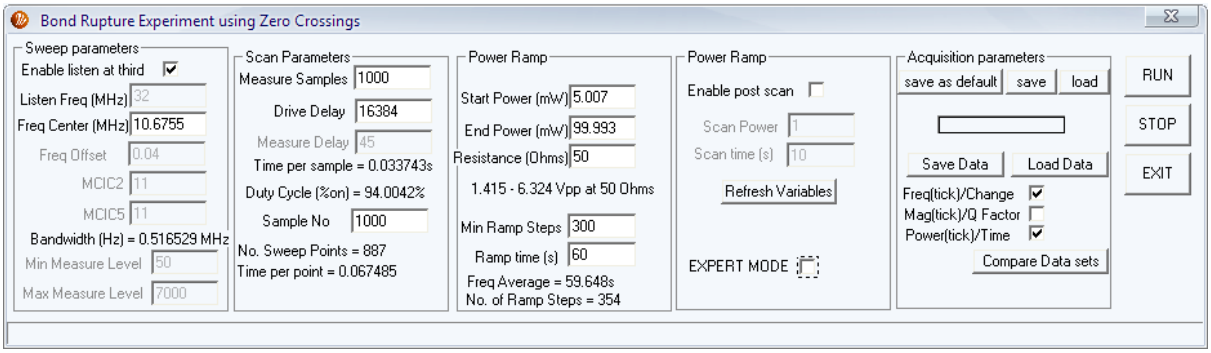


Figure 4.33: prospa Bond Rupture using Zero Crossings Experiment.

The Bond Rupture Zero Count GUI is called Bond Rupture Experiment using Zero Crossings or ‘BRzc_V2’ for the macro name – this is the last design iteration of which many improvements were made from the original program. This GUI macro (seen in Figure 4.33) has many controls which can appear confusing so to simplify the operation, the advanced controls that do not usually need changing are not accessible (greyed out) by default unless the expert tick box is

pressed. The most important variable that must be set is the Frequency Centre variable. This is for setting the initial frequency to try exciting the crystal at. This must usually be set within about 20 kHz of the real resonant frequency but this range is reduced when the starting voltage is very low as this reduces the signal size significantly making it difficult to detect the resonant frequency.

The ability to listen to Bond Rupture noise is one signification improvement over the 'Bond Rupture Experiment Using Magnitude' software from the previous section. This Bond Rupture noise method uses the driving period when the receiver is not needed to listen for Bond Rupture noise, at either a frequency specified, or at 3 times the fundamental. This listening at three times the fundamental frequency is set using the 'Enable listen at third' tick box. If this is not used the listen frequency can be set manually to the frequency of interest. This frequency is captured at the same bandwidth that the frequency tracking algorithm is using and then the magnitude of the signal is averaged over this period - the time this happens over is specified by the drive delay variable.

The values for setting the bandwidth of the frequency to be captured are the MCIC2 and MCIC5 variables. These values are the decimation values that specify how much the each of the decimation filters need to down sample. The bandwidth is calculated from the ADC operating frequency divided by the MCIC2 and MCIC5. For example 62.5 MHz divided by 11*11 is 516.529 kHz bandwidth. The frequency offset variable specifies the offset from DC in MHz that the captured frequency should be at. The value should be set to 10% of the bandwidth or less to keep within the linear CIC filtering region.

The minimum and maximum measure levels are important variables to keep the frequency tracking stable. The minimum measure level is the level where if the magnitude falls under this while trying to measure quality and zero counting then the measuring will keep whatever has been gathered and ignore the rest of the data. This was implemented as the quality of the signal will degrade and affect the accuracy of the measurement, if at a low level. The maximum measurement level variable is to limit the maximum input signal to prevent overflow. When this value is exceeded the input gain of the VGA is reduced by one gain value.

The 'Scan Parameters' group has a number of variables that set up the time spent doing the different tasks in within the scan period.

The number of the 'measure samples' variable (previously mentioned as M_Samples in 4.33) is the maximum number of samples to measure the zero crossing for. This value needs to be optimised depending on what amount of time the oscillations can be measured for which can be determined by the quality factor. This also will depend on the amount of down sampling, as fewer zero crossing my occur at a lower band width, but on the other hand an increase in accuracy due to signal processing gain will occur. This value is usually chosen based on experimentation.

The 'Drive Delay' variable (previously mentioned as the D_Delay in Figure 4.33) is the number of samples to drive for. The timing of this variable is based the sample rate, as during this time the DRP will be used to sample for the 3rd harmonic Bond Rupture noise. This must be a value that is a power of two to enable the 3rd harmonic averaging function to be efficient.

The ‘Measure Delay’ variable (previously mentioned as the M_Delay in Figure 4.33) is the number of samples to ignore after the driving has stopped. This variable needs to be changed every time the CIC filters decimation rate is modified as the number of samples is dependent on the CIC filters decimation rate. This is usually determined experimentally.

The ‘time per sample’ value is the amount of time that is needed to perform one set of driving and measuring pulse like shown in Figure 4.33. This is calculated from the bandwidth, measure samples, drive delay, and measure delay.

The ‘Duty Cycle’ value is calculated by the amount of time the output is being listened to, over the amount of time the output is being driven. This is useful as a measure of the amount of time the output is on.

The ‘Sample No’ is a roundabout number of samples that the user requires. The number of points that will be saved are shown below this box. There is also a time per point value which is the amount of time spent gathering each individual output point. If the ‘Time per point’ output is greater that the ‘time per sample’ output, then the points would have been averaged - to reduce the number of points that need to be saved in memory. This averaged amount is optimised to be close as possible to the ‘Sample No’ specified.

The ‘Power Ramp’ group enables the user to set the start and end power to ramp over the specified ‘Ramp time’. This ramp is linearly distributed over the power range with the number of steps changes specified by the ‘No. of Ramp steps’ output. A minimum number of ramp steps can be specified in the ‘Min Ramp Steps’. This number is used to specify the minimum number of steps so that the optimisation algorithm can get the closest to the time specified as possible. The peak to peak voltage is also shown which is calculated from the equivalent resistance. This is calculated from the calibration and does not take into account power lost from impedance mismatch.

There is a post scan ramp to enable a second scan at the end of the ramp to be performed using the same parameters except for time and only using a fixed voltage. There is also a ‘Refresh Variables’ button which is used to refresh the output values shown on the screen such as the voltage, and time per sample.

The acquisition parameters group is used to save or load parameters, or to save or load plot data, or change the data that is being displayed on the plot. There are three main plot settings. The first one is a tick box to change whether or not to display the actual frequency or the frequency offset from the frequency centre value. The reason this is needed is that it can be difficult to read the frequency change when you are looking for 100Hz change in offset of 10MHz. The next tick box is whether to display magnitude or quality factor in the second graph window. Magnitude is measured from the first point gathered from the quality measure algorithm. The Power or time tick box changes between displaying power or time on the x-axes on the plots. This enables both time and driving power to be analysed and for a fixed power output enables the data to still be plotted versus time.

The last button is the ‘Compare Data sets’ which is to compare two data runs (only can be runs of the same settings at this stage) producing a graph showing the first graph minus the second.

This is used to cancel out any effects due to the change frequency due to driving power. Two consecutive runs on the same blank QCM will produce identical results (apart from error in accuracy), which when compared will look as a flat line plot of frequency versus power. A Bond Rupture scan is run on a QCM with a sample on the surface and then two consecutive runs for comparison. The comparison tool is used to compare the three runs. The first run compared with the second run shows a frequency change depending on the bonds on the surface. Where the second and third runs when compared show that there is no relative change in frequency.

When the prospa macro is run a plot similar to Figure 4.34 is produced. This plot (Figure 4.34, bottom) shows the frequency change that occurs.

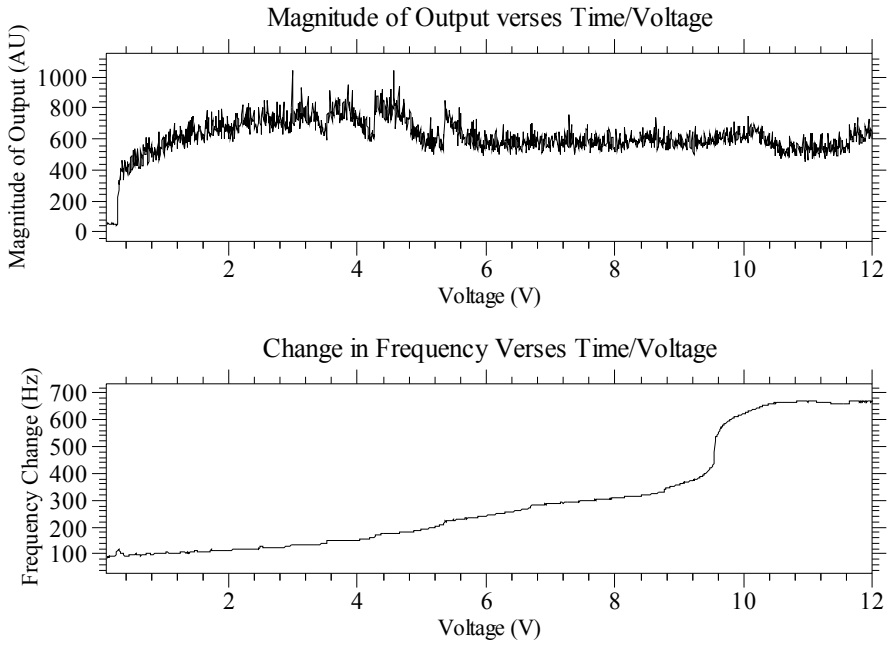


Figure 4.34: Bond Rupture Zero Count plot, with the 3rd harmonic noise (top), and frequency change (bottom)

The prospa BR Zero Count GUI as seen in Figure 4.33 variables are defined as follows:

- Freq Centre (MHz): is the starting frequency to be run. Care must be taken to choose the frequency to be close enough especially for low voltages as the signal may not be large enough if the crystal is not properly excited.
- Freq Offset: is the offset that the DRP will use to enable zero crossing to occur. Care must be taken to make sure this is less than $1/10^{th}$ the band-width, as if it is greater then too much filtering may occur degrading the signal resolution.
- MCIC2: is the first down-sample factor for the CIC filter and so determines the signal band-width.
- MCIC5: is the second down-sample factor for the CIC filter and so determines the signal band-width. The combined decimation factor should not be under 100.
- Band-width: displays the signal bandwidth that is outputted from the DRP to the DSP.
- Min Measure Level: is the value that if below the DSP will cease using as a valid value as below this the noise will be too low.
- Max Measure Level: is used to scale the input variable gain amplifier when this is exceeded.

- Scan parameters: is used to define all the scan parameters needed.
- Measure Samples: is the number of samples to measure for the listen stage.
- Drive Delay: is the number of samples that the QCM will be driven for before listening.
- Measure Delay: is the number of samples that are invalid and so are ignored before the measure samples are captured. This needs to be determined experimental every time variables MCIC2 or MCIC5 are changed.
- Time per sample: is the amount of time it takes to perform one zero count block which includes; drive delay; measure delay; and measure samples.
- Duty Cycle(%on): displays the percent of time that the QCM is driven for.
- Sample No.: is the approximate number of samples the user would like to gather.
- No. Sweep Points: displays the number of actual samples recorded.
- Time per point: is the actual time per recorded point, the points in between are averaged to save data memory.
- Voltage Ramp: is used to set all the voltage ramping related variables.
- Start Voltage: the voltage you want to start your BR scan from.
- End Voltage: the voltage you want to stop you BR scan at.
- Min Ramp Steps: the minimum number of steps that will be allowed in the ramp.
- Ramp time: the amount of time that the ramp/experiment will occur over.
- Freq Average: displays the actual time that will pass over the experiment.
- No. of Ramp Steps: this is the actual number of steps or times the voltage will change over the experiment.
- Enable post Scan: is a checkbox that enables you to select if you want to do a post scan at a fixed voltage.
- Scan Voltage: the fixed voltage for the second scan.
- Scan time: is the amount of time to perform this scan over.
- Refresh Variables: is a button for updating the displayed text if it does not do so automatically.
- Expert mode: when check enables all the greyed out boxes. NOTE: these should only be changed if you know what you are doing otherwise you may cause an unexplained error to occur, or damage the amplifiers.
- Save as default: saves all the GUI parameters for next time the GUI is opened.
- Save Data: Saves the experimental data to be opened later using Load Data.
- Load Data: loads the data from a previous save.
- Display Actual Freq: this checkbox when checked displays the absolute frequency, and when not checked it just displays the offset frequency from ‘Freq Centre’ box.
- Mag (tick)/Error: when checked this checkbox displays the averaged magnitude each frequency run performed, and when not checked it displays the error from the frequency tracking algorithm.
- Voltage (tick)/Time: when checked this displays voltage on the x scale of the plot and when not checked it displays time.
- Compare Data sets: This enables you to subtract one plot from another giving a more accurate frequency change output.

4.3.4 MAGNITUDE AND FREQUENCY CALIBRATION

The calibration process is an important one as it determines that the outputs and inputs are at the correct levels. It also enables the output to be set at the correct level without measuring every time – open loop output control. In a perfect world calibration would not be necessary as models could be used instead. Unfortunately there are many different non-linear effects that influence the magnitude across the frequency spectrum.

There are two main calibration types that are performed, frequency calibration, and input and output gain calibration. Frequency calibration is important calibrate for magnitude changes due to analogue filters on the DDS and other analogue non ideal components. To calibrate, the frequency sweep is run over a range of usable frequency values with no QCM (open circuit - infinite impedance), averaged to get accurate values, and then normalized to a known frequency as seen in Figure 4.35. A frequency and magnitude calibration should run whenever the system is changed such that the frequency response or magnitude response is changed – an example is when a new board is added, or the external amplifier is changed.

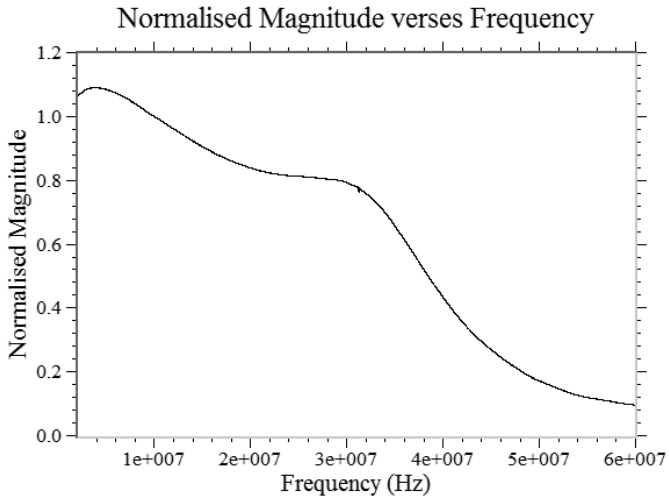


Figure 4.35: Digital system frequency calibration curve with magnitude normalized to one at 10 MHz versus frequency.

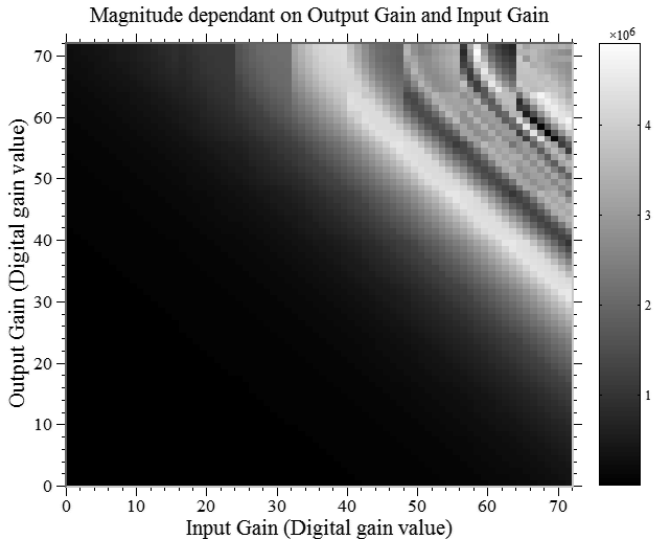


Figure 4.36: 2D Gain calibration graph, with magnitude (colour), dependant on the settings of output gains, and input gains. ADC overload can be seen as moving from left to right near the top as the gains increase and then become sporadic.

Gain calibration is used to calibrate the VGA's, as these are digital logarithmic amplifiers and the gain can vary significantly between chips and depending on the loading and external components accuracy. This is carried out by running through every input and output gain possibility (72*72=5184 possibilities) at a known frequency and measure the magnitude at

each gain as seen in Figure 4.36. This determines the correct scaling to get maximum dynamic gain from the ADC for a specified output gain as well as the scaling factor used to find the actual voltage input for given gains.

Voltage Calibration is used to find, given a desired output voltage, the correct output scaling factor for the DDS and the output gain values for the VGA. This is achieved by normalising the gain calibration values and scaling by measuring the output voltage (with an oscilloscope) at a known gain value. This gain plot is measured at the same frequency used for normalising the magnitude versus frequency plot. The whole graph can be scaled as shown in Figure 4.37. This graph can be converted from voltage to power as it is calibrated for 50 ohms.

To calculate the required output voltage for a specified frequency, a series of steps are run through. First the normalised factor for frequency is found by looking up the output frequency on the magnitude versus frequency graph. The normalised value is then used to scale the voltage versus gain graph. This gives a scaled voltage gain graph. From this a search algorithm is run to find the closest voltage to the voltage required. This value is rounded up to the nearest voltage and the gain value is returned. Final adjustments to get an accurate output voltage are performed using the scale factor on the DDS. This is done using a function within the calibrate macro called 'FindScale(Vout,Frequency)'. This function takes the two input variables output voltage and frequency. This function generates the output of the VGA output gain value, the VGA input gain value, the scale factor for the DDS and the actual voltage achieved. This function is used by the previous three GUI macros to determine all the input and output gain values for the voltages required.

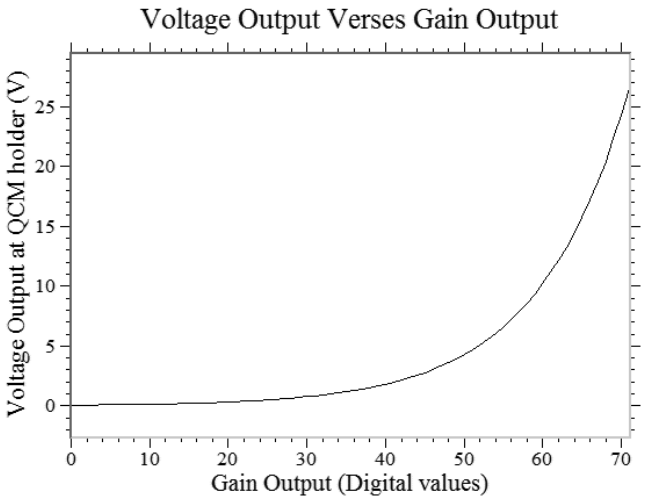


Figure 4.37: Output gain versus output voltage at 10.7MHz.

The main purpose of this equipment is to monitor for Bond Rupture. To perform this we had to vary the amplitude of the oscillation of the QCM by changing the power/voltage which is being applied to the QCM. So calibration to a value was needed so a range could be set to scan over. Voltage was the value chosen to calibrate the output to the QCM to. Unfortunately this is a complex issue to calibrate for, because the impedance of the QCM changes depending on the frequency that it is run at. Some contrasts can be seen as follows; at the exact 1st resonant frequency of the QCM in air the impedance will be around 20 ohms whereas away from resonance it could be up to 1000 ohms; at the 1st resonant frequency running in air it will be

around 20 ohms but in water around 100 ohms. When the impedance changes so will the voltage being applied - as in a potential divider circuit – and so it is hard to calibrate the output in real time as the voltage measured will change depending on QCM quality factor and resistance.

The prospa GUI (see Figure 4.38) has many different variables that can be set. The Run button when pressed runs a scan over first the frequency specified then over the gain ranges specified producing a 1D magnitude versus frequency plot, and a 2D input gain versus output gain plot showing magnitude as the image.

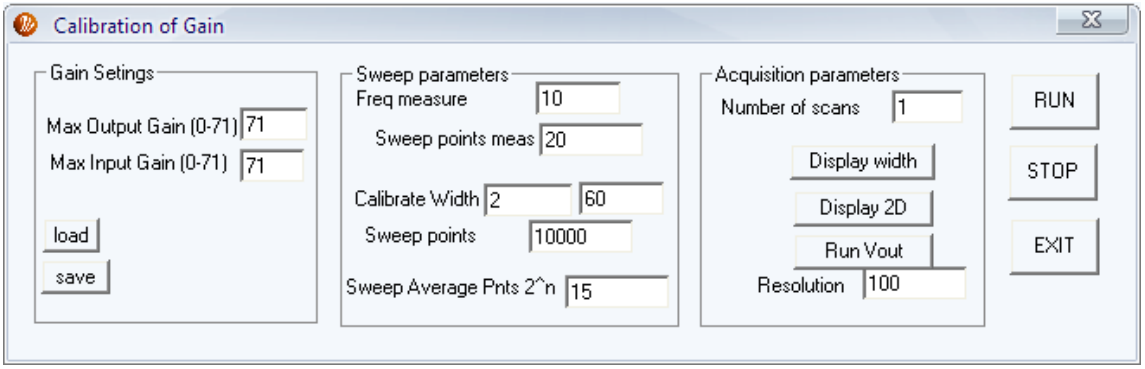


Figure 4.38: Calibration GUI.

The prospa Calibration GUI variables are defined as follows:

- Max Output Gain: this is the max setting for the output amplifier to end the calibrate scan at.
- Max Input Gain: this is the max setting for the input gain for the amplifier to end the gain calibrate scan at.
- Freq measure: the frequency to measure all the gains at.
- Sweep points measure: this is the number of points to capture for each gain calibrate.
- Calibrate width: the first box is the start frequency and the second box is the stop frequency.
- Sweep Points: the number of points to measure for the frequency calibration scan.
- Sweep Points Average: the number of averages to the power of 2 to use for each sample.
- Number of Scans: not used.
- Display Width: displays the frequency scan.
- Display gain values as a 2D colour figure.
- Run Vout: Runs the output at a specified voltage for a specified amount of time to be measured by a scope for voltage output calibration. See Calibration macro and procedure RunVoutCal within this macro.
- Run: runs the frequency and magnitude gain calibration.
- Stop: stops a run make sure a new run is done after this to recalibrate.

4.4 RESULTS AND DISCUSSION

The design and testing of the equipment was performed in a much more iterative approach than is indicated from the layout of this chapter. Since the DSP and Transceiver was developed by Dykstra the testing of the devices is contained in his thesis, so for further information on

the specifications see (Dykstra 2006). The testing was started with the initial development and testing of the wobble centre algorithm. After this an initial Bond Rupture program was implemented using the wobble software and performing the resonant frequency calculation on the PC. This was a very slow method with around one sample every second. This was used for initial testing with the consensus being to continue developing the next stage Bond Rupture method due to the low time resolution. This was the magnitude Bond Rupture algorithm. This was used to verify the Bond Rupture method but it was found that this method is not very useful unless the starting frequency is very near to the real frequency, producing many errors from experiment to experiment. To get around this problem Bond Rupture using the zero crossing method was used. This method has the added advantage of measuring the quality factor of the QCM. Also due to the DRP not being necessary during the driving time, it was able to be tuned to look for Bond Rupture noise. Two problems were found with the zero crossings Bond Rupture method. Firstly that it does not operate well in low signal environments and so it can run into problems when the signal is very low at the start. This in turn required the starting voltage to be limited to around 0.1V to ensure that the QCM could always be measured. The zero crossing method was used as the standard algorithm by IRL (Industrial Research Ltd) for further Bond Rupture experiments. The rest of this section contains the testing and experiments carried out using this software.

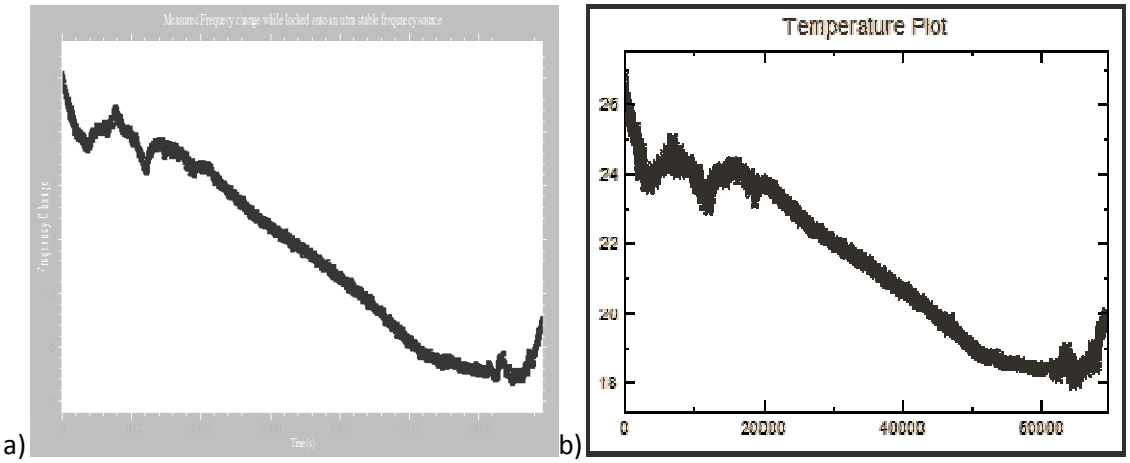


Figure 4.39: Temperature stability of the DSP and transceivers timing crystal. (a) Frequency change measured with a temperature stabilised frequency source as the signal being tracked 10Hz (Measured in MATLAB). (b) Measured air temperature over same period (measured from custom program see section 3.3).

The first testing that was carried out was to see what the stability of the DSP and transceiver was. This was tested by using the temperature box from chapter 2, a stable frequency source (PS 250 Z002BM frequency synthesizer 0.1 Hz variation per day and over 0-50°C), and the wobble software with the QCM system. This experiment was carried out by measuring the external air temperature (see Figure 4.39 a), while also measuring the frequency change while also measuring the stable frequency source (see Figure 4.39 b). This was measured over night for around 19 hours finding a clear correlation between temperature and frequency change of the base frequency on the DSP. The measured change works out to be around 4 Hz/°C. As this falls well within the error for frequency measurement this will not significantly affect

frequency measurements. This small change will also not affect measurements unless a significant external temperature change occurs.

Wobble is used as an effective look at frequencies at which the resistance reduces significantly and so more power is absorbed by the QCM. This is where the resonant frequencies occur. If a scan is performed at around the resonant frequency such as can be seen in Figure 4.40, it will be found that other absorbance points occur such at ~10.004MHz, which if not careful the user could lock on to this frequency (using the magnitude method) instead of the actual resonant frequency which is around ~9.98MHz. Experiments also found that the resonant frequency's minimum point significantly reduced in magnitude when water is introduced onto the surface of the QCM. It can also be seen that the resonance magnitude curve widens, which is indicative of the quality of the QCM. Experiments were also carried out at the third harmonic resonant frequency see Figure 4.41. It was found that the resonant looks similar to the first but the amount of power absorbed was reduced significantly. Further investigation is required to measure whether this is caused by impedance mismatch or change in the quality factor at this frequency.

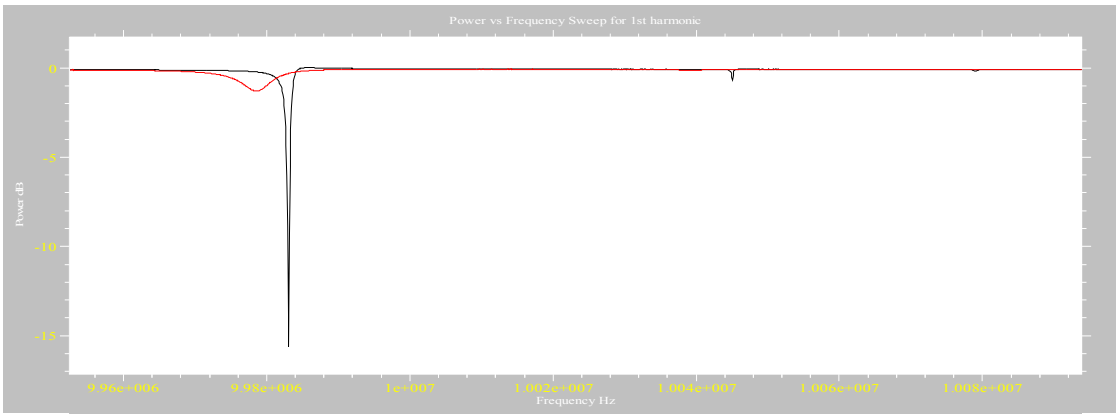


Figure 4.40: Wobble run using a QCM around fundamental resonant frequency with: (black) air environment, and (red) water environment.

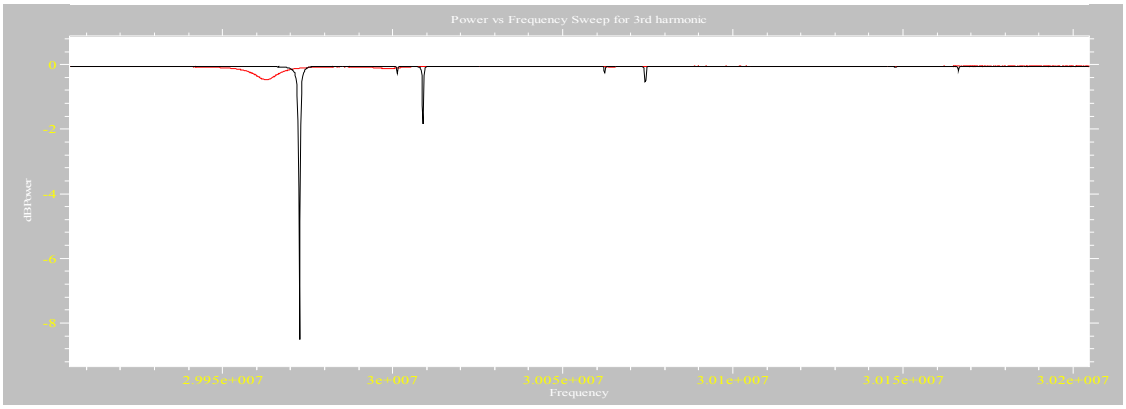


Figure 4.41: Wobble run using a QCM around third harmonic resonant frequency with: (black) air environment, and (red) water environment.

The magnitude Bond Rupture method was used to successfully verify Bond Rupture. It was found that there was a high frequency change that was induced as the voltage was increased. This can be seen in Figure 4.42 with a blank crystal with the frequency increasing significantly

as the voltage is increased. After a number of experiments it was found that this frequency change was not constant from one QCM to another, so requiring further investigation. It was found that the voltage distortion was constant for a single QCM from scan to scan so two subsequent scans could be used to remove the voltage effect by subtracting one scan from another.

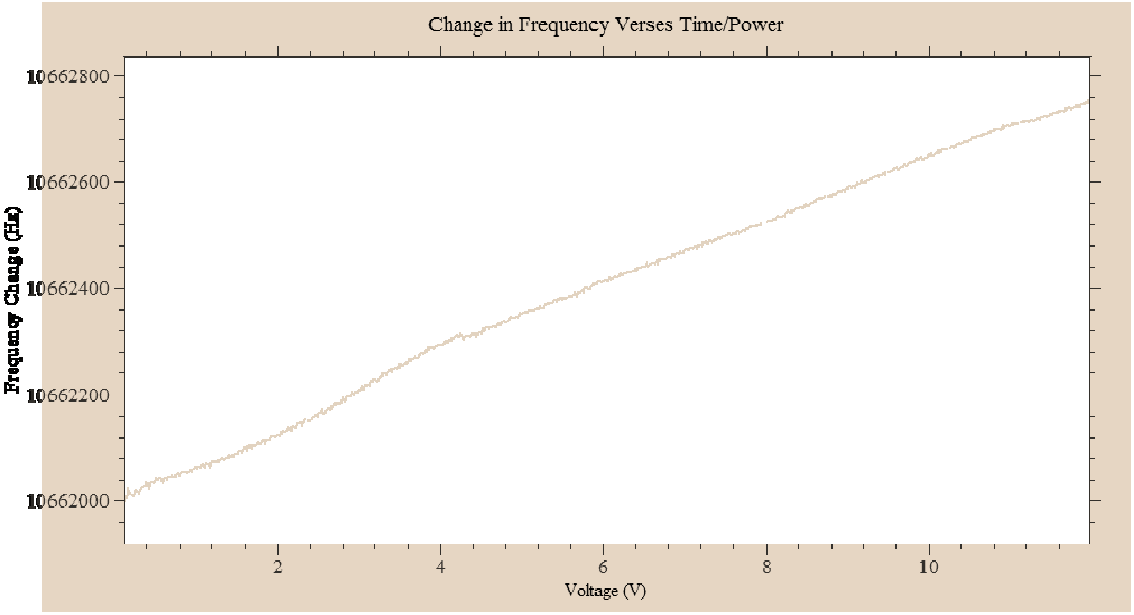
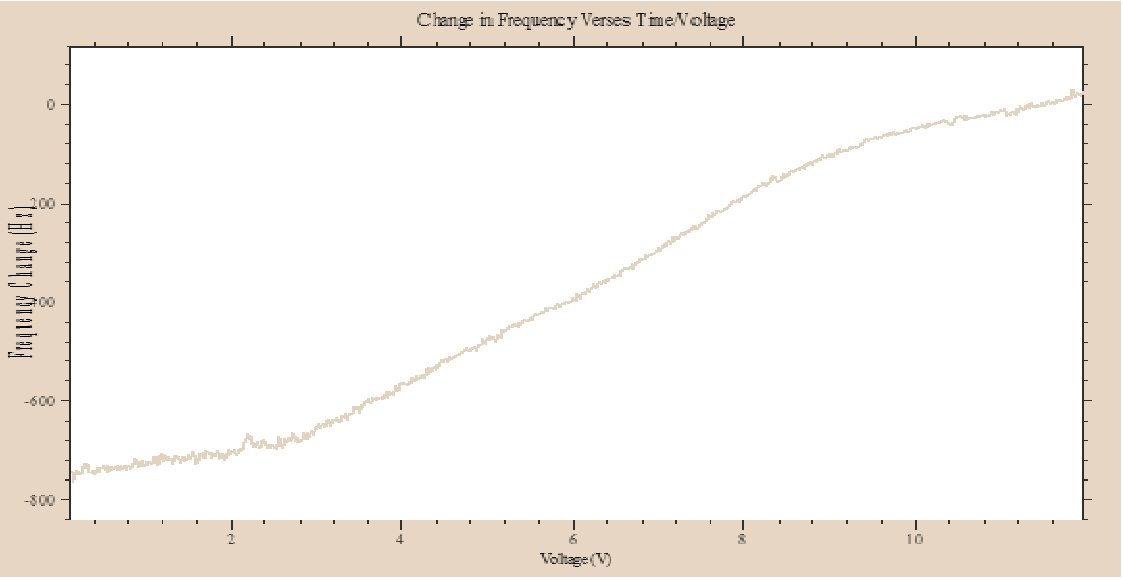


Figure 4.42: A Bond Rupture scan showing change in frequency versus voltage using the Magnitude measurement method.

Usually at least three scans are be carried out to confirm that the Bond Rupture has been completed. An example of the Bond Rupture comparison between the first and second run is shown in Figure 4.43 (a) and comparison between second and third in Figure 4.43 (b). This experiment was carried out with Streptavidin beads attached to the surface of a biotin treated QCM. It can be clearly seen that a frequency change has occurred in Figure 4.43 (a) whereas no frequency change has occurred in Figure 4.43 (b). The beads were also verified to have broken from the surface using a microscope. This leaves no doubt that Bond Rupture has been detected via the frequency change magnitude method.

The magnitude method is by no means perfect as the scan occurs there is a large amount of frequency noise present. This short term noise as can be seen in Figure 4.43 (b), is around 10Hz with the long term noise being around 60Hz change. This measured long term noise could be a small amount of un-ruptured beads rupturing from the surface; it could also be due to rebinding occurring.

a)



b)

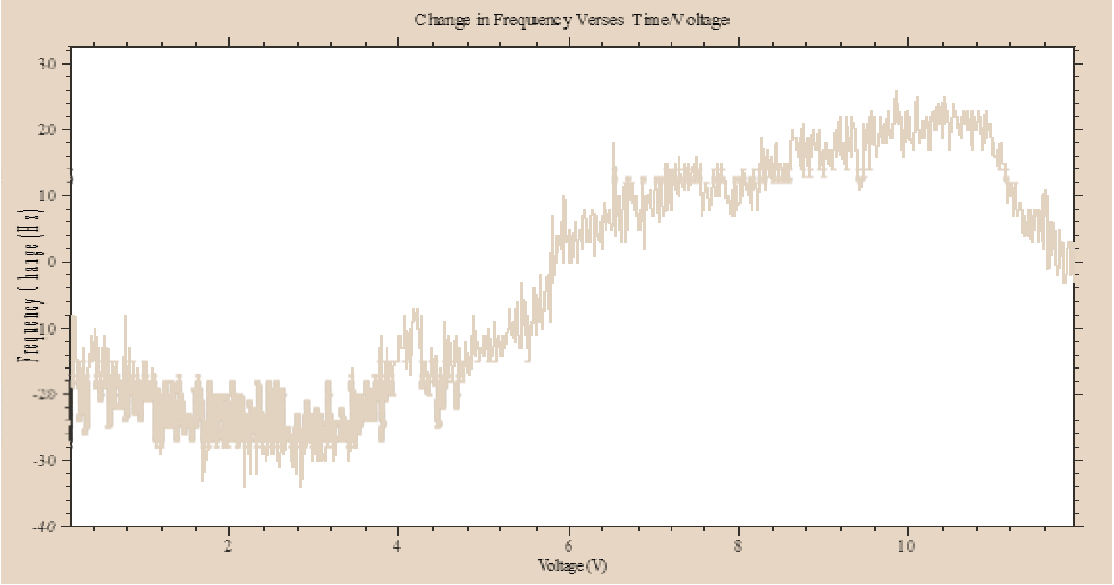


Figure 4.43: Bond Rupture scan performed over 30s using the magnitude method. (a) First scan minus the second scan. (b) Second scan minus the third scan.

The Bond Rupture magnitude method suffers from a problem of not being able to find the frequency lock when the frequency too far away from the resonant frequency at the start. This can be solved by performing a frequency sweep is run at the start of every run to ensure that the resonant frequency is close to correct. Another problem is that if a sharp change occurs such as from liquid being introduced or a possible sharp Bond Rupture the frequency lock could also be lost. For this reason, as well as out of interest into the other quality variable that has thus far not been measured, the Zero Count method was developed.

The Zero Count method drives the QCM at near the resonant frequency and then turns off the driving signal to examine the free frequency response of the QCM. The frequency that the QCM operates at this time is the true resonant frequency. The quality factor of the QCM can

also be attained by examining the decay of the signal gathered. This method enables a value to be taken from the noise signal at the third harmonic by examining this frequency while the run is in the driving mode.

A number of experiments were carried out using the zero count method. The first experiments were using 2µm and 4µm Streptavidin beads attached to the surface of a QCM with biotin attached to the surface. The QCM was ramped from a voltage of around 0.1 to 15 Volts over a period of 30s during the experiments. A Bond Rupture frequency change and the quality factor were measured as shown in Figure 4.44 for 2µm beads. Though the quality of the QCM does not seem to change much over the Bond Rupture time, more accuracy in this measurement would enable a better comparison. The frequency measurement accuracy is not as good as the magnitude method and is more dependent on the size of the signal received, so accuracy is decreased significantly at the start of the voltage run.

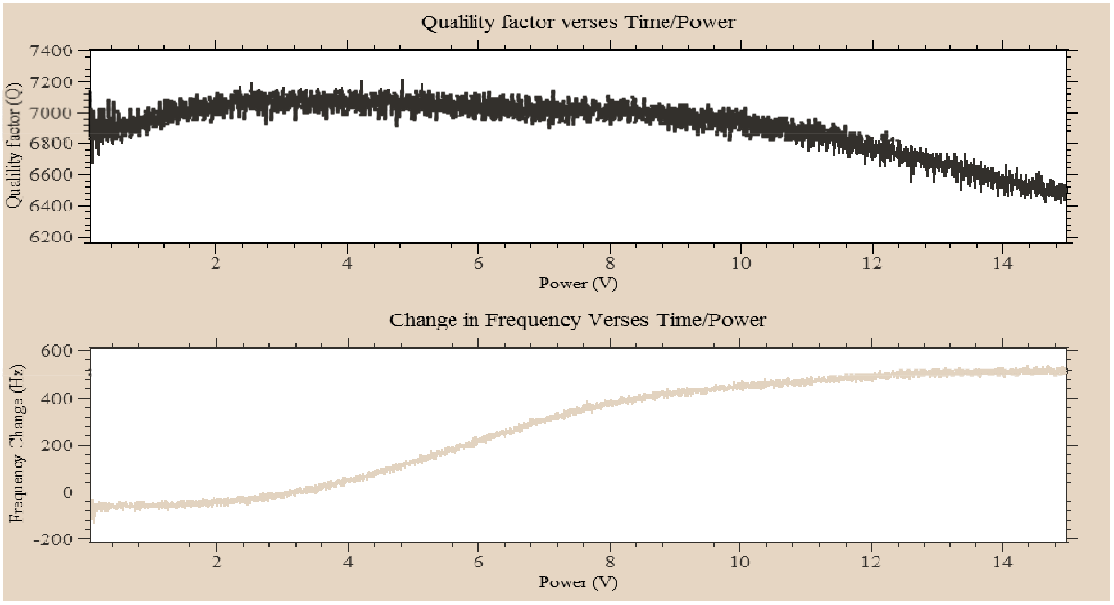


Figure 4.44: Bond Rupture experiment using Zero Crossing method and 2µm beads.

An experiment was carried out comparing the 2µm bead in Figure 4.44 and 4µm beads in Figure 4.45. This found that the size or weight of the beads changed the average point at which the rupture occurs. With the 2µm beads the average rupture point occurred at around 6 Volts whereas with the 4µm beads this point was around 2 volts. The significance of this is that this sensor could be used to detect multiple bonds rupturing from the surface so enabling it to be used as a multipurpose antigen sensor.

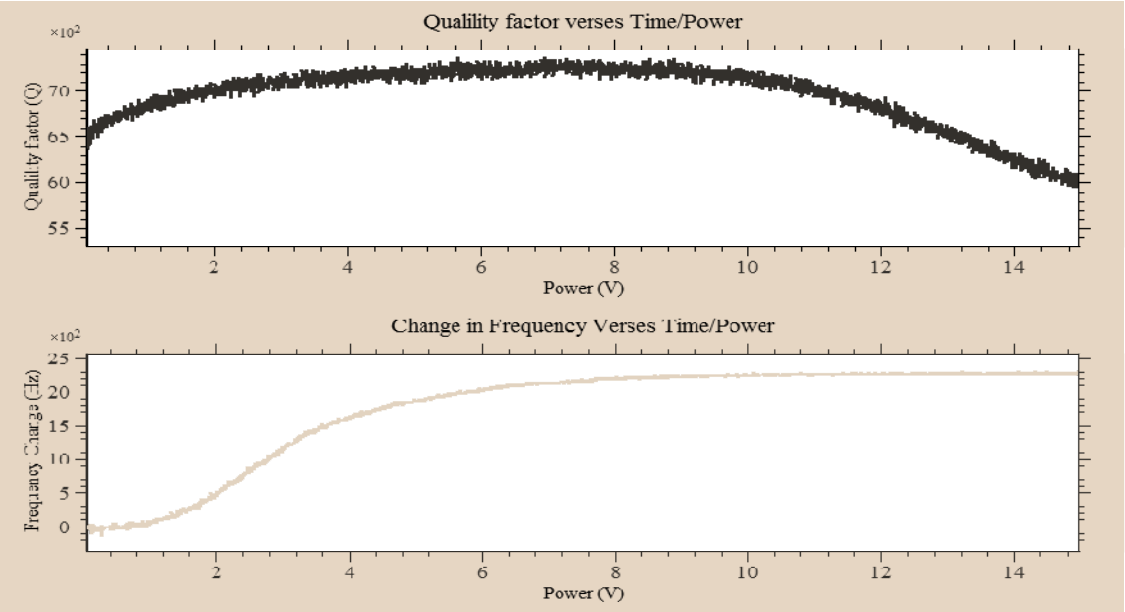


Figure 4.45: Bond rupture experiment using zero crossing method and 4µm beads.

The reliability of this sensor was also tested to see if this experiment was reliably rupturing around this point. This experiment was carried out using the 4µm beads and can be seen in Figure 4.45. Three experiments are shown in this figure which shows that the rupture starting point seems to stay the same, but the rate of rupture changes as well as the total amount of frequency change. The rate of rupture changing could be due to the impedance of the QCM's changing from one to another as this could change the amount of power that is transferred to the QCM. It could also be due to any liquid left on the surface after the nitrogen drying process has been carried out. The total amount of frequency change is due to the number of bonds that form during the binding process and it seems to be difficult to get consistent binding results. More investigation is required into this problem with a focus on the impedance matching of different QCM's.

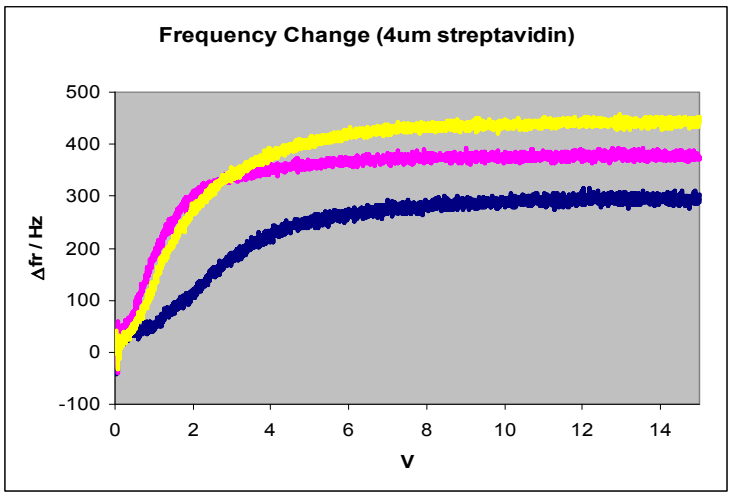


Figure 4.46: Frequency change with Zero Count Bond Rupture method of multiple experiments using different QCM's.

The zero count algorithm has provided important information to enable the Bond Rupture to be measured using a different method. Some problems were found with the zero crossing method. The first was that it does not operate as well at low signal levels as can be seen in the bottom of Figure 4.46. This was especially noticeable when the base signal was attenuated to increase the size of the 3rd harmonic noise signal. Another problem was that to measure the quality factor the time that the signal was sampled over needed to be enough for the signal to fall below the ~37% point needed by the algorithm to measure the time constant of the dissipation. This results in a loss in the ability to measure the quality in high Q environments as shown in Figure 4.47. This can be fixed by extending the sample time, but this can also have adverse affects on the frequency resolution at low signal levels.

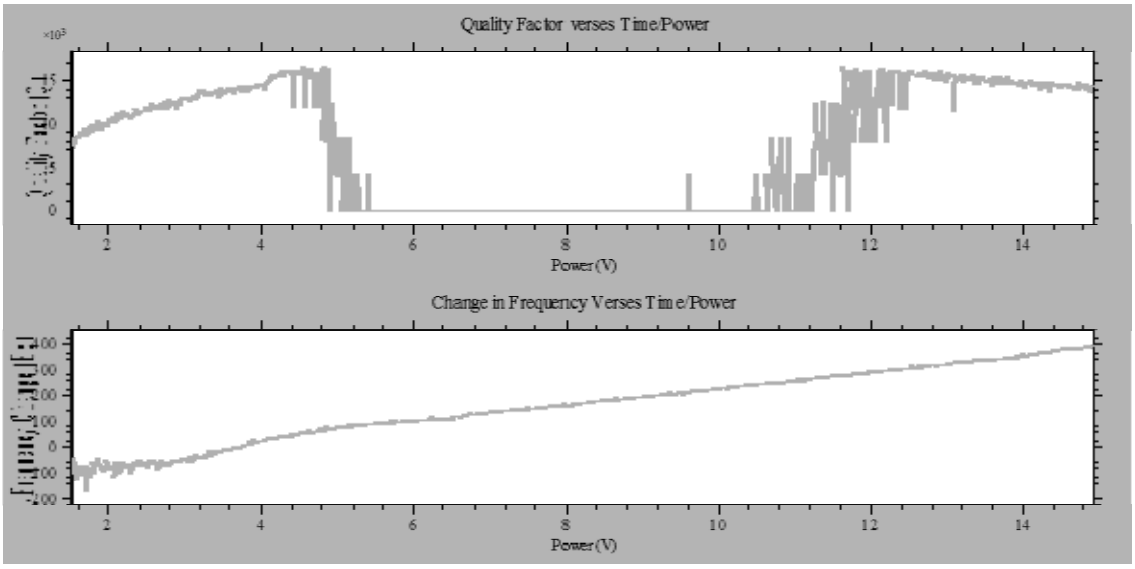


Figure 4.47: Frequency change using the bond Rupture zero crossing method, showing the quality error that can occur and the frequency change accuracy.

Overall the zero crossing is an improvement on the magnitude method but ideally an algorithm would use both methods (with some improvements to each algorithm) together to produce an improved algorithm.

The initial experimental results from using the Bond Rupture equipment was written up in the following paper (van der Werff, Yuan et al. 2005). After further testing at IRL a paper was written up with the results by Yuan (Yuan, van der Werff et al. 2007). This section will attempt to summarise the findings of the 2007 paper.

The experimental protocol for this is included in Yuan's (Yuan, van der Werff et al. 2007) paper. This involved attaching the biotin onto the surface of the QCM using a Self Assembling Monolayer (SAM). Streptavidin coated 6 µm polystyrene microspheres are diluted from 10 µL of 1.37% solid-latex-based microspheres solution into 5 mL of water then introduced to the surface by adding a drop (5µL) of this solution on the QCM's surface. This is then dried using a nitrogen gas source over the surface removing all liquid. The QCM is then placed into the equipment holder and the input voltage at resonant frequency is scanned from a voltage of 0.01 to 15 V over 60s. This is run twice and the second run is used for a reference for the first. If the first plot is subtracted from the second a plot as shown in Figure 4.48 is produced. This

shows the resonant frequency changing as the driving voltage increases. This shows that Bond Rupture is occurring as the frequency change indicates that mass has been lost. Another interesting effect is the large initial frequency change at between ~0.5 to ~2 V which, is theorised to be caused by beads that have not bonded to the biotin on the surface or non-specific bonds. There also appears to be a Bond Rupture in the range from ~3 to ~4.5 V. This would suggest that the non-specific bonds can be separated from the specific bonds so could be used to detect Bond Rupture occurring.

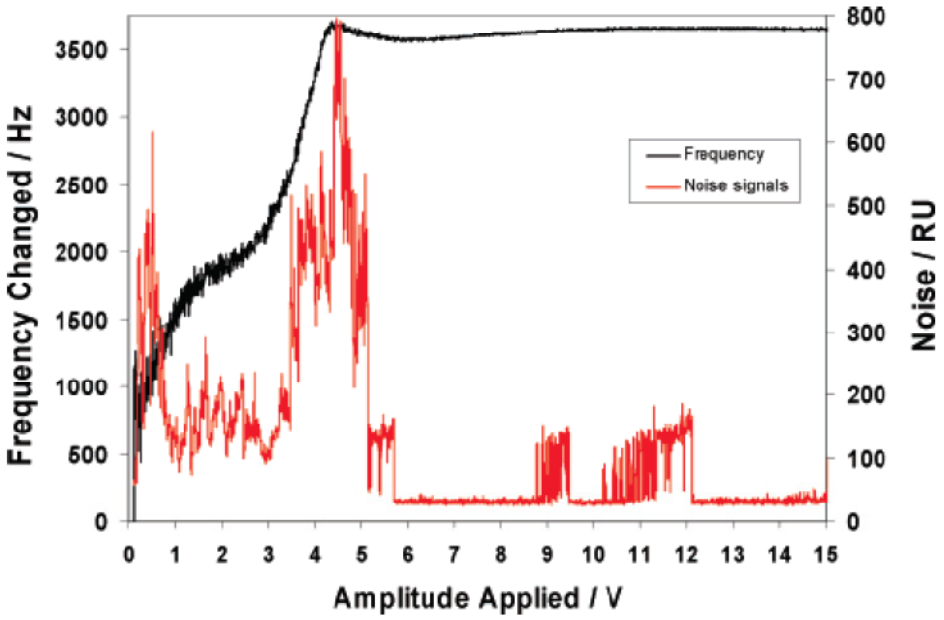


Figure 4.48: Profiles of specific and nonspecific bimolecular interactions by a dynamic Bond Rupture scanning mode. The black line is the resonance frequency change of the first harmonic frequency, while the red line is the signal level of the third overtone. (Yuan, van der Werff et al. 2007).

Another interesting feature in Figure 4.48 is the captured Bond Rupture 'noise' at the overtone of the QCM. Though this overtone graph shows noise appearing at the third harmonic, this did not seem to occur consistently throughout all experiments, which could lead one to question whether this is actual Bond Rupture 'noise' or some other effect introduced by the amplifiers distorting. To confirm Bond Rupture the amplifiers would need to be improved to lower the measured noise floor.

To confirm that the beads were actually rupturing from the surface a high powered microscope was used to examine the surface before and after Bond Rupture. This is shown in Figure 4.49 with (a) before and (b) after Bond Rupture respectively. It can clearly been seen that almost all the beads have been ruptured from the surface, proving that the Bond Rupture is occurring. This was also recorded on video which showed the beads initial rupturing from the centre and then rupturing out towards the edge of the QCM as the voltage was increased.

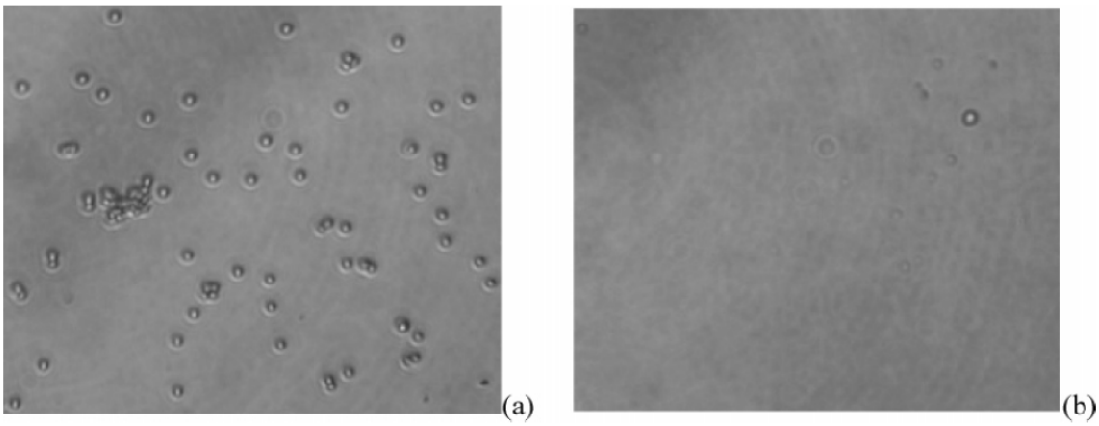


Figure 4.49: Photos (a) and (b) were taken before and after Bond Rupture scanning (Yuan, van der Werff et al. 2007).

In conclusion to this section the Bond Rupture effect was detected with the developed equipment with a few different algorithms finding that this effect can be detected using the frequency change method. The bond Rupture 'noise' may have been detected but further equipment development is required to confirm this is not some other effect or interference from the amplifiers.

4.5 EQUIPMENT LIMITATIONS AND PROPOSED IMPROVEMENTS

There are a number of limitations that were discovered within this digital transceiver system for the Bond Rupture application. The most significant limitation is that only one narrow band frequency can be examined at a time with this equipment. This limits the Bond Rupture 'noise' frequency to being able to be examined only within the time during which the frequency following algorithm is not using the digital receiver. Another major limitation is the number of data points that can be stored in memory - as if a wide band noise sample is to be examined a significant amount of storage is necessary to enable sufficient time resolution. The memory size also limits the number of resonant frequency samples that can be stored in memory, which limits the number of samples that can be gathered every second. These limitations and others have led to the decision to redesign the digital transceiver board for the Bond Rupture system.

If a redesign is to occur there are many aspects of this system that initially were adequate for the prototype now that more is known about the Bond Rupture effect there is a number of aspects that could be improved. One of the problems is the DDS's limited output resolution; it would be worth considering a DDS with a greater dynamic range and higher speed. It would also be useful to be able to generate a custom output waveform for other experimental techniques – yet to be conceived.

Another useful feature would be to be able to capture the ADC data directly into memory, as well as having a DRP, to enable a full view of the wideband input signal to the ADC. This would enable a significant amount of debugging capabilities as any problem of interfering frequencies could be quickly found and diagnose. This would also allow direct capture of what is happening at the time of Bond Rupture for offline signal processing. It would also be useful to increase

the speed of the ADC so to have a wider band-width input. It would also be useful for this device to be compatible with higher frequency SAW devices to coincide with the future development within this research group. There would also be a great advantage to having a higher dynamic range ADC to increase the accuracy of the data gathered and it would also be good to increase the bit range of the data out of the DRP device to enable the whole dynamic range to be taken advantage of. A secondary DRP is also required to capture the 3rd harmonic Bond Rupture 'noise' signal while simultaneously monitoring the resonant frequency of the QCM.

From a system developer point of view a significant amount of development would be required to improve and standardise the communication bus between the DSP and digital transceiver. This would include some sort of address management to simplify the addressing method of the digital transceiver system. Such a development could enable this board to be reconfigured for other sensor based applications. It would also be good to develop the new digital transceiver board with the idea to eventually develop it as a standalone Bond Rupture device.

The output and input amplifiers of the current Bond Rupture system are not adequate for the application so a new board is required to be developed. Among the problems is the output VGA is limited in the amount of gain range and also the number of output steps available. The output total power is also limited with a external amplifier required to achieve Bond Rupture. The new output amplifier would need to achieve the same output power as the current external amplifier achieves (~1Watt). The input amplifier is also limited in the gain range, noise figure, and the maximum input voltage that it can handle. Investigation needs to occur into more suitable devices for this.

Another aspect that needs to be examined further is the impedance matching and this needs to be examined in more detail to find a better solution than what is currently used. One aspect that should be further examined is filtering out the fundamental frequency so that it is at a similar magnitude to the 3rd harmonic. This could enable the Bond Rupture to be better detected.

There is a significant amount of additional development that could be performed into making the software user friendly and less error prone. Since there are a number of hardware changes necessary it would make good sense to implement the software changes on the new system. Among the changes required would be combining both the magnitude and zero crossings frequency measurement methods in order to take advantage of both their strengths together. Another improvement required is resonant frequency measurement resolution as this can be limited especially at low operating voltages and so improvements are required to the measurement algorithms. There is also a significant amount of improvement that could be done to the calibration methods as the current method used only works with one configuration. It would also be useful if we could calibrate for multiple different boards to be plugged in.

4.6 CHAPTER CONCLUSION

Within this chapter a Bond Rupture system has been developed, and Bond Rupture via the frequency change method has been verified. Frequency change has been measured using two main methods, magnitude method, and the zero count method. Both these methods were tested finding that the magnitude method tends to have better frequency accuracy over a large dynamic range but is not so good at following fast frequency changes. The zero count method has poor frequency measurement capabilities in low dynamic range situations but also can gather the dissipation value of the QCM. The Bond Rupture frequency methods when combined with voltage ramping, managed to produce the Bond Rupture effect as beads were measured rupturing from the surface. This proved that the Bond Rupture effect exists, but the Bond Rupture 'noise' has not been able to be measured as yet. This equipment has been used to perform further frequency Bond Rupture tests at Industrial Research Limited (IRL) for further verification and in the search of applications for this technology.

Though this equipment is working fairly well there are still a number of issues that cannot be addressed with software redesign alone. Therefore it was decided that the digital transceiver board and amplifier hardware would be redesigned to address all of these issues raised so far, using a FPGA device as the basis for the new system enabling any future problems to be addressed with software redesign as opposed to hardware.

5 REPROGRAMMABLE DIGITAL TRANSCEIVER SYSTEM

5.1 INTRODUCTION

Though Bond Rupture (via frequency change method) was verified with the digital transceiver system in chapter 4, the Bond Rupture noise had not been detected due to either a limitation in sensitivity or due the Bond Rupture noise effect not existing. As the limitations with the previous system were with the digital transceiver and amplifier boards a need was clearly seen for a redesign of these. There were also many other factors that need improving so this chapter attains to the further hardware and software development that was required.

The most important specification needed for the new transceiver board is the ability to measure two frequencies at once. There are three methods that could perform this. The first method is using two digital transceiver boards in parallel. This would require two ADC's, DRP's and all the other components that go with it, as well as the DSP having to be able to process both data streams simultaneously. It also does not allow the ADC data to be captured directly. The second method would be to use one ADC and a dual DRP (also called a digital down converter or DDC) chip, or two one channel DRP chips. This method though reasonably simple, it does lack the ability to capture the raw ADC data, and is limited in what commercially available integrated circuits exist. The third option is to create a dual DRP in software using a FPGA (Field-programmable gate array).

FPGA's are made up of thousands of reconfigurable high speed digital blocks. This enables FPGA's to perform large digital number operations in parallel, such as implementing a DRP. An advantage of an FPGA is that they can be configured exactly to the application of interest and many peripherals can be attached. One FPGA could have two DRP's, as well as the needed FIR filters, external and internal RAM, control a high speed DAC, and have an address translator to remove all external address logic required on the previous board. The greatest advantage of the FPGA is the fact that it can be reconfigured for different applications especially relevant for experimental laboratory equipment development. This is as it can be reconfigured if a new problem presents itself. All these features make it much more attractive than an ASIC (application specific integrated circuit) DRP used in the current transceiver board. So to meet the broad range of specifications needed for this system a FPGA was chosen.

Since a new board was being designed the limitations of the old transceiver board were considered when designing the new FPGA board. ADC resolution was the first to be considered as a higher bandwidth and resolution would be beneficial to improving the frequency and Bond Rupture 'noise' measurement capabilities. The output resolution and speed of the DDS portion would also need to be improved, to enable more accurate voltage ramp control, as well as a lower output noise floor. The amount of glue logic (logic for controlling addressing and communication with the DSP) on the previous board was significant, which could be reduced by incorporating almost all the glue logic into the FPGA. External memory was to be added to enable capture of the raw ADC and processed data. The size and type of FPGA device to be used was considered to ensure that the application required will fit into the device used. All these topics will be covered in the sections to come.

The software design was also looked at again enabling to the resonant frequency measuring method to be improved significantly. This will be looked at in the software section where it will be discussed in some detail including the advantages and disadvantages to the previous software algorithms.

The finished FPGA digital transceiver board that was designed has many advantages over the previous version – the features are compared side by side with the previous transceiver in Table 5.1. Among some of the features is the ADC used. The ADC on the previous transceiver board was a 14bit (84dB of dynamic range) 65 MSPS ADC with an operation range of between DC and 250MHz, whereas the FPGA transceiver used a 16bit (96dB of dynamic range) ADC with a 105MSPS operating frequency and operating frequency of between DC and 750MHz. The new ADC allows not only more sensitivity but also allows a wider frequency to be measured relaxing the requirements for the analogue filters. It also allows using sub-sampling, to sample at up to 750 MHz as long as the band-width is limited to the Nyquist bandwidth requirements. Another major improvement was the faster and higher precision output DAC – waveform generated by a DDS in the FPGA. The DAC has an improvement from 12 to 14bits of which the dynamic range is 72 and 84 dB respectively with the output speed increased significantly – which reduces the output filter requirements. RAM was added to the FPGA transceiver board enabling capture of the ADC or processed ADC data stream directly. A DSP board with increased RAM was used enabling the storage of more datum points.

	Bond Rupture V1	Bond Rupture V2 - FPGA
ADC bits	14	16
ADC Sample Rate	62.5 MSPS	100 MSPS
ADC operation range	0-250MHz	0-750MHz
DAC bits	12	14
DAC Update Speed	125MHz	200MHz
Number of DRP's	1	2
DRP data bit width	16	24
DRP CIC max decimation	512	16383
DRP min BW	122kHz	610Hz
RAW data FFT	NO	YES
DRP ram	0k	256k x 16
DSP ram	32k x 24	256k x 24
RAW data grab	NO	YES
Standby mode	NO	YES
Current Usage (running)	?	800mA @ 7V
Current Usage (standby)	?	300mA @ 7V

Table 5.1: Comparison of the two created systems.

The FPGA introduced a number of features to the transceiver including the ability to design two DRP's within the same chip – enabling two frequencies to be monitored at once and enabling one to be buffered directly into onboard RAM. The FPGA has also enabled the data width output to be customised to 24bit data, which is the standard data width within the DSP, enabling the higher precision data to be captured by the DSP. Another major feature is the ability of the FPGA's to down sample by 16383 (6.1 kHz bandwidth which is much lower sample rate enabling longer storage of a data signal. A ten times decimation FIR filter was also implemented within the FPGA to enable decimation by a total of 163830 (610 Hz bandwidth).

The FPGA software also has the ability to capture data directly from the ADC by routing the 16bit data directly to the RAM at 100MHz rate. This enables a maximum capture time of 2.6 ms of the ADC which over this time 4Mbits of data is stored.

There were a number of additions made to make the software programmer's job easier such as addressing in a logical simple way by reading the writing to registers. There are also options included to enable up to 48bit writes and reads for updating the frequency words within the DDS. This component is called the address translator and this controls the communication between the DSP and all the internal components.

The new FPGA design enables much more power saving to occur and so would be useful for a portable device. This is performed by shutting down segments of the FPGA when they are now in use. There is also a power saving using the new ADC as this enables it to be shutdown externally via the FPGA. Switched mode power supply was also implemented to enable the supply of the low voltage high current supply the FPGA demands without huge current drawing from the supply and huge heat dissipation problems that would occur. All these factors result in a significant power saving on the previous version of which it could be increased further by using a low power FPGA device.

Once the device was chosen a development board was purchased to evaluate whether or not the two digital receivers could fit within this device size, and whether using a FPGA for this application was possible considering the speed. This was investigated using the Memec development kit using the XC3S400 device in conduction with the Memec P160 expansion board – which contains two 12 bit 50 MSPS ADC's and two 12 bit 165 MHz DAC's. The ADC input data was used to test the two DRP's, and it was found that these could be fit within the FPGA device The DRP's were also tested using an internally generated waveform running at 100 MHz to test the speed capabilities. The DAC's were also tested by measuring the output with an oscilloscope when driven with an internal DDS device finding that it performed well. After this initial testing it was decided to develop the first prototype board using this FPGA chip.

There was also further development into a new power amplifier board and high gain input amplifiers. Also a low noise amplifier was used on the front end as an input to the system enabling the system to get very close to the theoretical thermal noise restraints for measurement of the third harmonic. There was also a amount of work looking at different ways to drive and measure the response of the QCM.

This remainder of this chapter covers a variety of issues pertaining to the development and testing of the Bond Rupture hardware. These changes include: Hardware changes, developer environments used, design and verification of FPGA transceiver board, software algorithm improvements, low noise threshold measurement, and experimentation.

5.2 HARDWARE

5.2.1 INTRODUCTION

The hardware is the backbone on which all the software is built and so getting the hardware right is important. This section outlines the hardware used and the reasons for using it. This includes the Transceiver board and brief description of the new DSP board used as well as the amplifiers and QCM impedance matching board.

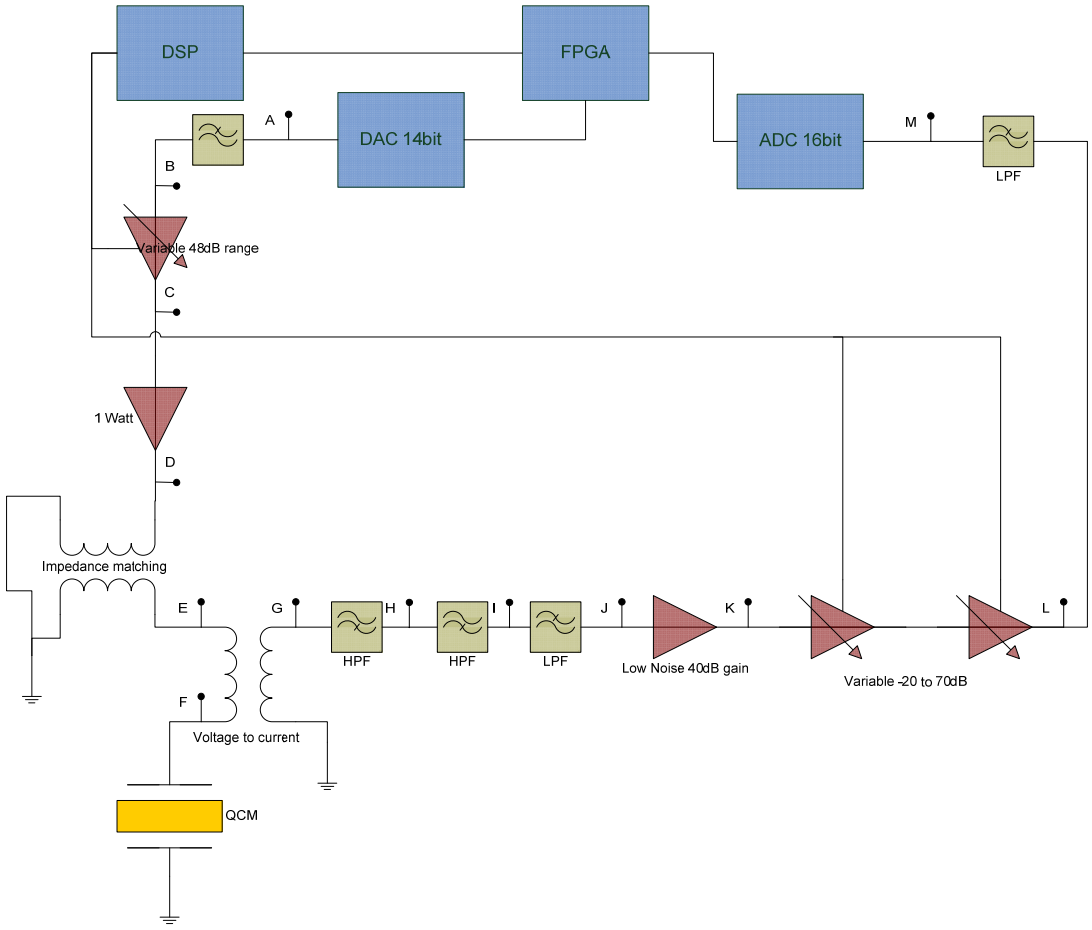


Figure 5.1: **System diagram of the new FPGA based system.**

The hardware diagram is included in Figure 5.1. This system is based around the Motorola DSP used in the previous chapter but with improved storage capabilities. The DSP communicates with the variable output and input amplifiers as well as with the FPGA digital transceiver board. The Digital transceiver board performs a number of different functions including generating the digital waveform using an internal DDS to output to the 200MHz DAC. It also handles down sampling and filtering of the incoming data from the 100MHz ADC. There is two output amplifiers in series enabling a total of 48dB gain range and a 30dBm (1 Watt) maximum power output. This is then connected to the impedance matching transformer and thru the current measuring transformer to the QCM. The tap off the current measuring transformer is then high pass filtered to remove some of the fundamental frequency while keeping the third harmonic for the Bond Rupture 'noise' measurement. This signal is then fed to the Low Noise

Amplifier (LNA) and then fed to two high variable gain amplifiers in series enabling a large total gain with a very low noise figure - for the purpose of 3rd harmonic Bond Rupture 'noise' detection.

5.2.2 TRANSCEIVER PCB

If the DSP is the brains of the Bond Rupture system, the transceiver board is the arms and legs, providing a way to interface to the outside world or in this case the sensor. The transceiver does this using a combination of a DAC drive the transducer and the ADC to listen to the response. Because of the high data rate that the DAC and ADC drive and return data at, this cannot be controlled by the DSP. To get around this the FPGA is used to generate the waveform for the DAC, and convert the high bandwidth data from the ADC to a narrow band data rate the DSP can manage. This is what the transceiver board does basically. There are a lot more components needed to make this FPGA based transceiver operate, of which the rest of the 'Transceiver' section covers the hardware components. Some of the other components that will be covered are the: RAM, DSP communication port, flash memory for loading the FGPA, JTAG for loading of the FPGA software and debugging, clock generation, and power supply.

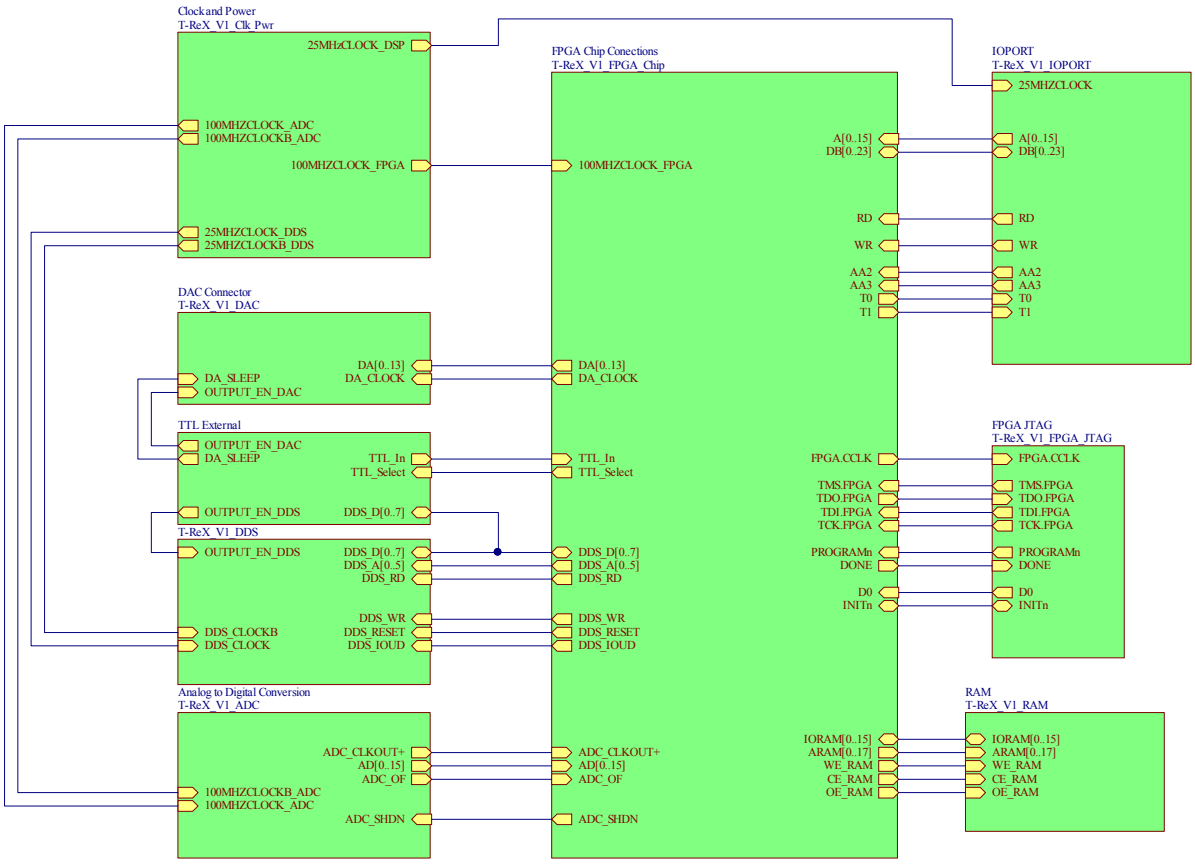


Figure 5.2: Schematic documents making up the T-ReX board.

The FPGA Digital Transceiver prototype board developed was named T-ReX (Transmit Receive eXtension board) and maybe referred to within the thesis from time to time. The main schematic for the T-ReX board developed is shown below in Figure 5.2. This shows all the

schematic documents and the connections between each of them. As can be seen the FPGA schematic is the central document with all of the other documents connecting to the FPGA. The individual schematic documents will be explained in the following sub sections.

5.2.2.1 FPGA

The digital transceiver device in chapter 4 was based on an Analog Devices™ DRP device. This device is limited to one quadrature channel at a time to be outputted. To create a digital receiver processor that can process multiple channels without needing multiple chips, a FPGA (Field Programmable Gate Array) was used. A FPGA is a grid of digital logic blocks that are programmatically routed (in Figure 5.3), enabling any almost any digital application to be implemented on a single chip. FPGA's are an affordable, and usually a low volume alternative to ASIC (application specific integrated circuits) such as a DRP chip. It has only been in the last few years that FPGA's have reached the point where their speed, size, and price has enabled them used in the place of tradition ASIC devices. FPGA's can also contain specific devices such as embedded multipliers, block ram, and even embedded microprocessors enabling them to contain a whole digital system on a chip.

There were a number of different FPGA manufacturers and devices that were considered when choosing a FPGA. There are some considerations to be made when looking for FPGA devices, among them are how readily available are the FPGA to obtain in prototyping quantities as some devices and packages are difficult to obtain. Another consideration is can the device chosen be routed and soldered with the tools and materials available, as many FPGA devices use BGA (Ball gate array) chips, which need to be placed on to the board with special methods and cannot be soldered traditionally. These can also require many layer boards for routing of the signal and power tracks. The leading FPGA manufacturers are Xilinx and Altera. Both of which have a large number different devices, ranging from one dollar per chip for the lowest device, to over a thousand dollars per chip for a top of the range device. The other important factor to consider is the software that they both supply, with extensive investigation finding that they supply almost identical basically free software and a wide variety of IP cores, of which most also are free. Basically due to the similarity of the products, the choice between the companies came down to previous experience that this author had had with programming Xilinx FPGA's. The choice of device was also reasonably challenging as there are a wide number of devices that are produced by Xilinx. The Choice was made to use a Spartan 3 device which was the fastest family of devices without having to spend an excessive amount of money. Within the Spartan 3 range there are a number of device sizes ranging from 50k to 5M of reconfigurable logic gates. The device needed to be able to be soldered easily for development purposes, so the largest device non BGA device was chosen. This was the 400k gate device (XC3S400), which has a 208 pin QFP (Quad Flat Pack) package and costs about $US40 for low quantities.

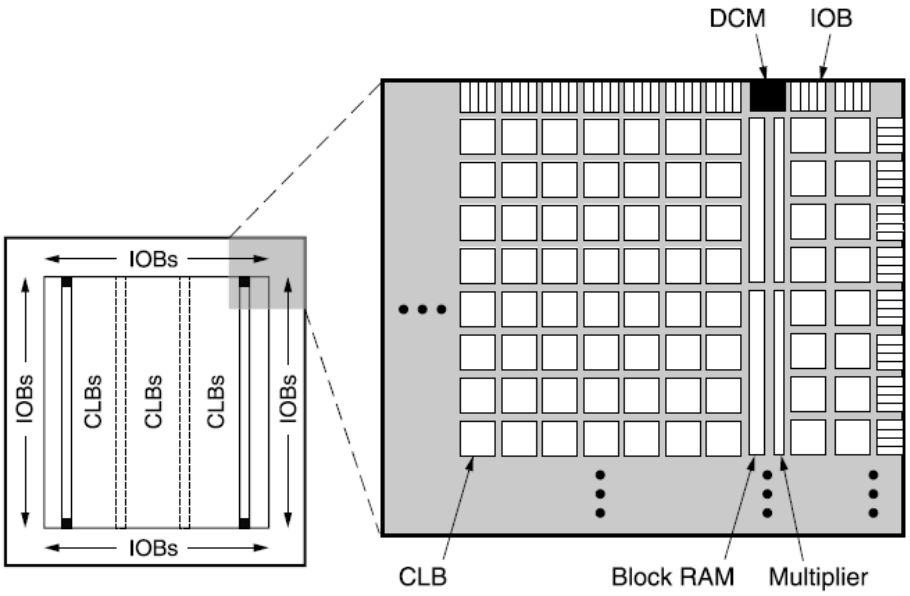


Figure 5.3: FPGA diagram from www.xilinx.com.

The FPGA device chosen is made up of 7 banks with a total of 141 pins of user IO with the rest of the ports used for power connections. The figure may sound like a lot but it is taken up very fast by the all the peripherals. The connected FPGA device is shown in Figure 5.3 with all the different banks and ports connected. Once all the devices were connected to the FPGA, there was a one free port. The separate banks correspond to the different segments around the FPGA, so to minimise routing and signal delays ports where kept as close as possible to each other. With a FPGA each bank can be set to a wide variety of different voltage standards, this board was all kept at 3.3V for communication making the bank assigning much simpler.

Other than the main I/O banks there are also the GND bank and power bank as well as the JTAG bank. Since the FPGA is basically made up of static RAM, the programs stored in the FPGA are lost when the power is switched off. To enable programs to be started up Flash memory is used to store and upload the program to the FPGA on power on. The Flash ram can be setup to upload serially via JTAG or some specialty IO ports can be used to load parallel Flash memory and there are many other types of memory interfaces supported. Serial Flash memory was chosen for this application so as to minimize the amount of IO used. For this application the Flash memory was connected serially by incorporated the memory into the JTAG chain. The JTAG chain can be connected to via a computer and the FPGA bit stream can be sent to the FPGA or to the Flash memory. Another option is that JTAG devices can be controlled within the FPGA as well as using built FPGA in debugging functions.

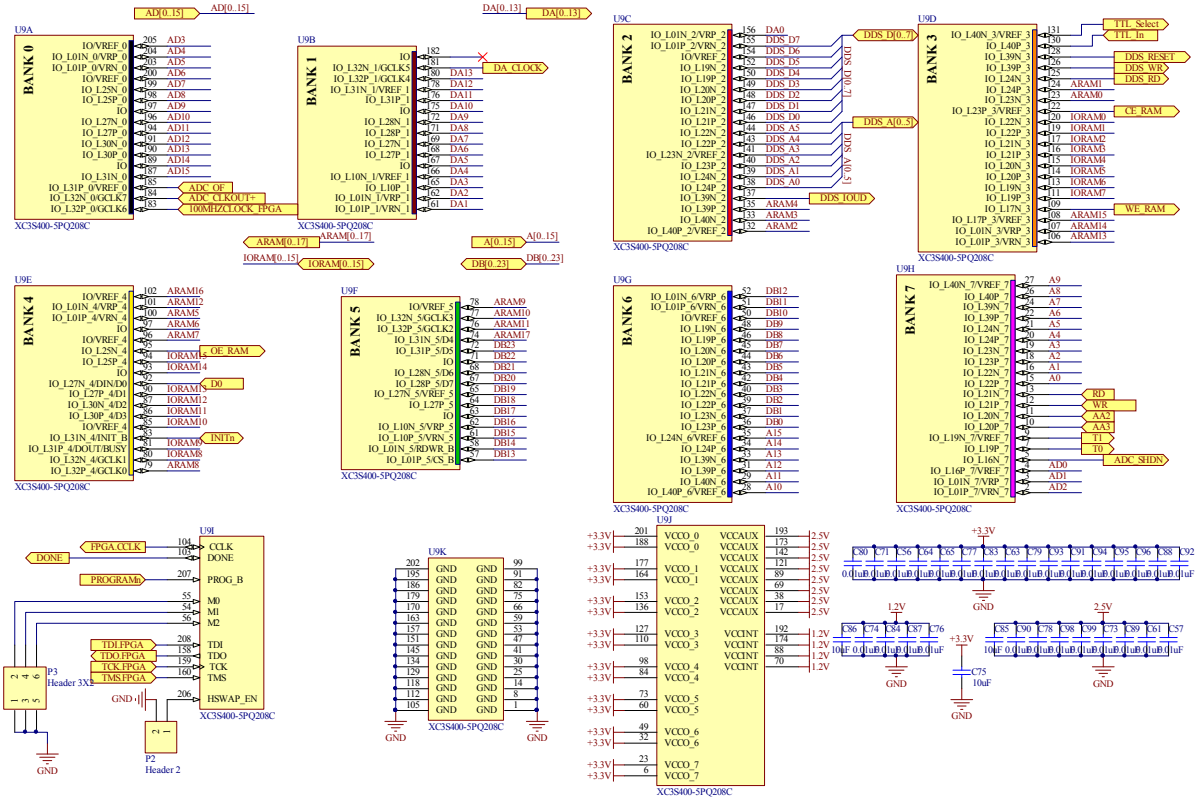


Figure 5.4: FPGA Schematic.

To enable extension of FPGA output ports a latch was implemented on the DDS data port. The data is clocked into this latch when the select pin (TTL_select) pin is changes from low to high (see Figure 5.5). This enables an extra 7 output pins. These pins are used for the DAC's sleep pin (DA_SLEEP), the analogue switches output enable for the signals coming out of the DAC and the DDS chip. There are also 5 extra external pins for arbitrary use. In addition there is one TTL input for a trigger.

The FPGA needs three separate power supplies and some times more when multiple voltage communication standards are used. The communication supply is usually 3.3V (though can be set to a different standard for each bank) and with 141 communication pins the current drawn can be quite high if all at once. For a device of this size at least 1A is required for the communication to be stable. There is also a supply used to communicate with JTAG and the Flash, this is set at 2.5V. This supply only needs to be low current and around 300mA is ample. The last supply is to supply the core of the FPGA which operates at 1.2V. For a device of this size around 1A supply is needed to keep the device operating consistently. The power requires of the system is examined more fully in the Power Supply section. The decoupling on the FPGA is very important as the FPGA can potentially have very high peak currents, so each power pin was given a 10nF decoupling capacitor with a 10uF tantalum capacitor for each supply close to the chip.

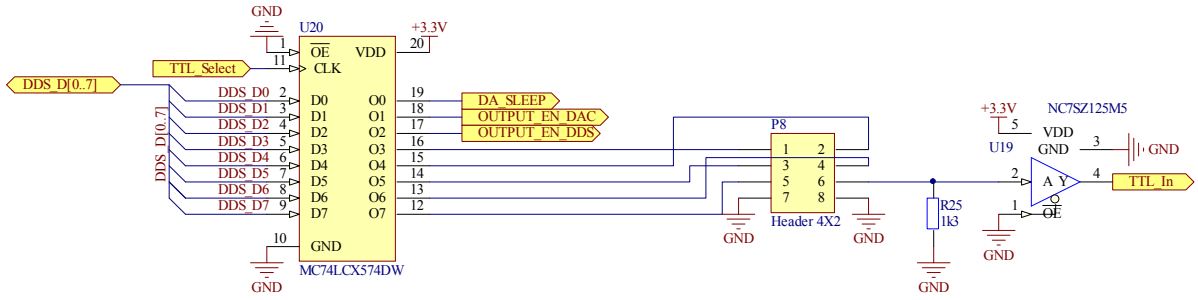


Figure 5.5: Latch to extend FPGA ports and TTL trigger input.

The FPGA device used is made up of SRAM so cannot remember its state after the power is turned off. To enable a program to be able to start up on the FPGA a Flash ROM is used to load it. The Flash memory used (see Figure 5.6) is made especially by Xilinx for this purpose. The chip used is the XCF04 which is a 4Mb flash chip, with a minimum data retention period of 20 years. The Flash memory is incorporated serially into the JTAG chain and so can be programmed via JTAG interface on the PC. When the FPGA starts up it automatically loads the program from the flash into itself and starts the program running.

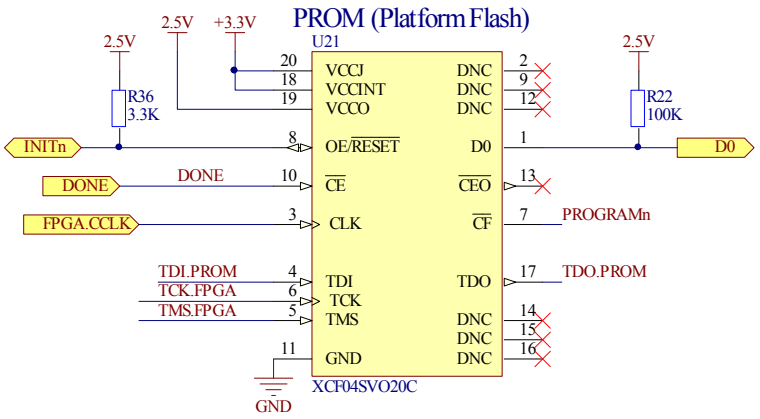


Figure 5.6: Flash chip connections - Xilinx XCF04.

The JTAG chain connector enables the communication between the PC and the FPGA or Flash. This is enabling a program to be loaded into the FPGA as well as the debugging of the internal hardware/software using Xilinx's chipscope. A Xilinx programmer was used which plugs into the back of the PC via the parallel port.

The FPGA has special ports for clock input to enable low skew and direct routing to the DCM (Digital Clock Manager) modules. There are two clock inputs for the FPGA, first the main 100MHz system clock, and secondly a clock from the ADC which is also running at 100MHz. These clocks can be converted internally using the DCM's, enabling almost any internal clock frequencies to be generated. An example of this is this 100MHz clock is converted up to 200MHz to run the DAC. This is not used for driving the ADC clock input due to the high jitter introduced within the FPGA. The clocking circuits will be examined in more detail in the next section Clock Generation.

5.2.2.2 CLOCK GENERATION

An important part of the transceiver system is the clock. The clock frequency, jitter, and line impedance are all important factors that need to be considered when designing a board as they can affect the speed of operation, accuracy of ADC acquisition, and reflection interference. To enable high frequency acquisitions the clock must have very low clock jitter. This is especially important when looking at under sampling as when the jitter is high bit resolution is lost - under sampling is where the frequency being examined is higher than Nyquist frequency. Under sampling would be important for use with high frequency SAW devices and in the interests of making this board as multipurpose as possible this was considered.

The oscillator used is a LVDS (Low-voltage differential signalling) output with a 100MHz output frequency – the oscillator circuit is shown in Figure 5.7. It has a maximum RMS jitter value of 1ps and to minimise any other circuitry such as buffer that add to the jitter noise this frequency is routed directly to the ADC. The oscillator is also fed to a buffer (MC100EPT23D) that converts from the LVDS to a single track TTL signal for the FPGA clock input. A 25MHz clock was also required for the DDS chip and also to fed back to the DSP. This was implemented using a divide by 4 clocking chip (MC100LVEL33D), so taking the 100MHz clock and converting it to a 25MHz clock. This 25MHz could have generated within the FPGA but due to pin restraints an external chip was used. A set of resistors were used to ensure that the clock impedance match was correct.

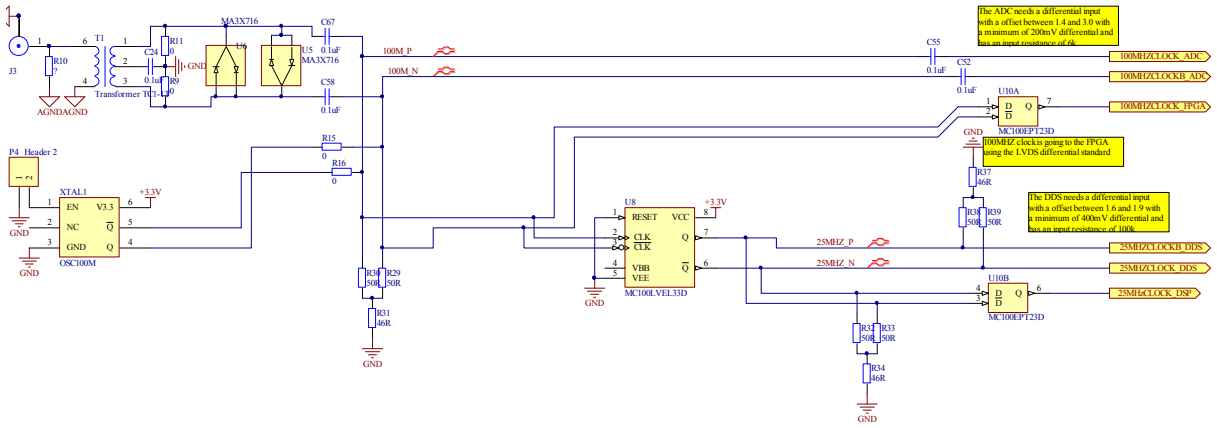


Figure 5.7: Clock generation circuit for the transceiver board.

There was also an input for an external clock to enable low jitter experiments to be carried out. This has a smb connector input to a mini-circuits transformer – to convert the single ended input to differential. It also has two clamping diodes (MA3X716) to limit and protect from overvoltage. If this is to be used the onboard Oscillator must be disabled using the jumper P4 and resistors R15 and R16 must be disconnected. The capacitors C67 and C58 are used to connect the external clock to be able to operate.

5.2.2.3 RECEIVER – ADC

There are many different ADC devices on the market from a large number of manufactures. The leading manufactures at the time for high speed and high resolution ADC's were Analog Devices, and Linear Technology. Both Analog Devices and Linear Technology produce fairly similar devices with in a similar price range but the linear technology device has more useful features.

The Linear Technology 105MSPS device (LTC2207 - see Figure 5.8) was chosen because among other features the ability to be able to shut down via a power down pin. This enables a significant power saving when the ADC is not running. This ADC also has a function to enable a semi randomisation of the output bits to prevent interference from the current draw when driving the output. Another nice feature is the dither function which results in reduction of unwanted tones at the small expense of a very slightly higher noise floor. The other useful feature is the PGA pin which when disabled the input range is $2.25V_{pp}$ and when enabled $1.5V_{pp}$ which is a gain of 3.5dB when enabled. The other useful function is the overflow pin (OF), which is updated at the same rate as the data outputs, enabling overflow to be recorded and gain to be adjusted accordingly.

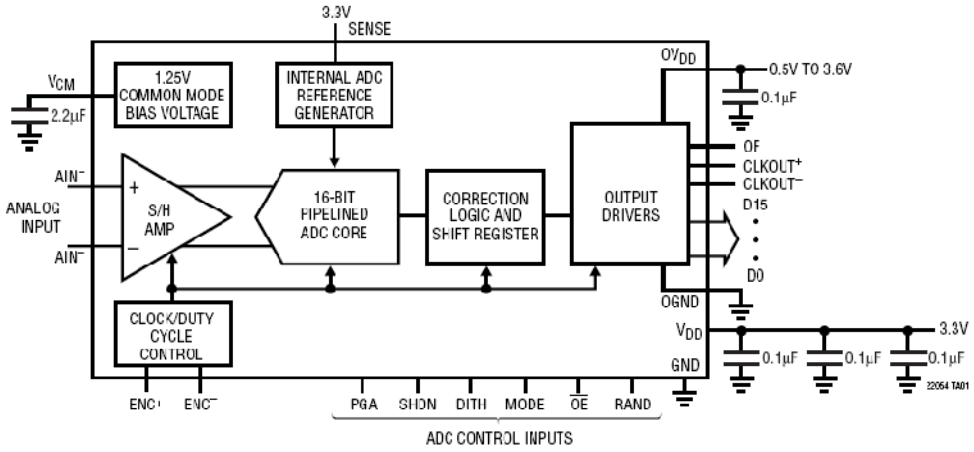


Figure 5.8: Linear Technologies internal makeup and external interface pins from the LTC2207 datasheet (ref).

The circuit schematic in Figure 5.9 shows the design of the ADC side of the circuit. The 16 bit data line (AD[0..15]) and ADC_OF are updated at the ADC input clock rate after a delay of eight clock cycles introduced via the pipeline ADC. The clock is also passed though the same delays as the sampled data and outputted as ADC_CLKOUT. The only other signal is an input that the FPGA uses to put the ADC into shutdown mode using ADC_SHDN. There are also another three pins for controlling dither, randomise, and PGA, which are controlled using jumpers on the transceiver board. These were not controlled via the FPGA due to the lack of spare ports.

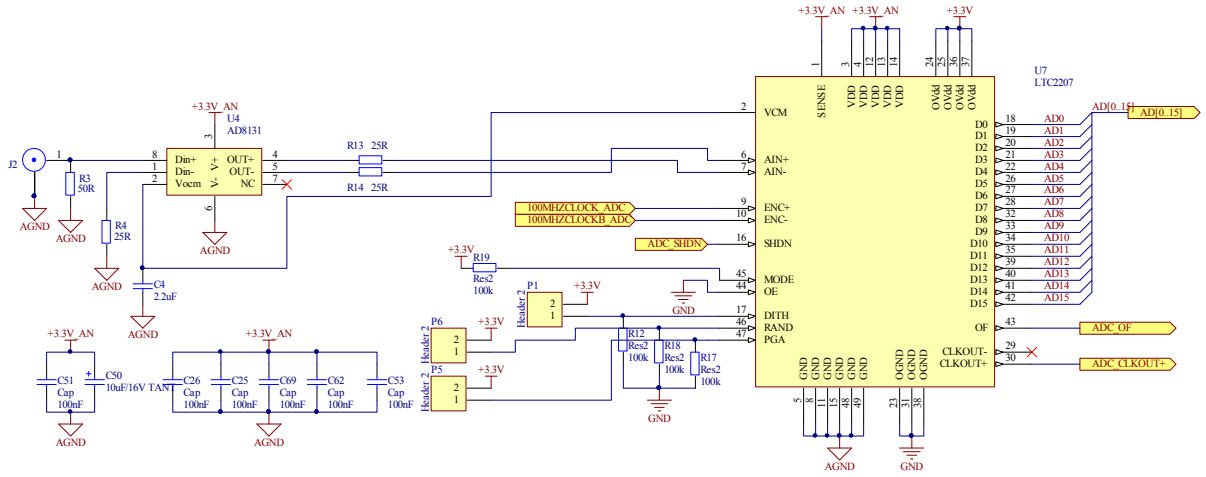


Figure 5.9: ADC Schematic.

To minimize noise added to the analogue supply from other components the ADC is supplied with two separate supplies and grounds to minimize the interference between the analogue and digital components.

The ADC is fed the 100MHz clock directly (see Figure 5.9) to minimize jitter introduced by any clock processing components. The sample and hold jitter on the ADC is $80fs_{RMS}$ which is the limit of jitter that is usually introduced by a top of the range clock. The jitter limits the bit accuracy of the ADC at high frequencies - as the higher the frequency the less time that a voltage can be accurately determined, and so the actual voltage is not known. Jitter is usually quite high phase lock loops like the DCM's within the FPGA, hence why the clock is fed directly to the ADC rather than though the FPGA. The clocking circuit has been designed so that an extra low jitter external clock can be used instead of the onboard oscillator.

The input to the system is a single ended 50ohm with a smb connector. This needs to be converted to a differential input before reaching the ADC at a voltage offset specified by the common mode voltage output (VCM). This conversion can be achieved with either a transformer or a buffer amplifier. A transformer has the advantage of very minimal signal distortion and also low noise addition. On the other hand the buffer amplifier does provide a certain amount of protection from over voltage, as the buffer amplifier will be damaged before the ADC. The buffer also enables the input to be amplified by a small factor. The buffer also enables measurements down to DC. The buffer does have the disadvantage of a limited bandwidth. Overall after considering the two options the buffer option was chosen due to being able to measure near DC and also because of the protection given to the ADC.

The input buffer amplifier used is the Analog Devices AD8131 – see Figure 5.9. This amplifier works up to 400MHz, with a fixed gain of 2 (6dB). This uses a common mode voltage to set the voltage around which the differential voltage operates. By using the common mode voltage from the ADC any small fluctuations in this voltage supply do not affect the measurement accuracy as these fluctuations cancel out as they occur in both the ADC and amplifier. This amplifier has a fairly high voltage noise figure of $25nV/\sqrt{Hz}$ which is not the best however this noise should be smaller than the noise being inputted into the system. The maximum input specifications to the receiver is 0.75V with PGA pin enabled or 1.125V with it disabled. These

values are not exact as a small gain change occurs depending on the impedance matching, these will be experimental obtained once the system is complete for calibration purposes.

There are some on board jumpers that can be set to change input range (P5), RAND (digital output randomization) selection (P6), and dither (P1). The input range can be either 0.75Vpp (P5=1) or 1.125Vpp (P5=0), which is set via P5. The RAND function is set by setting jumper on P6 enabling Data bits 1 to 15 to be exclusive-or-ed with data bit 0, which when enabled gives a slight increase in the noise floor for a reduction of unwanted tone signal due to digital interference. To use the RAND function the ADC_RAND_IN control pin must be enabled. Dither is set by setting jumper P1, which gives increase of SFDR for small signals at the expense of a 0.5dB elevation of your overall noise floor.

When designing the PCB for this section of the board the track size was set to an amount that kept the impedance at 50ohms between the smb input and the buffer amplifier. The impedance was also kept constant at going into the ADC with the differential tracks running beside one another with ground either side. Also other tracks were routed away from the analogue input as much as possible to prevent RF interference. The digital tracks were also kept at a constant length and as close to the FPGA as possible to minimize line difference delays and also minimize RF pickup, as the lines are running at 100MHz

5.2.2.4 TRANSMITTER – DAC AND DDS

The transmitter section of the transceiver design is what drives the sensor. This enables the excitation of the sensor and so the Receiver can measure the response. This needs to be at the frequency the sensor operates at. The QCM sensor operates at around 10MHz for its fundamental frequency and at least 30MHz for the 3rd harmonic. So this system needs to be able to perform this at least, furthermore it is also required to be able to measure the 5th harmonic which is around 50MHz and even potential the 7th harmonic at 70MHz. To enable this large frequency range the output must be able to be at least 140MHz to satisfy Nyquist. To loosen up the anti-aliasing filter specifications a number of 200MHz was chosen. The frequency generated usually only needs to be a single frequency component at one time, but it would be advantageous to be able to generate more than one frequency output at one time or a custom waveform from memory.

There were two main options considered to generate this output the first was using a DAC controlled by the FPGA enabling the output to be anything that the FPGA can generate, enabling multiple signal frequencies or custom waveforms. This is only limited by what the FPGA can be programmed to do. The other method is using a DDS such as was used in the previous system. If the DDS is used the FPGA is not required to generate a real time output, but using the DDS limits the system output to a single waveform. This reduces the pressure on the FPGA code, as not as many components are needed within the FPGA. One advantage is that the DDS is guaranteed to perform at the 200MHz operating frequency, whereas the FPGA has not been tested driving the DAC at 200MHz and so this may not work! Because of this possibility and rather than designing two boards, both the DAC and DDS were placed on the designed circuit board.

The DAC that was chosen was the Analog Devices AD9744. This DAC can operate up to 210MSPS output and has 14bits resolution. There are three inputs to the DAC from the FPGA – see Figure 5.10. First there is the 14 bit data bus running at the input rate specified by the second input which is the clock. The third input is the sleep input which enables the FPGA to turn the DAC off when it is not in use. This DAC is a current differential output so this must be converted to a single ended voltage signal before being outputted from the transceiver. The current output is controlled by the resistor R20 and is set to enable the output to be 9.6mA of peak output current. This works out to be around 0dBm output for the 50ohm system used.

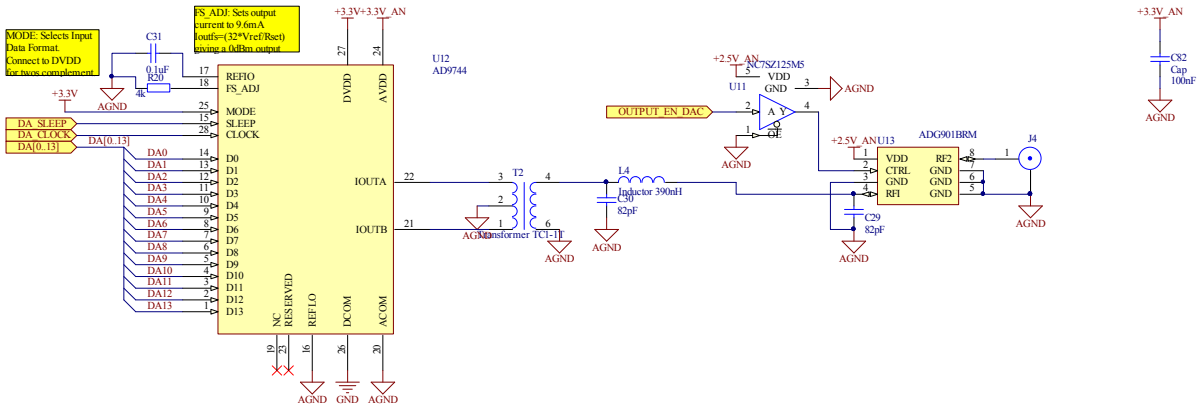


Figure 5.10: DAC schematic.

To minimise the error that could be caused by noise on the supply the DAC is supplied with two separate supplies and grounds to minimize the interference between the analogue and digital components.

Since the output from the DAC is differential current this is required to be converted to single ended voltage output to be compatible with the 50ohm system. There are a few methods of performing this. Usually either a transformer or buffer amplifier is used. Transformers main disadvantage is that they do not work at low frequencies however, since the frequency required would be greater than 1MHz this is not an issue. Premade transformers such as the TC1-1T can be purchased from mini-circuits having less than a 1dB variation over a range of 1 to 100 MHz, which is well within the range of interest. The buffer amplifier method on the other hand needs to be powered and also can introduce noise into the system. In the end the transformer method was chosen as there was no real advantage that would outweigh the extra circuitry required to implement using a buffer amplifier.

An anti-aliasing filter is required to filter the output signal from the DAC. This is as when a DAC is set to output a frequency this frequency is mirrored around the Nyquist frequency and is necessary to be filtered out so as not to interfere with the output. The filter used is a PI type filter which is a series inductor with a capacitor either side – see after the transformer in Figure 5.10. The filter used two 82pF capacitors and one 390nH inductor obtaining the magnitude filtering response shown in Figure 5.11. From this figure it can be gathered that if a 10MHz waveform is generated the 190MHz aliased region will be down 40dB providing significant filtering.

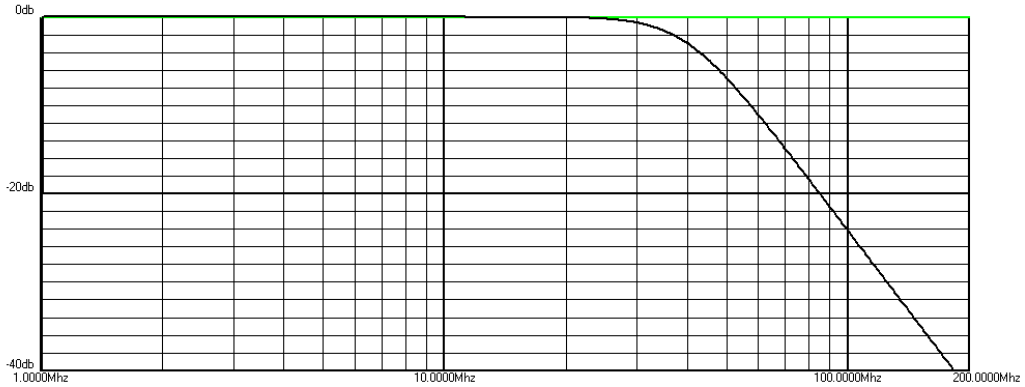


Figure 5.11: Magnitude response of the PI filter using AADE filter design software.

A digital switch is used to enable the turning on and off of the output without changing the DAC and can be switched on or off fast enabling generation of a sharp impulse. The switch used is the Analog Devices ADG901. This has an off Isolation (at frequencies less than 100MHz) of around 60dB. It also can be switched on or off within 10ns. This device switches between the output and 50ohms impedance on both the input and output side leaving the DAC still connected to a load.

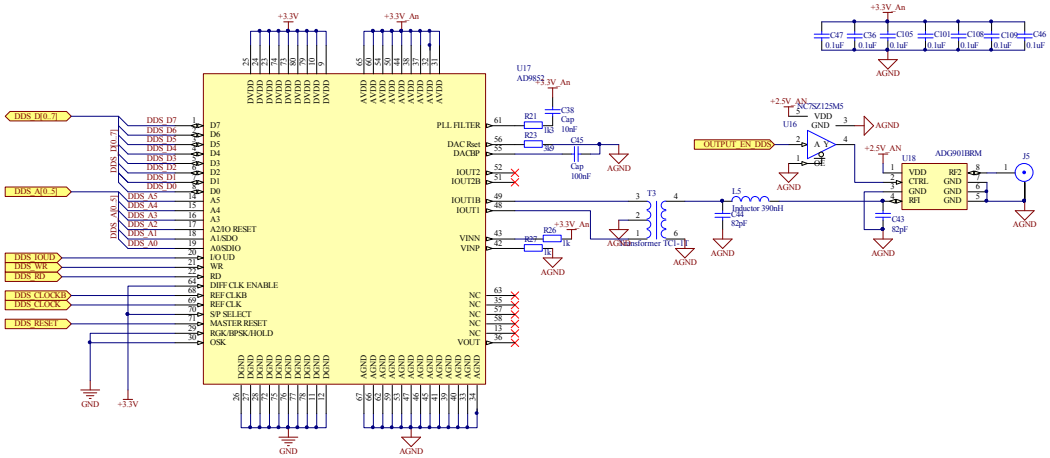


Figure 5.12: DDS schematic.

The DDS schematic is shown in Figure 5.12. This shows the ports that are connected from the DDS to the FPGA for control. The FPGA controls the DDS 8 bit data and 6 bit address ports as well as the DDS's IO pin, write, read, and reset pin. A 25MHz differential clock is sent to the DDS from the external clock circuitry. This 25MHz clock can be multiplied up to the frequency required by setting the correct registers in the DDS. The output current is set by R23 so the output will be around 0dBm. The DDS has exactly the same output hardware as the DAC - so previous paragraphs applying to the transformer, filter, and switch also apply to this.

5.2.2.5 RAM

It was decided that external RAM in addition to the RAM within the FPGA was needed to store the incoming data from the ADC and also DRP. To enable capture of the raw data from the ADC the RAM needs to be able to write at 100MHz x16 bits or at 1.6Gbits. To get these speeds

there was the option of either SRAM (static RAM) or a faster DRAM (dynamic RAM). SRAM is simple to use and uses direct addressing. DRAM can store more data though it needs to be periodically refreshed and is also not direct addressed. To get the same data rates with DRAM as with SRAM a much higher speed is needed to take into account the extra cycles required for addressing and refreshing. DRAM also needs a much more complex memory controller for the same reasons. Though DRAM has the advantage of being cheaper and larger amount of memory, to simplify the FPGA design SRAM was used.

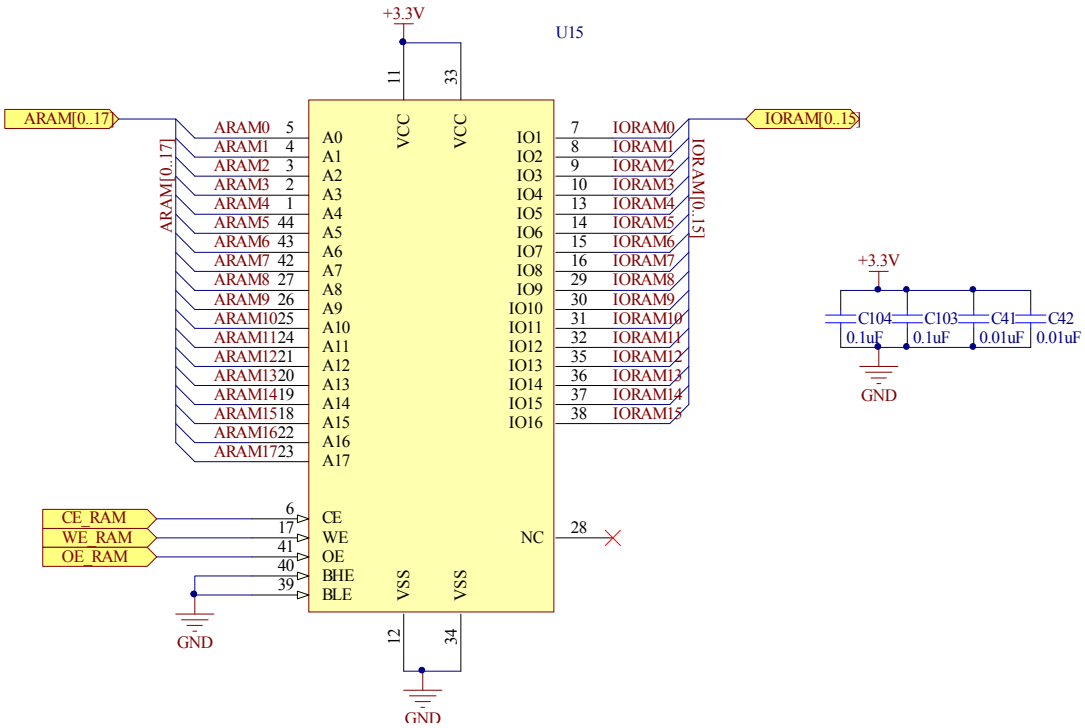


Figure 5.13: RAM schematic.

The minimum speed needed to store a data stream from the ADC is 100MHz with at 16bits. This requires 10ns write speed SRAM. The device chosen was the Cypress CY7C1041DV33 – seen in Figure 5.13. This device has 256k x 16 (4Mbits) of storage enabling the capture of 2^{18} (262144) samples. It is addressed using 18 bits and has a 16bit wide IO port. There are three control variables clock enable, write enable, and output enable. This device is connected to the same 3.3V supply that the FPGA runs off. The RAM used requires four decoupling capacitors two 100nF capacitors and two 10nF.

There are a total of 37 lines that are sent from the RAM to the FPGA which is about 25% of the FPGA used up by RAM connection. When routing these lines care must be taken to make sure the RAM device lines are similar lengths as well as not to far away from the FPGA.

5.2.2.6 EXTERNAL PORT – DSP COMUNICATION

To communicate to the DSP the transceiver was developed to be compatible with the backplane used by the previous transceiver device. This is a 96pin header is used which all the ports need for communication with the DSP are connected. The main communication lines to the DSP are the 24bit Data Bus and the 16 bit address bus. There is also some address attribute

lines (AA2 and AA3) that enable the DSP to select different address spaces to read or write from. There is the Read (RD) and write (WR) lines which the DSP uses to select whether to read or write to the FPGA. There are two interrupt lines which the FPGA can use to signal the DSP if need be (T0 and T1). The last line is the 25 MHz clock which is sent to the DSP. This ensures that the DSP runs at the same rate as the FPGA, preventing communication synchronisation problems. The connection circuit diagram is shown in Figure 5.14.

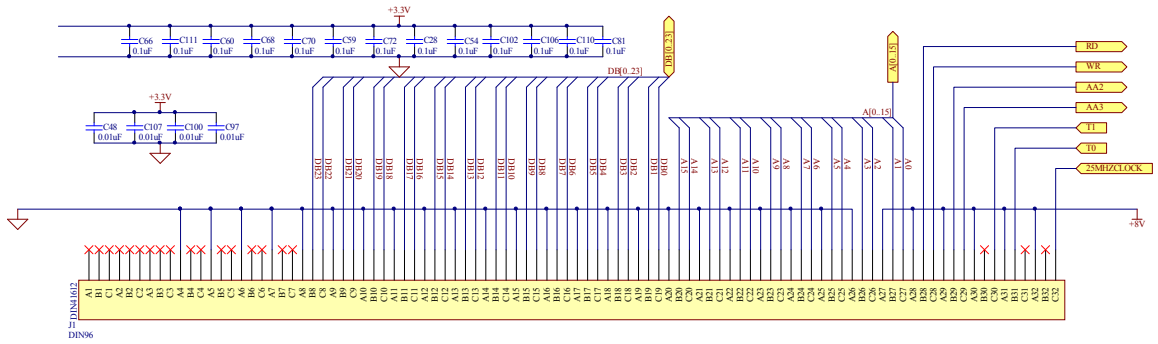


Figure 5.14: Circuit diagram for the connection to the 96pin header.

5.2.2.7 POWER SUPPLY

The FPGA requires three separate power supplies a 3.3V for peripheral communication, a 1.2V for the operation of the internal core, and a 2.5V for the JTAG communication start-up settings. The amount of current that the 3.3V supply needs source depends primarily on the number of output pins and their driving current setting and also a little on the number of input pins. To achieve a stable output for a worst case scenario the power supply selected to have an output current of 2 Amps. The current needed for the 1.2V supply depends on the amount of internal switching and so depends on the system clock rate and the amount of logic elements switching at one time. A worst case current usage can be calculated from the size of the FPGA chip. The design specification was set at 1 Amp. The 2.5V auxiliary supply only requires a small amount of current and so a design specification was set to 250mA.

There is a large variety of different devices that are for power supply use. The first device that was examined was a linear regulator, but because they dissipate the difference in voltage as heat and the FPGA requires a low voltage high current supply this would end up wasting too much power. In the case of 1.2V power supply and an input voltage of 8V 85% of the input power would be lost as heat. Obviously a better solution was required. So a SMPS (switch mode power supply) was chosen that was 95% efficient so only 5% of the input power is lost – a huge improvement. The down side of using a switched mode power supply is that more components are needed and also the components needed are expensive – a switching chip, a high current inductor, large capacitor with low ESR, MOSFET, and diode are needed to keep the voltage ripple small and current high.

There is a huge range of different SMPS controller chips available, from single power supply chips to multiple output controls, some even with built in MOSFETs and diodes. In the end a multiple output SMPS chip from Texas Instruments was chosen which is purpose built for supplying Spartan 3 FPGAs. This chip is the TPS75003 which has two SMPS controllers for up to

3A output and one linear regulator 300mA supply. The circuit diagram for connecting this chip is shown in Figure 5.15, with the 1.2V and 3.3V generated by the SMPS controller and the 2.5V supply by the linear regulator. The SMPS supplies were designed to supply 2A with the limitations being the inductor (value and current capability) and the capacitor (minimise the voltage ripple). This chip takes a input voltage range between 2.2V and 6.5V and so a linear regulator is still needed on the input to extend the range of the input from around 6.5V to 20V. This is done using a 5V linear regulator on the input to the SMPS – see Figure 5.16 5V supply. The 5V linear regulator used as the input voltage range to this board is 6.5V to 8V and to minimise ripple to the SMPS a voltage of 1.5V above supply is need. This voltage regulator can supply up to 5A of current so is within the parameters needed.

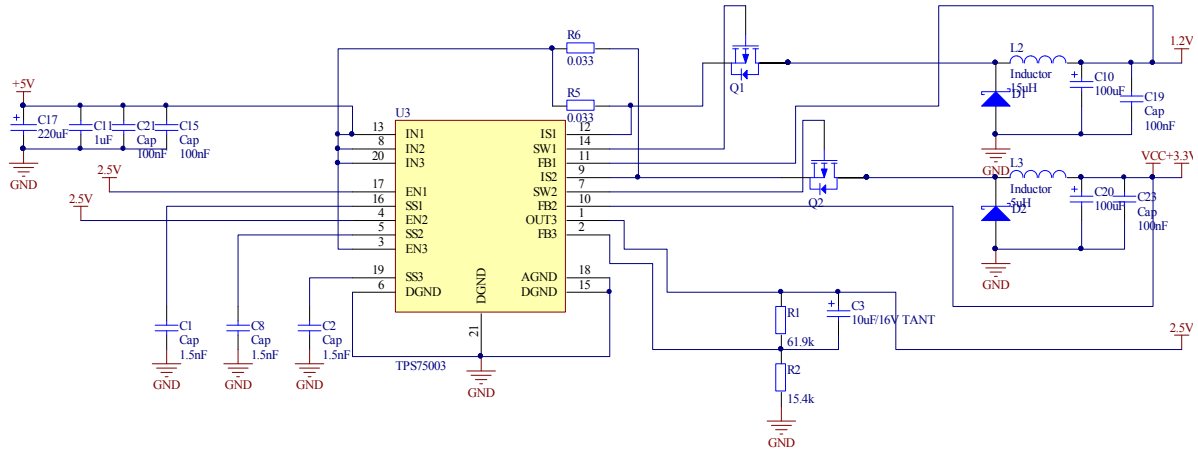


Figure 5.15: Switched mode power supply for FPGA and other digital components.

The start-up order is important for the FPGA as the right amount of current must be available so if you input current is limited the order in which the power supplies start-up can be specified using the enable input pins (EN1, EN2, and EN3) – see Figure 5.15. The ramp rate must also follow a certain path to enable the FGPA to start correctly which can be controlled by changing the capacitors connected to the soft start pins (SS1, SS2, and SS3).

Now that the power supply for the FPGA and other digital components has been designed the next part of the system is the supply for the analogue components – seen in Figure 5.16. The ADC and DAC both have two separate power supplies and grounds, separating out the analogue and digital components within the chip to minimize digital noise interference. A Linear regulator was chosen as they introduce much less noise than SMPS and the analogue side of the ADC and the DAC both operate off this 3.3V power supply - the digital side is also supplied by 3.3V supply but this is from the SMPS. The output from this linear regulator is further filtered by an inductor and capacitors to minimize introduced noise. The ground is also taken back to one entrance point to minimize the noise on the ground line. A 2.5V supply was also need to supply the analogue switches, this regulator is a very low noise that runs off the 3.3V analogue supply.

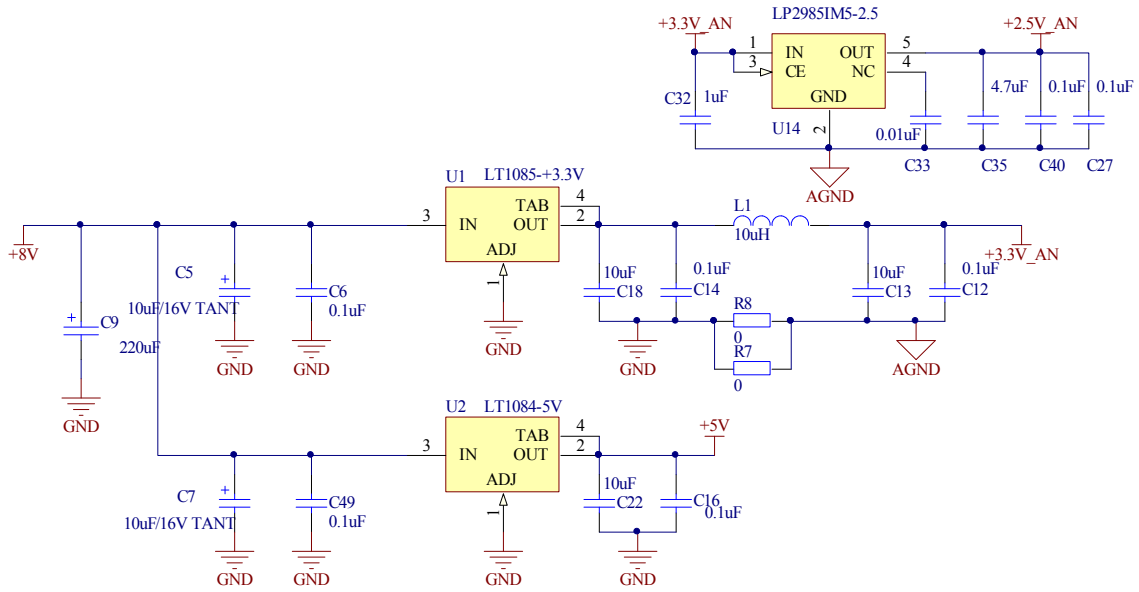


Figure 5.16: Linear regulators for analogue power supply and also 5V regulator for the switched mode power supply.

5.2.2.8 PCB DEVELOPMENT

Once all the schematics were designed they were exported to the PCB board design. There are many factors to consider when designing the PCB that are not relevant when designing the schematic. The first of which is the number of layers to have on the PCB. The PCB manufacturer that was used (Advanced Circuits Ltd.) can handle creating up to 6 layer board, but to reduce the cost a 4 layer board was chosen. A 4 layer board enables the middle layers to have power and ground reducing the amount of routing significantly as well as increasing the signal to noise on the power supply – assuming good decoupling. It is also important to stick to the manufactures tolerances for minimum track width, separation (0.152mm), and hole size (0.30mm). Another important factor to get right at the beginning is the component layout as this can affect the track lengths, the ease of routing and the amount of board area need. Using a FPGA has an advantage that most other chips do not in that the ports can be swapped with others to decrease track crossovers and improve the component placement aesthetics. It is also important to minimise the effects of interference from other tracks by separating the analogue and digital tracks as much as possible. When considering routing the tracks for analogue signals impedance of the tracks must be calculated to minimise reflection. Differential routing is important and the two tracks should be kept as close as possible to each other to minimise interference. Differential routing is used for a number of different lines including the ADC input, clock input, DAC output, and DDS output. Decoupling capacitors placement is another important factor especially for the FPGA as it can require a large peak current to remain stable, so capacitors are placed as close to the power and ground pins as possible to increase the peak power supply. There is also the ease of soldering to consider and so chips such as the ADC should not be placed too close to the FPGA to enable there to be room to solder it. Practicality of connector placements also needs to be considered and to simplify this and keep compatibility with the old transceiver board the Eurocard design was used. The PCB was designed considering all these factors and is shown in Figure 5.17.

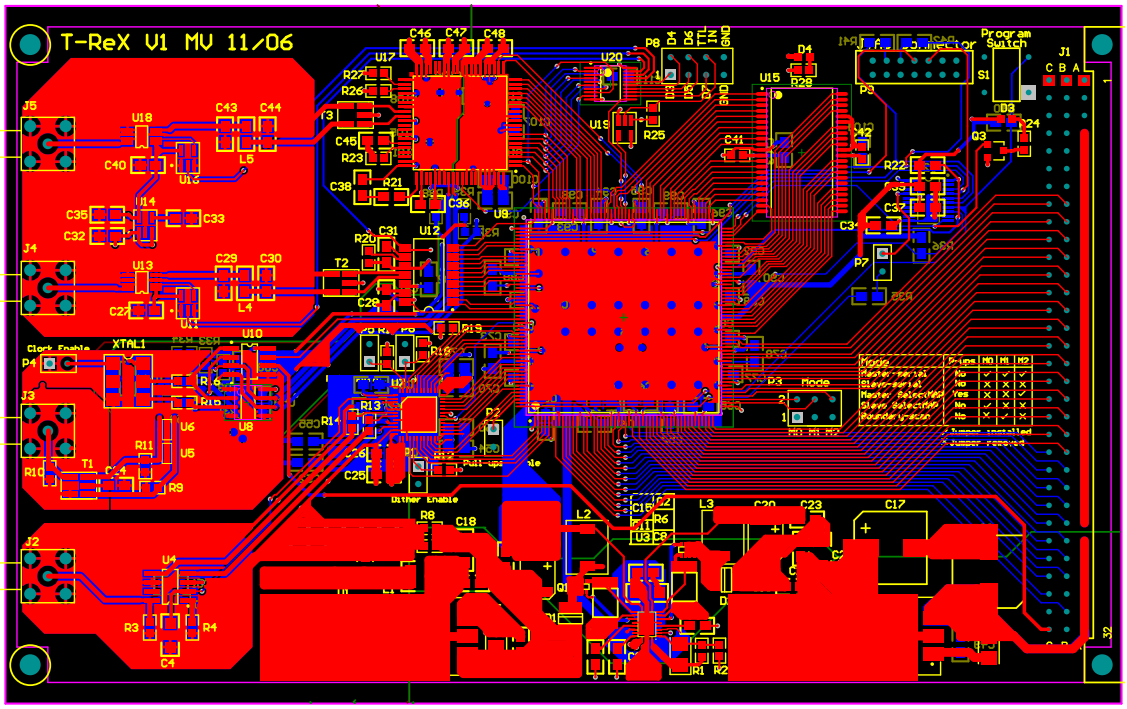


Figure 5.17: PCB Design in Altium Designer 6.3. Red - top layer tracks. Blue - bottom layer tracks. Yellow - overlay mask.

The Board once soldered and tested is shown in Figure 5.18 and Figure 5.19. Figure 5.18 has red circles showing the different components that make up the board including the DAC, FPGA, RAM, JTAG, clock, ADC, and power supply. The power supply was the first component to be assembled and tested. Once this was operating correctly the FPGA, JTAG, and clock components were added. Once this was correctly tested the ADC, DAC and RAM was added and tested. Once all the components were operating correctly the device was interfaced to the DSP unit.

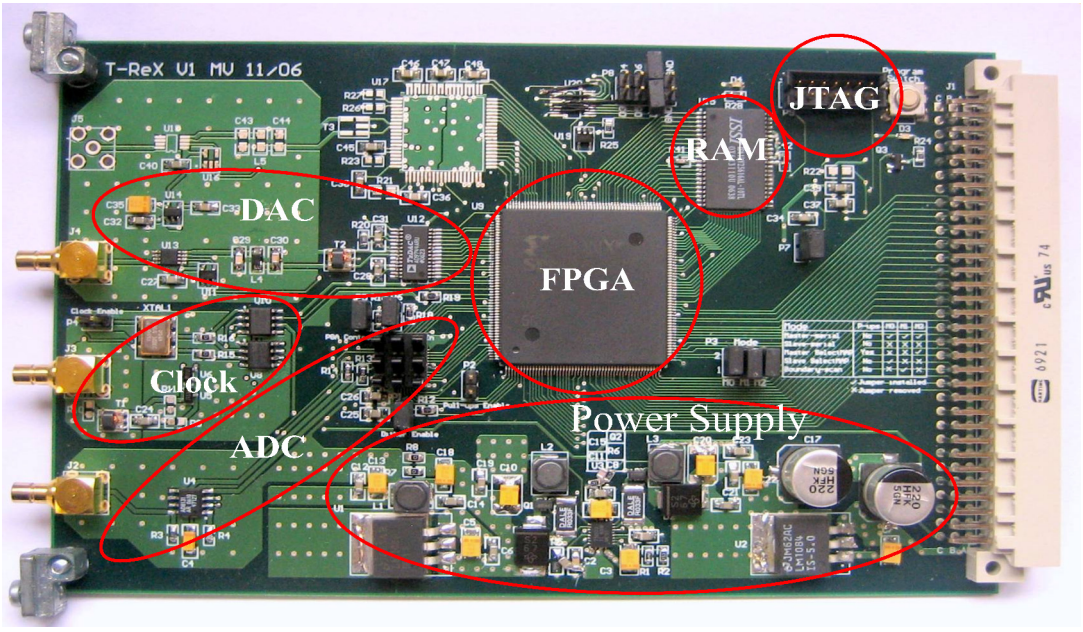


Figure 5.18: Created top side off FPGA transceiver board divided up into different blocks.

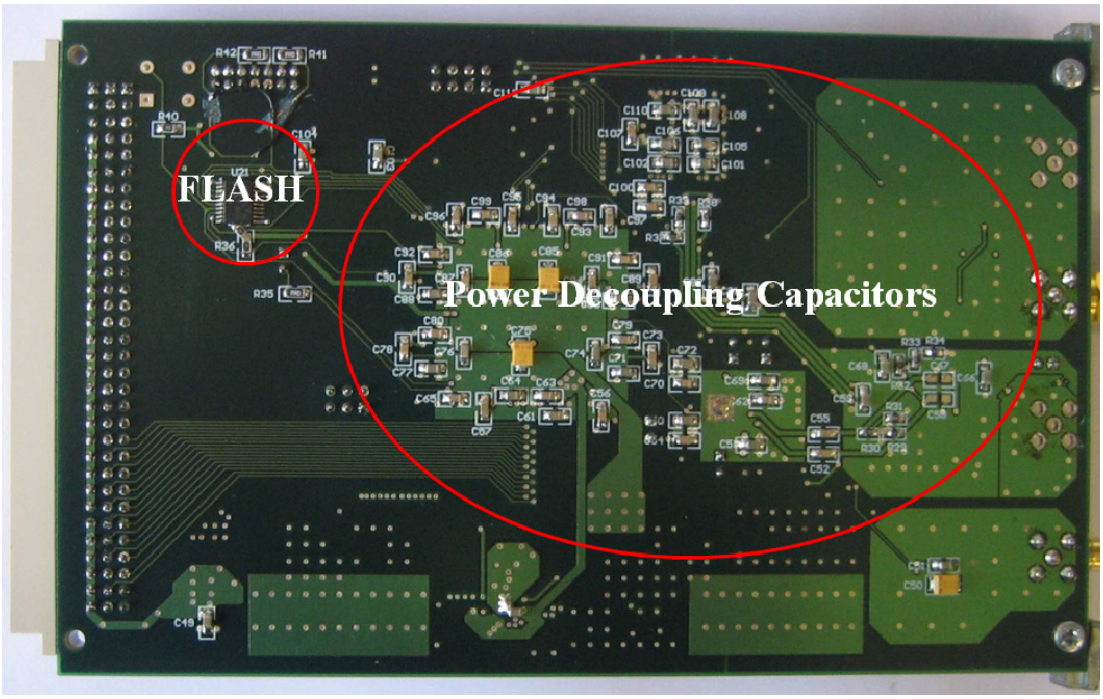


Figure 5.19: Created FPGA transceiver board back side.

There were many problems indentified in regard to the power supply. These were mostly caused by board manufacture problems as two out of the three boards manufactured had shorts between power layers. To solve this problem in future either a better board manufacture is needed to be used or the track clearance values for the centre layers need to be increased significantly.

Once the board had been manufactured the components were soldered onto the board and tested individually as much as possible. The power supply was first to be soldered and tested. A problem was found with the 1.2V power supply rail outputting 3.3V instead of the 1.2V needed. This was found to be caused by a lack of feedback resistors to the SMPS. This was fixed by soldering two resistors and a capacitor on the board and cutting one track. This can be seen in Figure 5.20.

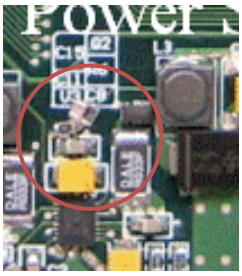


Figure 5.20: PCB showing resistors added to control the SMPS supply from 3.3 to 1.2V.

Once the power supply was soldered onto the board correctly the voltage and current seemed to be fine but the FPGA did not start-up. This was because the order that the power supplies start-up, the way the voltage is ramped up at the correct rate, and the start-up current are all important as if they are not correct the FPGA may not start-up. It was found that the ramping rate was needed to be decreased to minimise the current draw on start-up. This was

performed by varying the capacitors C1, C2, and C8 in Figure 5.15. To enable the FPGA to have a better peak current access for start-up a larger 100uF tantalum capacitor replaced the original 50uF capacitor. When these additions had been made the FPGA started up.

After the power supply had been fixed the next problem was the JTAG communication between the PC and FPGA. This was found to not be working and after many possible solutions were looked at it was found that it was a simple mix-up between the input and output communication line. This was fixed by cutting both the tracks and using a wire to cross them over – as seen in Figure 5.21. When this was done no further problems were found with the JTAG communication between PC to FPGA and PC to FLASH. It was found however, that when an inadequate current source was powering the FPGA there was sometimes a problem of the FLASH not loading the FPGA correctly. This is fixed by using an adequate current source as the main power supply input.

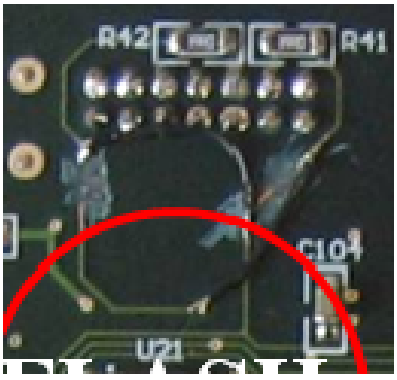


Figure 5.21: PCB error with JTAG Lines crossed.

After the FPGA communication was working correctly the next job was to capture the ADC data. To do this the ADC was needed to be soldered on the board. The ADC package is a QFN (Quad Flat pack No lead) which has no leads on the chip for use with reflow soldering – so posing a small problem as innovative soldering needed to be used. The package also has a ground pads on the bottom side of the ADC – so thru holes as were needed for heat dissipation and soldering. Both of these features created problems for the soldering of the package though success was achieved after much fine solder work. The ADC once running was found to need a heat sink.

The next job was to solder the clocking circuits on the board. The differential clock was measured with and oscilloscope the clock was not looking like it should and so the performance of the FPGA and ADC was not good. This was found to be a problem with the impedance matching circuit. The problem was that the resistors used were 50,000 Ohm rather than the 50 Ohm that was required. Once these were replaced with the correct resistors the response of the circuit was very good.

Much was learned with the debugging of the hardware board design, and the most important lesson was “don’t forget the simplest solutions first”, quite often the simplest solutions were left until the most complex ones were examined first – such as the JTAG in/out line mix-up took about a week to find as it was thought that the problem was with the FPGA chip or the PC.

The Transceiver board designed has passed our expectations in performance. For further testing see the software chapter.

5.2.3 DSP

The DSP used to control the FPGA Transceiver board is the basically the same as what is used in the previous version with a significant difference of more on-board RAM. The RAM on the old DSP board was 32k of 24bit memory whereas the new DSP board has 256k of 24bit memory. This allows the storage of a lot more data points and so enables longer experiments. The DSP board created is a pre-production board created by Robin Dykstra for the use of Magritek and it will be marketed by them in the near future.

5.2.4 AMPLIFIERS

Although the Transceiver has been designed move componentry is needed to provide the correct output power to drive the QCM and also to amplify the signal received from the QCM – to determine Bond Rupture noise as well as frequency change. This output and input power also needs to be able to be scaled to the correct amplitudes. This is done by using a combination of variable amplifiers controlled via the DSP by SPI, and fixed amplifiers - to get the output power required. An external low noise amplifier was also used to decrease the input noise for lisening for Bond Rupture.

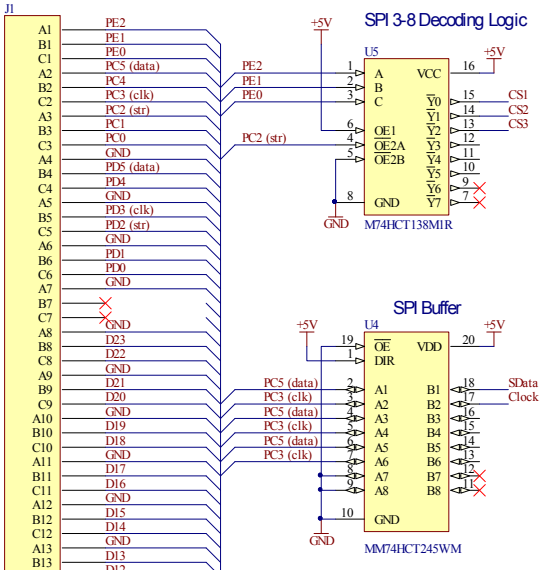


Figure 5.22: Connection of the SPI logic to the plug that connects to the backplane and DSP.

The communication protocol SPI uses three lines (for one way SPI) which are clock, data, and strobe. A basic description of SPI is as follows. The one bit data line is clocked into a shift register and shifted on a clock edge as many times as necessary and then the shift register is shifted in parallel into the internal register on a strobe rising edge. The SPI communication lines data and clock are buffered using an octal 3 state chip (M74HCT245) with the direction locked to output – see Figure 5.22. The other line used for SPI is the strobe line. To increase the number of devices that can be controlled with the same strobe line a 3 to 8 line decoder (M74HCT138) was used. This enables up to 8 lines to be controlled with the same SPI port of

which we are using 3. The main reason for using these buffers is for the isolation of the DSP board, especially important as we are using 12V for other devices on board and this could damage the DSP if a short were to occur.

5.2.4.1 DRIVING AMPLIFIER

The driving amplifier for exciting the QCM in the first version had a low power output of around 20dbm and so an external amplifier was required to attain the amplification necessary for Bond Rupture (<30dBm). This high power output was achieved in the second version using a surface mount amplifier developed by Mini-circuits the HELA-10. This amplifier has a gain of around 12dB and can have a maximum power output of just over 30dBm. Since the output power level should not exceed the amplifiers maximum limit of 30dBm, care was taken with the first stage amplifier to ensure that its output power was less than 18dBm - maximum level of 30dBm minus 12dB is 18dBm. The first stage amplifier used is Analog Devices AD8320 variable gain amplifier. This has a maximum output power of 22dBm and has a 36dB linear gain range (from -10dB to 26dB) with 256 ranges over this gain range. To limit the power into the HELA-10 from the AD8320 a 4dB attenuator was connected between them and to ensure that the AD8320 and the HELA-10 were not overloaded the input to the AD8320 was limited to -8dBm. With these chips both connected and working, a linear gain range of -6 to 30dBm (as seen in Figure 5.23) is available with any further attenuation needed performed by scaling the DAC output voltage.

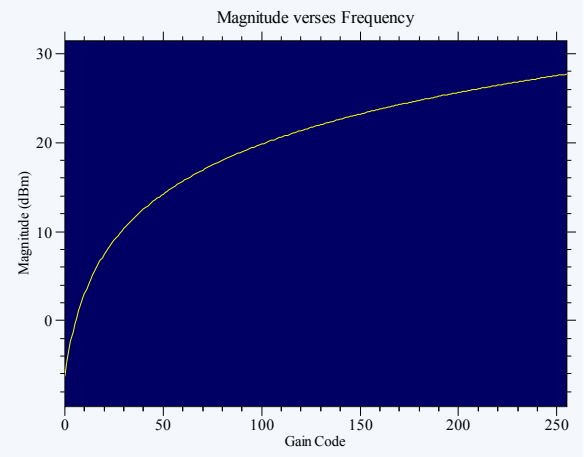


Figure 5.23: Effect of changing the gain code of the AD8320 on the output magnitude.

Both the first stage and second stage amplifiers used are run off a 12V input. This voltage input is not supplied on the back plane so a plug is used from an external power supply to the board. This supply is filtered using a inductor and capacitor to remove noise on the supply. There is also a protection diode to prevent the input voltage exceeding 13V. A 5V regulator is also used to buffer the input SPI ports. The AD8320 is normally used for 75Ohms systems so the input is adjusted to match 50Ohms system by using resistors. The HELA-10 is matched to 75Ohms on its input and 50Ohms on its output both single ended. The circuit schematic for the QCM driver amplifier is shown in Figure 5.24.

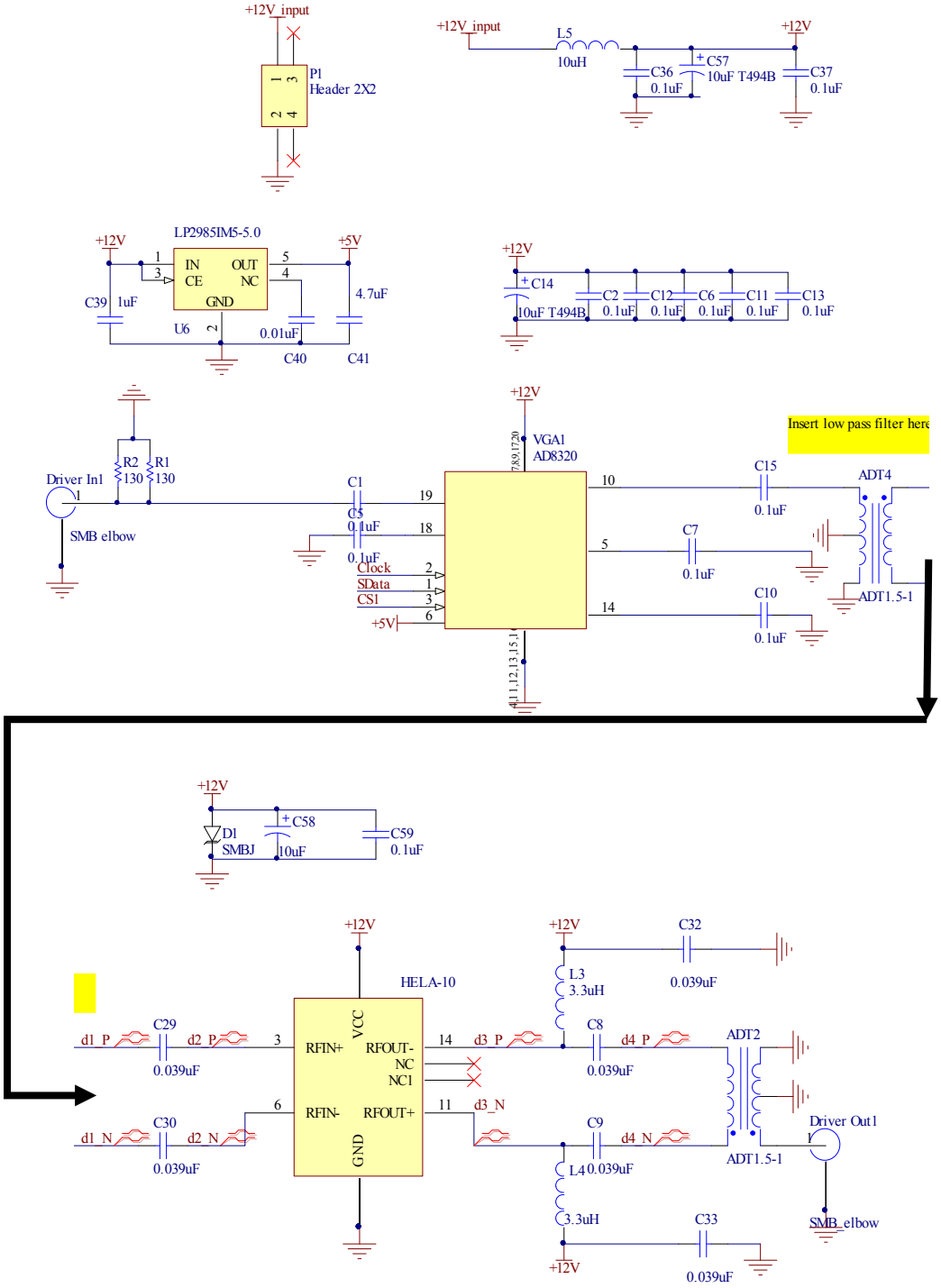


Figure 5.24: First and second stage amplifiers circuit schematic for driving the QCM - top right joins with the bottom left.

5.2.4.2 RECEIVER AMPLIFIER

The job of the receiver amplifiers is to amplify or attenuate the signal from the QCM such that the signal level maximises the dynamic gain into the ADC. To enable the listening for Bond Rupture noise the primary signal is attenuated significantly on the QCM board so high amplification is needed to restore this signal to monitor the primary frequency. Two stages of amplification were used enabling high amplification. This was done using two Analog Devices

variable gain amplifier's (AD8369). The AD8369 chips have an amplification range from -10 to 35dB and a total of 16 different gain settings over this range. When both chips are in series a total range of -20 to 70dB is present from the input from the QCM and the output to the ADC – a total gain range of 90dB. When the devices are combined a total 256 combination exist of which 32 different gain settings are useful. The schematic showing these amplifiers is shown in Figure 5.25.

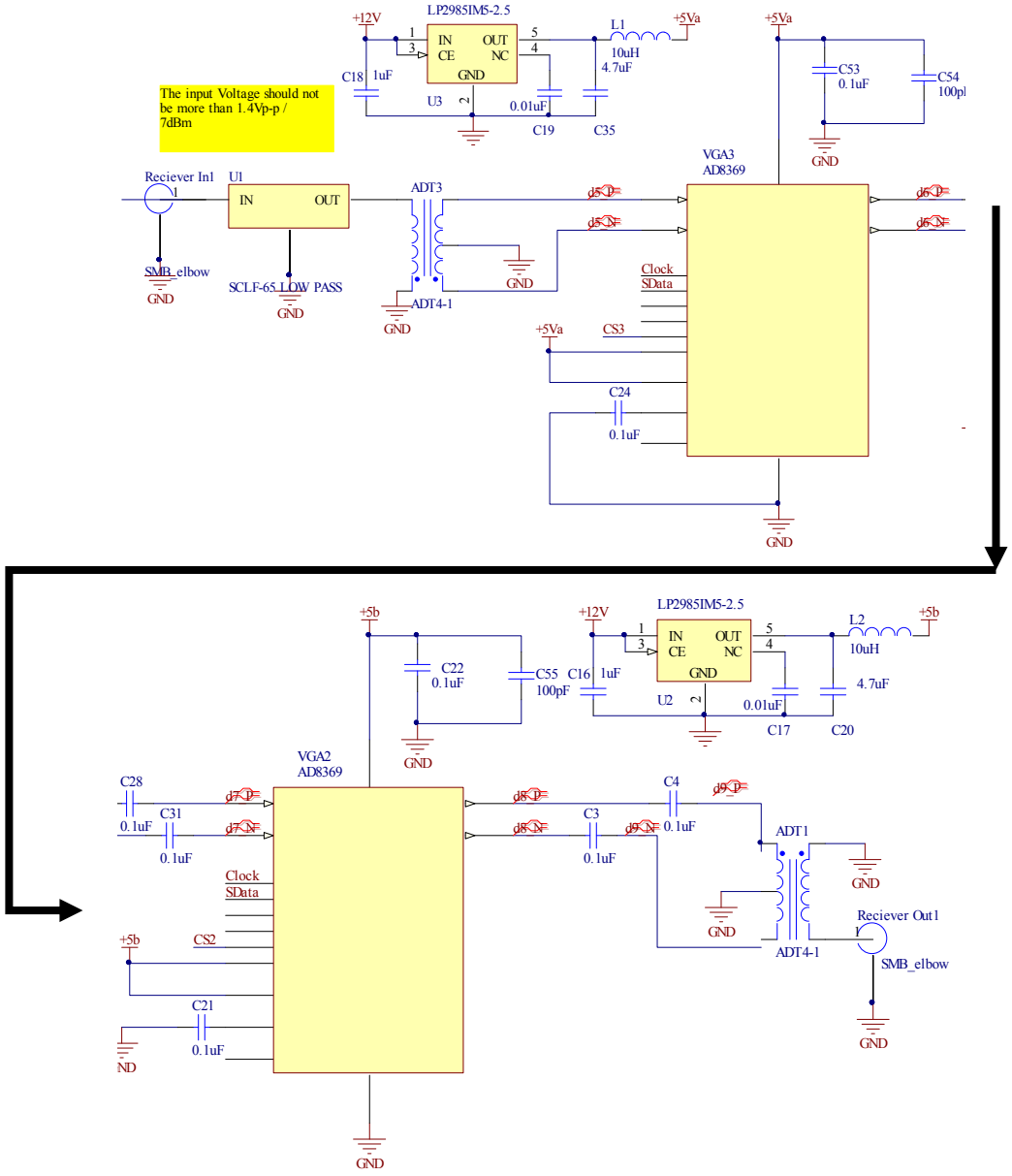


Figure 5.25: First and second stage amplifiers for the high gain input amplification - top right joins with the bottom left.

The amplifiers each were given their own power supply to minimise interference from each other. They were also put in an enclosed tin box to minimise external RF interference. The maximum input to the chips is 1.4Vpp

After the board was complete the input gain was tested using the calibration software. This found that the gain rage was around -30dB and 60dB. This is different from the specified 20 to

70dB gain due to attenuation cables, connectors, transformers, impedance matching and the filters combined effect. The combined gain effect is shown in Figure 5.26a with all of the 256 gain combinations. Of these a number of gain settings are chosen as shown in Figure 5.26b.

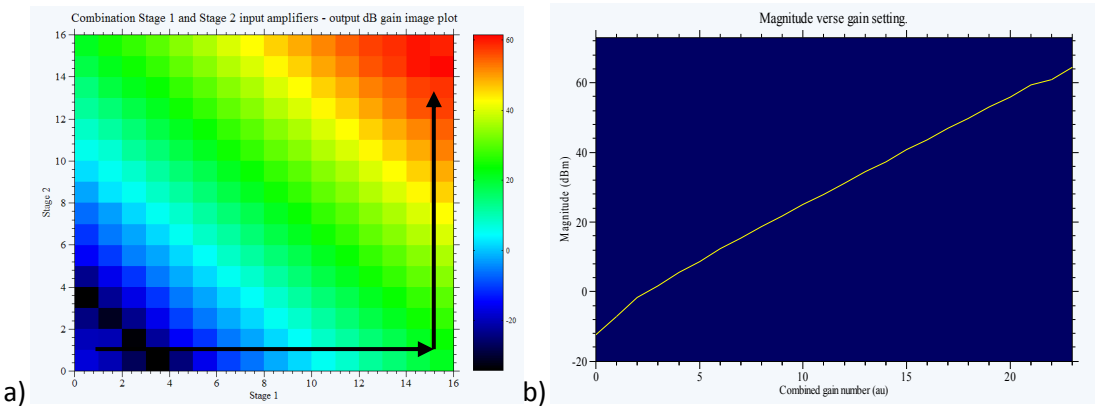


Figure 5.26: a) Image plot in dB of the gain of the input stages. b) Chosen gain path values selected by arrows shown in plot (a).

5.2.4.3 AMPLIFIERS PCB

The two amplifier circuits were combined onto the same PCB board – seen in Figure 5.27. The variable high gain amplifier for amplifying the signal from the QCM to the ADC is on the top left of the figure. The variable high power amplifier is shown in the bottom left and bottom centre. The SPI buffer circuitry is shown on the top right of the board.

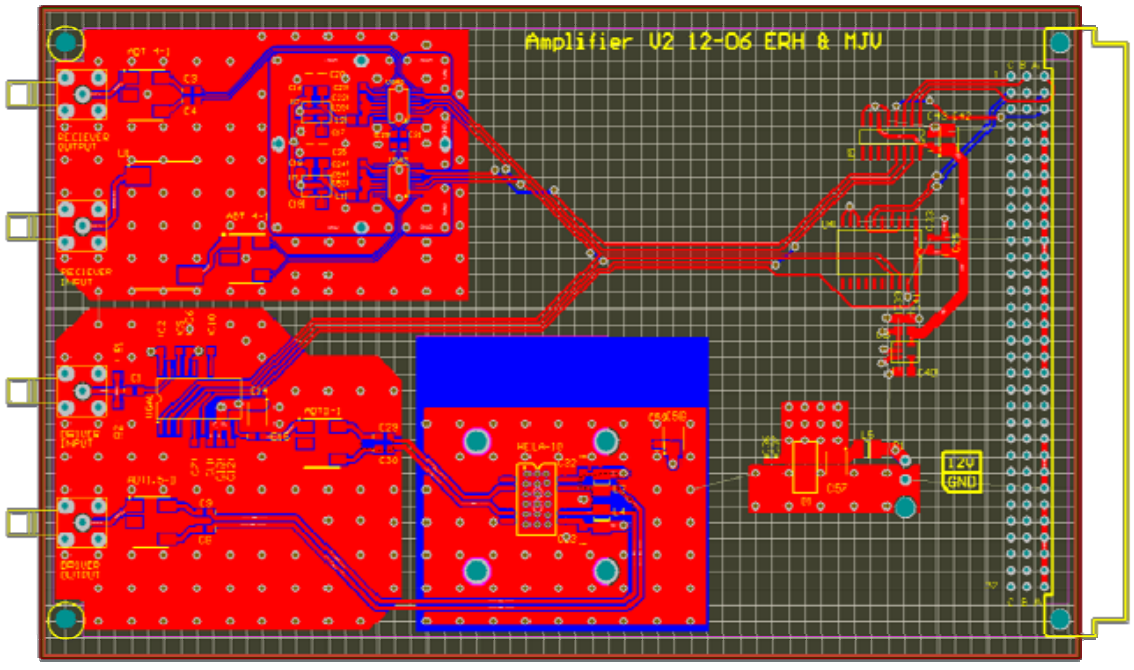


Figure 5.27: Amplifier Board designed in Altium designer – designed in collaboration with Evan Hirst.

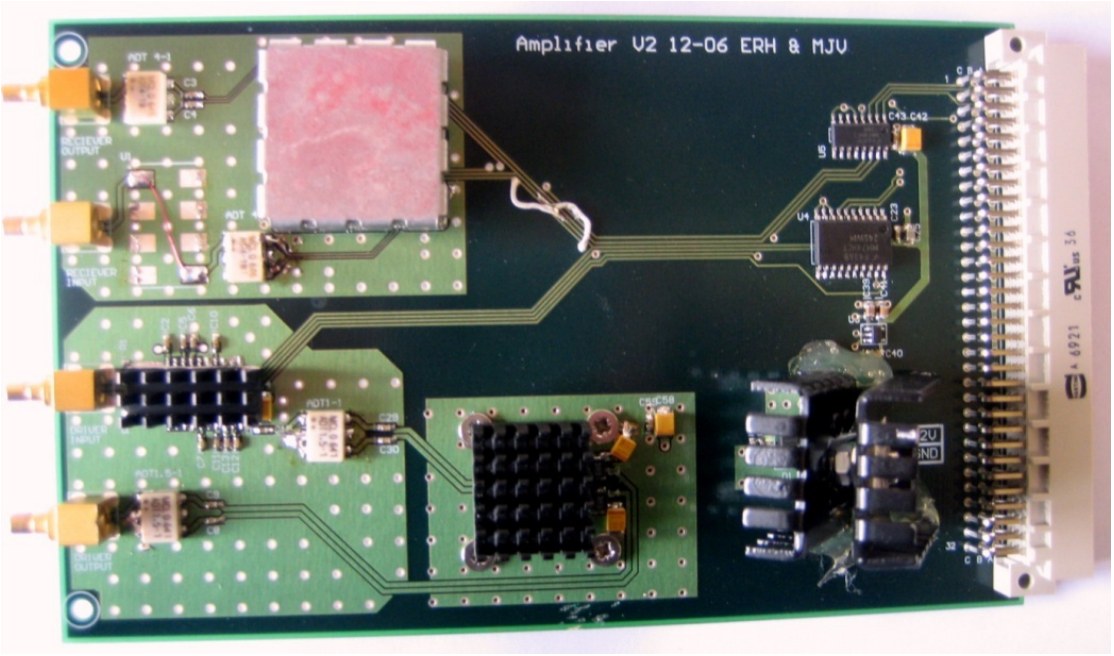


Figure 5.28: PCB board soldered and working.

There were a number of issues to consider when designing this device including the track impedance which must be kept to the correct impedance to minimise reflection and distortion problems caused by mismatching. The high gain amplifiers were placed inside a tin box to reduce interference from sources around. This was required as high gain of the amplifiers could amplify the smallest of signals to significant size with up to 3000 times (70dB gain) signal multiplication. It was also found that filling the thru holes on the back side of the board also significantly reduced the signal to noise ratio. The finished board is shown in Figure 5.28.

5.2.4.4 LOW NOISE AMPLIFIER

To decrease the input noise floor a low noise wide band amplifier was used. The amplifier used was a MITEQ AU1442 amplifier that has a noise figure of 1.1, a 37.5dB gain, and a frequency range of 0.01 to 200MHz. The pre-amplifier is used on the input to the two stage high gain amplifiers in a configuration as shown in Figure 5.29. Using a low noise preamplifier means that the noise figure on the other amplifiers has less impact on the noise of the signal and so enables a smaller signal level to be measured. Care however must be taken not to over load any of the amplifier stages as this will distort the signal. This is especially relevant as the low noise amplifier is adding another 37.5dB gain to the mix. The

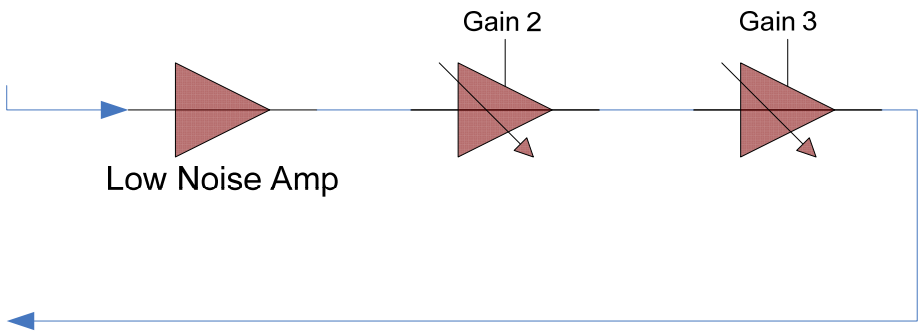


Figure 5.29: Configuration of low noise amplifier fed to the two high gain amplifiers.

To use the amplifier it was placed on a PCB Eurocard board and connected to smb connectors. On this board a 15V regulator was used to supply the amplifier which is run off a 24V supply.

5.2.5 QCM BOARD

As was mentioned in the literature review the QCM impedance varies significantly over its frequency range. The main frequency point of interest is when the inductance cancels the capacitance leaving pure resistance – this is called the resonant frequency. If only the impedance at the resonant frequency is considered, it is found that this varies depending on the QCM and also by what is on the surface and what the operating environment is (e.g. air, water). It is important to match the impedance from the driving amplifier to the QCM to minimise the power loss and also to minimise interference from reflection and various other factors introduced by impedance mismatch. Another factor that is important for this board is the measurement method whether to measure voltage, current, or power as all of these can be used to determine the resonant frequency of the QCM. It must also be taken into account how the 3rd harmonic 'noise' will be measured.

The impedance matching will be examined first. There are many different methods to impedance match the QCM all of which are a trade off of different factors such as bandwidth, noise floor, and whether purely resistive. Though this impedance matching could be done with an active device such as an operational amplifier, passive matching methods were chosen to minimise the noise inputted to the system. The first method considered was to use a transformer (N1:N2 windings) and match the windings directly – see Figure 5.30. This effectively is multiplying the QCM resonant impedance value by the square of transformer ratio. This enables any purely resistive load to be matched with only a small amount of inductance introduced by the non ideal transformer. This is by far the easiest method but it is not very flexible as if the QCM impedance changes the transformer must be rewound to match the new impedance.

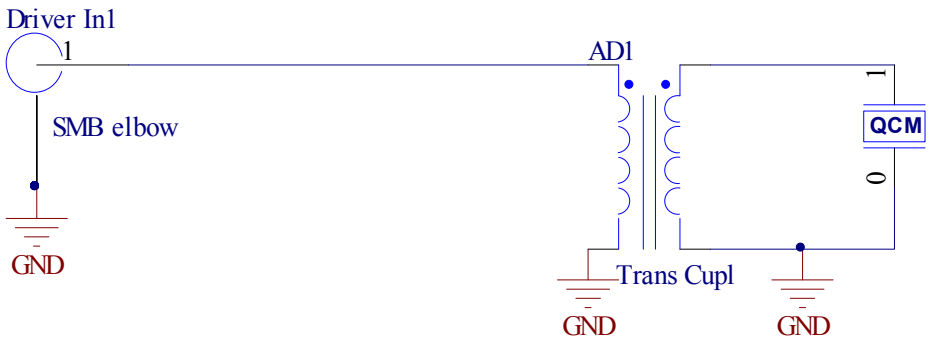


Figure 5.30: Transformer impedance match.

Another method that was used was to maximise the impedance matching was using a variable inductors and a variable capacitor to match the impedance. This can be matched by running the QCM at resonance then modifying the variable inductors and capacitor until the optimum impedance is achieved. This method enables excellent impedance matching to be achieved but suffers from other problems. One such problem is that the band width is limited by the capacitor and inductor values as they also act as a band pass filter. This limits the useful operating frequency. A transformer can also be used with this configuration (see Figure 5.31) to shift the impedance to within the matching impedance which enables the bandwidth to be maximised. Though this performs excellent matching for a QCM this requires a large amount of tuning per QCM so is not a good generic solution. It was instead decided to use the transformer only method. Further work could be done in this area to create a closed loop impedance matching solution, but this requires a significant amount of further research and development.

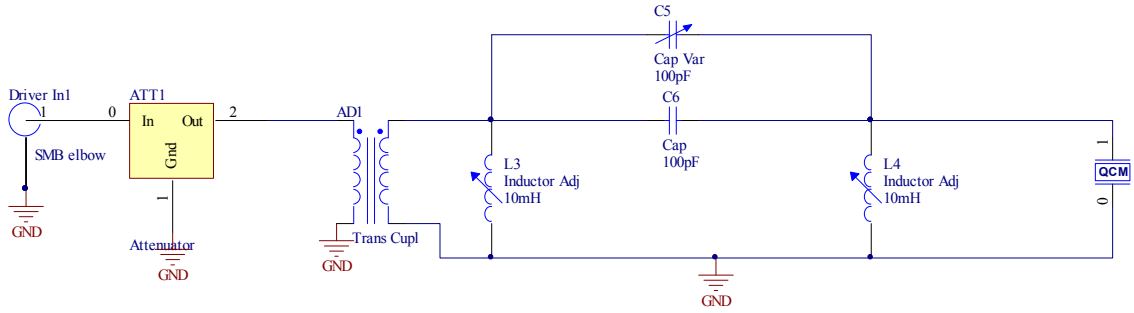


Figure 5.31: Combination transformer and PI network matching.

A number of solutions to the best way to measure the resonant frequency were investigated that would only require a single capture unit (ADC). One constraint that was introduced to the design from the beginning was that one side of the QCM was to be grounded to ensure the no voltage change was introduced on to the biosensor surface of the QCM – as it was proposed that this could cause unwanted interactions with chemicals on the surface. This affected meant that the QCM could only be used in parallel configuration limiting the circuit types and excluding the traditional oscillator configuration. Among the number of resonant measurement solutions there is power reflection, voltage, or current. If using voltage resonant frequency is measured at the minimum magnitude, current is measured at maximum magnitude, and power reflection is at a minimum magnitude but is dependent on how well the

QCM impedance is matched. The power reflection can be measured using a directional coupler which is made up simple circuit using two transformers – see Figure 5.32. This can measure the reflected power making it ideal for measuring Bond Rupture at the third harmonic as the driving signal would be at a minimum when at resonance and when the Bond Rupture 'noise' signal is at a maximum. This also incorporates only passive components so adds minimal noise. One problem with this configuration is that it is highly dependent on the impedance matching and it is hard to predict the response of it when it operates outside the standard impedance. The resonant frequency measurement can also be limited as it measures the reflected power not true resonance and if the impedance matching is too far out the wrong resonance frequency maybe used.

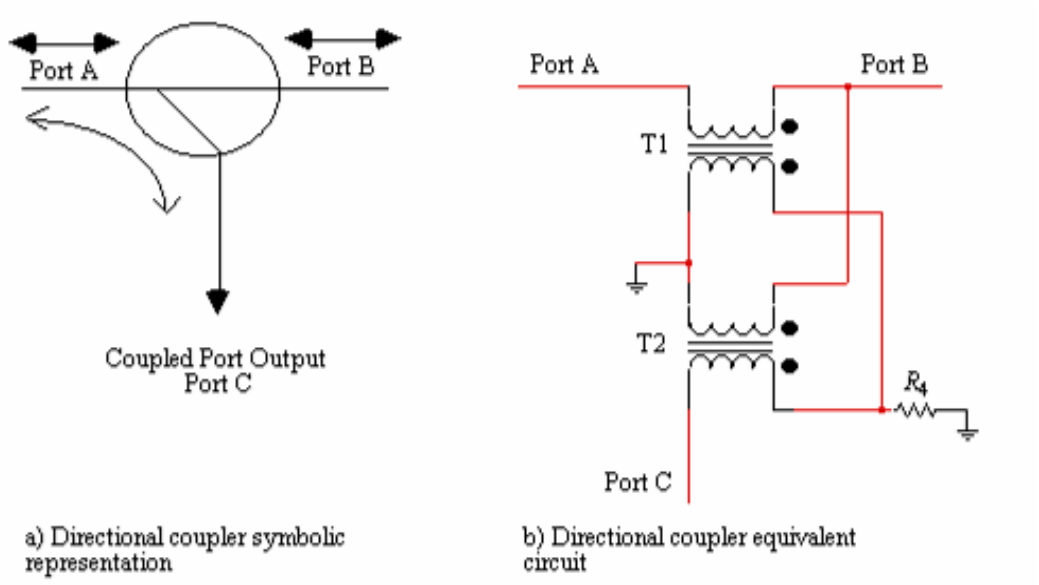


Figure 5.32: Directional coupler, http://michaelgellis.tripod.com/direct.html.

The method that was used in the final board was using an input filter to attenuate the third harmonic, and then a transformer for impedance matching to the QCM, with a current tap transformer enabling current passed to the QCM to be measured – see Figure 5.33. This output is then filtered thru up to three Mini-circuits 30 MHz high pass filters, enabling the fundamental frequency to be decreased so that the 3rd harmonic can be amplified higher without exceeding the amplifiers power range. Each filter adds around 60dB of attenuation to the fundamental frequency - Figure 5.34 shows the frequency response of one of the filters used. The circuit used enables the impedance matching by changing the windings of transformer T2.

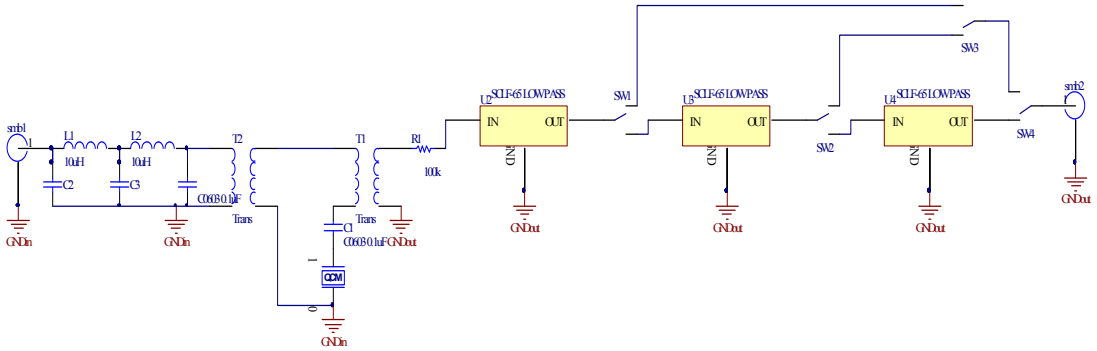


Figure 5.33: Impedance matching and QCM measurement board.

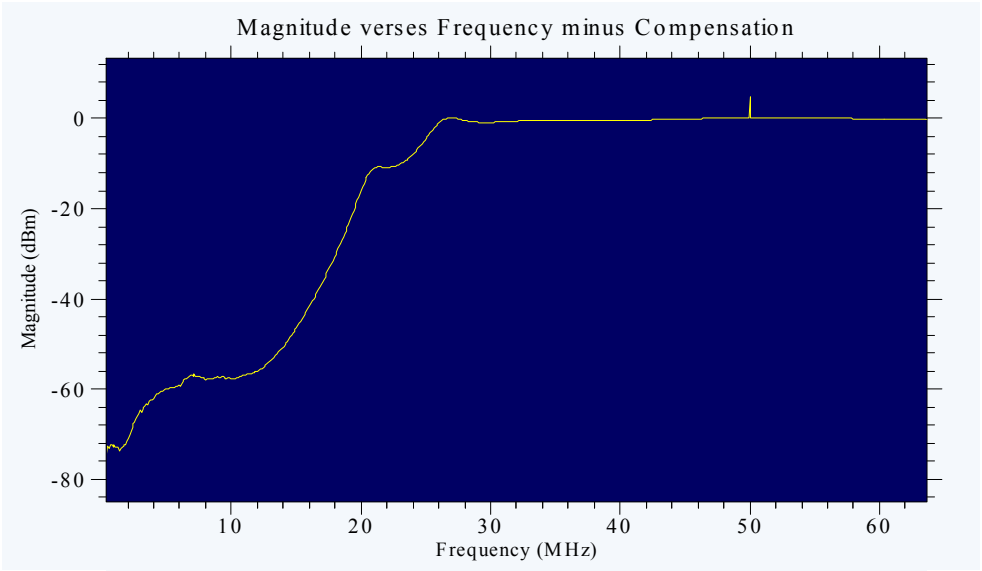


Figure 5.34: Frequency response of the 30MHz Mini-circuits high pass filter.

The current measurement was chosen as to maximise the signal at resonance to make the signal measurement simpler, while also enabling a significant amount of attenuation. The number of filters was experimented with but in the end one filter only was used as each filter added also affected the output impedance and signal level, so if too many filters were in series the resonance could not accurately be measured at low driving signal levels.

5.3 HARDWARE/ SOFTWARE - FPGA

5.3.1 INTRODUCTION - TRANSCEIVER SYSTEM OVERVIEW

The FPGA Digital Transceiver (T-ReX) device is a FPGA (Field Programmable Gate Array) based digital transceiver that can transmit via a DAC (Digital to Analogue Converter) and receive using an ADC (Analogue to Digital Converter) - with the FPGA as the glue logic to control the data flow, process the ADC and DAC data, and to send the data to an external DSP. The T-ReX has a 100Msps 16bit ADC, enabling sampling of wide dynamic range signals up to input frequencies of 300MHz. The input voltage range can be either 0.75Vpp or 1.125Vpp by on board jumper. The T-ReX also has a 200MHz 14bit DAC, that when combined with the DDS (Direct digital synthesizer) core implemented in the FPGA enables generation of waveforms up

to 100MHz. There is also an analogue switch which enables the DAC output to be turned off and on within 10ns. The T-ReX implements two DDC (Digital down Converters) in FPGA logic for the down-sampling of the 100Msps ADC to a minimum bandwidth of 600Hz. The T-ReX has a 256k x 16bit 100MHz memory for direct storage capability of the ADC waveform.

Figure 5.35: Diagram showing the three main design stages.

The T-ReX device has three main software development stages before an experiment can be carried out as seen in Figure 5.35. The first stage is the development of the FPGA software. To write FPGA based digital hardware the VHDL language is used - which stands for VHSIC (very high speed integrated circuit) hardware description language. This language is used to describe the how the digital hardware is connected up within the FGPA. This FPGA design would usually be compiled only once, but if needed more features can be developed and added later by simply uploading the new compiled FPGA internal hardware. The second stage of design is the DSP software for communication to the FPGA and the third the prospa GUI and macro design. This is done in the same way as previously described in 'Digital Transceiver system' chapter, except it the DSP is communicating with the FPGA device rather than the old transceiver system.

The FPGA software was developed primarily in 'Xilinx ISE 9.1' using VHDL and Xilinx cores. The FPGA software is seen in Figure 5.36 and is made up of a number of different components. This figure shows the internal FPGA components inside the FPGA box. The DSP communicates to the different components though the address translator. The address translator assigns all the blocks a different address. The address translator communicates with six different blocks being the: DDS generator which controls the generation of the DAC signal; trigger control which controls when to turn components on an off; ADC capture and conversion which controls the ADC input and the ADC status; both digital receive processors which process the ADC data stream; and the FIFO RAM controller which controls the external RAM.

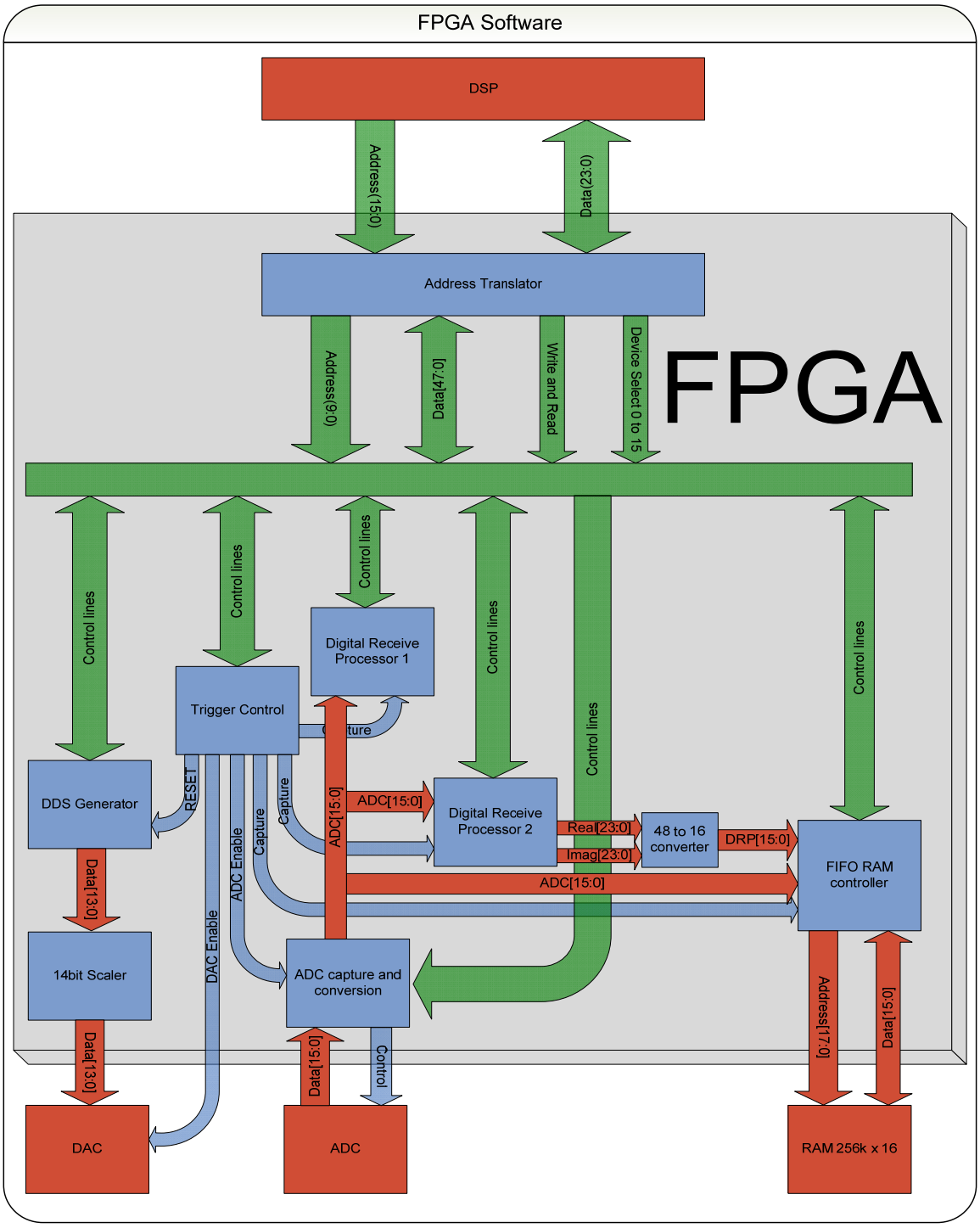


Figure 5.36: Simplified diagram of FPGA hardware interconnects. The external component are red, internal components are blue, control lines from the bus are green, control lines between components are blue, and Data is red.

The rest of this section covers the development of FPGA's device in VHDL for the development of a digital transceiver for the use in the Bond rupture system.

5.3.2 ADDRESS MAPPING

The address translator block enables the DSP to communicate with the individual devices within the FPGA. Many different types to standard address mapping was investigated but in the end a custom solution was chosen to enable the address translator to take up the least amount of logic within the FPGA while still supporting 48bit write and support for up to sixteen internal devices.

The address translator can be seen in Figure 5.37. It communicates with the DSP using a 16bit address bus, 24 bit IO data bus, write signal, read signal, and address attribute signal. These are the same ports as used to communicate with the old transceiver board except instead of a 16bit data bus a 24bit data is used. The devices within the FPGA are addressed using the top 4bits of the address input, enabling 16 devices to be addressed. The address translator enables up to 48bit communication between the DSP and the internal FPGA blocks using the 10^{th} address bit to select a 48bit read or write. This is carried out using the 24 to 48bit converter. The 10 bit address, write enable, read enable, and 48bit data is passed directly to all addressable blocks within the FPGA. The only thing that is different for each device is the device select bit that is passed to it.

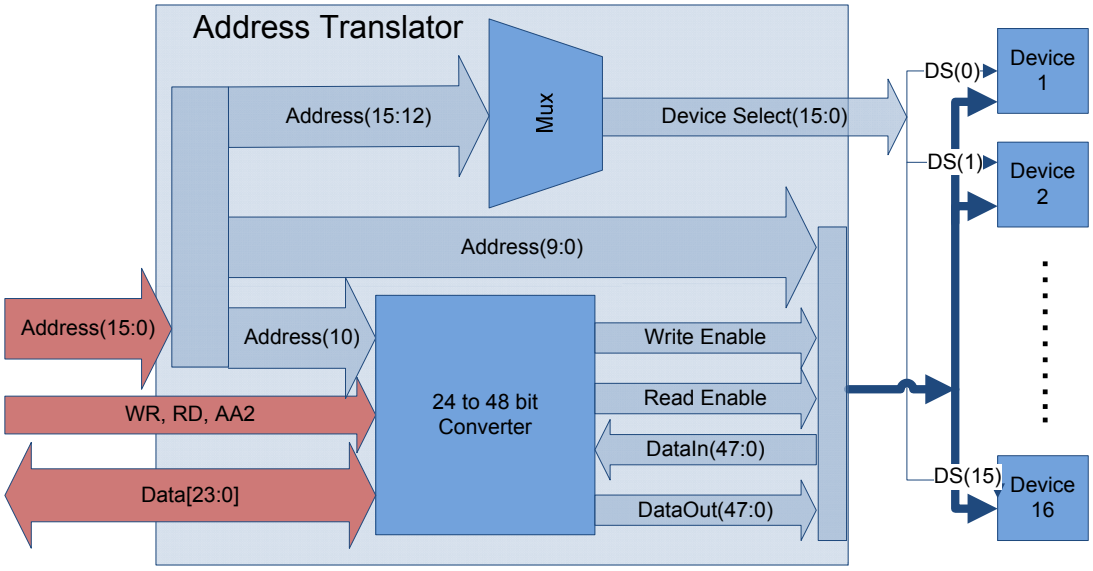


Figure 5.37: Address translator diagram.

The Address translator block is run at the 100MHz clock but the transfer rate is limited to how long a write or read cycle takes. This is set to be 7 clock cycles on the DSP to prevent errors. A write is selected by the WR and AA2 signals moving to low. This is then verified for 3 clock cycles and the IO data port is set to high impedance before write is performed. The waiting for 3 clock cycles ensures that the DSP output is not in error and gives enough time for IO port to be changed to high impedance (if the previous cycle was a read) and to guarantee that all the data to be presented properly to the IO ports. Once a write is confirmed address translator either saves the data for into the high buffer (if 48bit write is selected) or it outputs it to the 'DataOut' port and raises 'write enable'. The DSP selects a read by moving the RD and AA2 signals to low. The address translator then waits for 2 clock cycles and the 'DataIn' line is connected directly to the IO line. Once the write has been confirmed a read enable is sent out.

The Device then that is being communicated with then has 4 clock cycles to 'DataIn' the data to the 'DataIn' line so that the DSP can read the data correctly.

To enable the addressing within each device there is a 10 bit address line that is passed to each device enabling 1024 different addresses per device. There is also a write enable, read enable, data in for reading data, and data out for sending data. These are all sent to all the addressable devices with the exception of device select bus which is divided up into 16 lines and one sent to every device – see right side of Figure 5.37. The device select line enables the device selected to take control of the output port as well as enabling the selected device to listen for a read or write command.

The VHDL code for needed to instantiate a device to the address translator is shown in Table 5.2. This is the needed for every device to be addressed. The data input and output lines can be any number less than 48bit but to perform a 48bit write or read two DSP read or writes are needed. Because of this most ports are set to 24bit and the only addresses that need more than this are when a DDS is used (which are 32bit increment value is needed). The 'SEL' input is where the device number is set by selecting the line number of the 'AT_DeviceSelect' bus.

```
Inst_DDS_Wrapper: DDS_Wrapper PORT MAP(
      SEL => AT_DeviceSelect(1),
      ADDR => AT_Address,
      WE => AT_WRE,
      LD_DIN => AT_Dataout(47 downto 0),
      LD_DOUT => AT_Datain(23 downto 0),
      RD => AT_RDE,
      …
);
```

Table 5.2: VHDL code for passing the address to the internal device.

The VHDL code inside the device needed for communicating with the address controller is as is shown in Table 5.3. The first process shows that the output data port is only taken control over when the device is selected (when 'SEL' is high) and at other times the port is set in high impedance mode. In reality there is no high impedance status for the bits but the Xilinx compiler converts this into multiplexer. The second process controls what to do when a read or write event occurs. This is a synchronous process and only checks on rising edges for the SEL and RD or WR change. If the WR and SEL signals are both high at the same time the address pins are then compared with a case statement which determines what to write to. In this case when the lower part of the address is "001" the ADC control register is written to using the data in port. If the RD and SEL are high then a read is performed when the data that the address specifies is latched into 'DataFeedBack' register. The 'DataFeedBack' register is then outputted directly to the DSP while the DSP keeps the RD input high.

The T-ReX Board communicates via the DSP using a 16bit wide address bus and 24bit wide Data Bus using the address space 'AA2'. Both AA2 and AA3 are sent to the transceiver so that it could easily be possible to have two boards operating on the same bus just with different address spaces. The control signals are; "AA2" to signal that this address space is active; "WR" to signal a write is occurring; "RD" to signal a read is taking place. The signal "AA2" activates the FPGA to be read or written to, in the currently used DSP, AA2 is set up to correspond to a internal read or write to "x:$1XXXX" with the other 16bits passing on the external ports. The

data, address, and control lines need to be held for a specified amount of time, presently the DSP is configured to take 7 clock cycles to complete a write or read with the DSP reading the data line at the end of the 7 clock cycles. The main delay is the reading back of data. This is because the RAM requires three cycle pipeline delay to read the data back to the input port. This could be improved with further work but it was not a priority as speeds of 14MHz write or reads are plenty fast enough.

```
process(DataFeedBack,SEL)
begin
      if( SEL = '1' ) then
            LD_DOUT <= DataFeedBack;
      else
            LD_DOUT <= (others => 'Z');
      end if;
end process;

-- Clock input registors
DDS_Registor: process(clk)
begin
      if( rising_edge(clk) ) then
            if(WE = '1' and SEL = '1') then
                  case ADDR(2 downto 0) is
                        when "001" =>
                              ADC_Control_Reg <= LD_DIN(23 downto
0);
                        when others => NULL;
                  end case;
            elsif(RD = '1' and SEL = '1') then
                  case ADDR(2 downto 0) is
                        when "001" =>
                              DataFeedBack <= X_Control_Reg;
                        when "010" =>
                              DataFeedBack <= X_Info_Reg;
                        when others => NULL;
                  end case;
            end if;
       end if;
end process;
```

Table 5.3: VHDL template within the device to enable communication with the address and internal components.

The T-ReX Board communicates via the DSP using a 16bit wide address bus and 24bit wide Data Bus using the address space ‘AA2’. Both AA2 and AA3 are sent to the transceiver so that it could easily be possible to have two boards operating on the same bus just with different address spaces. The control signals are; “AA2” to signal that this address space is active; “WR” to signal a write is occurring; “RD” to signal a read is taking place. The signal “AA2” activates the FPGA to be read or written to, in the currently used DSP, AA2 is set up to correspond to a internal read or write to “x:$1XXXX” with the other 16bits passing on the external ports. The data, address, and control lines need to be held for a specified amount of time, presently the DSP is configured to take 7 clock cycles to complete a write or read with the DSP reading the data line at the end of the 7 clock cycles. The main delay is the reading back of data. This is because the RAM requires three cycle pipeline delay to read the data back to the input port. This could be improved with further work but it was not a priority as speeds of 14MHz write or reads are plenty fast enough.

The 16bit address input is shown in Table 5.4. This is broken up into the DSR (Device Select Register) which selects the device to be read or written to, the HWE (High Write Enable) which enables a 48bit write and the DDR (Direct to Device Register) which is the address passed to each device.

15	14	13	12	11	10	9	8
DSR3	DSR2	DSR1	DSR0		HWE	DDR9	DDR8

7	6	5	4	3	2	1	0
DDR7	DDR6	DDR5	DDR4	DDR3	DDR2	DDR1	DDR0

-- reserved should be written with 0

Table 5.4: Address input register.

The DSR register selects which internal device is to be written to or read from and is addressed as shown in Table 5.5. This enables a total of 16 devices to be addressed. To write to the DDS the DSP address would be 0x1XXXX. To select AA2 the address 0x11XXX would be written to. When a device is selected the FPGA raises the corresponding device select pin high.

DSR[3:0]	Device
0	Trigger
1	DDS
2	ADC
3	-
4	DDC1
5	DDC2
6	External FIFO
Others	-

Table 5.5: Devices that are selected for corresponding DSR value

The HWE bit when set high during a write, the 24bit data is stored in a register with no output write occurring until the next write with HWE low, at which time the second write is outputted normally on the lower [23:0] to the device selected but the 24bit data from the first write is outputted to the device on [47:24] enabling a 48bit write to be carried out. This is implemented with a basic state machine. Table 5.6 shows the table for what happens when the HWE bit is selected for 48bit write. It can be seen from this table that the output depends on the previous HWE value.

HWE	last_HWE	data_Register	data_device[48:24]	data_device[23:0]
0	0	0	0	Data_in
1	0	Data_in	0	0
0	1	0	data_Register	Data_in
1	1	Data_in	0	0

Table 5.6: demonstrates how 48 bit write works

To perform a 48bit write within the DSP start by writing the high data with HWE high then write your low data with HWE low (see Table 5.7). The 48bit word is only loaded into the register selected on the second write.

```
; Write to 48 bits to DDS
move   a1,x:$11500  ; Output high DDS_Phase_Increment
move   a0,x:$11100  ; Output low DDS_Phase_Increment
```

Table 5.7: ASM code for the DSP to perform a 48bit write to the DDS.

5.3.3 TRANSMIT DESIGN

A conceptual drawing of the transmitter can be seen in Figure 5.38. This is made up of the internal FGPA components for generating the digital waveform and two external components being a DAC and an analogue switch. The digital sinusoid waveform is generated by a DDS and then scaled by multiplying the waveform with a scalier value.

Direct digital synthesizers (DDS), or also called numerically controlled oscillators (NCO), are an important part of this system as it enables the generation of a digital waveform that is outputted to the DAC converts it to an analogue waveform. The DDS runs at a frequency of 200MHz, enabling any frequencies to be generated from 0 to 100MHz. The DDS used is implemented is a Xilinx core using Taylor series correction which is mapped into the FPGA. A block diagram of the implementation on the FPGA is shown in Figure 5.39 The DDS is then connected internally to a 14bit multiplier to enable scaling of the output and then to a 200MHz external DAC.

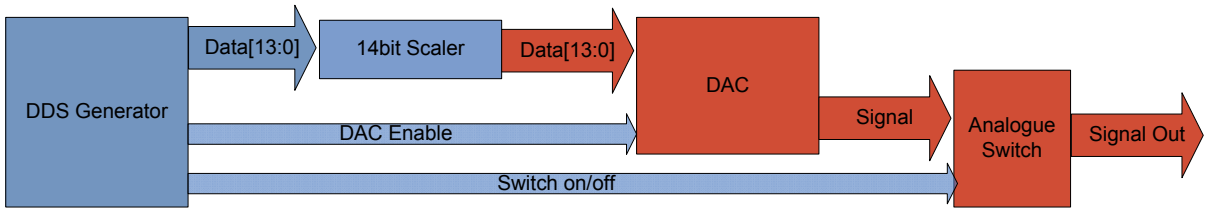


Figure 5.38: Transmit software (blue) and hardware (red).

The DDS shown in Figure 5.39 generates a waveform using a combination of accumulator to calculate what position on the sinusoid should be outputted and then using this with table of values to lookup the amplitude for the output. The lookup table stores the samples of a sinusoid, that when combined with accumulator on the input enables the frequency to be controlled by changing the value added to the accumulator. The DDS also changes the phase of the waveform by adding a fixed phase offset to the accumulated value (see Figure 5.39).

The DDS is part of the free IP supplied by Xilinx and is added to a design using the 'core generator' tool. This generated core is then instantiated within the VHDL code for the named DDS device. Since the DDS is running at 200MHz for the address translator (which runs at 100MHz) to communicate with the DDS a 100MHz to 200MHz data transfer conversion must be created. This is performed using the code shown in Table 5.8 which only outputs the writing output pulse for one 200MHz clock cycle. To address the DDS core the VHDL wrapper reconfigures the addresses and data bus.

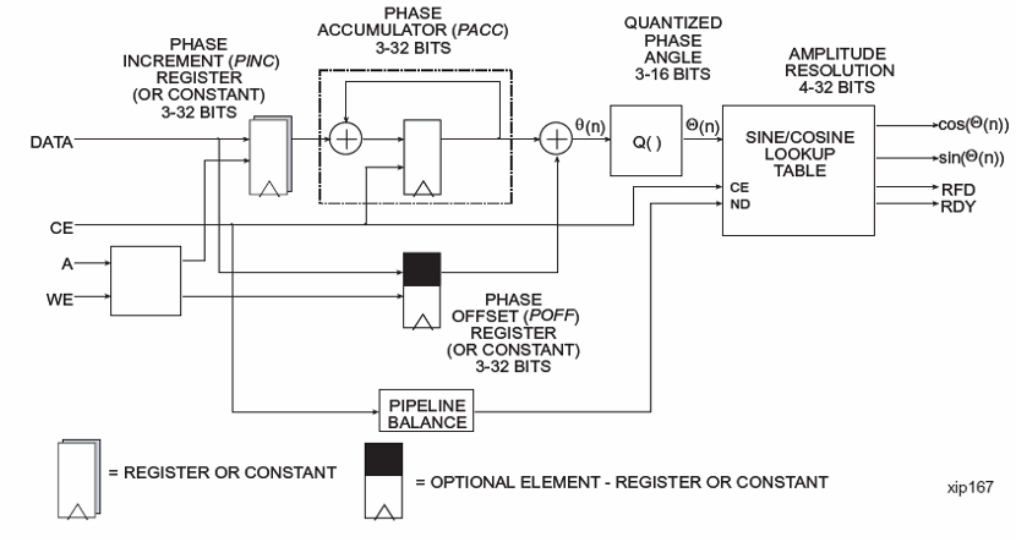


Figure 5.39: DDS Core (DS246 April 28, 2005, Xilinx DDS v5.0).

There are a series of calculations needed to be able to set the DDS to the correct value via the DSP. The Phase accumulator register is 32bits enabling a frequency precision of 32bits, and so gives a frequency resolution (Δf) of 0.046Hz as calculated from Equation (5.1) – where, f_clk is system clock of 200MHz, and B_θ is the bit precision of 32Bits.

$$\Delta f = \frac{f_{clk}}{2^{B_\theta}} = \frac{200x10^6}{2^{32}} = 0.0465661\ Hz \qquad (5.1)$$

```
-- Enable 100M to 200M conversion INTERFACE ----------
DS_Registor: process(DDS_CLK)
      variable last_WE :std_logic:= '0';
begin
      if(rising_edge(DDS_CLK)) then
            if( ADDR(9 downto 8) = "01" and ( (last_WE = '0') and
            (WE = '1') ) and (SEL = '1') ) then
                  DDS_WR_ENABLE <= '1';
            else
                  DDS_WR_ENABLE <= '0';
            end if;
            last_WE := WE;
      end if;
end process;
```

Table 5.8: 100 to 200 MHz data transfer conversion.

To read and write configuration values to the DDS controller the correct DSR (Device Select Register) value must be written to which for the DDS controller is '1'. Also when the correct device is selected the DDR (Direct to Device Register) address is passed to this DDS wrapper device. These are all addressed using a process similar to the one mentioned in the address mapping section. Table 5.9 shows what address the DDR needs to contain to read or write to certain registers. The different registers are used to change the way that the DDS and components associated with it behave.

DDR[9:0]	Type	Register Name	Register Bits
0x001	R/W	DDS_Control_Reg	24
0x002	R	DDS_Info_Reg	24
0x003	R/W	DAC_STARTUP_WAIT	24

0x004	R/W	DDS_Scale	14
0x100	W	DDS Phase Increment	32
0x101	W	DDS Phase Offset	32

Table 5.9: DDR functional description for when DDS controller is selected.

The DDS control register is for setting the DAC power, the output switch, and phase reset on write bit it can be seen in Table 5.10. The DAC Power On (DAC_PO or Bit 0) turns the DAC on (high) or off (low). When the DAC is turned on hardware wait of DAC_STARTUP_WAIT occurs before STARTED_UP pin goes high providing the DAC time to stabilise. The DAC Output Enable (DAC_OUT or Bit 1) bit enables the analogue switch to output when trigger control enables it. When the Reset Output on Write (ROW or Bit 4) is high it enables a reset of the phase of the DDS on a write to the phase increment or phase offset register.

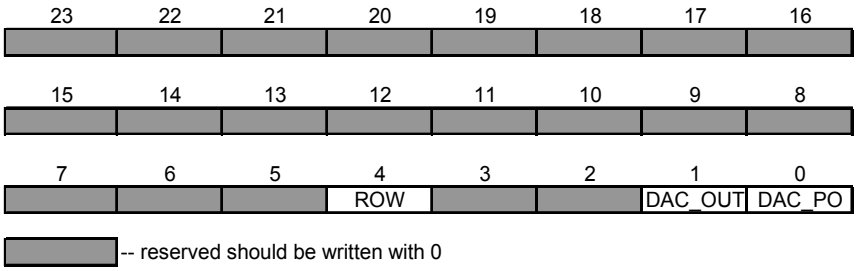


Table 5.10: DDS_Control_Reg.

The DDS Information Register (DDS_Info_Reg [23:0]) provides relevant information about the DDS and components that within it. It has outputs for when the DDS is ready and when the DAC is powered up as seen in Table 5.11. DDS ready (DDS_RDY or Bit 0) is high when the DDS has finished setting up after a DDS reset. The DAC power up (DAC_PU or Bit 1) is high the DAC is stabilised and ready to output – specified when the DAC wait count has finished.

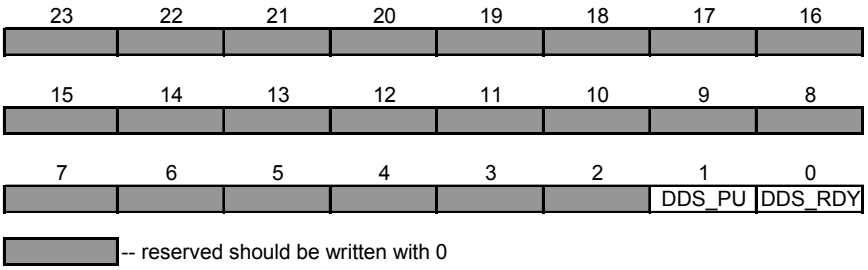


Table 5.11: DDS_Info_Reg.

The DAC start-up wait time register (DAC_STARTUP_WAIT [23:0]) is used to specify how long to wait before asserting the DAC_PU pin high after turning the DAC on. This is used to ensure that the DAC stabilization time is fulfilled before outputting DDS data. The time taken in real time is shown in Equation 5.2 – where, f_{clk} is 100MHz, and DAC_STARTUP_WAIT is DAC_{wait}. The default value for DAC_{wait} is 0x000800, which corresponds to a wait time of 7.81µs.

$$t_s = \frac{1}{f_{clk}} DAC_{wait} \tag{5.2}$$

The start-up wait is controlled using a VHDL counter process (as shown in Table 5.12) which starts counting when the DAC is turned on and outputs the ready signal when a certain count has been reached. This ensures that the DAC is stable before being used.

```
process( clk ) is
variable count : std_logic_vector(23 downto 0) := (others =>
'0');
begin
      if( rising_edge(clk) ) then
            if( DAC_POWER_ON = '0') then
                  count := (others => '0');
                  DDS_POWERED_UP <= '0';
            else
                  if( DDS_POWERED_UP = '1' or count =
                  DDS_STARTUP_WAIT ) then
                        DDS_POWERED_UP <= '1';
                  else
                        count := count + 1;
                  end if;
            end if;
      end if;
end process;
```

Table 5.12: Process for controlling the start-up time of the DAC.

The DDS scale register (DDS_SCALE [13:0]) is a 14bit constant that is multiplied by the 14bit DDS output. This gives a 28bit value of which the top 14bits are sent to the DAC. The Scalier enables the output to be scaled up or down at the expense of the dynamic range of the DAC. This is performed using a multiplier core from the Xilinx's core generator, which can either create a multiplier out of logic blocks or use the one of the hardware multipliers that the Spartan 3 contains. Because of the high speed at which it runs at, a hardware multiplier was used. The hardware multiplier adds a pipeline delay of two clock cycle on to the DDS output.

The DDS phase increment (DDSPI [31:0]) register controls the current output frequency. This is a write only register and is initialised to zero on FPGA start-up. Once set the Phase Increment value will change but is subject to pipeline delay. This is dependent on the pipeline delay of the DDS which should be 12 clock cycles at 200MHz so which is a delay of 60ns which is about the same as a data write from the DSP. The Phase Increment value is a 32bit number and so must be written using the HWE (High Write Enable) pin when performing the write (see Address Mapping section). If the ROW bit from the DDS control register is high when the DDS is updated it will reset with DDS_RDY going low for the pipeline delay.

To calculate a Phase Increment value ($\Delta\theta$) for a given frequency output Equation (5.3) is used. In this example a frequency (f_{out}) of 1.1MHz was chosen which when calculated gives a hexadecimal value to be loaded into the 32bit Phase Increment register of 0x016872B0.

$$\Delta\theta = f_{out}.\frac{2^{B\theta}}{f_{clk}} = \frac{1.1x10^6.2^{32}}{200x10^6} = 2.236223x10^7 = 0x016872B0\ (hex) \qquad (5.3)$$

The DDS phase offset (DDSPO [31:0]) register controls the current phase offset. This is a write only register and is initialised to zero on FPGA start-up. Once set the Phase offset value will change but is also subject to pipeline delay such as the phase increment register. The Phase Increment value is a 32bit number and so must be written using the HWE (High Write Enable) pin when performing the write (see Address Mapping section).

The phase offset is represented with 32bits over 2π. The phase offset output value P_{offset} can be calculated via Equation (5.4) with $\emptyset$ being the angle in radians that is required for the output phase.

$$P_{offset} = 2^{B_\theta}.\frac{\emptyset}{2\pi} = 2^{32}.\frac{\frac{\pi}{2}}{2\pi} = 2^{32}.\frac{1}{4} = 0x40000000\ (hex) \quad (5.4)$$

To set-up the DDS section of the FPGA using the DSP certain commands must be performed. The address mapping for the DDS section is shown in Table 5.13. This makes the assembler code more readable by replacing the addresses with names.

```
; DDS Registors
FPGA_DDS_CR   EQU   $11001     ; DDS Control Register
FPGA_DDS_IR   EQU   $11002     ; DDS Info Register
FPGA_DDS_SW   EQU   $11003     ; DDS Startup Wait Register
FPGA_DDS_Sc   EQU   $11004     ; DDS Scale Output Register
FPGA_DDS_PIL  EQU   $11100     ; DDS Phase Increment low bits Register
FPGA_DDS_PIH  EQU   $11500     ; DDS Phase Increment High bits Register
FPGA_DDS_POL  EQU   $11101     ; DDS Phase Offset low bits Register
FPGA_DDS_POH  EQU   $11501     ; DDS Phase Offset High bits Register
```

Table 5.13: The Register addresses setting for the assembler file

Table 5.14 shows the sequence that is usually followed in setting up the DDS section. This starts by setting the Start-up wait value, and then turns the DAC off then on again to ensure that a start-up wait occurs – the output is also set to be enabled by trigger. The scale output is set to full output and the output frequney is set to the programs start value. And lastly the phase is set to zero. Once the DDS and the ADC have been set-up a loop is set in the DSP waiting for the DDS and ADC wait value before running.

```
;Setup DDS
      move   #$000800,a1
      move   a1,x:FPGA_DDS_SW      ; Write DAC_STARTUP_WAIT value
      move   #$000000,a1
      move   a1,x:FPGA_DDS_CR      ; Turn DAC off
      move   #$000003,a1
      move   a1,x:FPGA_DDS_CR      ; Turn DAC on and Turn on Output
      move   #>8191,b1
      move   b1,x:FPGA_DDS_Sc      ; Set DDS_Scale to1 (2^14-1)
      move   x:TX1,a
      move   x:TX1+1,a0
      move   a1,x:FPGA_DDS_PIH     ; Output high DDS_Phase_Increment
      move   a0,x:FPGA_DDS_PIL     ; Output low DDS_Phase_Increment
      clr    a
      move   a1,x:FPGA_DDS_POH     ; Output high DDS_Phase_Offset
      move   a0,x:FPGA_DDS_POL     ; Output low DDS_Phase_Offset
```

Table 5.14: Example DSP code for setting up the internal DDS.

In conclusion for this section a DDS has been successfully created using the FPGA as to generate the digital output and the DAC to generate the output at frequencies between 1 and 100MHz. This has been developed to work with the DSP board.

5.3.4 RECEIVER DESIGN

The receiver is the most complex part of the FPGA design. It includes the ADC capture and de-randomising of the signal. It also includes the DRPs which are made up of a sine and cosine DDS generators and CIC and FIR filters. It also includes the FIFO data storage device.

The flow of the receiver can be seen in Figure 5.40. The signal is inputted into a buffer amplifier which is then amplified by a factor of two and converted to a differential signal which is then sampled via the ADC. The 16bit ADC digital data is then captured by the FPGA and sent to the ADC wrapper module. Once there the data is de-randomised and inverted (to account for the buffer inversion) it is then made available for use via the DRP's or the FIFO. There is also an ADC overflow detector. This monitors the overflow bit from the ADC and raises a flag high if an overflow has occurred any time since the overflow bit was cleared.

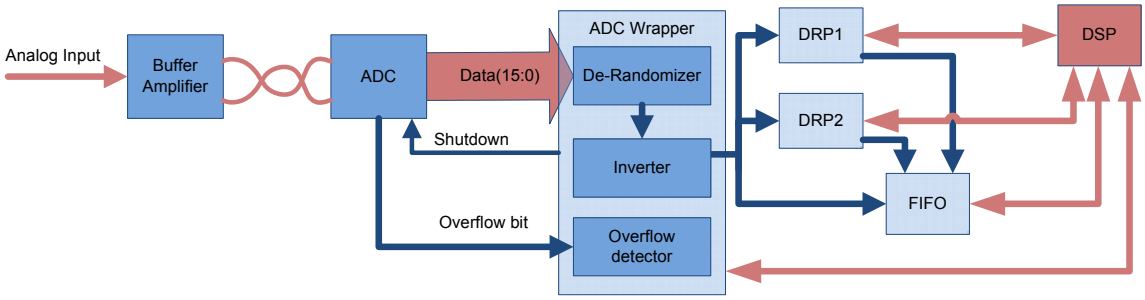


Figure 5.40: Analogue and digital data flow, internal signals (blue) and external (red).

The DSP provides independent control of all the components within the Receiver. This includes the ADC wrapper, the DRP1 and DRP2, and the FIFO. The rest of this section will explain the separate parts of the receiver design in detail and explain how the DSP communicates with each part.

There are two DRP modules implemented in this design that operate simultaneously. One DRP processor would be used for measuring the Bond Rupture noise at the third harmonic while the other DRP is used to monitor the first harmonic. The data from the first DRP can be sent directly to the FIFO for capture and storage so that noise data can be stored without the need for the DSP to perform much monitoring.

5.3.4.1 ADC

The ADC)used is the Linear Technologies LTC2207 with specifications of 16bit data resolution, a sample rate of 105Msps, and maxim input frequency of 700MHz. This has been fully discussed in the hardware section under Receiver ADC.

The ADC wrapper (see Figure 5.41) is the description of the hardware within the FPGA to latch the data at the internal FPGA clock rate. The wrapper also de-randomises the signal from the ADC and inverts the signal. Once the data is latched and converted, it can then be used by other blocks within the FPGA - such as the DRP's. There is overflow capture hardware which samples the overflow bit at the ADC data rate and if it is high the overflow bit will say high until cleared. The ADC wrapper also controls the start-up and shutdown of the external ADC. A start-up wait loop is included enabling the hardware to wait a specified amount of time until the ADC has stabilised at which time started up it held high. The ADC wrapper also has a pipeline wait which enables a certain amount of time to be waited after the trigger sets the ADC to capture, to enable the 7 clock cycles pipeline delay of the ADC hardware to be accounted for. These components are controlled with the DSP via the address translator module.

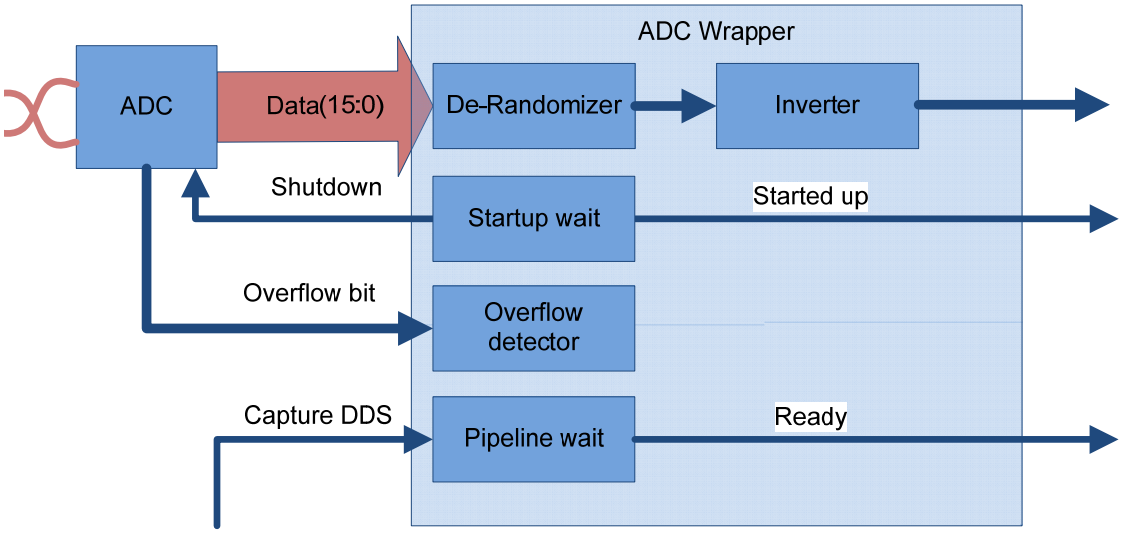


Figure 5.41: ADC Wrapper diagram.

The randomization function on the ADC is there to minimize the interference from the digital output with the analogue input. This is implemented by performing an exclusive-or on all the outputs with the lowest bit. This makes the digital outputs dependant on the lowest bit which is essentially random and so the output is randomized. The only effect on the data is to very slightly raise the noise floor but with better spurious peak responses. The data can be restored by simply performing the same exclusive-or function within the FPGA. The VHDL code used for this can be seen in Table 5.15 enabling the de-randomization to be turned off using the 'ADC_RAND_MODE_IN' variable.

```
process( clk ) is
      begin
            if( rising_edge(clk) ) then
                  ADC_OF <= ADC_in_OF;
                  if( ADC_RAND_MODE_IN = '1' ) then
                        ADC_Data(0) <= ADC_in(0);
                        ADC_Data(1) <= ADC_in(1) xor ADC_in(0);
                        ADC_Data(2) <= ADC_in(2) xor ADC_in(0);
                        ADC_Data(3) <= ADC_in(3) xor ADC_in(0);
                        ADC_Data(4) <= ADC_in(4) xor ADC_in(0);
                        ADC_Data(5) <= ADC_in(5) xor ADC_in(0);
                        ADC_Data(6) <= ADC_in(6) xor ADC_in(0);
                        ADC_Data(7) <= ADC_in(7) xor ADC_in(0);
                        ADC_Data(8) <= ADC_in(8) xor ADC_in(0);
                        ADC_Data(9) <= ADC_in(9) xor ADC_in(0);
                        ADC_Data(10) <= ADC_in(10) xor ADC_in(0);
                        ADC_Data(11) <= ADC_in(11) xor ADC_in(0);
                        ADC_Data(12) <= ADC_in(12) xor ADC_in(0);
                        ADC_Data(13) <= ADC_in(13) xor ADC_in(0);
                        ADC_Data(14) <= ADC_in(14) xor ADC_in(0);
                        ADC_Data(15) <= ADC_in(15) xor ADC_in(0);
                  else
                        ADC_Data <= ADC_in;
                  end if;
            end if;
      end process;
```

Table 5.15: VHDL code for latching and de-randomization of the ADC data.

The overflow detector detects the ADC overflow. This is cleared until the ADC is started up. Once the ADC is started up the overflow detects anytime the ADC's overflow bit is high and

then holds the ADC overflow indicator bit high until a reset is written or until the ADC is turned off then on again. The VHDL code for implementing the overflow detector is shown in Table 5.16.

The waiting functions for start-up wait and ready wait use a 24bit counter and compare it to a register to store the wait amount and since it is run at the 100MHz clock rate a total wait time of 167ms can be obtained. The wait time used for the ADC start-up was around half a millisecond. Though this wait could be done via the DSP it makes much more sense for it to be built into the transceiver system. This wait counter work in conjunction with the trigger module. The trigger module waits for the ADC and DDS to start-up before running.

```
process( clk ) is
begin
      if( rising_edge(clk) ) then
            if( (ADC_POWERED_UP = '0') or (OVERFLOW_RESET = '1')
)then
                  ADC_OVERFLOW_IN <= '0';
            else
                  ADC_OVERFLOW_IN <= (ADC_OF or ADC_OVERFLOW_IN);
            end if;
      end if;
end process;
```

Table 5.16: VHDL code showing the overflow detection.

To read and write configuration values to the ADC wrapper from the DSP the correct DSR (Device Select Register) value must be written to which for the ADC controller is '2', when this is selected the DDR (Direct to Device Register) is passed to ADC controller to be interpreted as shown in Table 5.17. There are four addresses within the ADC wrapper device: 'ADC_Control_Reg' controls the power to the ADC and whether to de-randomize the input; 'ADC_Info_Reg' is for reading back whether the ADC is powered up and ready for output and whether a overflow has occurred; 'ADC_STARTUP_WAIT' controls the amount of time to wait to for the ADC to start-up; and 'ADC_RDY_WAIT' controls the time to wait before outputting the ADC data after the trigger has signalled the ADC wrapper to run.

DDR[9:0]	Type	Register Name	Register Bits
0x001	R/W	ADC_Control_Reg	24
0x002	R/W	ADC_Info_Reg	24
0x003	R/W	ADC_STARTUP_WAIT	24
0x004	R/W	ADC_RDY_WAIT	24

Table 5.17: DDR functional description for when ADC controller is selected.

The ADC control register (ADC_Control_Reg [23:0]) controls two functions shown in Table 5.18. The first being the ADC Power On (ADC_PO or Bit 0). This turns the ADC on (high) or off (low). When the ADC is turned on a hardware wait of ADC_STARTUP_WAIT occurs before ADC_STARTED_UP pin goes high. The ADC Randomizer Input (ADC_RAN or Bit 1) enables de-randomization of the ADC input. This function exclusive-ors' ADC data bits 1 to 15 with ADC data bit 0 – this should only be selected if the randomise function is selected on the ADC with a jumper.

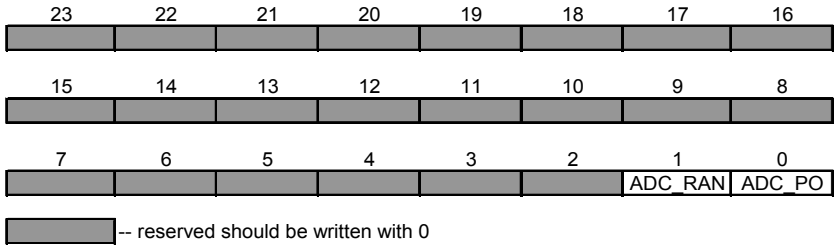


Table 5.18: ADC_Control_Reg.

ADC Information Register (ADC_Info_Reg [23:0]) in Table 5.19 provides relevant information about three functions within the ADC wrapper. ADC ready (ADC_RDY or Bit 0) signal moves high after ADC is set to running by the trigger module and ADC_RDY_WAIT time has occurred. ADC Power UP (ADC_PU or Bit 1) is raised high when the ADC is stabilised and ready to output. This occurs after the ADC_STARTUP_WAIT timer has counted up. The ADC over flow capture (ADC_OF or Bit 2) monitors the ADC's over-flow bit to determine if any over voltage conditions have occurred. This bit is reset on a write to the ADC_OF bit. So to measure if any overflows have occurred over a specified period, firstly you would perform a write to bit ADC_OF wait a specified period then check to see if the ADC_OF bit is high. If it is then an overflow event has occurred during that time period if not then no overflow has occurred or the ADC was not running.

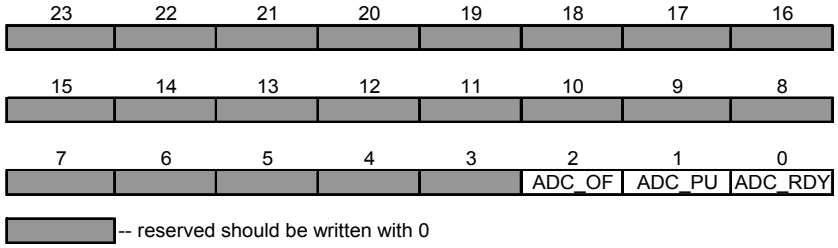


Table 5.19: ADC_Info_Reg

The ADC start-up wait time (ADC_STARTUP_WAIT [23:0]) is the register used to set how long to wait before asserting the ADC_PU pin high after turning the ADC on. This is used to ensure that the ADC has stabilised before capturing data. The time taken in real time is shown in Equation (5.5) – where, $f_{clk}$is 100MHz, and ADC_STARTUP_WAIT is ADC_{wait}. The default value for ADC_{wait} is 0x00D6D8, which corresponds to a start-up wait time of 550µs.

$$t_{\mathrm{s}} = \frac{1}{\mathrm{f_{clk}}}.\mathrm{ADC_{wait}} \tag{5.5}$$

The ADC ready wait time (ADC_RDY_WAIT [23:0]) is the register used to set how long to wait before asserting the ADC_RDY pin high after trigger tells the ADC to run. This is to compensate for the ADC pipe line delay. The time taken in real time is shown in Equation (5.6) - where, $f_{clk}$is 100MHz, and ADC_RDY_WAIT is$ADCR_{wait}$. The default value for $\mathrm{ADCR_{wait}}$ is 0x00000f, which corresponds to a ready wait time of 150ns.

$$t_{\mathrm{s}} = \frac{1}{\mathrm{f_{clk}}}.\mathrm{ADCR_{wait}} \tag{5.6}$$

To set-up the ADC wrapper section of the FPGA using the DSP certain commands must be performed. The address mapping for the ADC section is shown in Table 5.20. This makes the assembler code more readable by replacing the addresses with names.

```
; ADC Registors
FPGA_ADC_CR     EQU     $12001      ; ADC Control Register
FPGA_ADC_IR     EQU     $12002      ; ADC Info Register
FPGA_ADC_SW     EQU     $12003      ; ADC Startup Wait Register
FPGA_ADC_RW     EQU     $12004      ; ADC Ready Wait Register
```

Table 5.20: The Register addresses for setting up the assembler file.

Table 5.21 shows the sequence that is usually followed in setting up the ADC section. Once the DDS and the ADC has been set-up a loop is set in the DSP waiting for the DDS and ADC wait value before running.

```
; SETUP ADC
    move    #$00d6d8,a1         ; 55000*10e-9=550us
    move    a1,x:FPGA_ADC_SW    ; startup time from shutdown
    move    #$00000f,a1
    move    a1,x:FPGA_ADC_RW    ; delay from output to capture
    move    #$000002,a1
    move    a1,x:FPGA_ADC_CR    ; Reset ADC
    move    #$000003,a1
    move    a1,x:FPGA_ADC_CR    ; Start-up ADC
```

Table 5.21: Example DSP code for setting up the internal ADC.

5.3.4.2 DIGITAL RECEIVE PROCESSOR

The DRP's (Digital Receive processor) are needed to convert the high rate ADC data into more manageable data rates at the frequency of interest – to give the DSP time to process the data. The DRP converts a high frequency, large band-width data down to a low frequency, small band-width signal in real time. The DRP system as can be seen in Figure 5.42 includes; a DDS generating a real (cosine) and imaginary (-sine) output; two multipliers to multiply the DDS waveforms with the ADC input data; two CIC filters to down sample the input data; and two scalars to adjust the bit precision to 24bit output. The DRP four variables that can be changed; frequency of DDS; phase of DDS; decimation rate R; and the scalar value for specifying the what 24 bits to look at from the 61bit output from the CIC filter.

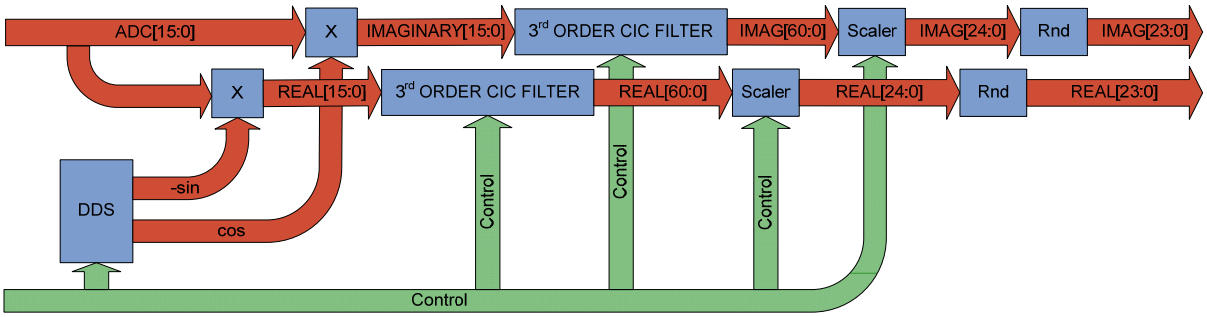


Figure 5.42: Digital Receive Processor.

The high frequency waveform is converted to a low frequency waveform via the multiplying the input data with a sinusoid and also cosine waveform. This produces to data streams which each have two frequency components shown in Equation (5.7) and Equation (5.8) – with f_1

being the lower frequency we are interested in and f_2 the high frequency that will be filtered out in later stages.

$$f_1 = (f_{input} - f_{DDS}) \tag{5.7}$$

$$f_2 = (f_{input} + f_{DDS}) \tag{5.8}$$

Figure 5.43 shows what happens in the frequency domain with f_1 being the low frequency signal generated and f_2 the high frequency that will be filtered out. Having both real and imaginary data enables the magnitude of the waveform to be determined, also enabling the phase data to be calculated, and also when a FFT is performed enabling twice the bandwidth as the negative/imaginary frequencies can be found.

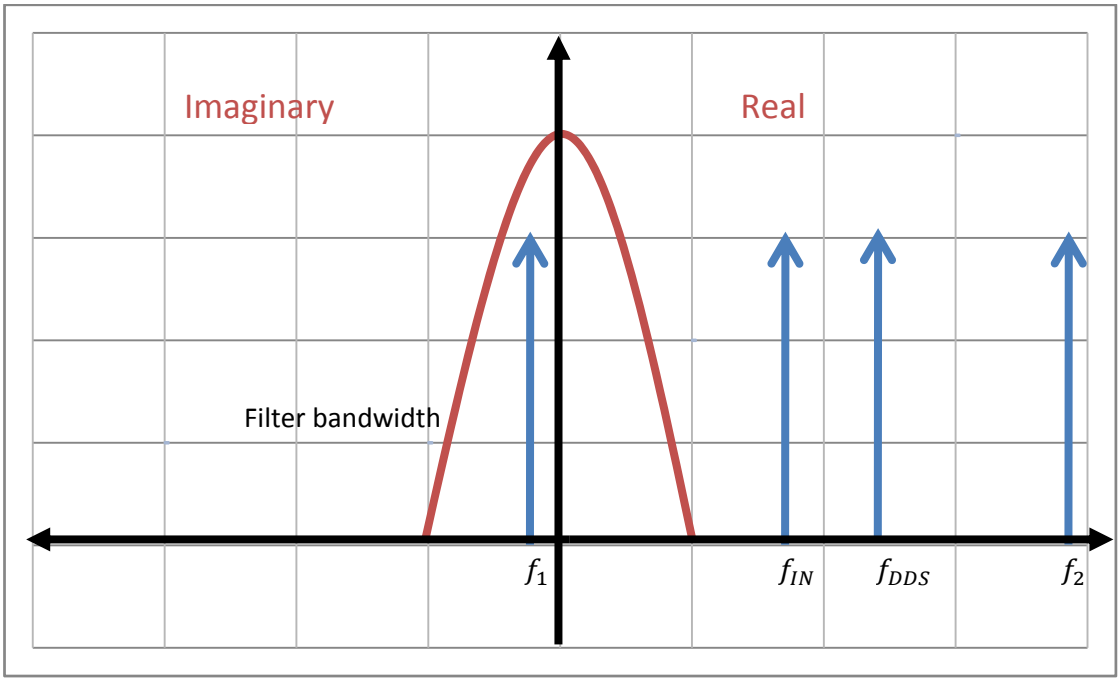


Figure 5.43: Real and imaginary output data from CIC filter, showing how f_{IN} plus f_{DDS} is equal to f_1 and f_2 and with the filter leaving only the f_1.

Once the frequency is closer to DC, the frequency bandwidth can be reduced. The bandwidth is reduced by filtering out any frequencies not in the low frequency band of interest and then throwing away redundant samples. This method is implemented using a decimation filter. The filter that was used is a CIC (Cascaded Integrator-Comb) filter. It was used because it does not use any multipliers, only adders and so has a much smaller resource foot print on the FPGA than an equivalent FIR filter. The CIC filter basically consists of N integrators on the input operating at the input sample rate. Next a Decimation counter R which outputs every R'th data sample. After this is N Comb blocks which runs at the lower Decimated data rate 1/R. Once implemented the data outputted is at the frequency of interest and the selected bandwidth. An example of the result of the CIC filter is shown in Figure 5.44. The data rate at the input to the CIC filter was 100MHz and the output data rate is 100kHz.

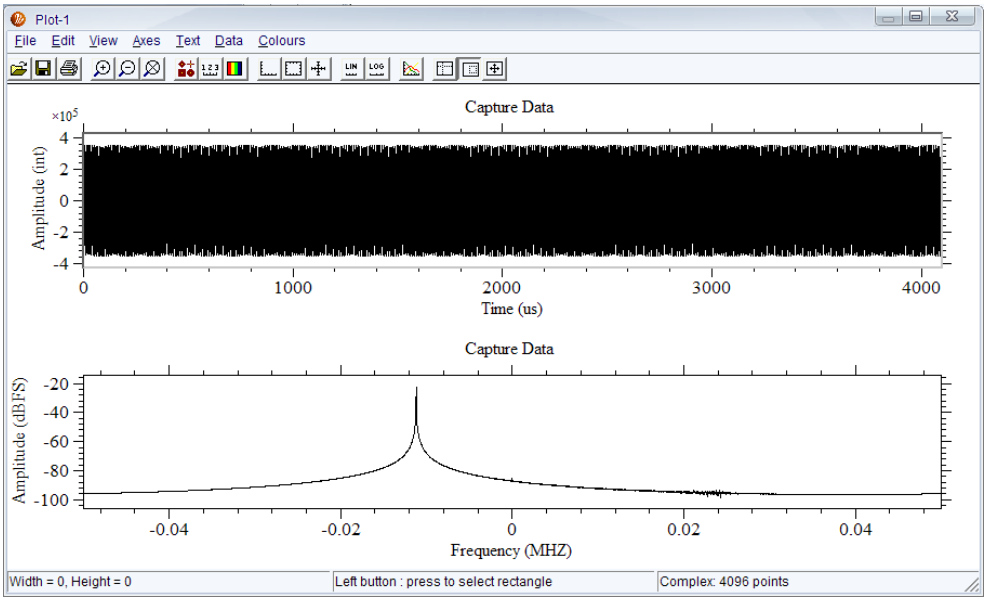


Figure 5.44: Real and imaginary data captured from the DRP (top) then a FFT taken of that data (bottom).

A FIR filter has also been implemented within the FPGA on the first DRP only. This is to increase the amount of time that can be sampled for on the FIFO by down sampling the data further, while also compensating for some of the CIC filter discrepancies.

To read and write configuration values to the DRP controller the correct DSR (Device Select Register) value must be written which will be either '4' for DRP1 or '5' for DRP2. When selected the DDR (Direct to Device Register) is passed to DRP controller to be interpreted as shown in Table 5.22.

DDR[9:0]	Type	Register Name	Register Bits
0x001	R/W	DRP_Control_Reg	24
0x002	R	DRP_Info_Reg	24
0x003	R	Real_Data	24
0x004	R	Imaginary_Data	24
0x005	W	FIR_Control (DRP1 only)	32
0x100	W	DDS Phase Increment	32
0x101	W	DDS Phase Offset	32
0x120	W	Decimate_factor	24
0x130	W	Scale	24

Table 5.22: DDR functional description for when DRP controller is selected.

The DRP Control Register (DRP_Control_Reg [23:0]) controls the enabling of the device, enabling memory to be written to, and resetting of the internal DDS on a write. The DRP control register is shown in Table 5.23. The DRP enable (DRP_EN or Bit 0) whilst high enables the DRP to run after the ADC ready is asserted. The memory output enable (MEM_Out or bit 1) enables a ready output for the FIFO data capture enabling the FIFO to capture the outputted data when also enabled. This is only relevant to DRP1. The DDS synchronous clear on set (DDS_SOS or Bit 4) when high resets the DRP's DDS on a write to the DRP.

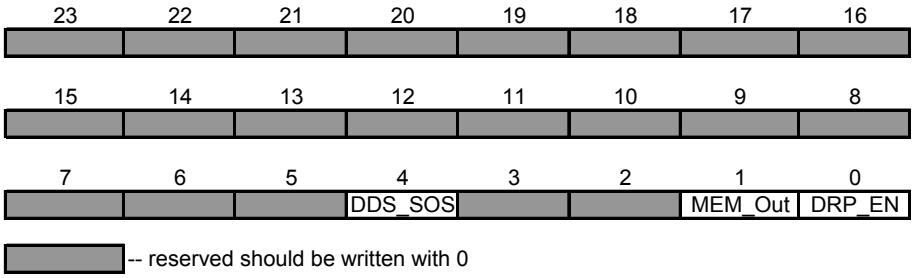


Table 5.23: DRP_Control_Reg.

The DRP information register (DRP_Info_Reg [23:0]) indicates when the DRP has data available for reading. This bit is the DRP ready (DRP_Rdy or bit 0). This bit is high when new data has become available for read from the DRP. This is reset by a read to REAL_DATA.

Real Data (Real_Data [23:0]) reads the real data from the outputted from the scaler DRP block and also a read to this resets the DRP_Rdy bit. A read also latches the Imaginary_Data output so it is essential that the Real data is read before the Imaginary data.

Imaginary Data (Imaginary_Data [23:0]) reads the Imaginary data from the outputted from the scaler DRP block.

DDS Phase Increment (DDSPI [31:0]) is the address for the frequency control of the DDS contained within the DRP. The DDS's Phase Increment register control's the current output frequency. This is a write only register and is initialised to zero on FPGA start-up. Once set the Phase Increment value will change but is subject to pipeline delay. The Phase Increment value is a 32bit number and so must be written using the HWE (High Write Enable) pin when performing the write (see Address Mapping section). For more information on setting the frequency of the DDS see the Transmit Design section.

DDS Phase Offset (DDSPO [31:0]) is the address for the phase control of the DDS contained within the DRP. The DDS phase Offset register control's the current phase offset. This is a write only register and is initialised to zero on FPGA start-up. Once set the Phase offset value will change but is subject to pipeline delay. The Phase Increment value is a 32bit number. For more information on setting the frequency of the DDS see the Transmit Design section.

Decimate Factor (Decimate_Factor[23:0]) controls the down sampling that the CIC filter will perform. The decimate factor must be set between 4 and 16384. This is set with the lower 14 bits of the Decimate factor register. The output sample rate will be the input sample rate (100MHz) divided by the Decimate factor producing a bandwidth range that can be set between 6.1 kHz to 25MHz. By default the start-up of the decimate factor is set to 10. For more information see the CIC filter section.

The Scaling Factor (Scale[23:0]) is needed because as the CIC down samples the input more precision is gathered though averaging which generates a high precision output. That output needs to be scaled to a 24bit output to match the DSPs data with. The output of the scale factor will be CIC[(Scale + 23) : Scale] so to a Scale value of 20 would give the output of the CIC filter of CIC[43:20]. To calculate the optimum value to maximise your dynamic gain Equation (5.9) should be use where; R is Decimation factor; N is the number of integrator and comb

elements and is set to 3; M is the differential delay which is set to 2; and ceil is the function for the next highest integer.

$$B_{max} = ceil(N.\log_2(R.M) + 16) \quad (5.9)$$

For the actual Scale value to use Equation (5.10) should be used.

$$Scale = (B_{max} - 23) \quad (5.10)$$

To setup both the DRP's using the DSP the following address equates are used as shown in Table 2.24. These are set in the FPGA equates file for the FPGA board. This enables the registers to be addressed easily as the actual addresses are not needed to be known only the names.

```
; DRP1 Registers
FPGA_DRP1_CR   EQU   $14001     ; DRP1 Control Register
FPGA_DRP1_IR   EQU   $14002     ; DRP1 Info Register
FPGA_DRP1_DR   EQU   $14003     ; DRP1 Real Data Register
FPGA_DRP1_DI   EQU   $14004     ; DRP1 Imaginary Data Register
FPGA_DRP1_FIRC EQU   $14007     ; DRP1 FIR Control
FPGA_DRP1_PIL  EQU   $14100     ; DRP1 Phase Increment low bits Register
FPGA_DRP1_PIH  EQU   $14500     ; DRP1 Phase Increment High bits Register
FPGA_DRP1_POL  EQU   $14101     ; DRP1 Phase Offset low bits Register
FPGA_DRP1_POH  EQU   $14501     ; DRP1 Phase Offset High bits Register
FPGA_DRP1_Dec  EQU   $14120     ; DRP1 Decimation factor Register
FPGA_DRP1_Sca  EQU   $14130     ; DRP1 Scale factor Register

; DRP2 Registors
FPGA_DRP2_CR   EQU   $15001     ; DRP2 Control Register
FPGA_DRP2_IR   EQU   $15002     ; DRP2 Info Register
FPGA_DRP2_DR   EQU   $15003     ; DRP2 Real Data Register
FPGA_DRP2_DI   EQU   $15004     ; DRP2 Imaginary Data Register
FPGA_DRP2_PIL  EQU   $15100     ; DRP2 Phase Increment low bits Register
FPGA_DRP2_PIH  EQU   $15500     ; DRP2 Phase Increment High bits Register
FPGA_DRP2_POL  EQU   $15101     ; DRP2 Phase Offset low bits Register
FPGA_DRP2_POH  EQU   $15501     ; DRP2 Phase Offset High bits Register
FPGA_DRP2_Dec  EQU   $15120     ; DRP2 Decimation factor Register
FPGA_DRP2_Sca  EQU   $15130     ; DRP2 Scale factor Register
```

Table 5.24: Registers for setting the DRP1 and DRP2 settings.

To setup the DRP's in the DSP assembly, code similar to Table 5.25 would be used. This sets up all the registers within the DRP.

```
; Setup DRP 1
    move   #$000001,a1              ; Turn DRP1 on
    move   a1,x:FPGA_DRP1_CR
    move   #$000005,a1
    move   a1,x:FPGA_DRP1_FIRC
    move   #$000901,a1
    move   a1,x:FPGA_DRP1_FIRC      ; Turn DRP1's FIR fiter on
    move   x:RX1,a
    move   x:RX1+1,a0
    move   a1,x:FPGA_DRP1_PIH
    move   a0,x:FPGA_DRP1_PIL
    move   x:Dec1,a1                ; Update CIC Decimation 100
    move   a1,x:FPGA_DRP1_Dec
    move   x:truncDec1,a1           ; Update Scale
    move   a1,x:FPGA_DRP1_Sca
; Setup DRP 2
    move   #$000001,a1              ; turn DRP2 on
    move   a1,x:FPGA_DRP2_CR
    move   x:RX2,a
    move   x:RX2+1,a0
    move   a1,x:FPGA_DRP2_PIH
    move   a0,x:FPGA_DRP2_POL
    move   x:Dec2,a1                ; Update CIC Decimation 100
```

```
move    a1,x:FPGA_DRP2_Dec
move    x:truncDec2,a1              ; Update Scale
move    a1,x:FPGA_DRP2_Sca
```

Table 5.25: Example DSP code for setting up the internal DDS.

To read data from one of the DRP once setup and running first the ready signal is check and once it is high the data is read first from the real then imaginary port. This is shown in Table 5.26.

```
    .LOOP    x:IgnoreCyc
last1:  move    x:FPGA_DRP1_IR,a
        jclr    #0,a1,last1        ;Wait for Rdy flag
        move    x:FPGA_DRP1_DR,a
        move    x:FPGA_DRP1_DI,a
    .ENDL
```

Table 5.26: Assembly code for reading data from a DRP.

The rest of this section continues on to explain how the multiplier, CIC filter, Scalier, and FIR filter were created and how they operate.

CASCADE INTEGRATOR-COMB (CIC) FILTER

Within wide bandwidth system such as ours an efficient filter method is needed to extract a narrow band signal from a wide bandwidth source. FIR filters are one method but this requires many multiple and accumulates all running at the input data rate - which in this case is 100MHz. FIR filters are not very flexible as a new filter would need to be design for each decimation rate. To solve this problem a novel filter was devised by (Hogenauer 1981) which uses only accumulators. This is the cascade integrator-comb (CIC) filter.

The CIC filter can work as either an up-sampler or down-sampler depending on the order of integrator and comb components. The application of interest in this application is the down-sampler CIC filter which is made up of first a integrator element, then a decimator and then a comb – as seen in Figure 5.45. The integrator is made up of an accumulator which adds the registered value from the last output of this accumulator. The decimator is a counter which outputs a value for processing at the output sample rate. The comb is the current value minus the previous input value. This takes a while to propagate though so there is a sample delay. A simple introduction to CIC filters and how they are made is seen in Donadio's paper (Donadio 2000).

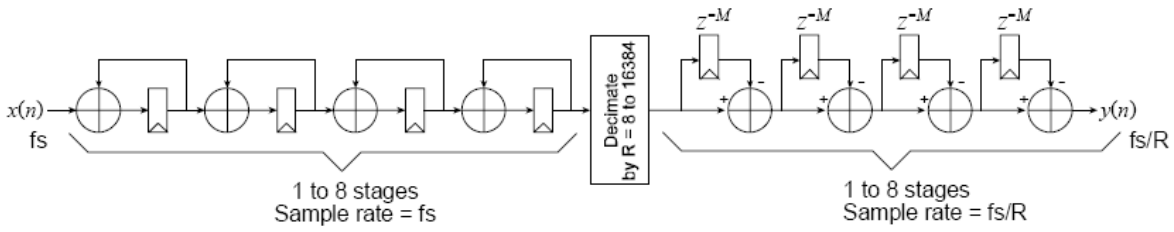


Figure 5.45: CIC filter from Xilinx Cascaded Integrator-Comb (CIC) Filter V3.0 datasheet from Xilinx.

The CIC filter has a magnitude frequency response as shown in Equation (5.11) where: M is the delay; R is the decimate value; N is the number of stages; and f is the frequency. This formula is from the Xilinx CIC filter datasheet. The greater number of stages (N) then the lower the alias

bands are so the less aliasing interference. But for each increase in N a nonlinear increase in the implementation size. This limits the size of filter that can be used. This can value can be increased by using a FIR decimating filter after the CIC filter.

$$H(f) = \left[\frac{\sin \pi \mathrm{Mf}}{\sin \frac{\pi \mathrm{f}}{\mathrm{R}}}\right]^{N} \tag{5.11}$$

Different decimations values change the response of the filter significantly. This is shown in by looking at Figure 5.46 with a decimation of 10 and Figure 5.47 with a decimation of 16384. As can be seen the band-width is significantly decreased with a higher decimation rate and aliasing rejection is also increased though it is basically the same for the side lobes.

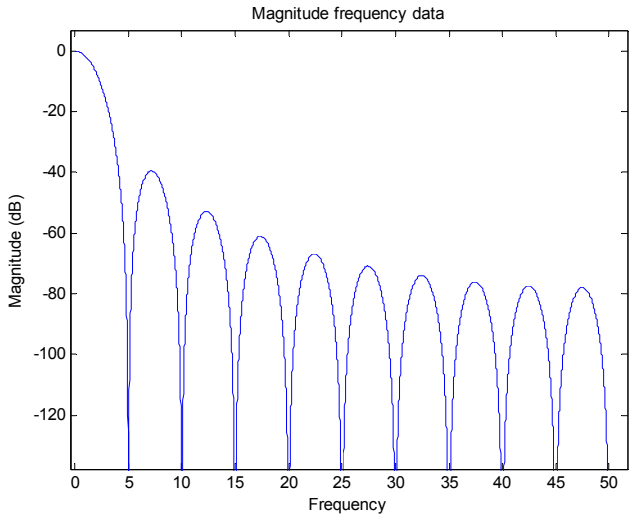


Figure 5.46: CIC filter magnitude frequency response normalised to 100 MHz sampling rate for N=3, M=2, and R=10.

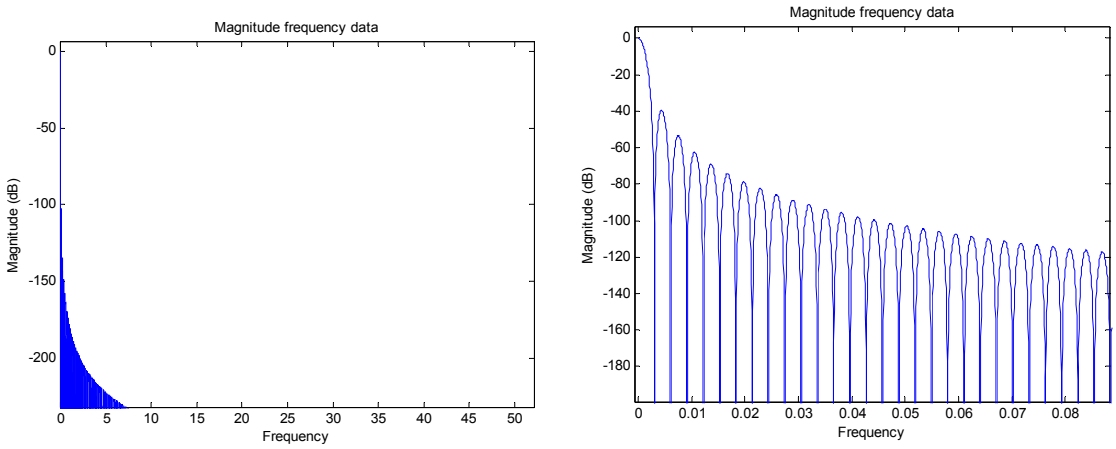


Figure 5.47: CIC filter magnitude frequency response for N=3, M=2, and R=16384. (left) frequency response normalised to 100 MHz sampling rate. (right) Magnified to 90kHz.

To calculate the optimum value to maximise your dynamic gain Equation 5.12 should be used where; R_{max} is the maximum decimation rate; N is the number of integrator and comb elements; M is the differential delay; ceil is the function for the next highest integer; and B_{max}

is the number of bits that the CIC filter data path must be contain – including all the accumulators and registers. This data path must be the correct size otherwise overflow can occur which corrupts the data.

$$B_{max} = ceil(N.\log_2(R_{max}.M) + 16) \quad (5.12)$$

There are many factors to consider when designing the ideal CIC filter for the application specified. Some of the design considerations are the size of the filter versus the precision and decimation amount required. A table of the calculated B_{max} values and estimated size on the FPGA for certain N and R_{max} values can be seen in Table 5.27. This shows that the as the N elements increase the size significantly increases as well. And also that for a high decimation rate the size also increases.

N	R_{max}	B_{max}	Est. size on FPGA
2	127	32	128
2	16383	46	184
3	127	40	240
3	16383	61	366
4	127	48	384
4	16383	76	608
5	127	56	560
5	16383	91	910

Table 5.27: Calculated B_{max} and estimated size on the FPGA for N and R_{max} values for generation of a CIC filter.

Each CIC filter is required to run at 100MHz and though the CIC filters support two channels for quadrature the CIC filter could not run at the 200MHz required for this so two CIC filters are needed per DRP. Also since two DRP's are needed in this design a total of four CIC filters were needed all up. Since there are four CIC filters the design values chosen for the CIC will affect the size on FPGA by a factor of four.

Another factor is the amount of time to propagate though a CIC filter which depends on the number of elements N, the delay M, and the decimation rate R. This is shown in Equation (5.13), where N_{delay} is the number of cycles at the input sample rate before the output is valid. The larger the decimation rate the longer the propagation time but the N and M values also affect the time significantly.

$$N_{delay} = N.M.(R + 1) \quad (5.13)$$

The values that were used after much consideration of all the factors was N as 3, M as 2, and R_{max} as 16383. This gives a resultant width of 61bits with side lobe noise rejection of 40dB. This is not ideal but using a FIR filter after the CIC filter can significantly reduce the noise rejection. Also the FIR filter here can be implemented much more efficiently as there is the sample rate out of the CIC is low compared to the ADC output.

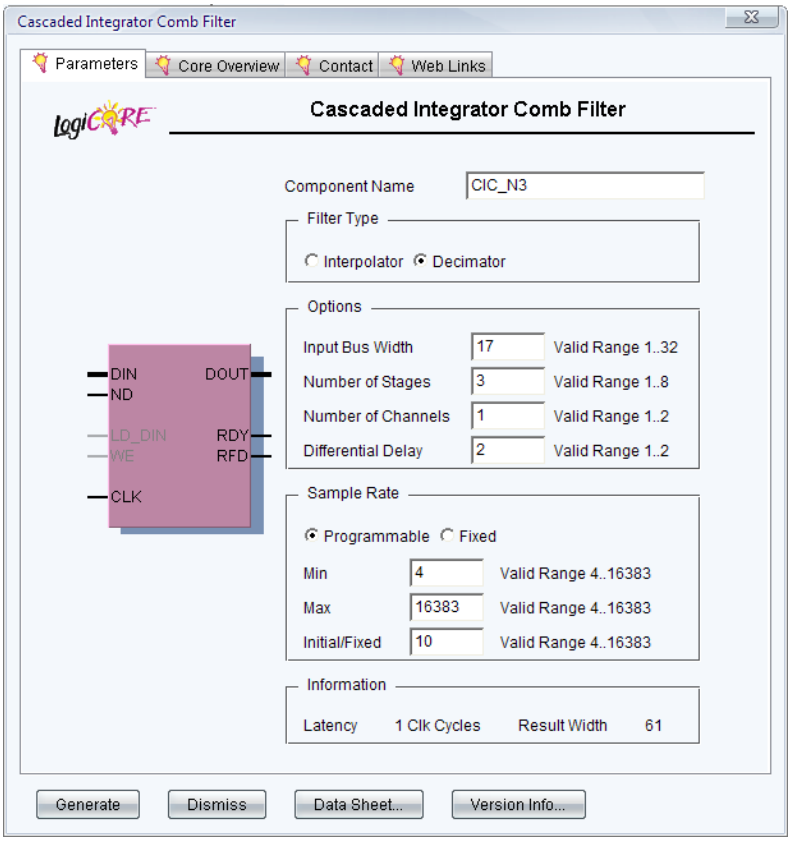


Figure 5.48: Xilinx CIC filter core generator with implemented values.

The CIC filter was implemented using the Xilinx core generator shown in Figure 5.48. The CIC filter generator has some disadvantages over creating a custom VHDL solution. One disadvantage is that there is no clear pin to reset the CIC. Though this reduces the amount of space used by the CIC filter by taking advantage of the FPGA internal structure it means that the CIC will respond from the last data unless enough time is waited to flush it. Also the CIC generator has a bug that to fix an input width of 1 more than needed must be used. This is an unknown quantity and the effect of this on the output a much better option would be to design the CIC filter from scratch so that the structure is known. It is fairly simple to make a CIC filter and one was created but not implemented due to time constraints.

SCALER

A scaler component was created to convert the 61bit CIC output to a 24bit output. This is needed because of the range of decimation values that are used which when changed modify the maximum number of bits to represent full scale. For example: with a decimation value R is 4 the B_{max}value is 25, or R is 16383 B_{max}value is 61. So the maximum bit is required to be between bit 25 and 61 to prevent overflow which is a range of 36.

To implement a scalier to options were considered. The first and simplest would be to use a multiplexer for each of the 24 bits to select which bit out of 36 different combinations. This is quite inefficient use of FPGA resources so another method was proposed using a 60bit shift register and a controller. This method works by shifting the 60bit output up to 36 times to obtain the correct output at one shift per clock cycle. This shift amount does not affect the

data path apart from a small delay as for a decimation value of 4 no shifts are required so that the data can propagate though this device much faster when less decimation is needed and when more shifting is required the data rate is much lower so also easily achievable without affecting the output data rate.

This is controlled using a scale controller which manages how many shifts to perform depending on what control line value. The scale controller uses an internal counter to set how many shifts should occur and when to move on to the next stage in the data path. This is shown in Figure 5.49.

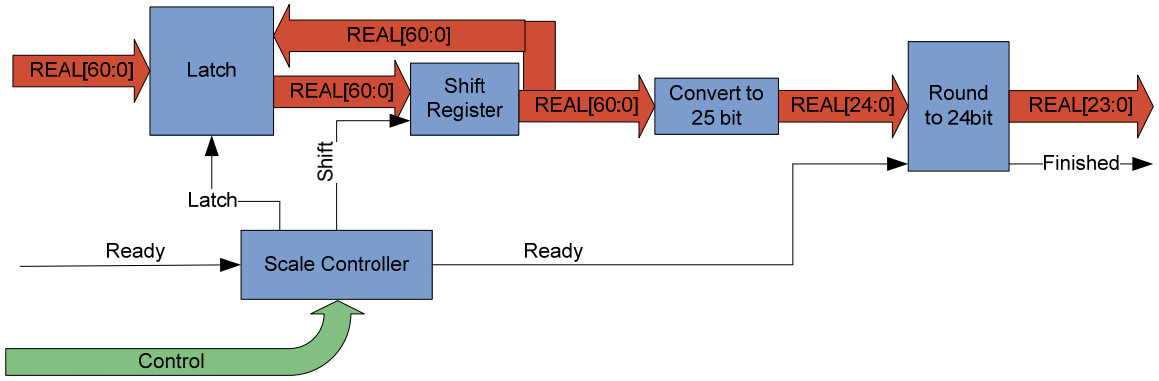


Figure 5.49: Functional diagram of the 61 to 24 bit Scalier and rounder.

Another part to the scalier is a rounder. This was implemented after a DC offset at about -110dBFS down on the FFT of the data was found. Once the rounder was implemented this DC offset disappeared.

The limitations of using a shift register scalier is that the scale value must be correctly set by the DSP software otherwise the data sample rate may not be met.

FIR FILTER

A FIR filter was also added to the first digital receiver to improve the alias rejection as well as to increase the amount of time that the data can be captured for by reducing the band-width. The FIR filter was implemented on the FPGA using the Xilinx core generator. This filter is optimised for dual channel operation so is ideal for our application.

To enable the FIR filter used to be compatible with our data flow setup a dual channel to single channel converter was used on the input and output. This can be seen in Figure 5.50. This enables the same filter to be used to for both channels. This filter can also bypassed if that is selected by the control line.

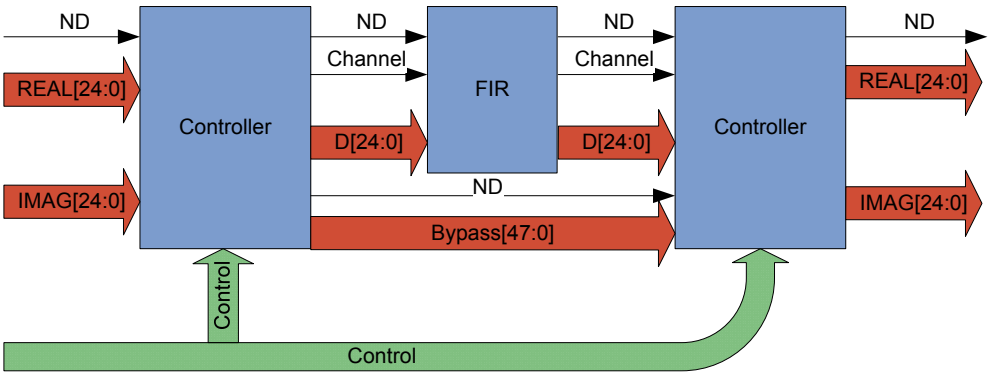


Figure 5.50: functional diagram of FIR and controller.

The filter was created in Matlab using the 'fdatool'. The filter used has 180 taps and has a ten times decimation. This has a response as shown in Figure 5.51. As can be seen in this figure the isolation between the signal and the alias band is about 140dB with a very sharp cut-off frequency. Also the phase change is linear and minimal phase change over the region of interest.

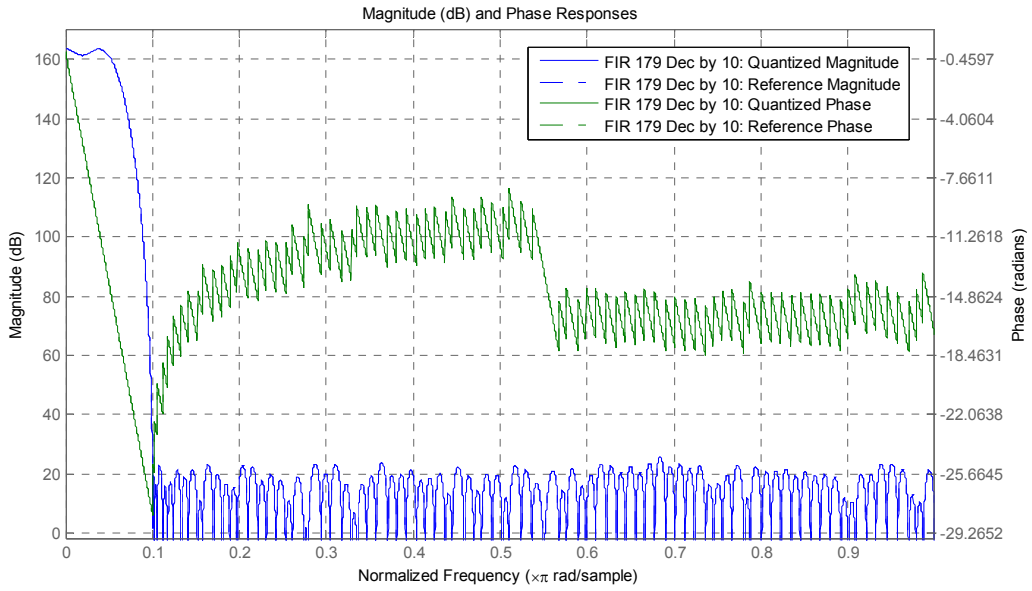


Figure 5.51: Magnitude and phase response for the 179 tap decimation by 10 FIR filter generated by Matlab.

The Xilinx filter generator optimizes the amount of multiply accumulate elements and the amount of memory used depending on the clock rate that the FIR runs at and the input clock rate. Because the input data rate is variable a maximum data rate was chosen for the FIR filter to run at to minimize the logic resources that the FPGA will use. This maximum input data rate was chosen to be 200 kHz. This enables the FIR filter to operate on an input rate between 6.1 kHz and 200 kHz giving an output range of between 610 Hz and 20 kHz.

Using a FIR filter after the CIC filter enables some of the aliasing affects of the CIC filter to be cancelled out due to the way the CIC filter folds back. This can be seen in Figure 5.52 which demonstrates the aliasing folding back. The top curve in this figure is the CIC filtering of the data and the others are the folded back aliased data. If a 10 times decimation FIR is applied to

this data it can be seen that the first aliased interference is reduced to 77dB which is an improvement of 37dB on the original CIC filter. It also has the advantage of looking at the region where the CIC filter data is flat so the output FIR response is much better with less attenuation in the region of interest.

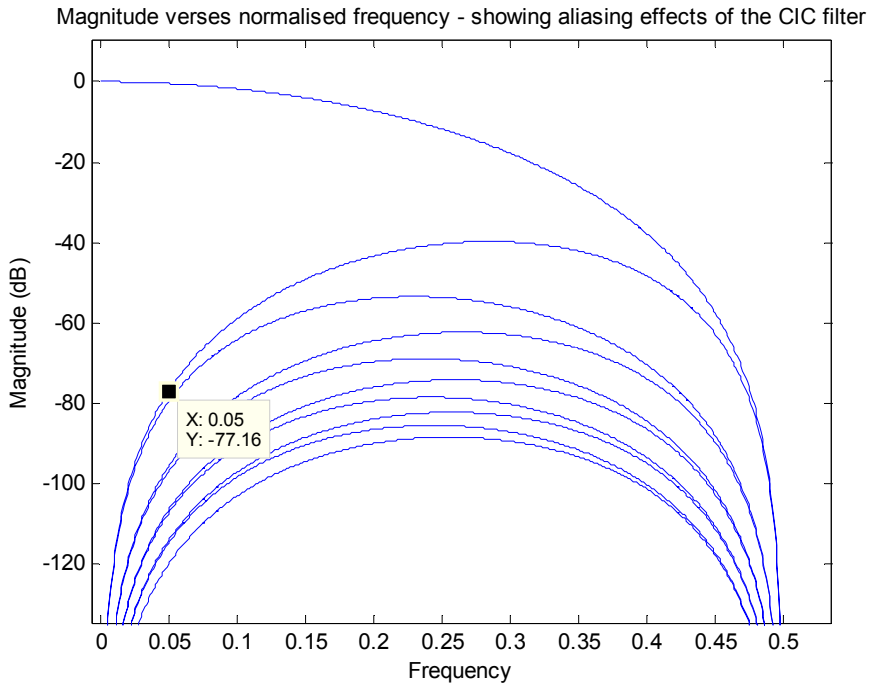


Figure 5.52: Filter response of the CIC filter with the alias response folded back.

DATA STREAM CONVERTER – FOR MEMORY STORAGE

An important component of the DRP is the data stream converter - which converts the outputted data to the correct format for writing to the RAM. This operates by converting the two 24bit data streams from the CIC or FIR filters and converts them into three blocks of 16bit data. This is achieved using a state machine and it starts running whenever an output is available. This block can run at decimation rates down to as low as four.

5.3.4.3 STORAGE

The storage of data into the RAM is controlled using a FIFO (First in First out) device and also a memory controller. The FIFO device is created within the FPGA logic using two address counters for the top of the memory address and the bottom address which then are connected to a memory controller which communicates with the external SRAM (Static Random Access Memory). The FIFO has indicators available for external DSP read back to show when it is ready for read/write, when it is full or empty, as well as a level indicator to indicate how many elements are stored within the RAM. The FIFO is controlled via DSP using the standardised address interface. The FIFO controller, interface to the memory controller, and the RAM connection is shown in Figure 5.53. The FIFO enables the capture of the raw ADC data or data directly from the first DRP.

The RAM used is a 256k x 16bit memory. This means that the FIFO needs an 18bit address counter to address all the memory elements. The FIFO is implemented using two 18 bit grey code counters to one two address the writing of the memory and one two address the read point. These are updated on a successful read or write. Grey code counters were used to minimise the number of output bits switch enabling less power supply interference from switching, line interference and also decreasing the timing constraint slightly for the FPGA. The FIFO is limited to the memory space so that overflows do not occur. The FIFO also has a register showing the present level of data contained in the FIFO so that that the DSP can attain how many elements are in memory.

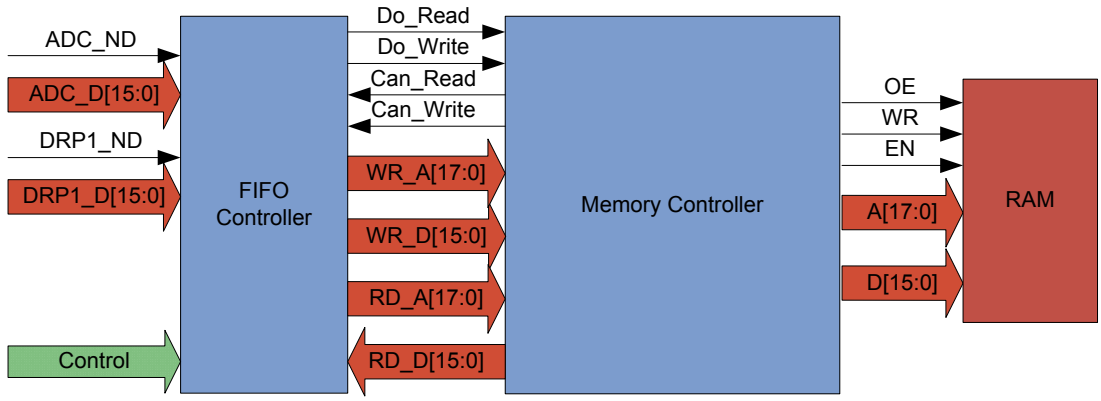


Figure 5.53: Functional description of the FIFO memory controller for external RAM.

The FIFO can be set to either record the RAW data from the ADC or from the first DRP. The memory can be read at the same time if the data is from the DRP but if RAW ADC capture mode is set the memory cannot be read at the same time - as this operation requires all the available memory bandwidth. The memory capture can be set up to run until the memory is full requiring no intervention from the DSP during capture. This is useful as the data can be recorded independently from the capture of data from the second DRP. The reading from the FIFO to the DSP is implemented via a standard DSP read instruction to a specified memory element within the FPGA.

The external RAM is controlled using a RAM controller. The RAM controller is essential a state machine which requires tight control of the timing of the read and write cycles. The external memory can written at speeds of up to 100MHz (with a one cycle turnaround from a read cycle) and has an 18bit address bus allowing storage of 262,144 x 16bit words. Because of delays though the FPGA a pipeline was implemented on the write cycle – this is to enable the FPGA to stick to the tight timing restraints needed to enable a write. This means that there is a one cycle delay between a write being requested until it is carried out to the SRAM. The read instruction takes two clock cycles (50MHz), as a fast read is was not essential as the DSP can only read at much lower than this. This was implemented using a synchronous state machine seen the following Figure 5.54.

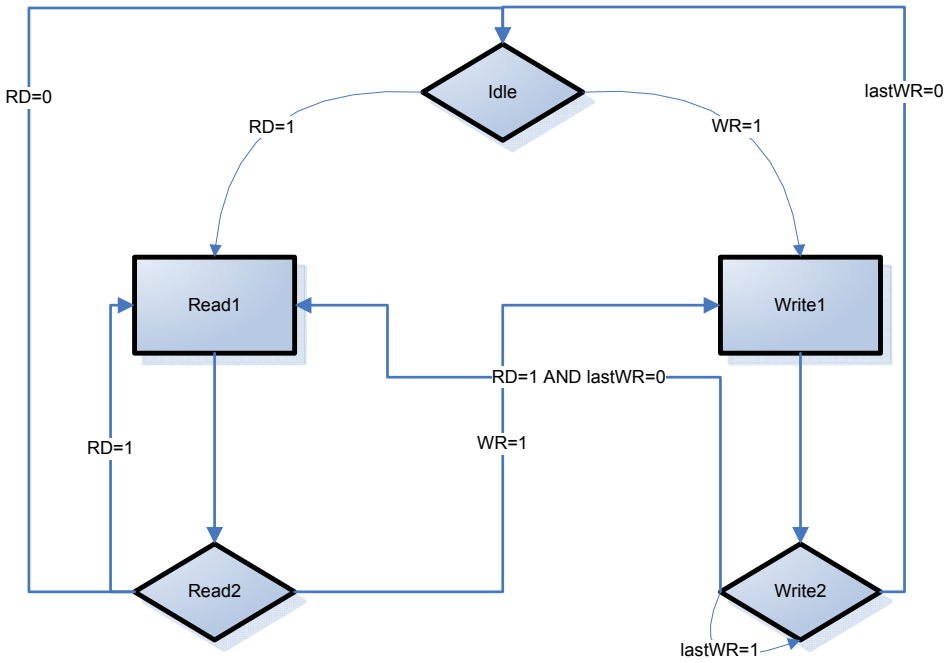


Figure 5.54: State diagram of SRAM controller running at 100MHz inside FPGA.

To read and write configuration values to the FIFO controller the correct DSR (Device Select Register) value must be written - which is '6'. When selected the DDR (Direct to Device Register) address is passed to FIFO controller to be interpreted as shown in Table 5.28.

DDR[9:0]	Type	Register Name	Register Bits
0x000	R/W	Data_Read	24
0x001	R/W	FIFO_Control_Reg	24
0x002	R/W	FIFO_Info_Reg	24
0x003	R/W	FIFO_LEVEL	18

Table 5.28: FIFO Device Select register address.

The data read register (Data_Read [23:0]) is for reading data out of the FIFO. This register reads the 16bit data from the FIFO and sign extends it to 24bit 2's complement. In the case of the FIFO being empty the register will be zero. When reading from the RAW ADC data no further processing is needed after reading but when using the DRP data capture the data must be reconstituted from sets of three 16 bit blocks to two 24 bit blocks as can be seen in Figure 5.55. This is done within the DSP by reading in blocks of three and shifting the data as the data is read using the DSPs 48bit registers, then saving each half of the 48bit registers independently.

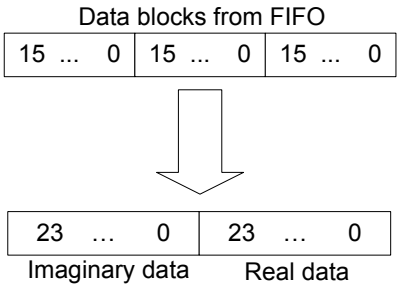


Figure 5.55: Conversion from three 16bit blocks to two 24bit block performed within the DSP.

The FIFO Control Register (FIFO_Control_Reg [23:0]) controls the enabling of the capture of data and the selection of the input to use and also the resetting of the data. The FIFO control register is shown in Table 5.29. The reset bit (RESET or Bit 0) whilst high resets the FIFO counts and levels to zero so while not actually resetting the data it makes the old data inaccessible and effectively so is reset. The FIFO capture enable bit (Capt_EN or Bit 1) enables the FIFO to start capturing as soon as new data becomes available from the selected data source. The select input bit (Select_IN or Bit 2) enables the selection of the input to the FIFO. When this bit is low the RAW ADC data input is selected and when high the data from the DRP is selected – though this is 48bits of data it is converted to 3 times 16bits by the memory converter within the DRP. The FIFO is always reset automatically on this bit changing.

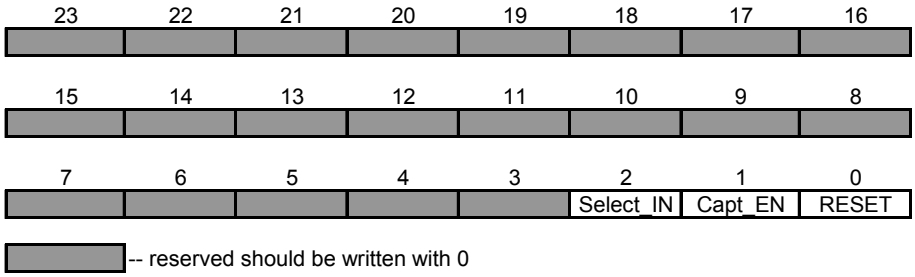


Table 5.29: FIFO_Control_Reg

The FIFO information register (FIFO_Info_Reg [23:0]) indicates whether the FIFO is empty or full. This register can be seen in Table 5.30. The full indicator bit (FULL or bit 0) is high when the FIFO is FULL and so cannot accept any more data. This is cleared by reading data or resetting the FIFO. The empty bit (EMPTY or bit 1) is high when the FIFO is EMPTY and has no data written to it. This is cleared when data is set to the FIFO and on reset will be high.

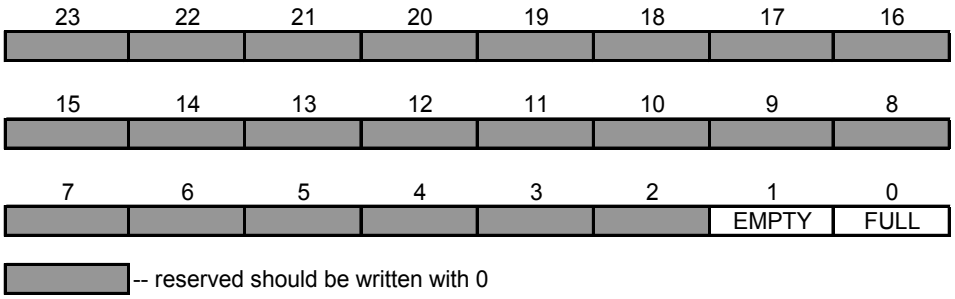


Table 5.30: FIFO_Info_Reg

The FIFO level (FIFO_LEVEL [17:0]) is a 17 bit register containing the current level of the FIFO. On a reset this will be zero and when full this will be the decimal value 262,143.

Table 5.31 is the example code for the DSP56803 board setting up the FIFO on the T-ReX board, setting to capture the RAW ADC data then waiting a set amount of time and then copying all the captured data into the DSP memory for read back to the PC.

```
; Setup FIFO
move    #$000001,a1
move    a1,x:$16001   ; Reset FIFO
move    #$000002,a1
move    a1,x:$16001  ; Set to Capture ADC data
move    #10,r7
bsr     wait       ; wait 10us
.loop    x:$16003  ; Read all FIFO data
          move    x:$16000,a  ; Read Capture data
          move    a1,y:(r5)+          ;Write to DSP memory
.endl
```

Table 5.31: Example DSP code for setting FIFO and reading back RAW ADC data.

Table 5.32 shows the example code for reading back of data captured by the FIFO from the DRP.

```
.LOOP    x:samples
          move    x:FPGA_FIFO_D,x0
          move    x:FPGA_FIFO_D,a0
          asl     #8,a,a
          move    x0,a1
          asl     #8,a,a
          move    a1,y:(r5)+          ;Write to memory
          move    a0,x0
          move    x:FPGA_FIFO_D,a1
          and     #$00ffff,a
          or      x0,a
          move    a1,y:(r5)+          ;Write to memory
.ENDL
```

Table 5.32: Example DSP code for reading back DRP captured data from the FIFO.

5.3.5 TRIGGERING

The trigger control block controls the waiting for components to be ready then the synchronisation of the components that require it. This removes from the DSP the burden of ensuring the devices are started up and synchronised in the correct order. This also enables the FIFO to start capturing at the correct time and not before or after as the run of the ADC output is only enabled once the trigger allows it. This is important as the DSP can only communicate with FPGA at rates of ~20MHz (50ns) whereas the trigger state machine is operating at 100MHz (10ns) enabling devices to be synchronised at the FPGA internal clock speed.

The trigger state machine runs in the sequence shown in Figure 5.56. First there is one main control register which is called GO, which starts the state machine sequence. Once in go mode the first action performed by the trigger control is to wait for the ADC and the DAC to start up. The next action is to synchronise all the DDS's (output DDS and also the DDS within the DRP) by holding their synchronous clear pin high for a clock cycle. The state machine then waits for all DDS to raise their ready flags. Once ready the ADC_RUN pin is held high and the output is

enabled for the DAC. Finally after the ADC_RDY pin is high the Global ready signal is raised high and passed to the DRP's and FIFO capture blocks.

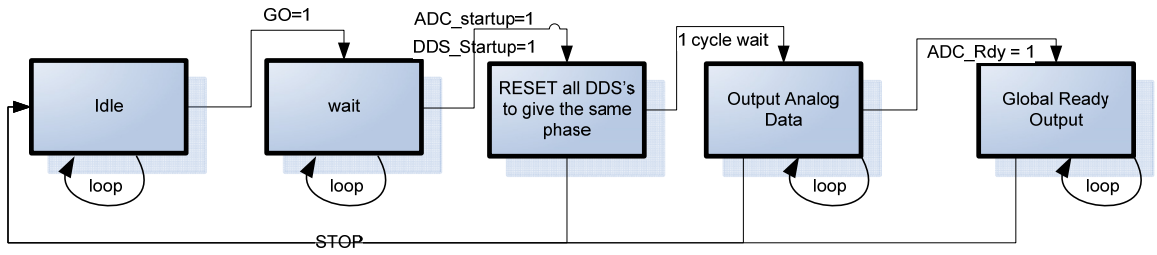


Figure 5.56: Trigger control state machine

To read and write configuration values to the Trigger controller the correct DSR (Device Select Register) value must be written to which for the Trigger controller is '0', when this is selected the DDR (Direct to Device Register) is passed to Trigger controller to be interpreted as shown in Table 5.32.

DDR[9:0]	Type	Register Name	Register Bits
0x001	R/W	Trigger_Control_Reg	24
0x002	R	Trigger_Info_Reg	24

Table 5.32: DDR functional description for when Trigger controller is selected.

The trigger control register (Trigger_Control_Reg [23:0]) controls only the starting of the state machine. This is shown in Table 5.33. If go (GO or bit 0) is high the state machine is started if GO is low then the state machine is reset with its default values, which is; disable analogue switch output, turn ADC_RUN off, and disable GLOBAL ready.

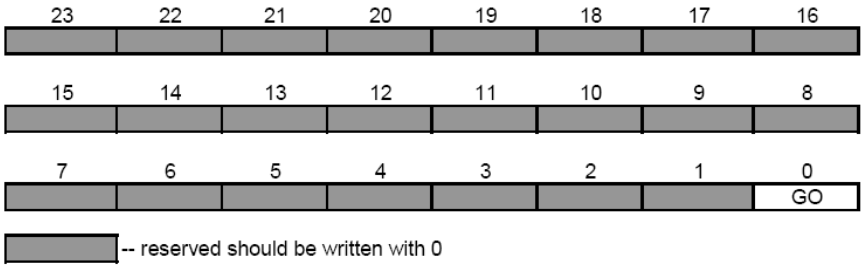


Table 5.33: Trigger control register.

The trigger information register (Trigger_Info_Reg [23:0]) is used to indicate all the input variables into the trigger state machine as shown by Table 5.34. DDS powered up (DDS_PU or bit 0) is high when the DDS is powered up. ADC powered up (ADC_PU or bit 0) is high if the ADC is powered up. DDS ready (DDS_RDY or bit 2) is high if DDS is ready for output. ADC ready (ADC_RDY or bit 3) is high when the ADC is ready for output. Global ready (GR or bit 4) is an output which is high when the trigger state machine has completed its procedure. The DDS output enable (DDS_OUT or bit 5) is high when the DDS output is on.

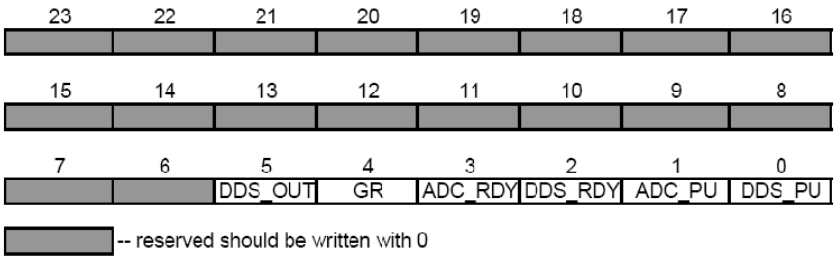


Table 5.34: Trigger information register.

5.4 SOFTWARE

5.4.1 INTRODUCTION

This section covers the design of the software algorithms convert a generic FPGA transceiver into a useful tool to examine QCM for the use as a biosensor. The software section includes the DSP assembler code and also the prospa GUI's. The interaction between the DSP and prospa is the same as described in the "Digital Transceiver system" chapter under "Software" and so this will not be repeated. However there are a number of new programs and also improvements of which this chapter is all about.

An important improvement to the T-ReX system is a calibration program with which the calibration data collected is used to set up the correct input and output levels on all the other programs. The program is called "Calibrate". This program enables the calibration for the specific setup used and enables the output voltage to be calibrated with multiply different amplifiers and attenuators and still enable the correct input and output voltage to be indicated.

A major addition to the system with the T-ReX device is the ability to capture the RAW ADC data directly to RAM. This enables the ADC spectrum to be analysed – effectively working as a spectrum analyser. The prospa GUI used to perform the data capture is called "FFT Capture". Which enables a frequency to be excited as well as measuring the DRP data output.

Another program created is called "Frequency Scanner". This program enables the equipment to work as a network analyser by outputting a range of frequencies and looking at the magnitude change over that range. This is used to find the resonant frequency of the QCM performing essentially the same job as the "Wobble Centre" program in the previous version.

The final program developed is a holistic program for Bond Rupture excitation and frequency and noise monitoring. It is called "Bond Rupture". This combines the magnitude methods with zero crossing methods to measure the QCM's frequency as well as also using original methods in the monitoring for Bond Rupture noise.

5.4.2 SYSTEM CODE DIRECTORY STRUCTURE

The prospa and DSP files are stored within a single directory structure as shown in Figure 5.57 below.

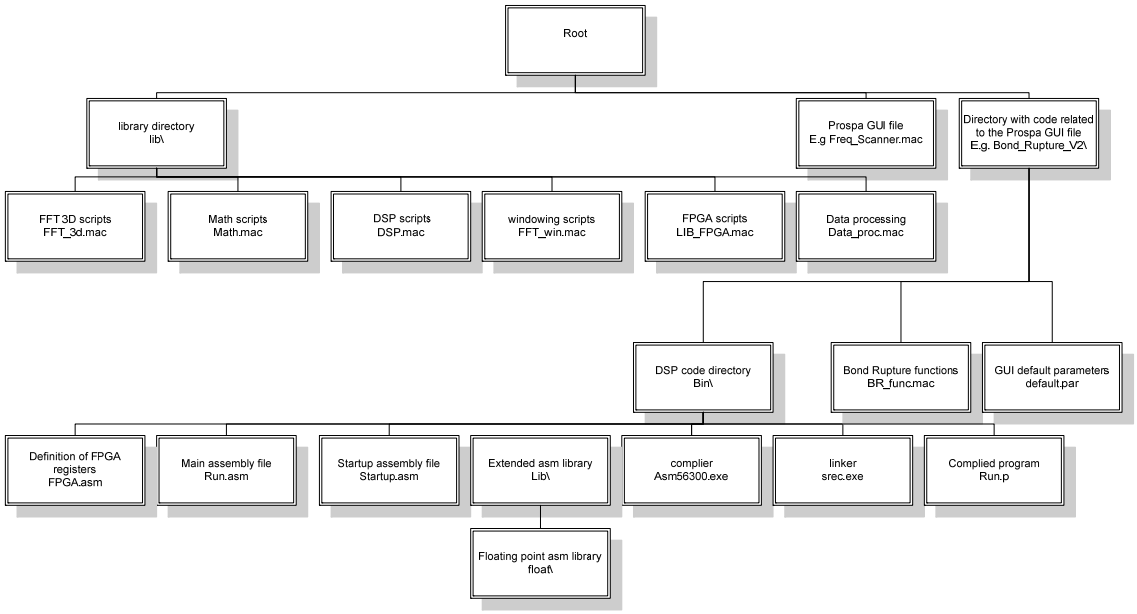


Figure 5.57: Simplified file and directory structure of the code directory developed for prospa GUI and libraries as well as the DSP files and libraries.

5.4.3 CALIBRATE

The Calibration program is required to correctly calibrate the frequency and magnitude response of the DAC, ADC and the amplifiers and filters that make up the system. The calibration method used enables the calibration of the input and output separately and enables multiple different configurations within the same system. This enables the user to select the a configuration when running an experiment then when the configuration is changed the user can create a new configuration file without over writing the old one. This configuration is then used to set the output voltage/power and scale input voltage/power. The calibration data and functions are used by the other prospa programs in the sections to come.

The data captured from the calibration runs is stored in a file system based database – see Figure 5.58. This starts with the primary scan data. The primary scan saves data such as the frequency centre, the frequency span, the number of frequency points to use for the scan, and the scan bandwidth each time a new parameter is created. This data is then used as parameters into the Frequency scan software to generate the scan data required. This ‘Scan Data’ is used as a baseline frequency response for comparison with other runs and to determine the voltage scaling required given a certain frequency. The Attenuation scan uses the same primary parameters to determine the frequency response but is run after an attenuator is added into the system. The data gathered is then subtracted from the primary ‘Scan Data’ to determine the attenuators frequency response. Multiple attenuation scans are able to be stored. The ‘Drive calibration’ run is run using the primary scan and attenuation value as primary values. The Attenuation limits the output of the DAC to the amplifier so as not

to overload the ADC input. The drive calibration has a number of stored values needed for calibration including: Frequency Scan – this is the frequency scan at a specified gain value used to calibrate the output voltage for a given frequency; gain scan value - ; Gain min – the minimum value to use to set the gain of the VGA; Gain max – the maximum gain value for the VGA; Gain scan freq – the gain value that the 'Frequency scan' is run at; Voltage Array – the voltage straight line polynomial values; Optimisation scale array – array of optimised 14bit scale values to achieve optimum voltage output. The 'Input Scan' data values stored are as follows: Attenuation – the attenuation value used; Gain2 Max – the maximum number of gain values for the first input amplifier; Gain3 max – maximum number of gain values for the second input amplifier; gain scan 2D – values stored from scanning over each possible gain2 and gain3 value creating a 2D image array. Gain Settings list – is the gain2 and gain3 array values chosen to use. Voltage – is the array corresponding to the 'setting list' showing the input scaling values as a voltage gain. These are settings are stored in a directory structure similar to a simple database enabling multiple configurations. They are generated using the Calibration GUI which shall be discussed next.

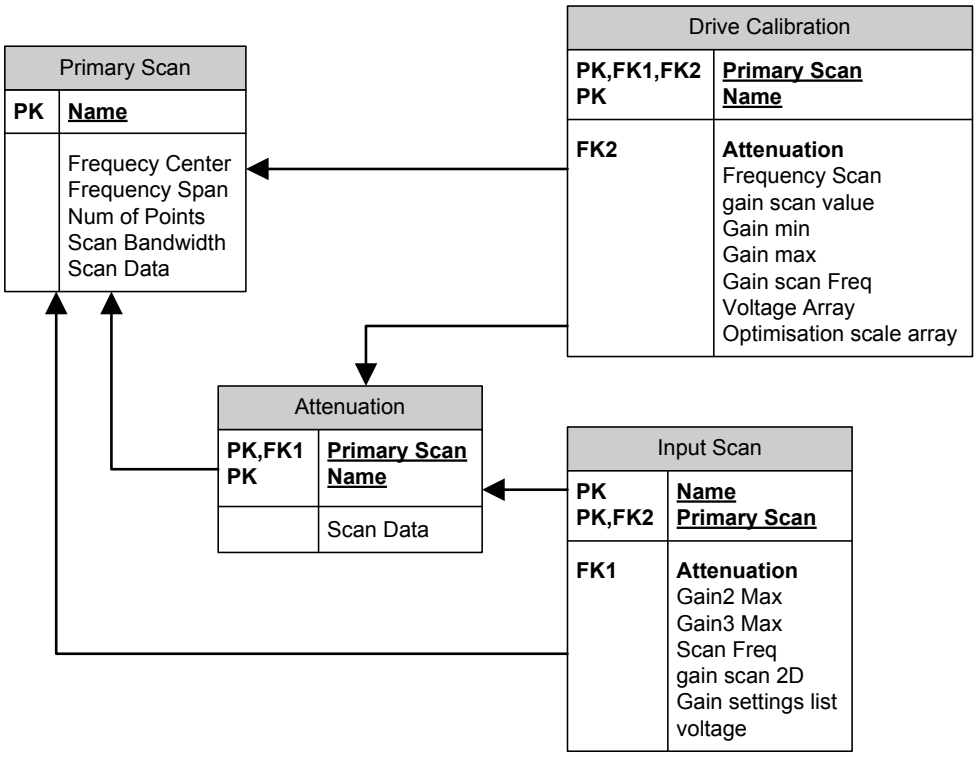


Figure 5.58: Calibration software database.

The Calibrate prospa GUI is made up of three tabs: Primary Scan – for calibrating the baseline frequency response of DAC and to calibrate the Attenuator blocks; Drive Calibration – for calibrating the output VGA's; and input calibration – for calibrating the input VGA's.

The Primary scan tab is used for setting the baseline frequency response and the attenuation settings. This GUI tab is also used for setting the scan variables used for the rest of the frequency scans. This GUI can be seen in Figure 5.59. The first component of this is the primary scan. This scan is run with the DAC connected to the ADC (with any other DAC filters included) and it is assumed that the ADC frequency response is linear (there should be very minimal

effect from the ADC bandwidth as low frequencies < 50MHz). This is run using the Frequency scan variables set. The scan data can then be used to determine the frequency response of the DAC output. This scan is shown in Figure 5.60. The ADC's accuracy is limited using frequency scanner mode to about -80dB as below this the frequency magnitude cannot be accurately determined. This poses a problem when comparing the primary scan data with attenuated data when the filter is present as the data can fall below what can be measured accurately and cause errors in the measured frequency.

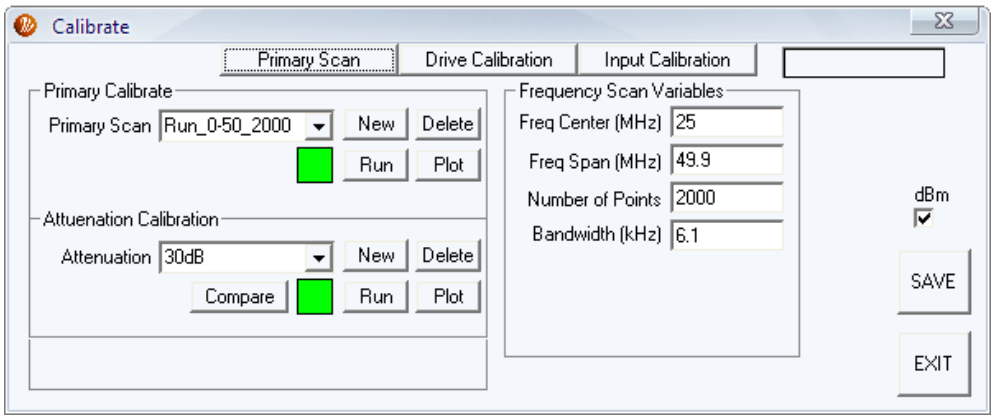


Figure 5.59: Calibration GUI - Primary Scan tab.

The interface is fairly simple as a new primary scan value can be created by simply click on the pull down menu and writing in a new name. When the enter key or new button is pressed this text value is saved as the new primary scan variable and the pull down menu is updated with the new value. This can be deleted by selecting a variable and pressing the delete button. All the other data is linked to the primary scan selected so if this is changed the linked values will change. The coloured square below the primary scan pull down menu is used to show if a scan has been run using that name yet. It is green if it has run and red if it has not. The run button is used to run the experiment. The plot button plots the stored data from the frequency scan experiment. An example of this can be seen in Figure 5.60.

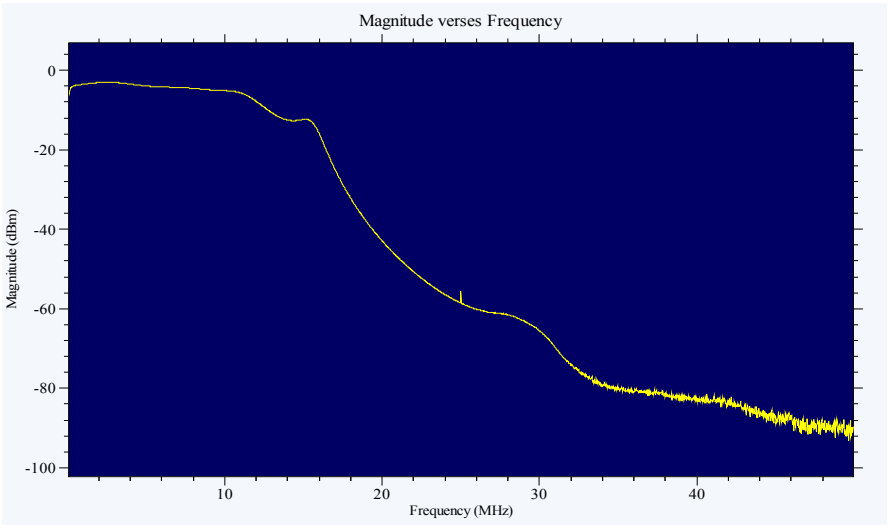


Figure 5.60: Primary Scan with a 10.7MHz filter on the output of the DAC and then connected to the ADC.

The attenuation box is used to measure the frequency response of the attenuations needed to limit the amplitude from the output and input amplifiers. The experiment consists of the same setup as in the primary scan with the exception of an attenuator in-between the DAC and the ADC. The 'Compare' button can be used to see the difference between the primary scan and the attenuation scan showing the frequency response of the attenuator – see Figure 5.61. It should be noted that that while to about 15MHz the magnitude seams accurate above this it will be affected significantly by the accuracy of the recorded frequency.

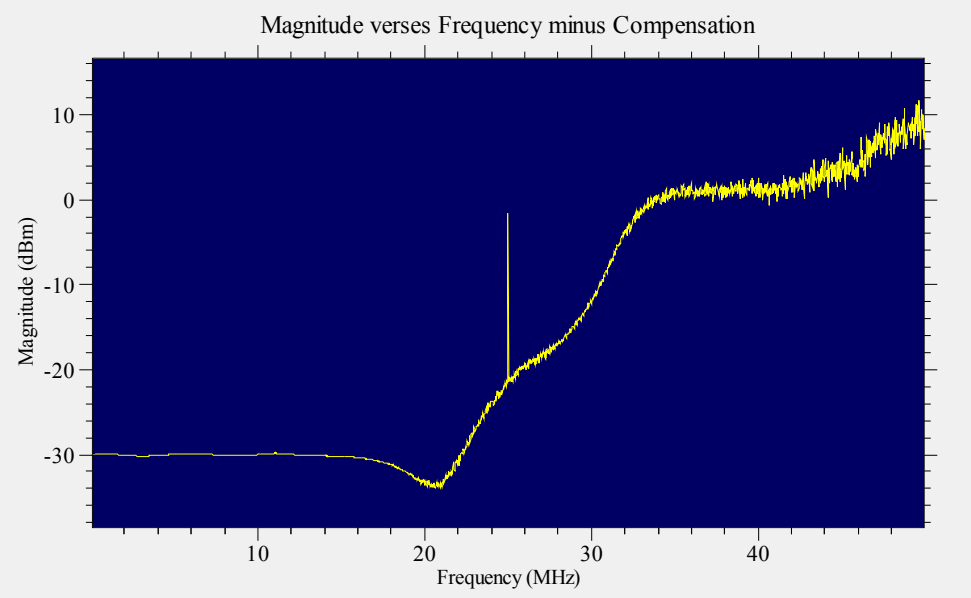


Figure 5.61: Attenuators scan data minus primary scan with a 30dB attenuator.

The Drive calibration tab is used to calibrate the driving output from the VGA output amplifiers – see the GUI in Figure 5.62. The calibration is achieved by feeding the DAC output to an attenuator, then to the amplifiers and then fed back to the ADC. Care must be taken to ensure that the maximum output power does not exceed the ADC maximum input power. This is achieved by selecting an attenuator of the size of the maximum amplification that can be achieved by the amplifiers. Next a new name is created and the attenuation value selected. The frequency scan is then calibrated by running at a known gain value. This is used by the software when an output voltage is required at a specified frequency to scale the actual voltage depending on the frequency attenuation/gain at the frequency specified. The next step is to run a scan over the gain range at a chosen frequency. This is used to calibrate accurately the gain of the VGA. This is performed using an optimisation algorithm.

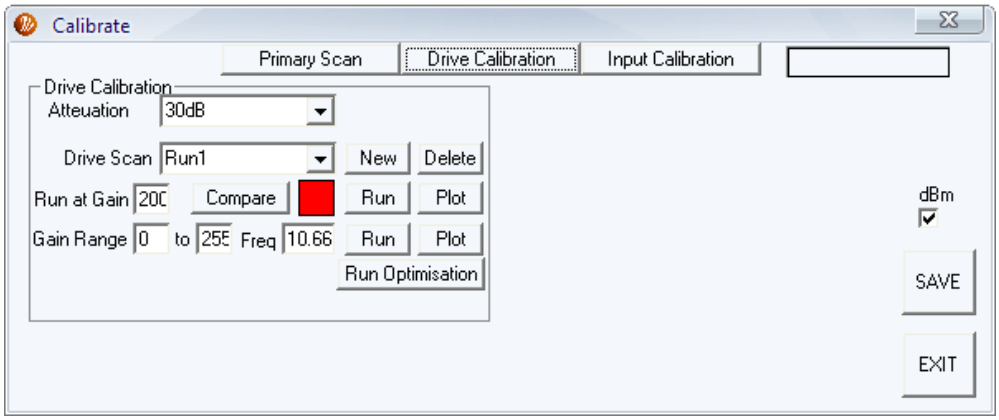


Figure 5.62: Drive Calibration tab of the Calibration GUI.

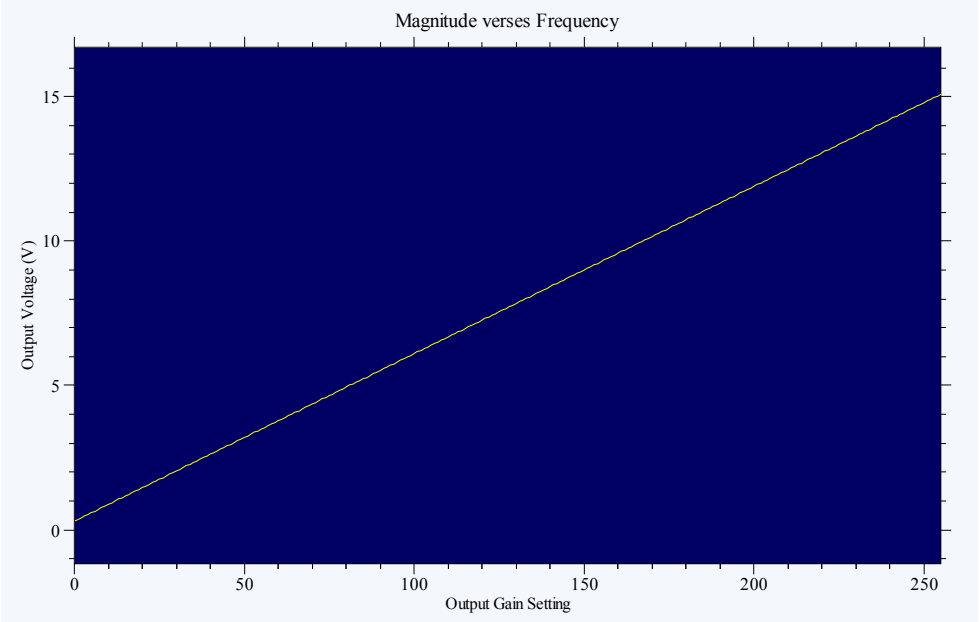


Figure 5.63: Voltage output plot versus gain setting.

The optimisation algorithm is started by pressing the 'Run Optimisation' button. The steps that follow after this are: first the magnitude is measured at each gain value at the frequency specified; next a first order best fit polynomial is fitted to the gain to voltage line; the scale array is adjusted for each gain value using a P loop to attempt to approximate the polynomial fit better. These steps are repeated until the error between the fitted line and the measured line is at the limits of accuracy. This scale array is then used to more accurately specify the voltage and is used to calculate the output voltage. Note that the Bond Rupture DSP function also uses this array to calculate the accurate output frequency. The voltage after the calibration can be viewed using the plot button directly above the run optimisation button, and this displays the calibrated output voltage into 50Ohms as can be seen in Figure 5.63.

The input calibration tab is used to calibrate the input voltage coming to the ADC – see Figure 5.64. This enables the correct scaling value to be used to accurately determine what the voltage was inputted into the amplifier whatever the gain setting. To perform the calibration gain the input to the amplifiers must be sufficiently attenuated to ensure that the amplifiers do not overload on the maximum gain setting. An attenuation of 66dB is used as the gain of

the amplifiers can be up to this. The calibration configuration consists of the DAC connected to the attenuation then to the input amplifier then to the ADC. The maximum gain values are required to be set in Gain2 max and Gain3 max and the test frequency also needs to be set. Once these variables are set and the 'Run 2D' button is pressed the software tests each combination of the gain2 and gain3 values creating a 2D image of the total gain. This is displayed in prospa under 2D plot and can be viewed after it has been run by pressing the 'Plot' button. An example of the 2D plot that is produced can be seen in Figure 5.65. This graph is then used to pick the gain2 and gain3 values to use. The Gain change 1 and 2 box's are used to specify the starting values in which the values will increment starting from gain change 1 to gain max and then starting from gain change 2 to gain max. These gain numbers are then stored in 2D array and a corresponding scale value with each. This data is plotted total gain amount (in dB using both amplifiers) versus the gain number (arbitrary number with is an index to the gain array values for Gain2 and Gain3) – see Figure 5.66. The scale number can then be used to scale input voltage to the real value given a gain amount or can be used to work backwards to calculate the gain values needed for a required amount of gain/attenuation.

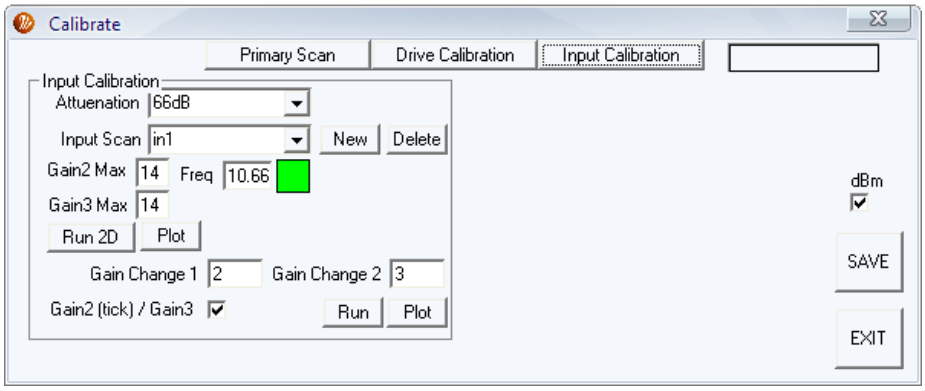


Figure 5.64: Input Calibration tab of the Calibration GUI.

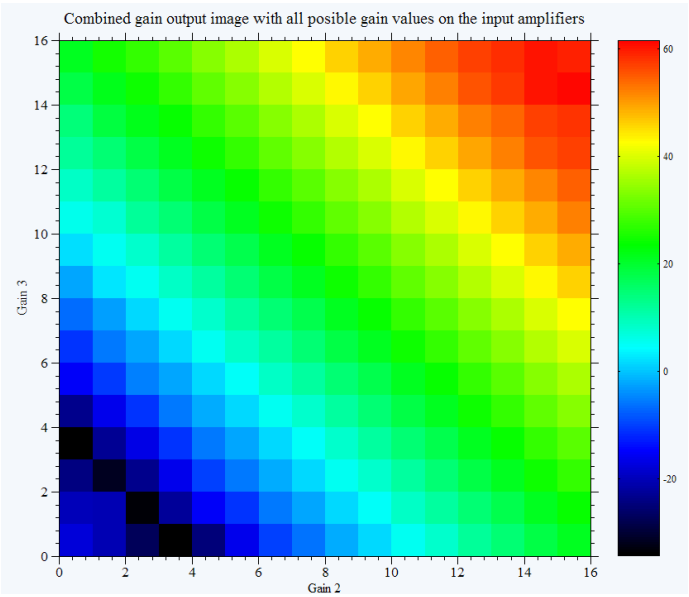


Figure 5.65: The 2D gain plot of gain2 versus gain3 combined magnitude output; (side) is amount of gain in the format of dB.

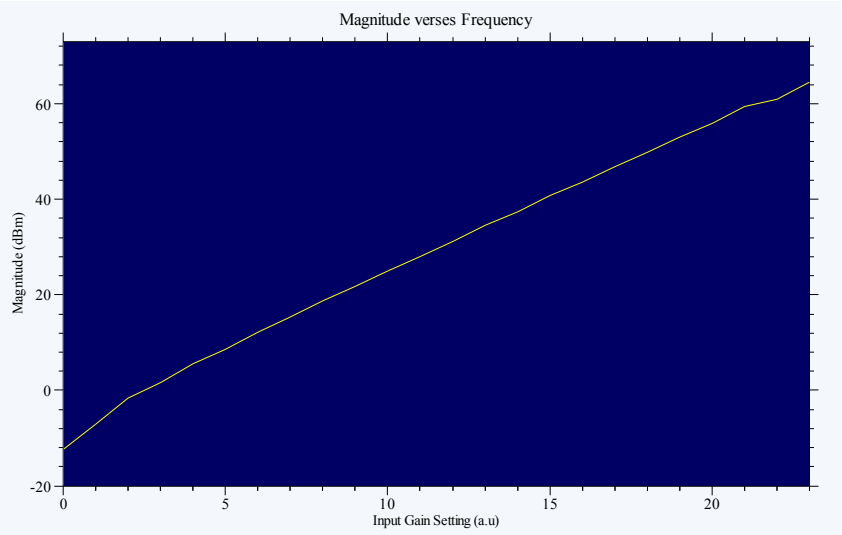


Figure 5.66: Chosen gain settings (which are made up of a combination between gain2 and gain3) and corresponding output gain.

Once a calibration is complete the calibration data is then used in the FFT capture, Frequency scanner, and Bond Rupture to calibrate the required output, and set the input gain. These are set in the gain settings box's (see Figure 5.67) which include; 'Output gain model' – for setting the calibrated gain model data to use; 'Output voltage (Vpp)' –for selecting the output voltage required (calibrated into 50Ohms); 'Input Gain Model' – for setting the calibrated data to use; 'Input Start Gain' – the gain amount that the input amplifiers will be set to.

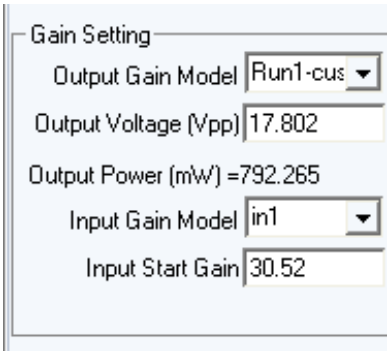


Figure 5.67: FFT capture GUI showing the gain settings.

5.4.4 FFT CAPTURE

The FFT Capture program is used mainly as a diagnostic program. This is used to ensure that the FPGA's ADC and DRP data is being captured correctly – so checking that there are no problems with the FPGA's software operation. Another useful function is the ability to capture the ADC's raw data and by performing a FFT on the data this performs the same job as a spectrum analyser (50MHz Band-width). The Spectrum analyser function (or the FFT of the data) enables the detection of interference on the ADC's input signal. This can enable problems to be identified such as input distortion. This can also be used to analyse the DAC's output spectrum to identify any problems with the signal quality.

The FFT capture flow diagram of the software is as shown in Figure 5.64 with prospa functions in blue and DSP functions in red. As can be seen in the diagram four different capture functions can be selected. The first is capturing the ADC raw data directly into the FPGA's RAM and after it is captured copying it to the DSP's memory. The second option is to capture the DRP1's data directly to the FPGA's RAM after which it is copied from the FPGA to the DSP and converted from 3x16 chunks into 2x24 chunks after which it is stored into DSP memory. The third option is to copy the DRP1's data as it comes in directly from the FPGA to DSP by polling the information register for when new data is available and then reading the two information registers. The fourth method is basically the same as the third except the DRP2's data is used instead of the DPR1's. Once the data has been read back to the DSP's memory the FPGA is put into power saving mode and the data is then copied into prospa. Once the data is in prospa the data is convoluted with a hamming window is to improve the spectral response of the FFT. Next the FFT is performed on the captured data and the data and FFT is displayed.

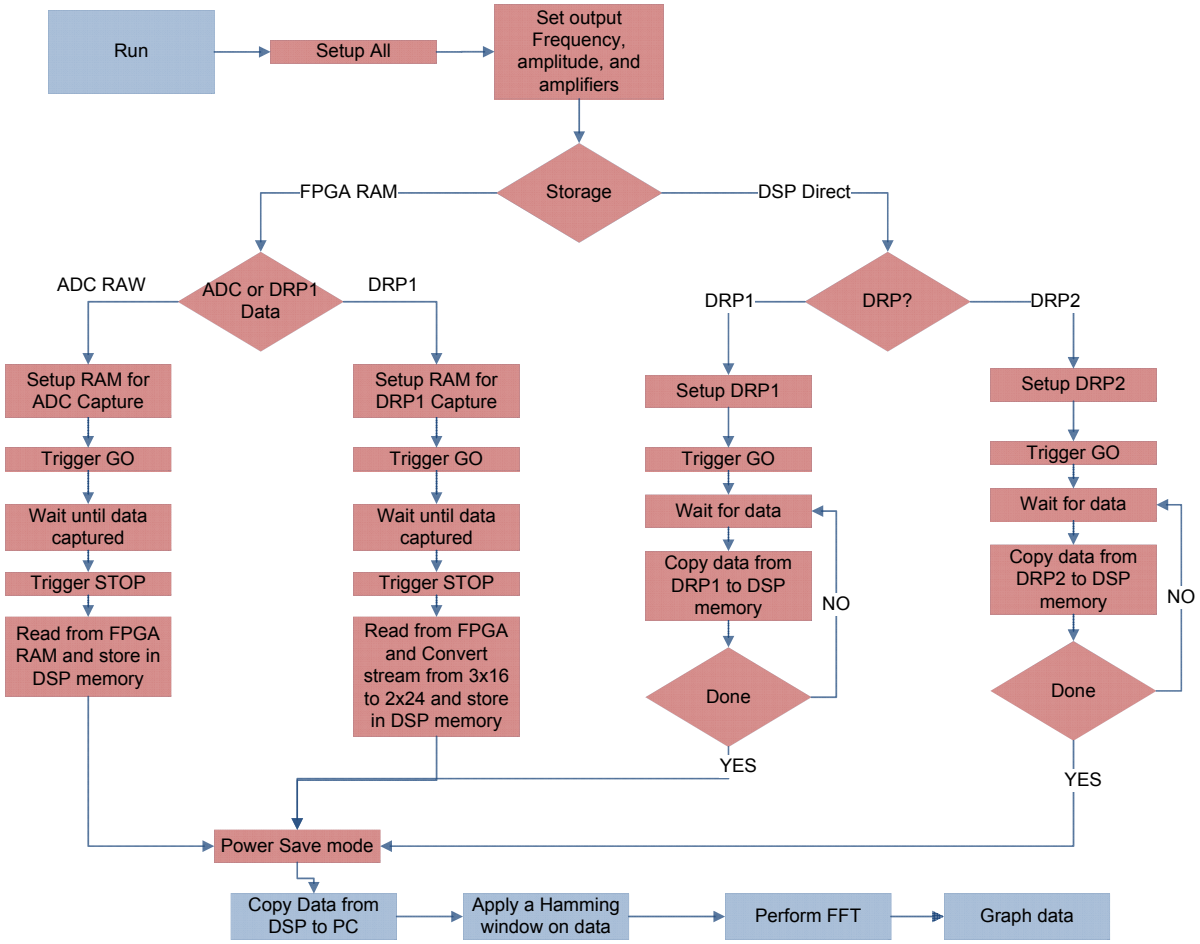


Figure 5.68: FFT Capture flow chart. Blue is for PC processing and red is for processing on the DSP.

The prospa GUI for the FFT Capture program is shown in Figure 5.69. There are a number of different boxes that control the data capture. The first box is called Gain Setting for the using calibration data to ensure that the output and input gains are correct. The Sweep parameters box is for setting the FPGA frequency, DRP, and capture amount settings. The Acquisition box is for setting what data stream to capture and what to use to capture it.

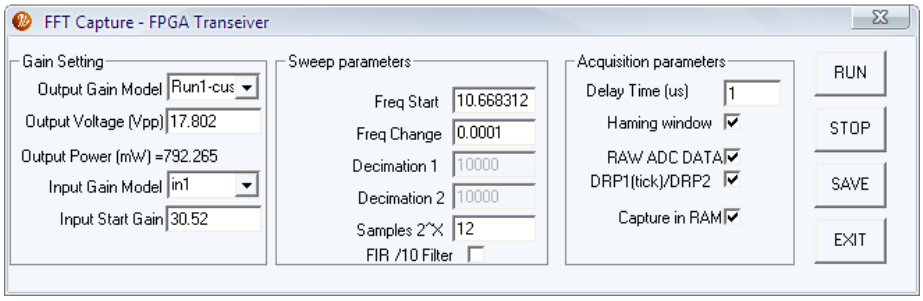


Figure 5.69: FFT Capture prospa GUI – for data capture from the ADC or the DRP's.

The gain setting group box uses the gain model made in the calibration setup to ensure that the correct output amplifier gains and scaling factors are used based on the voltage and frequency required. The Input gain model also uses the calibration setup to ensure that the input start gain is correct though the end power into the ADC must stay at around 0dBm. For example if the input signal from the QCM to ADC was -30dBm a gain of 30dB would be required to restore the input to around 0dBm ensuring that maximum dynamic gain is achieved. Note that if an ADC overflow condition occurs anytime during the capture a message box will indicate this to the user.

The next group box is sweep parameters. The first setting in this group is the frequency start which is the frequency that the DDS will output to the DAC. The second box is the frequency offset from the frequency start value that the DRP will be set to operate at. The next two boxes are the two decimation values for the both DRP's. The second to last box is setting the number of samples to capture (as a power of 2), and whether to capture the raw ADC data. The last is a tick box for whether the ten times decimation FIR filter should be used to further process the data stream from the DRP1.

The last group box is the Acquisition parameters. The first box called delay time is the setting for how long to wait before starting the output running until capturing the data. The second tick box is used to turn the hamming windowing on or off. The RAW ADC DATA tick box is whether to capture the RAW ADC data or the DRP data. The DRP1/DPR2 tick box is for selecting what DRP to use – this is not relevant when RAW ADC DATA tick box is enabled. The capture in RAM tick box is for selecting whether to capture the data to RAM or copy off as it comes in – this must be ticked for the raw ADC capture and cannot be selected when DRP2 is selected.

Once all the correct values are inputted the RUN button can be pressed to start the program running once finished it will return something similar to Figure 5.70.

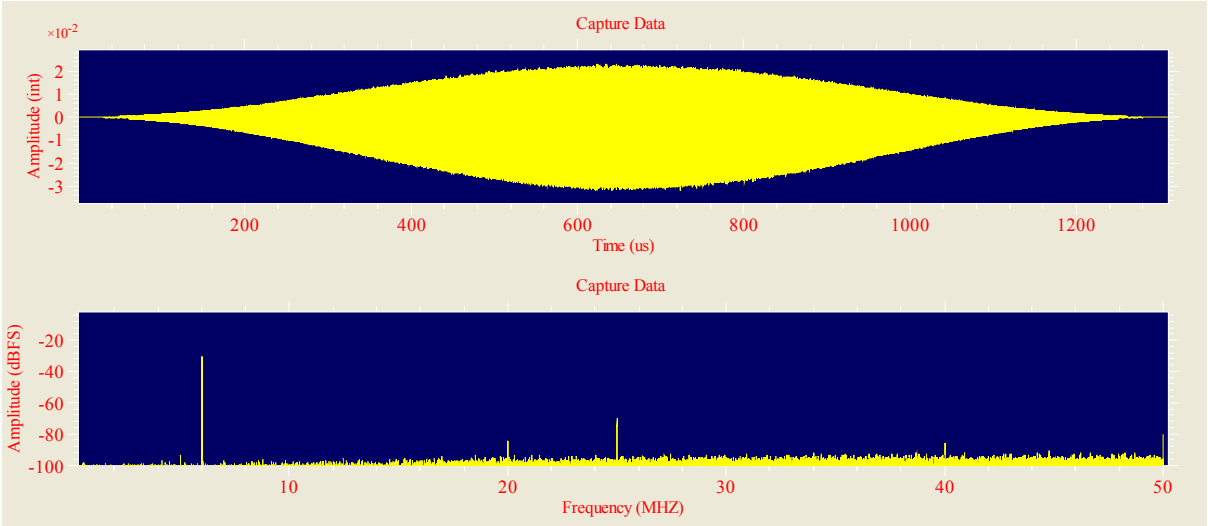


Figure 5.70: FFT Capture plot.

5.4.5 FREQUENCY SCANNER

The frequency scanner program is important as it is the basis for determining the resonant frequency of the QCM. The frequency scanner works by changing the frequency outputted to the QCM while simultaneously measuring the input amplitude and phase at that frequency. This operates similarly to the wobble program in version one with a few key differences including: the simplification of the prospa GUI; the integration of the calibration box; correct scaled graphs using dBm or Volts; increased number of possible capture points; and correct phase output. This program also updates the Frequency centre value once the graph has run by two methods one if it is at a width greater that 2kHz the maximum or minimum value (set in the script program depending on whether it is in current or voltage mode) is used to determine the resonant frequency or else if less than 2kHz width a the maximum or minimum value from a polynomial fit is used to improve the accuracy.

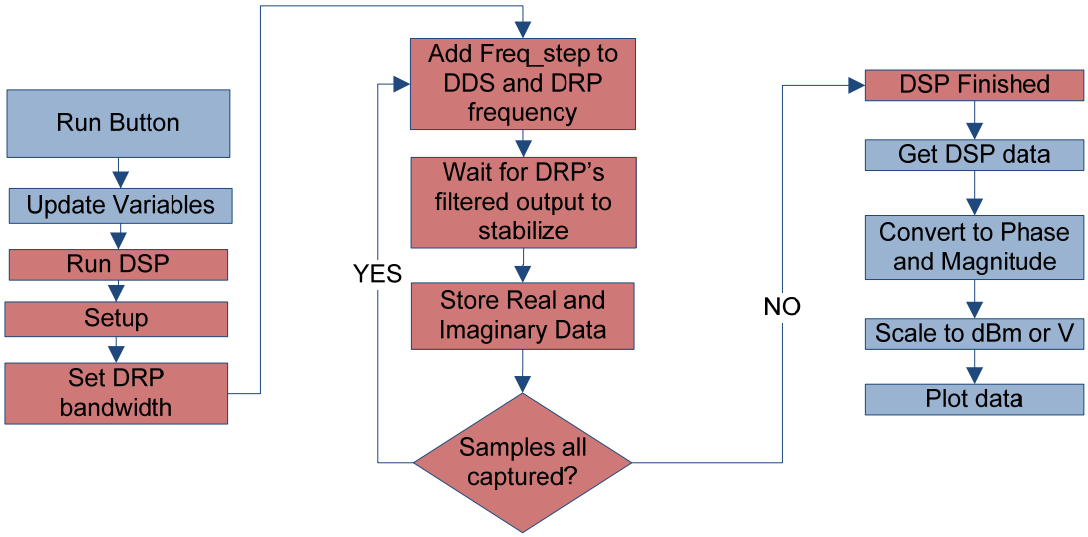


Figure 5.71: Frequency scanner software flow chart. Blue is for PC processing and red is for processing on the DSP.

The Frequency scanner operation is relatively easy as most of the digital processing is done automatically within FPGA using the DRP, all that the DSP needs to do is to setup the FPGA transceiver and then capture the data. The basic software flow diagram is shown in Figure 5.71. This starts with the run button being pressed in the GUI after which the variables that need to be passed to the DSP are calculated and sent. Then the DSP is set to run the frequency scanner assembly file. This starts by setting up the FPGA transceiver with: output frequency; band-width; number of samples; and frequency step – which is what is added after each new sample to the frequency. Next a loop around a series of steps is carried out until all the samples required are all gathered. These steps are first to add the step frequency to the current DDS and DRP frequency, secondly wait for the DRP data to stabilize by ignoring the first x number of samples, and thirdly to store the quadrature to the DSP's memory. Once all the samples required are gathered, prospa copies the data from the DSP and converts it to Magnitude and phase data.

The GUI is made up of two boxes for settings Gain setting and frequency scan variables as well as some buttons to control the starting and stopping of the DSP – see Figure 5.72. The Gain setting box is identical to the gain setting box in the FFT Capture program. It is worth noting that the output voltage setting is based on the calibrated output into 50 Ohms so this will change depending on the load impedance. The Frequency scan variables are used for setting the frequency centre, frequency span range, number of points, and the band-with of the scan. The Frequency centre variable is the centre frequency point of the scan with half of frequency span value on either side. The number of points to capture is the number of points to capture over this frequency range. The band-width is after the DRP has processed the ADC data at the frequency point specified. To get a band-width of 10 kHz from the 100MHz ADC a decimation rate of 10000 within the DRP is needed. There is also the 'width run' button which when pressed enables the user to select a start and end value from the graph and from this the prospa scripts calculates the new frequency centre and span values based on the selected width. After this the DSP program is run over this width. The 'centre run' button when pressed runs a macro that enables the user to select a centre point on the plot which updates the frequency centre variable and then runs the DSP program with this setting. Another setting is the dBm check box. This is used to scale the graph in dBm if checked or volts if not. The Run button is used to run the prospa GUI. The SAVE GUI button is used to save any GUI variable setting as default for the next run.

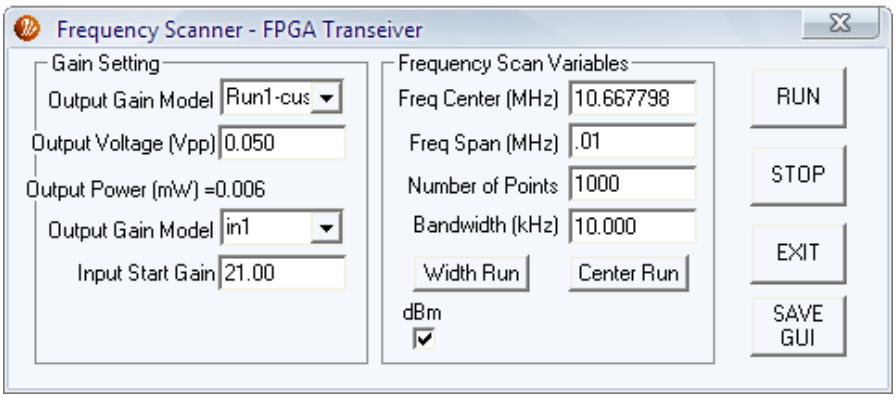


Figure 5.72: Frequency Scanner prospa GUI.

Once the run button is press and the DSP has returned the quadrature scan data further processing is needed to convert it to magnitude and phase outputs. The magnitude is calculated by taking the square root of the R (real data) value squared and added to the I (imaginary data) value squared. The Magnitude is a measure of the total RF voltage contained within that frequency band-width so any other frequency spurs within this bandwidth can affect the magnitude value. The Magnitude data is further scaled to be in units of either dBm or Volts and is scaled depending of the decimation and scale rate used (see functions scalier). The phase data is calculated by the inverse tan (function atan2) of R (real data) over I (imaginary data). The phase must then be run though an algorithm that un-wraps the phase data from the limit of –180 to 180. This operates by looking at two sequential phase values and if previous value is greater than 90 and changes to less than -90 a value of 360 is then added to rest of the phase array (or in the opposite case a value of 360 would be subtracted). Once the magnitude and phase data calculated it is correlated with the correct frequency data for plotting.

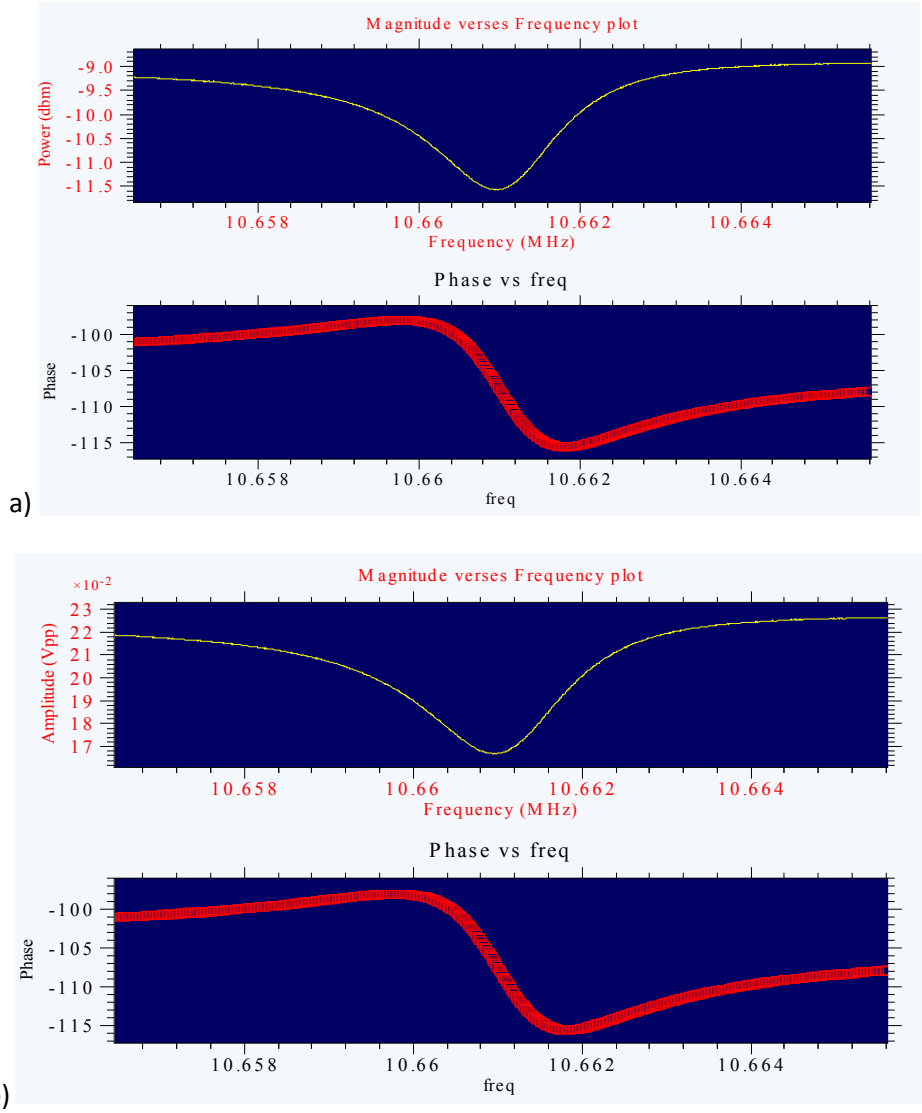


Figure 5.73: Frequency scanner plots of a QCM using voltage measurement method. a) Scaled to dBm. b) Scaled to Volts peak to peak.

The frequency scanner software's primary use is to measure the QCM's resonant frequency. This possible as the impedance of the QCM changes significantly at resonance and so changes the voltage and current measured for a constant power input. If the voltage is being examined the voltage will decrease at resonance where as if the current is being examined the current will increase at resonance. To find the true resonant frequency with the voltage measurement, the minimum point of the graph is used or the maximum point for current measurement. Using a polynomial fitting method the frequency can be determined within sub hertz resolution. The plots generated from running of the frequency scanner software with a QCM connected using the voltage measurement configuration is shown in Figure 5.73.

5.4.6 BOND RUPTURE

The Bond Rupture (BR) software is the essential part of the system, for which everything else was created. Experiments have shown that the bonds, rupturing off the surface can be detected by either the frequency change as the bonds break, or via measuring the signal magnitude at around the third harmonic. The BR software operates by increasing the RF power output into the QCM, while simultaneously tracking the resonant frequency of the QCM. While this is occurring, the noise data is also being captured at the 3rd harmonic. The resonant frequency is measured primarily using the magnitude method but, the software also incorporates the zero crossing and quality measuring method, so as to get information about the loading of the crystal, and providing another method of measuring the resonant frequency. Using this software Bond Rupture can be measured at driving voltages as starting as low as 10mVpp to as high as 20Vpp with QCM frequencies supported from 500 kHz up to 50MHz – note: to support the full frequency range some filters may need to modified.

This software has significant improvements over the previous version developed in chapter 4. One of the most substantial improvements is the combination of both the Magnitude frequency tracking with the zero crossing method. This is combined into the same DSP software enabling the advantages of both to be brought together. Another major improvement is the ability for the FPGA to capture data from the DRP without the DSP needing to process it - until it needs to be downloaded to prospa for plotting. This frees up the DSP to perform the involved frequency tracking and decreases the complexity of the DSP program. Other improved features include: accurate calibration of the output voltage/power setting; active gain changes implemented by the DSP; the breaking up of runs into separate sub runs which the data gathered up to such point can be plotted in prospa plot window; detection of frequency tracking errors; improved GUI for separating the run, plotting, and advanced variables to simplify the operation; and finally advanced post processing methods for frequency comparison and spectral analysis.

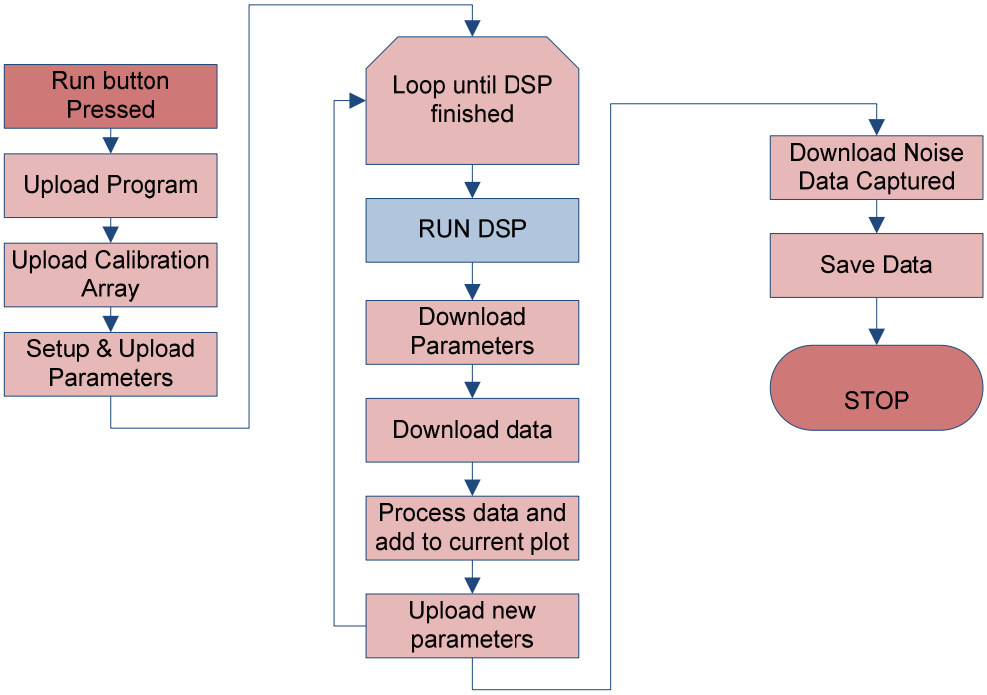


Figure 5.74: prospa script flow diagram once the run button has been pressed.

There are a number of components that make up the software for performing BR. Among them is the prospa GUI and a number macros. The process which is put into place when the run button is pressed can be seen in Figure 5.74. This entails the setting up of variables that need to be passed to the DSP then running a loop with the DSP returning when the time specified is complete. The loop period is specified by the 'refresh time' variable in the GUI window and is controlled by the DSP stopping after this time and returning any data gathered thus far. This data is then plotted on the prospa plot screen (format of the plot depends on which plot mode is selected) and once this is done the loop updates the DSP parameters and continues to run the DSP. While in the refresh period the DSP outputs a fixed frequency and amplitude sinusoidal waveform. Once the total time is up the DSP returns a stop value and the loop breaks. The data captured by the FPGA (usually around the third harmonic) is then copied from the FPGA to DSP to prospa for further processing and plotting.

The DSP software uses a millisecond internal timer to keep track of when for example the frequency is measured and also when to pause or stop the program and return to prospa. The DSP program will run until the time specified by the 'refresh time' (GUI variable in Figure 5.95). Once this is reached the program will pause while continuing to drive the QCM and return any data captured to prospa. When the time specified by the 'experiment time' (GUI variable in Figure 5.75) is reached in the DSP the program turns the output off and returns any data gathered so far to prospa. This enables data to be plotted as it is gathered and provides feedback for the user to the experiment progress while also enabling the acquisition of more data points. However, there may be times when it is preferable not to have any breaks in the experiment (which can be up to 500ms at times depending on the amount of data) and so the refresh time can just be set to greater or equal to the experiment time to disable it.

The DSP programs flow is shown in Figure 5.76. This starts by selecting the mode which it is to run out of start-up, loop, or the data read back. The start-up mode is the same as loop except

it sets up the registers within the FPGA and the DSP needed for operation. The loop mode uses the start-up variables from the previous run and continues from where the program left off. The read back mode is to copy the stored data from the FPGA onto the DSP so that prospa can copy the data off. The program continues on in the start-up or loop mode to choose whether or not to perform a zero crossing at the start or at the start of each run. The next choice is what settings to load into the second DRP for data capture including, the bandwidth and frequency specified. Once this is started the processed data is loaded directly into the FPGA's RAM to be processed later – if in the loop mode there will be no effect to the capture settings. After this the output and input gain settings are updated and a loop is started to run the magnitude algorithm – with the gain setting updated every cycle. When the internal time reaches either the 'refresh time' or the 'experiment time', the program will break from the loop and will either keep a constant output frequency and voltage, or turn the output off. If the program is in the stopped mode the output turned off.

The software consists of a number of functions including: time based voltage ramping and gain calculation; zero crossing algorithm for quality and frequency measurement; magnitude curve fitting algorithm for frequency measurement; and a number of smaller less significant functions. The first three functions shall be covered in the sections to come.

The three main functions all output to a stream in memory within the DSP which is copied to prospa once the DSP program has finished its run. This data stream is processed in prospa to separate out all the different types of data. Though this may not be the most efficient method to save the data, it does have the advantage of prospa not needing to know what order the data is to come in. This is especially relevant as multiple different types of data is been sent to prospa in varying orders.

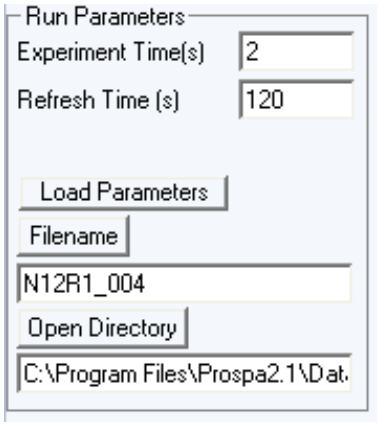


Figure 5.75: Bond Rupture GUI Run tab run parameters.

The controlling GUI for the Bond Rupture Software is broken up into three sub tabs to simplify the operation and enable more options to be present such as graphing. The first tab is called 'Run' – see Figure 5.77. This is used to specify all the simply run parameters such as experiment run time, operating frequency, and output voltage. The next tab is called 'Graphing' – see Figure 5.78. This enables the graphing parameters including opening of any passed data set and the data that goes with it. It also has functions for comparing the frequency change between runs for the Bond Rupture detection. The last tab is called 'Advanced' – see Figure 5.79. This includes advanced parameters that the standard user would

not have to deal with. This includes advanced settings for the ZCQ algorithm and the MLMS algorithm. The different GUI tabs shall be described in detail in the sections they are related to.

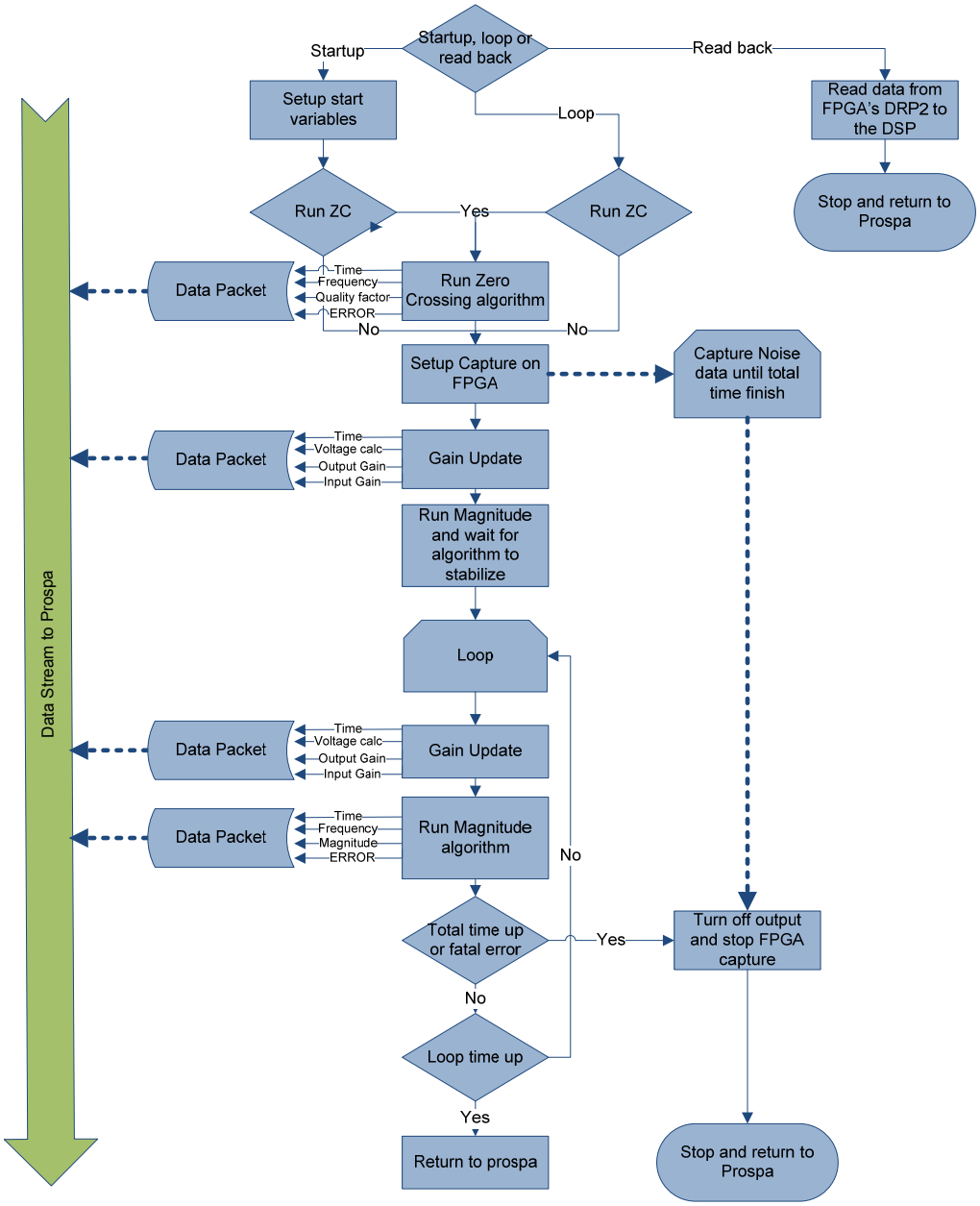


Figure 5.76: Simplified diagram of the DSP algorithm to perform Bond Rupture experiment.

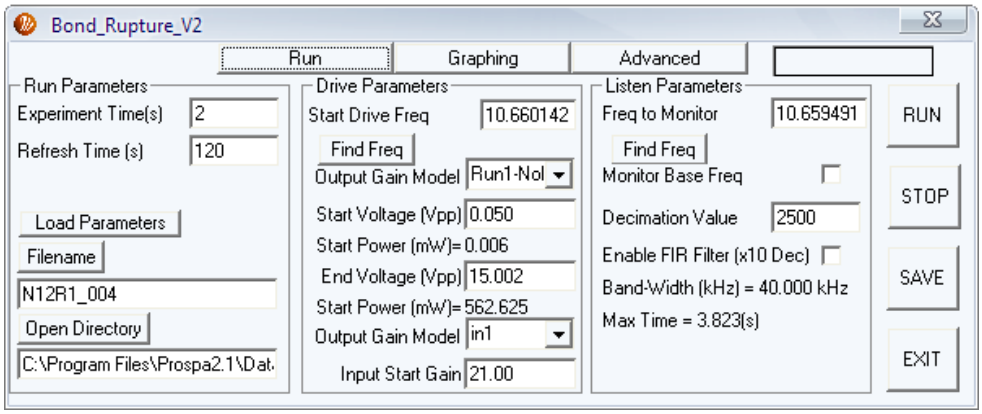


Figure 5.77: Bond Rupture Run tab – for setting the simple run parameters.

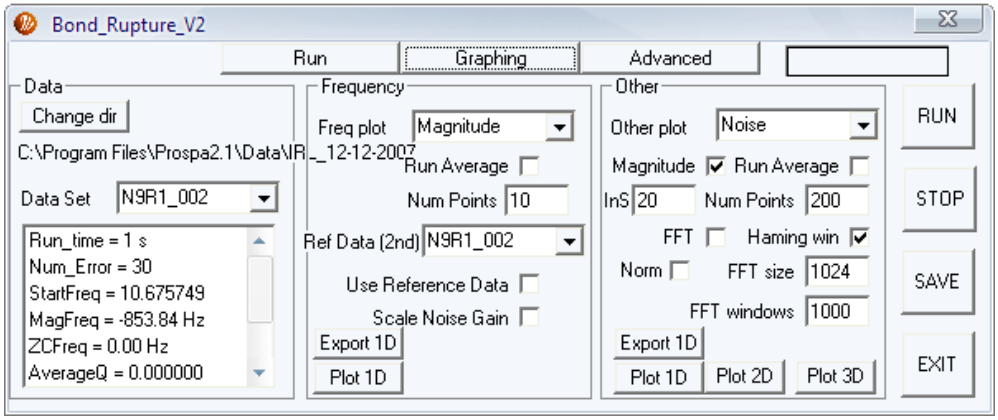


Figure 5.78: Bond Rupture graphing tab – for setting the graphing parameters.

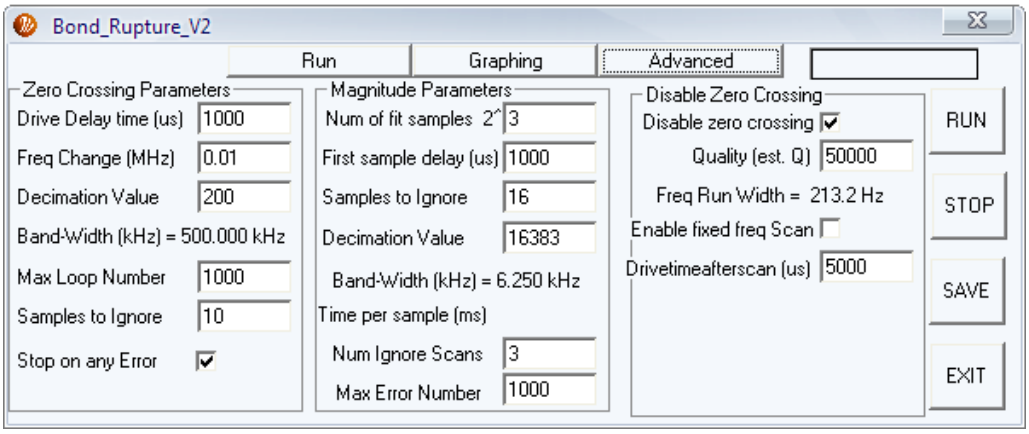


Figure 5.79: Bond Rupture Advanced tab – for setting the advanced run parameters.

5.4.6.1 MAGNITUDE LEAST MEAN SQUARES

The method for the previous system used three recorded magnitude points to find the minimum or maximum point on the magnitude curve. This method required several runs before the correct frequency value was found and also had difficulty running when the points where too close together or the signal was too noisy. To enable better tracking a new method was proposed. This method uses a simplified Least Mean Squares (LMS) algorithm to find the equation of the line and will be referred to as the Magnitude LMS method or MLMS for short.

The MLMS method relies on the curve region that being examined to be close to a 2nd order polynomial. If this condition holds by differentiating so that it is a straight line (1st order polynomial) with the zero crossing point being the minimum or maximum point of the curve. Now using the straight line's equation the point at which the zero crossing has occurred can be simply calculated and the exact point at which the minimum or maximum occurs is known. Of course this must be converted so that is can be performed with noisy data and a non ideal curve which is what the MLMS algorithm does.

The MLMS algorithm starts by dividing the frequency range into a number of equally spaced values over which the magnitude is sampled at each point. This data is then differentiated numerically and processed using a LMS algorithm to determine the best line fit for the data presented. The LMS algorithm is broken up into 3 parts to simplify the processing: firstly the initial setup, secondly the running calculation after each new data point, and thirdly the calculation of the line equation. The first initial setup includes the calculation of the step width and the initial X value set to minus half the width – assuming the middle frequency value is zero. It also needs to capture one magnitude point at the start so as to have a starting point for to enable the derivative to be calculated. The second running calculation operates after each magnitude capture within the loop. The first calculation is the numerical derivative of the current magnitude value minus the last magnitude value. After this the X value must be updated so that it represents the new derivative point. Next the sum of X multiplied by Y, sum of X multiplied by X, and sum of Y needs to be calculated – Y is the magnitude. This calculation can be seen in Equation (5.14).

$$XY_{sum} = \sum_i^{steps} X_i Y_i\,, \qquad XX_{sum} = \sum_i^{steps} X_i^2\,, \qquad Y_{sum} = \sum_i^{steps} Y_i \tag{5.14}$$

Next the next step frequency is calculated and the loop continues around until the number of steps specified is reached. Once the loop is complete the line equation values b_1 and b_0 can be calculated using the sum values and the number of samples. The b_1 value is calculated by dividing the sum of X^2 by the sum of XY. The b_0 is calculated by taking the sum of Y divided by the number of steps. See Equation (5.15).

$$b_1 = \frac{XY_{sum}}{XX_{sum}}, \qquad b_0 = \frac{Y_{sum}}{steps}\,, \qquad y = b_0 + b_1.x \tag{5.15}$$

The mathematical derivation of the LMS algorithm is described by (Weisstein 2007) from the Math World website. Once the line equation is calculated the equation can be solved for y is equal to zero to calculate where on the x scale the maximum or minimum point occurs on the polynomial curve. See Equation (5.16).

$$x_{new} = -\frac{b_0}{b_1} \tag{5.16}$$

This x_{new} value can then be used to work out the new centre frequency by adding this value to the previous centre value. But before this value is updated first the centre point is checked to make sure it occurs within the frequency scanned over and if outside it, the new frequency is set to the either the minimum width value if before or the maximum value if after. An error is also sent to the data stream to inform the user that this has occurred as the value that will be outputted will not be a correct frequency value and if too many of these errors occur the

program will be stopped. This range checking code is required to ensure that the tracking does not become too unstable and prevents one off read errors causing instability. A figure of main components of the algorithm can be seen in Figure 5.80.

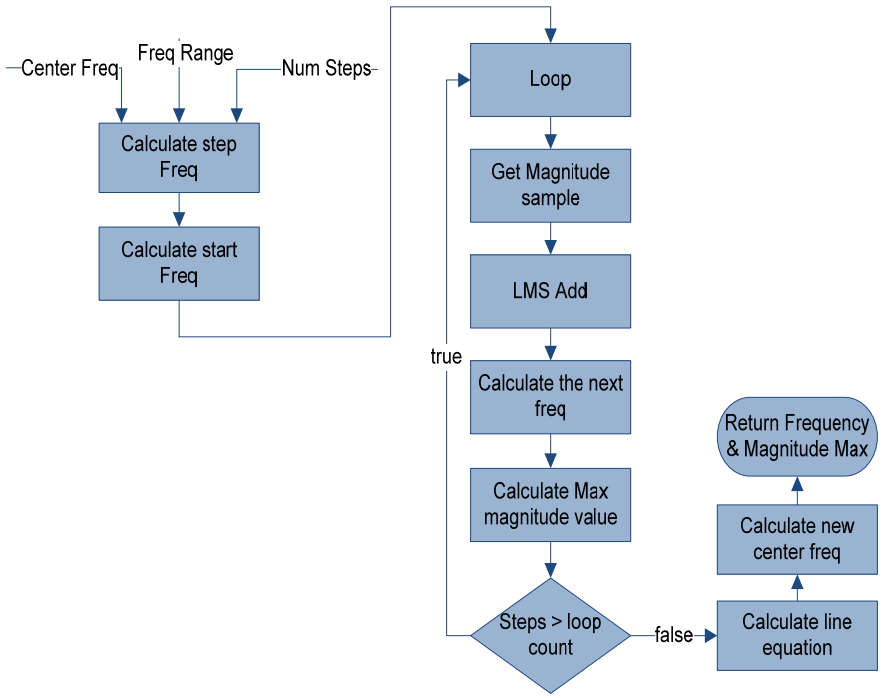


Figure 5.80: Software flow for MLMS algorithm.

The MLMS method works well in low signal level and noisy environments. It is also very fast with one frequency sample able to be taken every 5ms – this value is dependent on what the input bandwidth is set to as well as the number points used in the LMS algorithm. There are situations which can cause problems with this algorithm. One of potential problems is that tracking will not occur if the starting frequency is not within a few kHz of resonant frequency. This can be addressed by using the resonant frequency finder macro before running the GUI. It can also be minimised by using the zero crossing method which has less strict start frequency restrictions. Another problem is when the frequency range is not set within a part of the curve that can be approximated to a 2^{nd} order polynomial problems may occur, with a small amount of offset from the actual resonant frequency appearing. This is addressed by using the quality value from the algorithm described in the next chapter to calculate a frequency width that will contain only a portion that approximates a 2^{nd} order polynomial.

There are also a number of parameters required to be configured for the operation of the magnitude algorithm – see Figure 5.81. The first parameter is the 'Num of fit samples' which is the number steps to divide the frequency range into and so also the number of parameters that will be fed to the LMS algorithm. The number of samples is specified using the 2 to the power of numbers – powers of two are used to simplify and improve the speed of the MLMS algorithm. This value is usually set to around 3 (16 samples) and will double the amount of time spent running the algorithm each time though increase the accuracy significantly. The next parameter is the 'First sample delay'. This is used to specify the amount of time to wait after starting driving the QCM. This helps stabilise the measured magnitude data, as the QCM will take some time to respond to the input – one millisecond is a standard amount. The

'samples to Ignore' box is not used in the current version so can be ignored. The 'Decimation Value' box is used to specify the amount of down sampling that the DRP performs on the input data. The Band-width that is generated from the decimation value is shown in the text below it. The 'num Ignore Scans' box is used to specify the number of magnitude scans to ignore on the first running of the magnitude algorithm. This is because if the frequency is significantly different from the specified start frequency, it can take a number of scans to track to the correct resonant frequency. The value to be set here depends on whether the ZCQ algorithm is running and the frequency measurement accuracy. The 'Max Error number' is the number of errors to ignore before exiting the DSP program as a fatal error. A Magnitude error is if the resonant frequency calculated falls outside the frequency width area specified. If such an error occurs it is added to the error count and if it exceeds the max error amount the DSP program finishes.

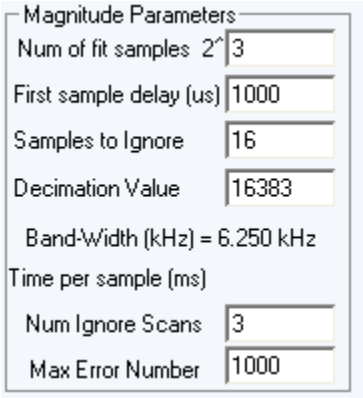


Figure 5.81: Rupture Advanced GUI parameters for Magnitude algorithm.

The 'Disable zero crossing' tick box as seen in Figure 5.82 is used in situations where it is not possible to use the zero crossing algorithm. An example would be when the user wants to start the driving voltage below what is recommended to acquire a ZCQ measurement – not recommended to operate below 50mV starting voltage depending on the accuracy of the start frequency and the quality of the QCM. Another example would be when only the Magnitude algorithm is data is required. If ZCQ algorithm is disabled certain parameters must be passed to the Magnitude algorithm. The first one is the 'quality' of the QCM. This is required to estimate the frequency width that the magnitude algorithm should sample over.

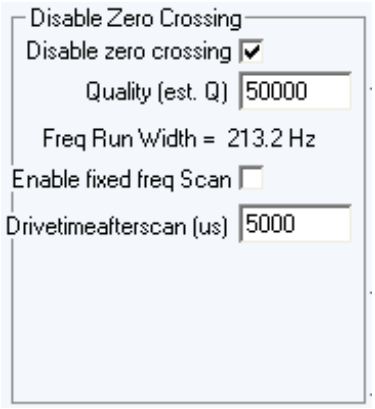


Figure 5.82: Bond Rupture Advanced GUI parameters for the ZCQ disable function.

The ‘disable zero crossing’ box also has a tick box for enabling a fixed frequency scan. This is when the frequency specified is outputted and the voltage ramps at that. This is useful if no frequency tracking is required only the Bond rupture driving and frequency noise listening. The ‘Drive time after scan’ is the time to drive the QCM for after the magnitude scan has occurred.

One parameter that has been left out is the ‘Start Drive frequency’ as seen in Figure 5.83. This is the frequency used to either as ZCQ algorithm start point or if that is disabled it will be passed as the centre frequency to the MLMS algorithm. The ‘Find Freq’ button is used to perform a quick frequency sweep (using the Frequency Scanner software) around the specified frequency to find a more accurate start point. It is recommended that this is run with each new QCM used or after the frequency has changed significantly.

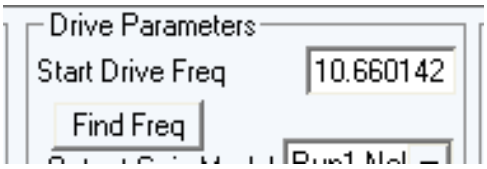


Figure 5.83: Frequency settings GUI box.

5.4.6.2 ZERO CROSSING & QUALITY

The basis for the ZCQ (zero crossing and quality) algorithm has been described in section “Bond Rupture Experiment using Zero Crossings” including how the frequency is measured via counting the zero crossing and quality is measured by measuring the time constant. Though the ZC (zero crossing) measurement worked well in the previous algorithm there were problems with the quality measurement that were evident once experimentation was carried out. One of these was that if there was not enough sample time to reach the time constant value the value could not be determined. The other major problem was that the accuracy could only be determined using two points either side of the time constant point. So majority of the data was not used and accuracy was lost. To improve on the time constant measurement in the new ZCQ algorithm a modified LMS algorithm was used to fit a exponential line to determine the time constant resulting in a much improved measurement result.

The ZCQ algorithm operates by first driving the QCM for a set amount of time at a user specified frequency - which is performed to excite the QCM. Next when the output is turned off the QCM will oscillate at the resonant frequency in an exponentially decaying fashion. This is measured using the first DRP which offsets the frequency by a set amount so that it is possible to count the zero crossings that occur. The DRP also provides a quadrature output which can be used to calculate the magnitude, as can be seen in Figure 5.84. The ZC algorithm uses both the quadrature data channels and the magnitude algorithm uses the calculated magnitude. Each time a sample data point is gathered the zero crossing and the quality LMS algorithm is run. Once a user specified number of samples are captured the frequency and quality is calculated and outputted to the data stream that will be returned to prospa.

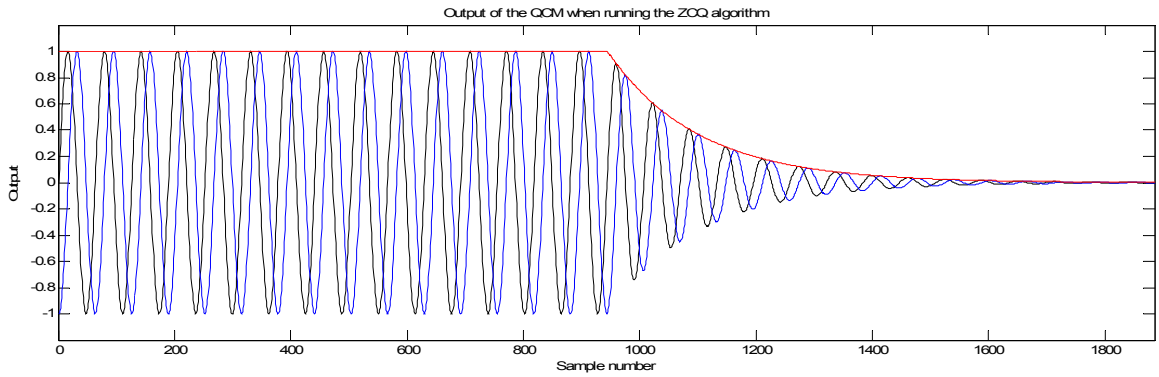


Figure 5.84: Simulated example of the QCM's signal recorded by the DRP and running the ZCQ algorithm. Black - real quadrature data. Blue - imaginary quadrature data. Red - magnitude calculated from real and imaginary data.

For purposes of simplification the ZC and quality measurement algorithm will be described independently. The ZC algorithm operates by measuring the first time one of the quadrature signals cross zero, then counting the number of times, each of signals cross until the last crossing occurs enabling the total time and number of zero crossings to be found. This can then be used to find the average period between ZC's. The QCM's resonant frequency with respect to the frequency offset of the DRP can then be calculated. The accuracy of the first and last zero crossing time measurements is improved by using the sample before and after the zero crossing, to gather via linear interpolation an more accurate total time value. To prevent false crossings (which can occur when there is a lot of noise on the signal measured) the zero crossing will only be counted if they occur in the correct channel sequence. This works by measuring the first zero crossing and noting what quadrature channel this occurs on. Once this is known this channel is ignored until a zero crossing occurs on the other channel and vice versa until the end of capture. This improves the operation of this algorithm significantly in noisy environments. To further protect this algorithm in low signal situations such as QCM's that have a very low quality factor there is a check to make sure the signal is above a certain level.

A series of calculations are needed for the ZC algorithm. The first is the total time as seen in Equation (5.17).

$$t_{total} = t_{end} - t_{start} \tag{5.17}$$

Where t_{start} is the first zero crossing point measured and t_{end} is the last point, both measured using the linear interpolation method. To calculate the average time Equation (5.18) is required.

$$t_{av} = \frac{4.t_{total}}{(n_{count}-1)} \tag{5.18}$$

Where n_{count} is the number of zero crossings measured on both quadrature channels. This is then used to calculate the average frequency – as shown in Equation (5.19).

$$f_{av} = \frac{1}{10e^{-9}.Dec.t_{av}} \tag{5.19}$$

Where Dec is the decimation amount the DRP is set to perform and $10e^{-9}$ is the sample time of the ADC. The last Equation (5.20) is used to calculate the actual frequency the QCM was operating at.

$$f_{qcm} = f_{DRP1} + f_{av} \tag{5.20}$$

Where f_{DRP1} is the offset frequency of the DRP and f_{qcm} is the frequency returned to prospa as the QCM's resonant frequency. This frequency is also passed to the magnitude algorithm so that it has an accurate start frequency.

The quality algorithm operates by taking the natural log of the magnitude data, so that the decay (which is an exponential curve) is converted to a straight line. Once a straight line exists the LMS algorithm described in the previous section can be used to find the formula for this line and from that determine the time constant of the exponential curve. To normalise the line so that it starts at zero, the first value is saved as a reference point. To calculate the points for the LMS algorithm the following normalisation Equation (5.21) is used.

$$Mnorm_i = ln(M_0) - ln(M_i) \tag{5.21}$$

Where the first measured magnitude value is M_0, and is the current magnitude value is M_i, and function $ln(x)$ is the natural log of x. The natural log was implemented using a 6th order polynomial approximation function – of which the optimisation parameters were designed in Matlab. This natural log algorithm is accurate 16 bit output and can accept a 48bit input value. This algorithm had to be fast so not to hold up the other processing needed and so it takes 20 clock cycles. The value $Mnorm_i$ is what is passed to the LMS fitting algorithm. This formula is run after new magnitude sample and after this the LMS algorithm is run with $Mnorm_i$ passed as Y and sample time passed as X. The following sums in Equation (5.22) are added each time a new magnitude sample is presented.

$$XY_{sum} = \sum_i^{steps} X_i Y_i\,, \qquad XX_{sum} = \sum_i^{steps} X_i^2\,, \qquad Y_{sum} = \sum_i^{steps} Y_i \tag{5.22}$$

Once this has been run though to the last captured sample the line equation values can be calculated (b_1 and b_0). This is done using the following Equation (5.23).

$$b_1 = \frac{XY_{sum}}{XX_{sum}}, \qquad b_0 = \frac{Y_{sum}}{steps}\,, \qquad Y = b_0 + b_1.X \tag{5.23}$$

The time constant (τ) can be calculated by making Y equal to negative one and then finding the value of X. This because the log of the exponential is taken negative one is equal to the time constant. Then the to calculate τ from Equation (5.24).

$$\tau = -\frac{(b_0+1)}{b_1} \tag{5.24}$$

The time constant is the value that is returned to the PC as a data point and can later be used to calculate the real quality value. The width Δf is needed to determine the width the frequency to run the magnitude function over. This is calculated in Equation (5.25).

$$\Delta f = \frac{f_0}{Q} = \frac{f_0}{\pi.f_0.\tau} = \frac{1}{\pi.\tau(s)}(Hz) \tag{5.25}$$

This value is multiplied by a width factor scaling variable to make the frequency width at a ratio suitable to be used by the magnitude algorithm.

The quality measurement algorithm enables the measurement of the time constant without needing to reach that point in time. This is because of the LMS calculating the line equation directly so the time constant could be calculated with only the first few samples. The algorithm also has the advantage of using all the points gathered versus the previous algorithm that only uses three points and so the more points gathered the greater the accuracy of the time constant measurement. One problem with this algorithm is that as the signal decreases so does the accuracy and this may affect the fitting of the line. Because of this, a minimum signal value comparator is used to ensure that the signal is not used if it falls below a minimum value.

The ZC and quality algorithm are both combined in assembler code to form the ZCQ algorithm. This is a huge improvement over the previous version of this algorithm but does have some limitations in comparison to the magnitude method. The magnitude method however cannot provide the quality factor capture. This is why a combined method was used. The other advantage of the ZCQ method is that is can usually find the resonant frequency from an input starting up to 20 kHz (depending on the settings) away from the actual resonant frequency.

There are a number of parameters that are required to be setup in the Bond Rupture GUI's advanced options under zero crossing parameters – see Figure 5.79. These parameters should not need to be touched by the general user but maybe needed if a different type QCM was used or for advanced optimisation. The first item is the 'drive delay time'. This is for specifying the amount of time needed to drive at the specified frequency before turning off the output and listening. This can be increased if the QCM is not being excited enough to get adequate signal output. The next parameter is the 'Freq Change'. This is used to set the offset on the DRP from the driving frequency so that the zero crossings can be counted. This provides a trade off between lots of zero crossings and time accuracy. This value the frequency is normally set to 10 kHz. The 'Decimation value' is the next variable. This is used to set the sampling bandwidth, which is shown in the 'Band-width' text below it. The 'decimation value' is the amount that the input sample rate is decimated by. This can be an integer number from 200 to 16384 which is a bandwidth from 500 kHz to 6.1 kHz. The bandwidth must be chosen carefully as it is a trade off between wide and narrow. With a wide bandwidth time there are a lot more samples to use but, with less signal processing gain from the DRP and more noise from other sources. On the other hand with a narrow bandwidth there is less noise from other sources and more processing gain from the DRP but, there is less samples to use and if the QCM signal dissipates to fast the signal may not be able to seen. The QCM's time constant for dissipation is usually around 0.1 to 1 milliseconds and so values must be chosen so most of the sample is within this time while also enabling the maximum number of samples to maximise the time constant measurement. The 'maximum loop number' is the number of samples to run the ZCQ algorithm over – though the measurement will stop if a fixed minimum magnitude level is crossed. The 'samples to ignore' is the number of samples to wait for the DRP output to stabilise after the driving – this is dependent on the decimation value and is due to filter response. The last tick box 'Stop on any Error' is to specify the DSP to stop if an error occurs where the frequency and quality cannot be detected.

5.4.6.3 NOISE CAPTURE AND PROCESSING

The capture of the frequency Bond Rupture noise from the QCM is a relatively simple task for the DSP. This is because the FPGA board handles the processing and storage at the frequency and bandwidth of interest. The only job the DSP has is to set the FPGA's DRP up to capture and store the data into RAM and at the end of an experiment, stopping the capture and copying the captured data from the FPGA to prospa. The typical noise floor is less than -100 dB (depending on the bandwidth set) and so a very accurate measurement of the noise within the bandwidth is possible.

The capture frequency can be set anywhere from 100 kHz to 50 MHz (up to 700 MHz if correct anti-aliasing filters are installed). This would usually be set to around 3 times the fundamental operating frequency so with a 10.7MHz QCM the frequency could be set around 32.1MHz. The bandwidth that the data is captured with can also be varied significantly, with a range of 610Hz to around 10MHz. The bandwidth setting affects the amount of time the data can be captured for as there is only a finite amount of memory. It may be recalled from the hardware section that there is 256k x 16bit memory on the FPGA board. This is only used for data capture and so all this memory is available for storage. The DRP outputs a two channel 24bit signal for every sample gathered so the total number of samples that can be stored in memory is 87381 samples. So a maximum capture time of 143.3 seconds for the 610Hz bandwidth or about 8 milliseconds with a bandwidth of 10MHz. As a standard experiment time would be around 30 seconds a narrow bandwidth should be aimed for to enable the capture of the full experiment time.

The Listen parameters are set on the first tab (Run) on the Bond Rupture GUI – see Figure 5.85. The 'Freq to Monitor' is the frequency that the DRP will capture data at. The 'Find Freq' button is used to perform an automated frequency scan and resonant frequency measurement, and is accurate to around 10 Hz. The 'Monitor Base Freq' tick box selects the ZCQ algorithm to find the resonant frequency and quality at the specified 'Freq to Monitor', this value is saved for users to view as well as updating what frequency the DRP2 records. When selected this operates at the start the DSP program and the start of each sub run. Although this is good at tracking the exact frequency as it changes it introduces frequency spikes in the noise capture data because to measure the frequency it must drive around the frequency in question – this feature is currently disabled in the 'run.asm' file. The 'Decimation value' is the decimation setting that the DRP2 is set to for the capture. The 'Enable FIR Filter' tick box enables the FIR filter block that is situated between the DRP2 and the capture RAM. If this is enabled an extra ten times the decimation amount is selected. The captured bandwidth amount is shown in the text box below it, which combines the decimation and FIR filter setting to calculate the bandwidth. The 'Max Time' text box shows the maximum amount of time that the capture can run for using the FPGA memory.

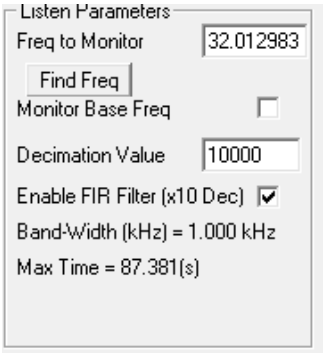


Figure 5.85: Bond Rupture GUI Run tab listen parameters.

5.4.6.4 GAIN UPDATE

The GU (gain update) algorithm is used to set the output voltage to the correct value and ensure that the input voltage does not cause an overflow on the ADC. This is performed using the calibration data gathered using the 'calibration' GUI. This algorithm is operated after every ZCQ or Magnitude run.

The output part of this algorithm controls the Bond Rupture voltage ramping for the purpose of inducing Bond Rupture. This is done using a linear ramp between the start voltage and end voltage. These values including total time is passed to the DSP, which then uses the current time value from the internal timer to calculate the current output voltage. This voltage value is then used in the DSP to calculate the output gain setting and also the DDS output scale setting. The gain and scale settings are dependent on the gain model used which is which is required to be calibrated when ever filters or amplifiers are changed in the system. The calibration process when run, generates a model of the variable gain amplifier (combined with other system components) to be used via the DSP to generate a very accurate output voltage. It should be noted that this is essential an open loop control of the voltage calibrated as into a 50 Ohms. This means that the output power will be constant but the voltage may change significantly. The voltage setting is used for conveyance rather than what is actually being outputted.

The input gain controller starts with a specified input start gain from the BR GUI. Then every time the GU algorithm is operated the program checks for an overflow on the ADC. When an overflow occurs the input gain is decreased by 3dB. More complex algorithms were experimented using the inputted magnitude values, but because it is only looking at a narrow frequency spectrum if an overflow appears in a different part of what is been examined there is no way to know. One solution for this would be a hardware maximum magnitude level monitor on the FPGA, but this would require further development to implement this but, for now the overflow bit monitoring will suffice.

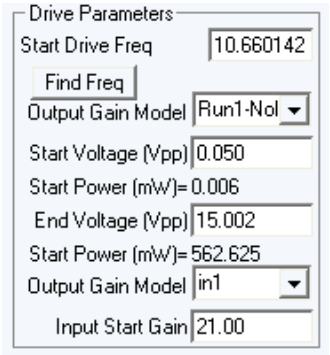


Figure 5.86: Bond Rupture GUI Run tab drive parameters.

The GUI parameters that must be set for the gain control is set in the Bond Rupture GUI's Drive parameters box on the first run page – see Figure 5.86. The first parameter related to the GU algorithm is the 'Output gain model'. This is a pull down menu for selecting from one of the calibration models created in the Calibration GUI. For further information on this see the calibration section. The next parameter is the 'Start Voltage'. This is the starting voltage output (calibrated for 50 Ohms) for the ramp settings, with the text box below it showing the amount of power output that this is. The next parameter is the 'End Voltage'. This is the voltage (calibrated for 50 Ohms) that the voltage ramp will finish on, with the text box below it showing the equivalent power output at this voltage. The next two parameters are related to the input voltage gain. The first parameter is the 'Output Gain Model' from which the input gain model is chosen from a pull down menu. This is setup in the calibration section. The last parameter is 'Input Start Gain'. This is the Gain value to start with using the calibrated input model. This start gain should be changed depending on the start voltage and the number and type of filters on the output to the QCM at this frequency. If there are no filters an attenuation block maybe required on the output from the QCM as the input to the ADC must be less that 1Vpp. When a gain change occurs (triggered by the ADC overflow) this will be shown in the graphing window under other plot input gain. If the graph is changing significantly at the start the gain start value should be decreased, if it is not changing at all or until the end the gain value should be increased. In future versions if the filters after the QCM are known this setting could be removed, to decrease confusion.

5.4.6.5 DATA STORAGE

There is a need to store each experiment in a logical way so that the user can view past runs and also compare different runs. This section describes the data storage method for storing experiments to file. To run an experiment the user must specify a filename and base directory for the data storage. Once this is done "_001" is appended to the end of the filename for the numbering system. This is incremented after each run. If the filename already exists the filename appendage will increment until a spare filename is found. This supports up to 999 data sets under the same name.

The run parameters are stored within a file in the base directory of the format "\filename_XXX.par". This is a text file that saves all the GUI parameters used to run that experiment in one data file. The data from the experiment such as frequency change is stored in a sub directory named "\filename_XXX\...". There are a number of data files stored here

including: important text data, “data1.par”; ZCQ frequency and quality data, “ZC_Q.2d”; ZCQ 3rd harmonic (capture frequency) frequency and quality data, “ZC_Q3.2d”; MLMS frequency and magnitude data, “Freq_MAG.2d”; Gain Update voltage output and input gain setting, “GainC.2d”; Bond Rupture noise capture, “Noise.2d”; and lastly the error data, “Error.2d”. These files all store data that can be accessed within the graphing interface and displayed or plotted on screen.

There are two GUI parameters that are required to be set for the data storage – see Figure 5.87. The first is the filename. This must be one word with no spaces and only one underscore followed by three numerals. These will be set automatically if a single word is typed in. The filename can also be set by using the ‘Filename’ button which opens file browser window to select the filename and directory. The directory can also be selected using the ‘Open Directory’ button which opens the directory browser window to select a directory. Alternatively the directory can be entered directly using the box below it. Once enter is pressed in the box the directory will be created if it does not already exist.

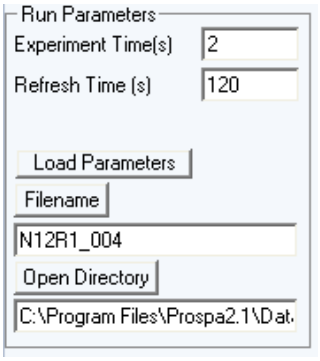


Figure 5.87: Bond Rupture GUI Run tab run parameters.

5.4.6.6 GRAPHING INTERFACE

The Graphing interface enables the data from the current and previous experiments to be plotted on the screen. The graphing options and tools are selected by pressing on the graphing button at the top of the Bond Rupture GUI – see Figure 5.88. The data is selected by first setting the directory to read the data from using the ‘Change dir’ button. This causes the software to build a list of all the data sets within that directory and can be seen in the ‘Data set’ list box. The experiment that the user is interested in can be selected from the pull down box ‘Data set’ - after an experiment this is set to the current run data. Once this is selected the box below this is updated with all the experimental data values such as the run time the number of errors the start frequency, etc. The graph is also updated on the screen with the selected plot settings.

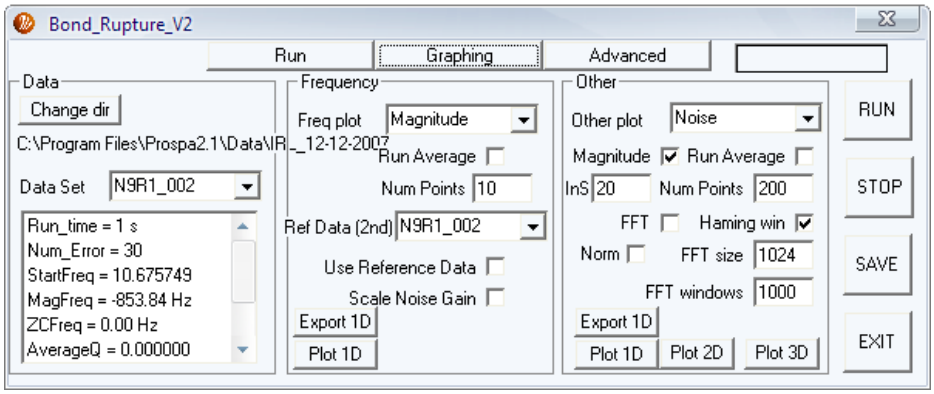


Figure 5.88: Bond Rupture GUI - Graphing controls.

The first plot displays the frequency change data from the magnitude, or zero count programs – see the top plot in Figure 5.89. This is controlled using the variables in the frequency group box. The 'Freq plot' pull down box is used to select the out of the Magnitude, Zero Crossing 1st, or Zero Crossing 3rd. Whichever option is selected will be plotted (if there is data available) on the top of the two plots in the 1D window. The next option is a tick box for enabling the 'Run Average' which performs a moving average on the frequency data with a width specified by 'Num Points'. The next two parameters are used to compare the frequency change between experiments. This is used to determine where the Bond Rupture occurred and is required as the frequency change of the QCM's is not linear with change of voltage and this cancels out the common mode values. The 'Ref Data (2nd)' value used to select the second run for comparison to the original run which is specified in Data Set. These are compared when the 'Use Reference Data' tick box is enabled. The data plot is updated automatically whenever a value is changed and if it does not change the 'plot 1D' button can be used. The 'Export 1D' button is used to save the frequency plot data on screen as a CSV file which is also opened by excel for external processing and plotting.

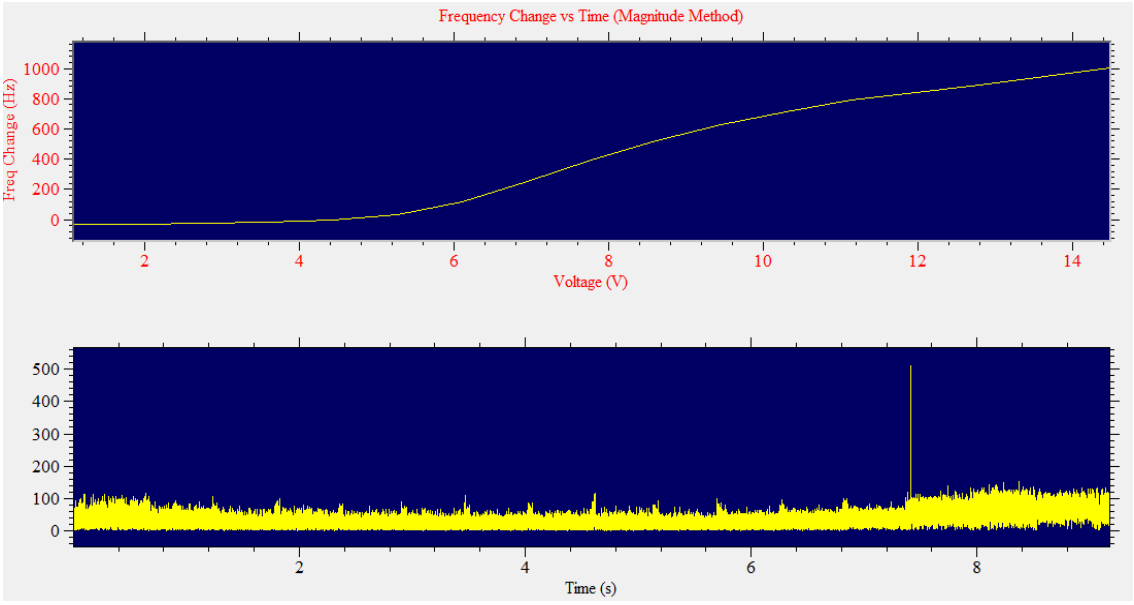


Figure 5.89: Plot generated from the Bond Rupture GUI, (Top) frequency verse time plot, (Bottom) captured magnitude noise data.

The second part of the 1D plot is controlled by the parameters in the 'other' group box – see the bottom plot in Figure 5.90. The 'Other plot' pull down menu enables a whole range of data to be selected including: Bond Rupture captured Noise, 'Noise'; quality of the crystal at the first harmonic, 'quality at 1st'; Magnitude of the driving signal, 'Mag at drive'; Selected voltage driving output, 'Voltage out'; input gain of the amplifiers, 'Input Gain'. Though all these data sets can be displayed only the 'Noise' can be further processed using the other GUI parameters.

The purpose of the rest of the parameters is to enable the post processing of the Bond Rupture noise. The first method for performing this is to convert the data from real and imaginary to the absolute magnitude. This is performed by ticking the 'Magnitude' tick box. The 'Run Average' tick box enables a moving average to run of a size specified by 'Num Points'. The 'Ins' box is the number of samples to ignore from the start of data set. This is needed as it can take some time for the CIC and FIR filters to stabilize. The 'FFT' tick box is used to perform a FFT on all the data captured. The 'Haming win' processes the data before the FFT using the hamming windowing method. The 'Export 1D' button is used to save the current other plot data on screen as a CSV file which is also opened by excel for external processing and plotting. These variables were all related to the 1D plot where as the next parameters are related to the 2D or 3D plot. The 2D or 3D plots are plotted when the '2D plot' or '3D plot' buttons are pressed.

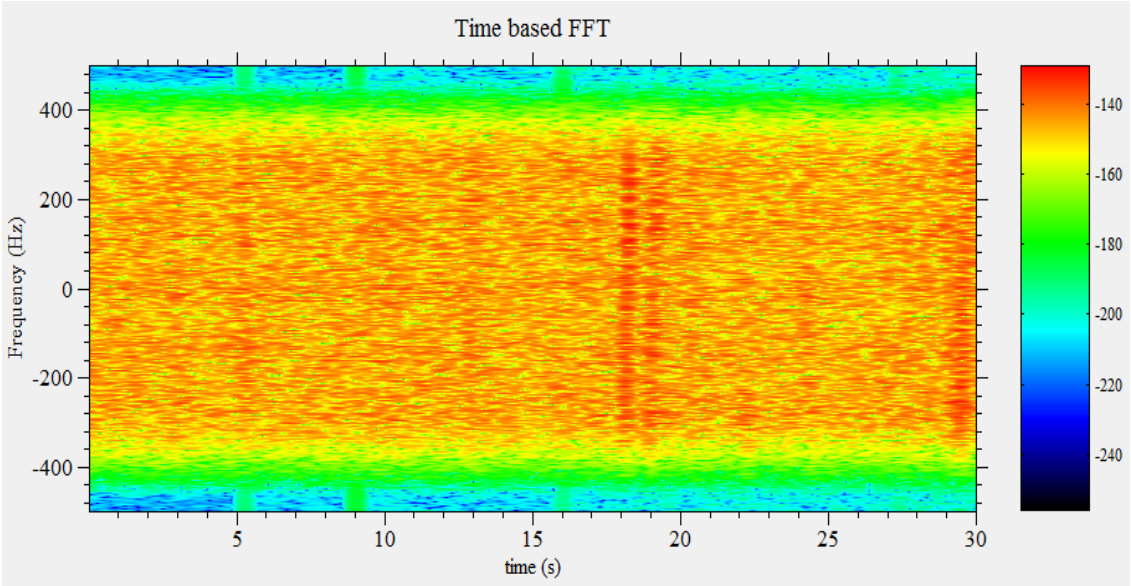


Figure 5.90: 2D FFT plot- Frequency versus time, (side) the colour table in dBFS.

A new plotting method was needed to show the frequency and time data on the same plot. This was performed using a 2D image plot (or 3D surface plot), which enables the time to be shown on the X axis and frequency data to be shown on the Y axis – see Figure 5.110. To generate this plot a 2D FFT is used. The FFT size used is specified by the user in GUI variable 'FFT size' and the number of time samples is determined by the GUI variable 'FFT window' size. The tick box 'Scale Noise Gain' scales the input data depending on the input gain used. This levels out the data so that jumps are not seen when the input gain changes. There is also a tick box called 'Norm' which when ticked takes an average of the FFT frequency data and uses this

as a baseline measurement subtracting this from each frequency line. This normalises the data so that variations due to the filters are compensated for leaving only differences – see Figure 5.91. These plots should enable Bond Rupture to be detected when it occurs and also enable the frequency data to be examined for interference and differences between narrow and wide band noise.

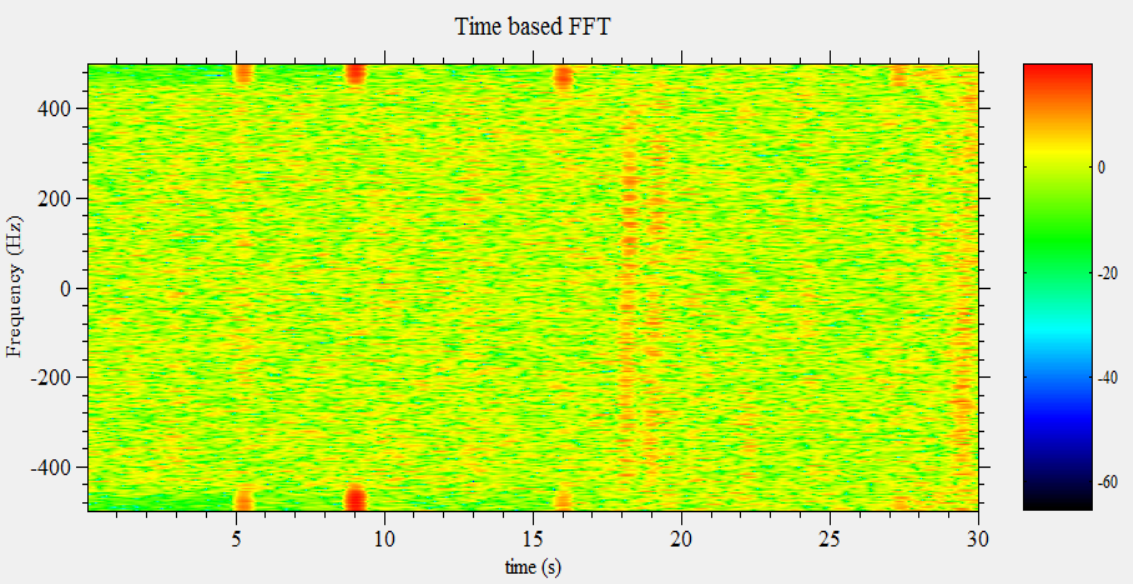


Figure 5.91: Normalised to 0dB 2D FFT - Frequency versus time, (side) the colour table in dB.

The graphing interface developed enables the user to fully investigate the data captured for Bond Rupture frequency change as well as examining the noise and a whole variety of other important variables. This is a significant improvement on the last Bond Rupture system and enables subsequent experiments to be easy kept track of. This also enables the user to export the data captured so further customised processing can be performed.

5.4.7 CONCLUSION

This version of the Bond Rupture software is much more powerful and user friendly than the last version and takes advantage of the new features within the embedded hardware. It also has some significant improvements in the software algorithms and therefore has much better QCM frequency measurement capabilities. This software is also a lot more fault tolerant and will inform the user when serious errors occur. This software provides a significant step forward for generation of Bond Rupture and the subsequent monitoring of the QCM frequency change and Bond Rupture noise detection.

5.5 RESULTS AND DISCUSION

5.5.1 INTRODUCTION

In the development stages of the device hardware and software a significant amount of testing was required to ensure that it was performing as required. This section documents some of the testing methods and results. There was a series of stages of development starting with the

hardware, then moving to the FPGA software, and then to the DSP and prospa software. This section starts by outlining the hardware testing (including FPGA software) then moves on the testing of the high level software and the results with real experiments.

5.5.2 TRANSCEIVER - RECEIVER

First test to be performed is testing the ADC ensure that it is working correctly. This is performed by setting the FPGA's FIFO to capture the RAW ADC data while looking at a good frequency source. Then the transceiver signal must be captured from the FPGA to the computer. The signal used was for initial tests were a 0.9MHz waveform generated using the Agilent (33220A) 20MHz waveform generator and attenuators were used to vary the input level. The data was captured initially using the Xilinx software Chipscope Pro – Chipscope Pro uses blocks within the FPGA to be used to capture internal FPGA data and send it to the computer via the JTAG interface used for programming the FPGA. This data was used to verify the data before the FPGA to DSP interface was complete – Chipscope Pro was also used for debugging the FPGA to DSP interface. Once data was captured the signal integrity is examined to ensure that there are no problems with the latching of the data from the ADC to the FPGA. The results from this can indicate whether the FPGA software is working correctly and if not a recompile maybe required with tighter timing restraints. Once this is verified and working correctly a FFT can be performed on the captured data to verify the dynamic rage of the ADC and find if there are any unexpected spurs. Figure 5.92 shows data captured using Chipscope which is then transferred and analysed using Visual Analog (free Analog Devices software) software. The calculated noise floor from this was -88.83dBFS. One problem was found that there was a large amount general noise and harmonic distortion which is introduced from the Agilent waveform generator and so a better waveform generator was required. Another interesting effect is the DC offset introduced which is around -40dB or 7mV. This is introduced by the ADC itself and also the input buffer amplifier. This poses not real problem as we are not interested in the DC components but it does reduce the maximum input voltage that can be reached without distorting – as the maximum input is reduced by the DC offset amount.

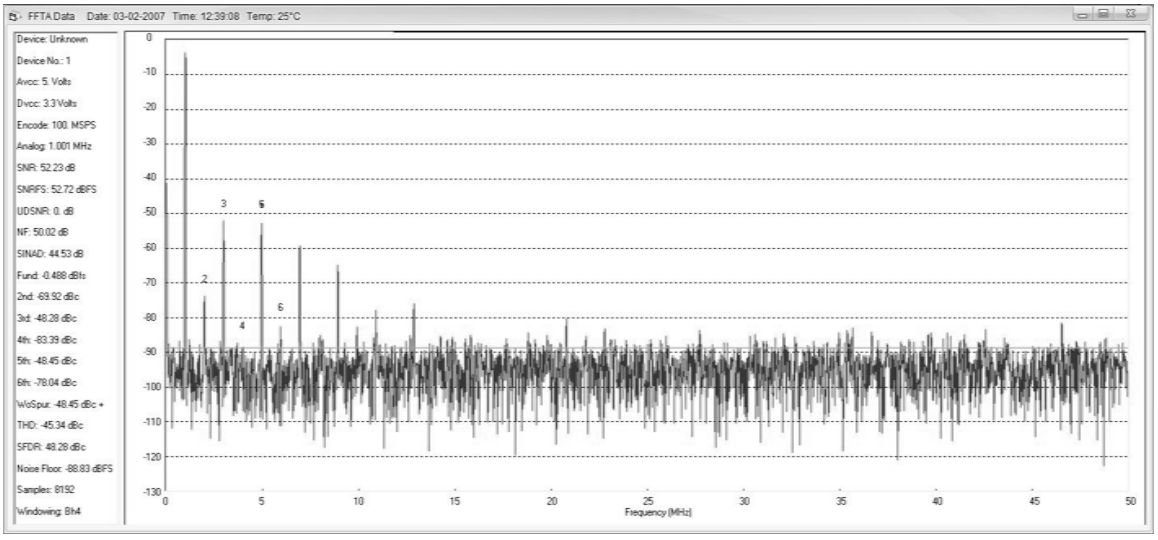


Figure 5.92: FFT data capture of an 8192 point FFT at an input frequency of 6MHz and input voltage of 500mV, using Analog Devices Visual Analog Software for analyzing ADC's.

After FPGA to DSP interface was completed further testing was carried out using the prospa to control the DSP and transfer data from the FPGA. A frequency generator was used to create the signal waveform. This waveform is then captured by the ADC then sent into the FPGA's memory, transferred to the DSP, then the prospa on the PC. The data captured is then post processed via prospa using the fast Fourier transform (FFT) to convert the time domain data into frequency domain data for further analysis. The prospa program used to perform this is called 'FFT capture'. This program is described in detail within the algorithm design section. An example plot is shown in Figure 5.93. The plots are divided up into two subplots. The top plot is the time domain ADC data scaled to be in units of volts, which is convoluted with a hamming window to enable the FFT data to be displayed more accurately. The bottom plot is the FFT of the data captured with the units displayed in Frequency (MHz) and amplitude in dBFS (decibels full scale) – most of the FFT's operate on 128k of data unless otherwise stated.

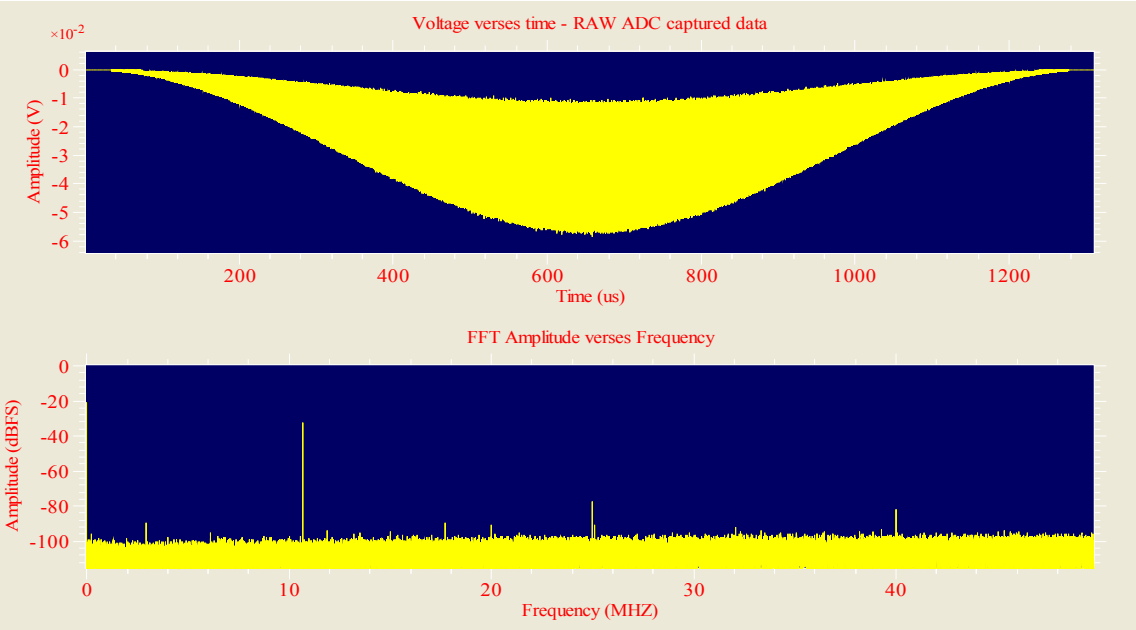


Figure 5.93: Captured RAW ADC data (128k) using an input signal of 10.7MHz at level of ~32dBm generated by the Agilent 33220A (expe3b). (Top) is a portion of the 128k points captured at a frequnecy of 6MHz. (Bottom) is the the FFT plot of the Hamming window of the above data.

The response of RAW ADC using the 'FFT Capture' program is shown in Figure 5.93. This was tested first for a small signal input (-32dBm) at 10.7MHz generated using the Agilent signal generator. The noise floor of this signal is around 96dBFS. The Harmonic distortion for a small signal input is less than 70dBc. A significant difference between this and the previous figure is a spur at 25MHz which is at -73dBFS. This is caused by poor shielding of the DSP's 25MHz clock and is a hardware fault that could be easily improved on in later board versions. This does not affect the system much as this is far away from any frequency of interest and so will be able to be filtered out easily with the DRP.

The next test uses the DRP to extract a signal of interest from the wideband ADC data. To perform this decimation rate of 50 was used which equates to a bandwidth of 2MHz. The signal input used is 10.7001MHz with the DRP set to look at 10.7MHz giving an offset frequency of 100Hz. Figure 5.94 shows the signal captured after the DRP (with hamming

window applied) shown as a quadrature signal. This is data is processed using a FFT and the magnitude data is examined after scaling by to be in units of dBFS. The spurs can be seen to be below the -100dBFS line giving excellent signal to noise response with the noise floor around -120dBFS.

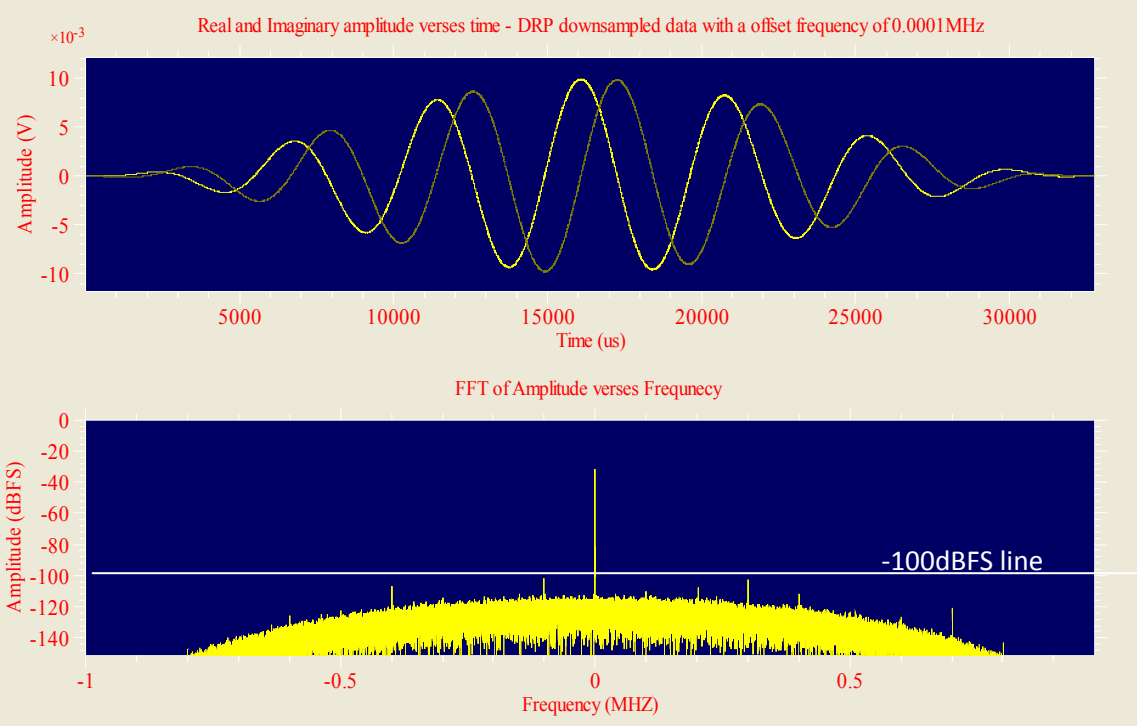


Figure 5.94: Captured data (128k) from DRP with decimation by 50, using an input signal of 10.70001MHz at level of ~-32dBm generated by the Agilent 33220A (expe4). The base frequency of the DDS within the DRP block is set to 10.7MHz. (Top) voltage versus time plot with a hamming window applied. (Bottom) FFT of above data with the correct offset frequency shown.

The baseline noise of the FPGA based transceiver was tested by using connecting a 50 Ohm resistor up to the ADC input and capturing 128k of data. This creates a baseline noise - see Figure 5.95. From this experiment it can be seen that there are spurs at 5MHz, 20MHz, 25MHz, and 40MHz. Though the 25MHz spur is known to come from the DSP clock where the other spurs have appeared from requires further investigation. Fortunately though these are not relevant as the frequencies of interested are not near any of these and so the DRP will filter any of these spurs out.

To measure the viability of the transceiver for high frequency operation it was tested using a 244MHz input signal to the ADC. This signal was at -38dBm input power, generated by a PTS250 synthesizer – see Figure 5.96. This was performed using a method called sub sampling, where the sampling frequency is much lower than the frequency of interest. The bandwidth of the displayed data is still the same but the higher frequencies fold around the Nyquist frequency – so 244MHz signal is at 44MHz. This does require analogue filters to ensure that only the bandwidth of interest is presented to the ADC otherwise the frequency cannot be truly known as it may have come from any one of the aliased frequency points. The FFT in Figure 5.96 shows that even at this high frequency the baseline noise is still very low and should have an adequate response once band limited by the DRP.

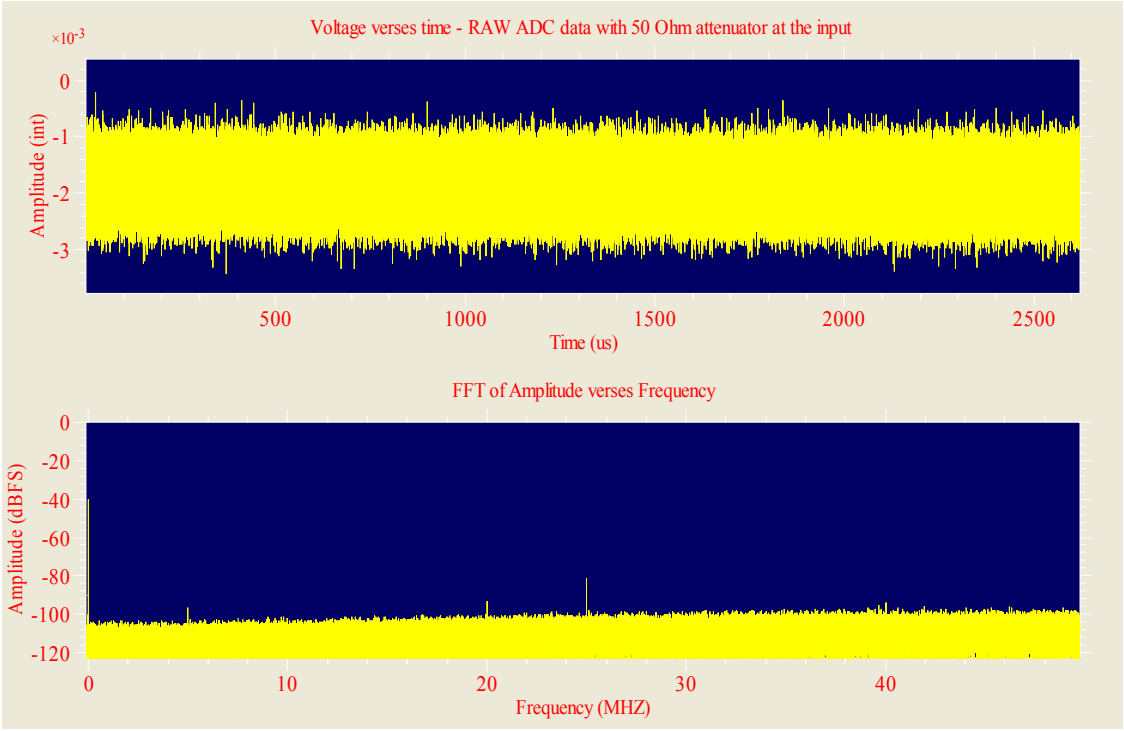


Figure 5.95: Captured RAW ADC data (128k) using a 50Ohm attenuator attached to the input – so only showing interference. (Top) is 128k points captured. (Bottom) is the the FFT plot the data.

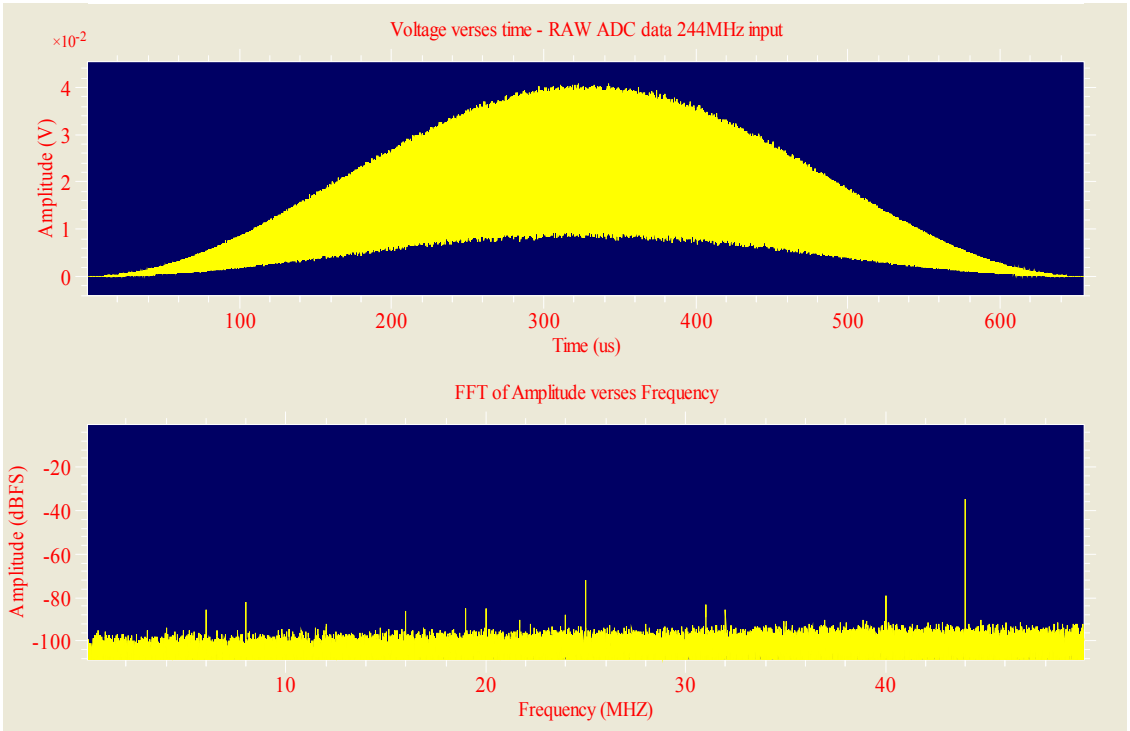


Figure 5.96: Captured RAW ADC data (128k) using an input signal of 244MHz at level of ~-38dBm generated using the PTS250 frequency synthesizer (expg1b). (Top) voltage versus time plot with a hamming window applied. (Bottom) FFT plot data.

The noise floor of the processed signal is very important as this will determine what level of Bond Rupture noise can be detected. Noise floor, gain of the amplifiers, is addressed in the noise measurement section.

5.5.3 TRANSCEIVER - TRANSMITTER

The Transmitter part of the system includes the DDS and scalier within the FPGA, the DAC, and filter and analogue switch. The response of this system is dependent on all of these components response. The DDS response is specified in the Xilinx DDS data sheet (DDS Compiler v2.1) and should behave as specified except if there are compiling problems. The scalier changes the dynamic range of the system by decreasing the number of bits the DAC outputs with. The DAC standard response can be found in the Analog Devices AD9744 datasheet, but this will depend on a number of different factors including: noise on the analogue power supply, temperature of the chip, and the amount of driving current. The output filters response is highly dependent on the component values and so can vary from board to board. The affects the system response significantly in amplitude but should add no noise to the system. The analogue switch should not affect the noise of the system but when it is off it will still pass a specified amount of signal thought to the system (~70dB isolation) the specifications for this can be found in the Analog Devices ADG901 datasheet.

The system dynamic range should be somewhere in the order of 84dB assuming that no scaling is occurring. To measure the combination of all the components in the system a 10.7MHz tone was outputted and captured using the ADC – as this is much more accurate than any oscilloscope. This can be seen in Figure 5.97. The system was also tested by processing this signal with the DRP to look at the quality of signal that could be captured. This can be seen in Figure 5.98. There are however a number of harmonics generated by the DDS block within the FPGA which explains some of the spurs appearing around the primary frequency. There are also a number spurs at the harmonics (2^{nd}, 3^{rd}, 4^{th}, 5^{th}) of the 10.7MHz waveform.

It is also relevant to look at the DAC output after it has been processed by the DRP. This is shown in Figure 5.98. This was captured with a 1 kHz bandwidth and an offset of 15 Hz from the base band. It can be seen that there are some spurs still present but the signal the processed signal to noise ratio is at least 100 dBFS.

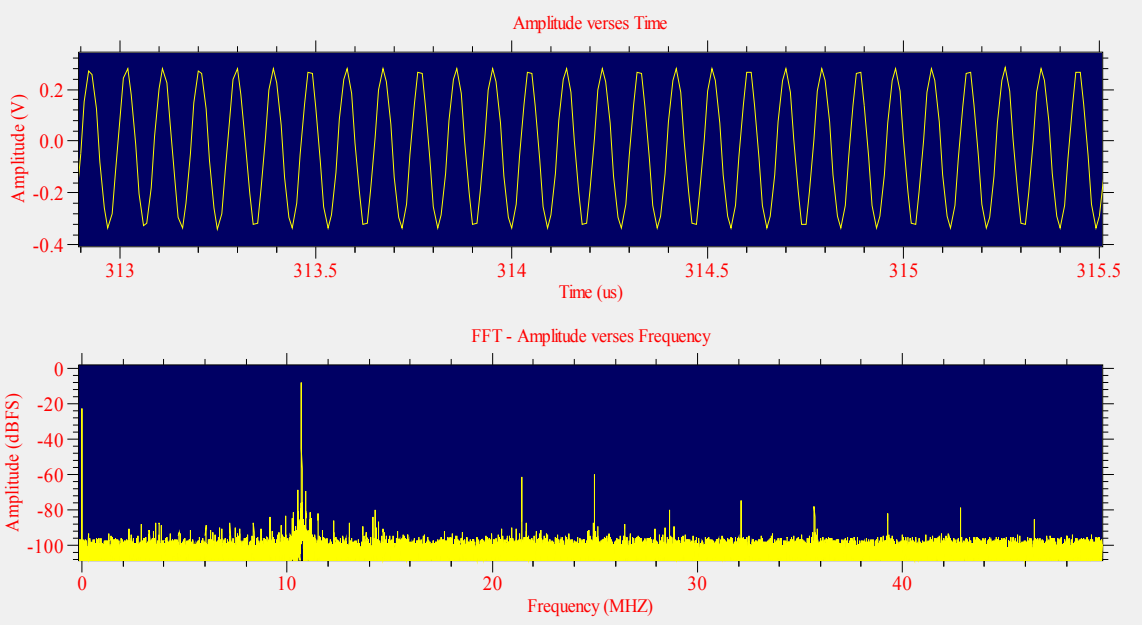


Figure 5.97: DDS running at 10.7MHz which is captured by the ADC directly using a 64k FFT – no filtering. (top) small region of the captured data. (bottom) FFT of the captured data using a hamming window.

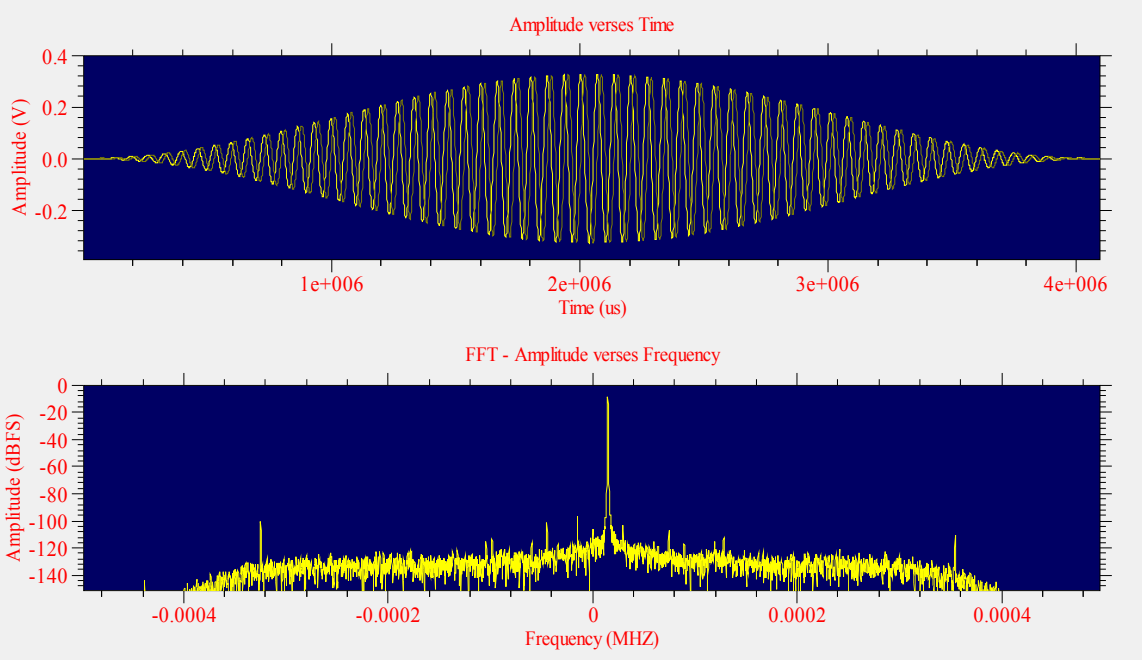


Figure 5.98: DDS data capture using the FFT (64k) Capture Software and 1kHz Bandwidth (decimation set to 10,000 and 10x FIR enabled). (top) captured data with a hamming window applied. (bottom) FFT of captured data.

The overall performance is adequate for the application needed, giving close to 100dB dynamic range for frequency scan measurements. The 'noise' level at the frequency of interest for listening for noise should be very low once the low pass filter is placed into the signal line. This response could possibly be improved with a better DAC or a DDS ASIC chip could be used, but this response is excellent for the application required.

5.5.4 FREQUENCY ACCURACY

Frequency accuracy is important as this informs the user to the changes that are occurring on the surface of the QCM. Acquisition speed is also important in for Bond Rupture measurement as this can improve the speed of the measurement. There is always a trade off between speed and accuracy as this data can be averaged to find a more accurate end value, though this can also cause problems as though a more precise value is seen important variations from sample to sample. Also it requires experiments to be carried out over long periods of time.

The frequency accuracy of the data points gathered is limited by the resolution of the frequency data returned. The frequency is represented by a 32bit number within the FPGA varying from 0 – 100 MHz (set by the DDS output variable setting). Once the frequency is calculated using the MLMS algorithm the value is scaled to this 32bit representation. This provides an accuracy of 0.047Hz for representing the frequency. The accuracy of the actual frequency run will depend on a number of factors. The first of these is the bandwidth of the captured magnitude data, and the number of the points to use in the MLMS algorithm, both of which affect the acquisition time. The other main factor is primarily up to the user to ensure that QCM's with a high quality factor are used as this will significantly affect the frequency accuracy. Another factor to consider is the environment that the QCM is operating in as if the QCM is exposed to air, fluctuations in air temperature and humidity can cause changes up 100Hz. An example of a small section of a Bond Rupture frequency scan can be seen in Figure 5.99. From this we can see that for this section almost no fluctuations in frequency are occurring with the accuracy close to the 0.047Hz set in the QCM. If we then look at the overall fluctuations of the frequency due to Bond Rupture this change appears to be insignificant.

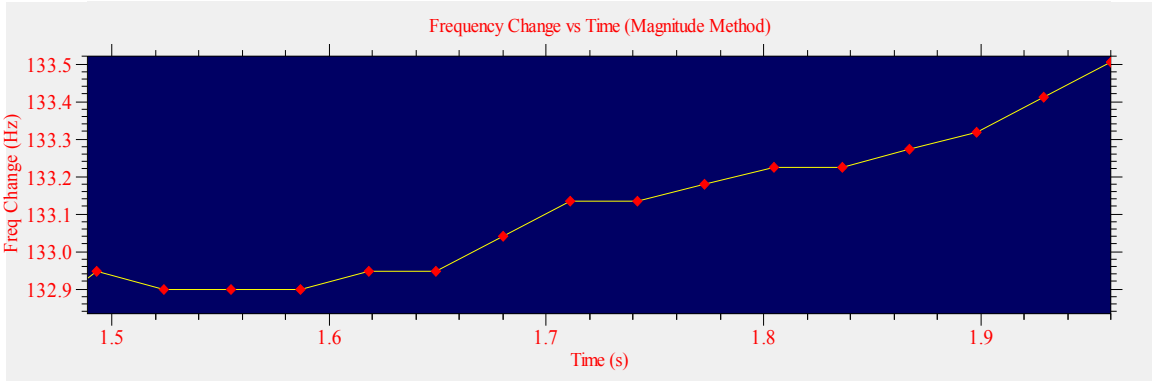


Figure 5.99: Frequency change versus time using the MLMS method.

Traditional frequency measurement devices can vary in acquisition time from 1s to 60s per sample. The Bond Rupture measurement requires the frequency data to be acquired fast a one experiment maybe carried out over 30s or less with over 1000 frequency points returned. If less frequency data points are required, the number of data gathered can always be reduced for higher accuracy, but it would be impossible increase the resolution. The designed acquisition speed is using the default parameters is set to 30 millisecond intervals (or 33Hz) which can be seen in Figure 5.99 as well as boasting an accuracy that rivals any traditional frequency method. This acquisition speed could easily be increased to around 10 milliseconds but this will in turn decrease the frequency resolution.

Another issue with this algorithm is the amount of frequency change that the device can continue measuring without loss of the frequency tracking. The frequency will be locked as long as the resonant frequency of the QCM does not move outside of the specified frequency measurement width, from one frequency measurement to the next. If it does jump outside the specified frequency width, the frequency change will not change more that the width specified. This should only be a problem in cases such as liquid being introduced onto the surface, and as long as the width is still on some part of the slope of the magnitude versus frequency curve the resonant frequency will be found, but may take more than one sample to find it. Whenever the frequency falls outside the width range an error will be created to inform the user of this. If too many errors occur so the frequency is lost the program will stop with a fatal error, so if the QCM is disconnected while running the program will stop running returning an error to the user.

The frequency accuracy could be improved further by decreasing the amount of filtering that is being used to enable the accurate capture of the Bond Rupture 'noise'. The frequency tracking resolution and acquisition speed is however still much better than the previous version of this device and significantly better than traditional QCM measurement methods.

5.5.5 BOND RUPTURE NOISE MEASUREMENT

Bond Rupture noise was first mentioned in literature by the Akubio group (does not exist as of 7-2008). The Bond Rupture noise effect has not yet been verified by independent sources. The research published by the Akubio group showed that noise was generated as the bonds ruptured from the surface. This Bond Rupture noise was shown to occur at a very specific driving voltage. This noise is reported to occur at very low amplitude but there has never been any data to indicate what actual level this noise occurs at. This has made specifying the Bond Rupture device rather difficult.

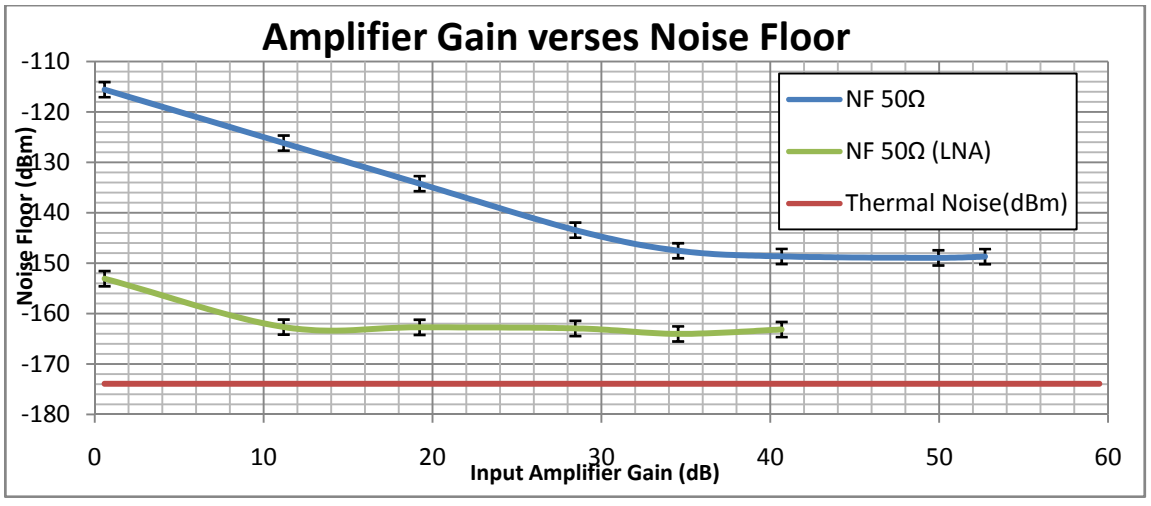


Figure 5.100: Noise Floor analysis of a 50Ohm resistor input to the transceiver input using a Bandwidth of 1kHz and looking at a the noise floor of the 12bit FFT of this data and listening at 32.1MHz.

The device was designed for worst case Bond Rupture noise signal level by using the high speed 16bit 100MHz ADC which is then processed using a DRP, FIR for further signal processing gain and stored as a 24bit value. The 24 bit precision enables a minimum noise floor of -

144dBFS. But in reality this is much larger depending on the signal processing gain (so decimation setting) and the noise in the system and the bandwidth that is being examined. Using the maximum decimation amount (minimum band width of around 610Hz) with a 50 Ohm resistor on the ADC input, the noise floor is down to around -140dBFS when performing a 64k FFT.

The noise floor of the digital transceiver was measured with firstly a 50 Ohm resistor and then with the QCM connected to the input. This is run with the output turned off. This can be seen in Figure 5.95. This experiment was performed using a decimation value of 100,000 using the DRP and the FIR decimation filter which is a bandwidth of 1000Hz. The DRP was set to listen at 32.1MHz as this is near to where the 3rd harmonic Bond Rupture noise should be occurring on a 10.7 MHz QCM. It was found that using a 50 Ohm resister as the input to the 'input amplifiers', the Noise floor is as low as -148dBm or down to -164dBm with the low noise amplifier. The noise figure on the input amplifiers is around 9dB whereas the noise figure for the LNA is around 1.2dB so providing a significant noise measurement advantage. The thermal noise floor is shown as the red line in Figure 5.100.

There are further issues for Bond Rupture detection which are related to the end user rather than the design of the system. One of these issues is the bandwidth in which the Bond Rupture noise will occur in, whether it is over a wide bandwidth or narrow band. There is not much known about the Bond Rupture noise and thought the system is developed there is a significant amount of experimentation to prove that it exists and also to investigate what level it occurs at. The graph in Figure 5.100 shows the thermal noise which is around 10dB away from the theoretical thermal noise limit. From this it can concluded that this device could detect a bond rupture signal as long as it is 3 times larger than thermal noise floor.

Bond Rupture 'noise' measurement is an important part of the experimental process. The process involves filtering out the first harmonic significantly while increasing the 3rd harmonic. Many different configurations frequency measurements were experimented with such as fixed offsets from the third overtone, at the third harmonic, or even using the exact 3rd harmonic frequency was found using a frequency scanner though this causes problems as it changes with mass. The data was then set to capture at this frequency while the Bond Rupture algorithm is running. The bandwidth of the captured data was set to a number of different values to see if this noise is occurring as wideband or narrow band noise.

A number of Bond Rupture experiments were carried out while monitoring for Bond Rupture 'noise' at a fixed frequency, with no conclusive results. An example of this experiment carried out is shown in Figure 5.101. In this experiment the 'noise' was monitored at 3 times the fundamental frequency plus 25 kHz while also performing a Bond Rupture experiment finding though a significant frequency change occurs, still no Bond Rupture 'noise' is observed. This data was also processed with a 2D FFT algorithm to confirm what is happening to see if this sheds any further information about the noise, though also finding none – see Figure 5.102.

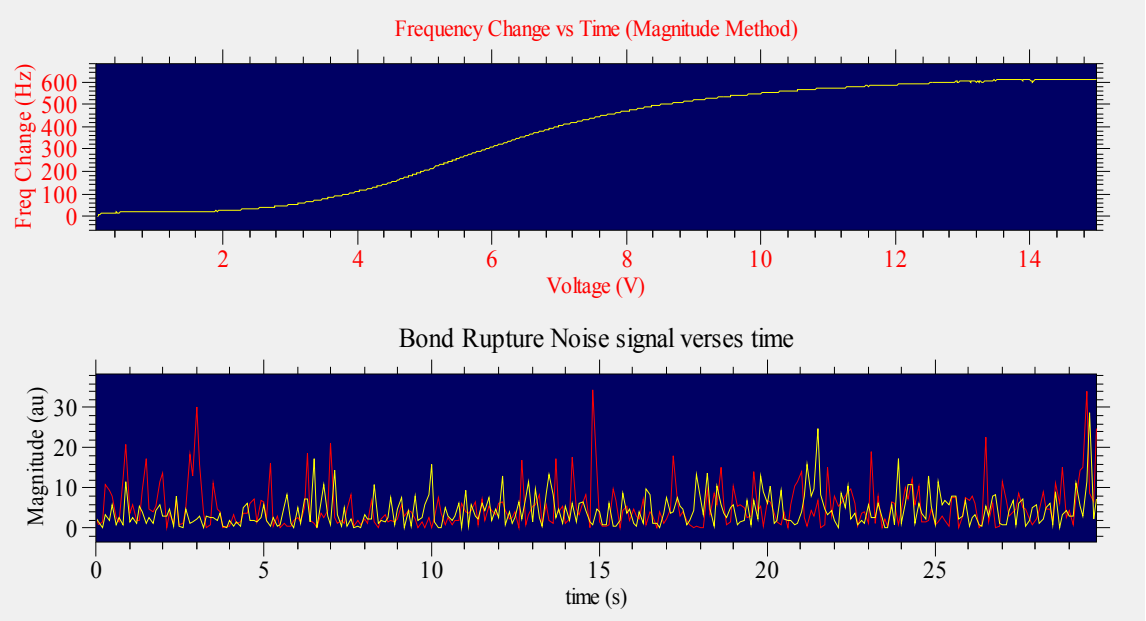


Figure 5.101: Bond rupture experiment with captured FFT data. (top) Bond rupture frequency change confirming Bond Rupture has occurred. (bottom) processed noise from 3 times the first harmonic plus 25Hz, (yellow) is the first run, and (red) is the second run.

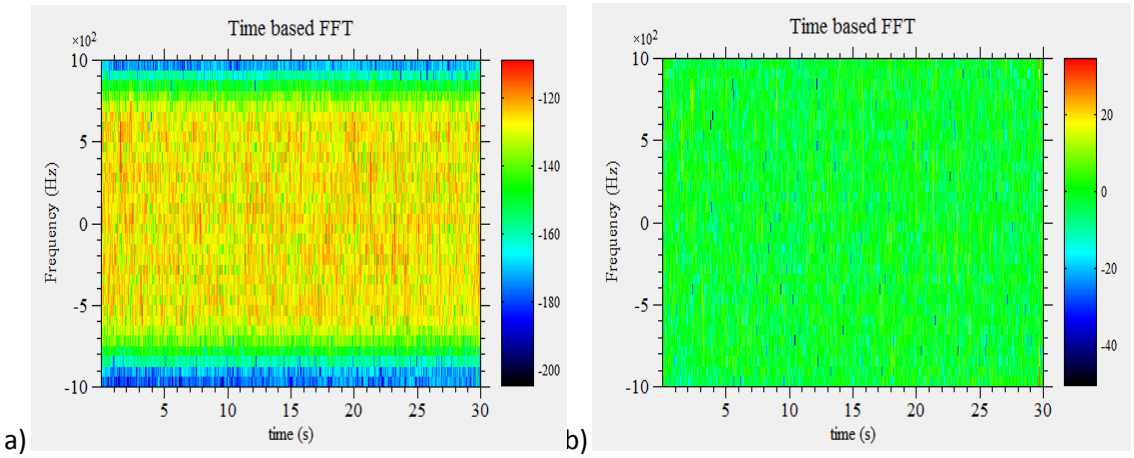


Figure 5.102: Captured noise data processed using a 2D FFT. (a) A 32 bit FFT per section. (b) Normalised FFT of (a).

To confirm that Bond Rupture could be detected with these algorithms the measurement frequency was set to at the third harmonic, where there still exists a significant signal generated from the harmonic of the driving signal. This shows a signal that we should be expecting if the Bond Rupture noise did indeed exist – see Figure 5.103. This is caused by the third harmonic passing thru the DC region of what the DRP is set to, which can be seen in Figure 5.104 showing that the when the frequency passes the DC part of the frequency the signal in found on the corresponding graph. So this is depending on the frequency change on the first harmonic.

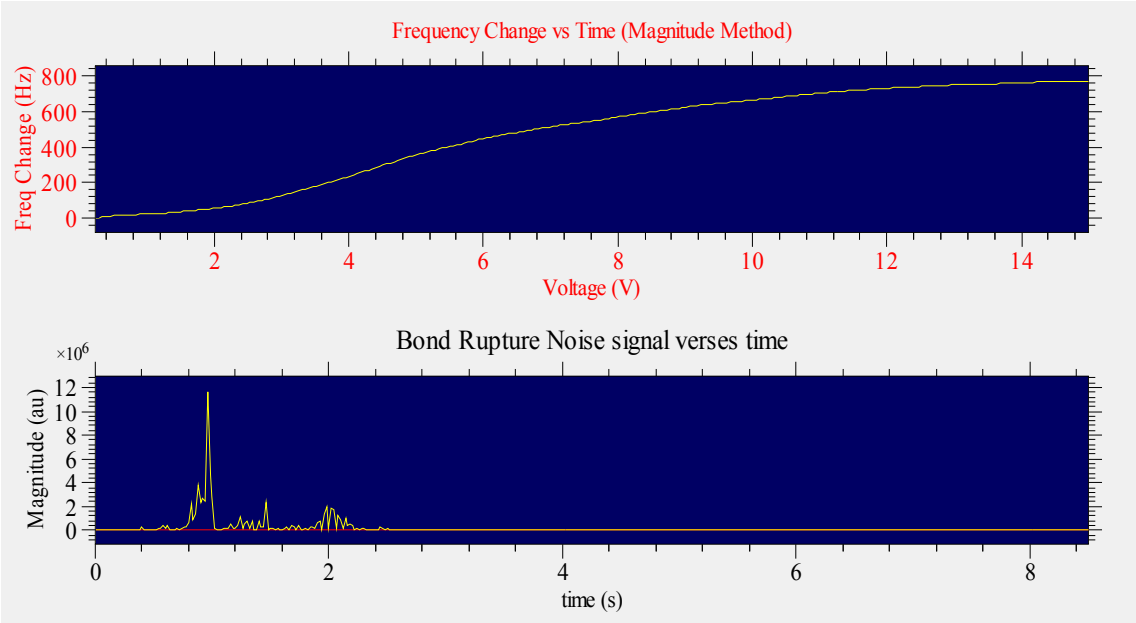


Figure 5.103: Bond rupture experiment with captured FFT data. (top) Bond rupture frequency change confirming Bond Rupture has occurred. (bottom) processed noise from 3 times the first harmonic plus 25Hz, (yellow) is the first run, and (red) is the second run.

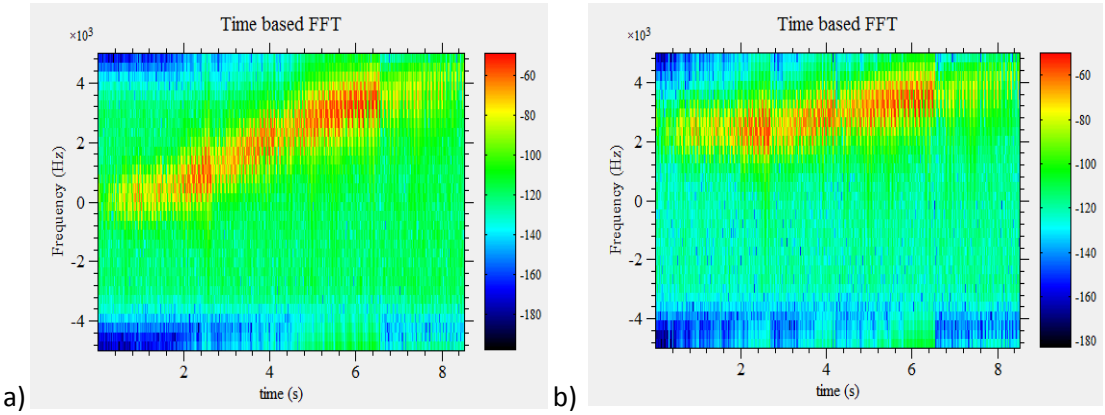


Figure 5.104: Captured noise data processed using a 2D FFT using a 32 bit FFT for the frequency resolution (a) first run (b) second run.

So far no Bond Rupture noise has been detected even thought the signal being examined can go down to within 10dB of thermal noise (using the LNA). This would lead us to believe that Bond Rupture 'noise' is either smaller than within 10dB of thermal noise, or lost though poor impedance matching, or it does not actually exist. It is possible that the Bond Rupture 'noise' could be improved though further investigation into the QCM impedance matching and also changing to a sensor device that the surface as a whole moves rather than the large spread of movement that is evident on the QCM as this would increase the bulk rupture effect.

These results are best case scenario results and demonstrate that if the QCM system is setup correctly Bond Rupture noise could be detected at better resolution than the previously published experiments.

5.5.6 BOND RUPTURE EXPERIMENTATION

A number of different experiments were carried out for the verification of Bond Rupture detection using our equipment. The experiments were carried out using Biotin attached to the surface of a QCM via Self Assembling Monolayer's. A solution of beads (2µm, or 4µm) coated in Streptavidin is then added to the surface to approximate the antibody-antigen interaction. For experimental protocol see Appendix A. It was found that if the voltage ramped up (while operating at the resonant frequency of the QCM) a frequency change occurs which is due to the bonds rupturing from the surface. A frequency change also occurs due to temperature and other individual QCM dependent factors when the voltage is increased. This is adjusted for by running the Bond Rupture scan multiple times and subtracting one run from the next run, creating a curve that represents only the frequency change due to Bond Rupture. Further scans are performed showing almost no frequency change, so it can be assumed that the initial frequency change was caused by bonds rupturing from the surface. This has also been verified by using a microscope to verify the beads have been ruptured.

The Bond Rupture starts from the centre of the QCM and moves out to the edges due to the Gaussian distribution of movement of the surface extending from the centre. This causes the frequency to change over a large range of voltages, but a peak rupture point can be recovered by differentiating the rupture curve or by various other curve fitting methods. This provides information relating to the size of the molecule, strength that the bonds are holding onto the surface, and the number of bonds ruptured.

The accuracy of the Bond Rupture scans is dependent on a number of factors. One is the stability of the QCM from run to run as if the QCM's frequency changes significantly this can lead to false frequency change due to the need for two runs to make up a scan. The frequency accuracy is also decreased by fluctuations in temperature, pressure and humidity. The voltage at which the Bond Rupture occurs also varies between QCM's due to the quality factor, and impedance. Further investigation is needed to stabilise this either by stabilising the QCM's and chemical process better, and by developing active impedance matching circuit for driving the QCM.

An experiment was performed using a linear ramp from 0.1 to 15 Volts over 30s. The beads were attached to the surface of the QCM using the procedure outlined in Appendix A, using Biotin on the surface and then introducing a solution of Streptavidin 4µm beads. This experiment was operated over a series of 5 runs of which 4 scans can be created. This plot is shown in Figure 5.105. The peak rupture point occurs at around 5V and subsequent scans showing no significant other frequency changes. This would suggest that no other Bond Rupture is occurring in subsequent scans. Using a microscope to verify the beads left on the surface it was found that most beads had been ruptured from the surface.

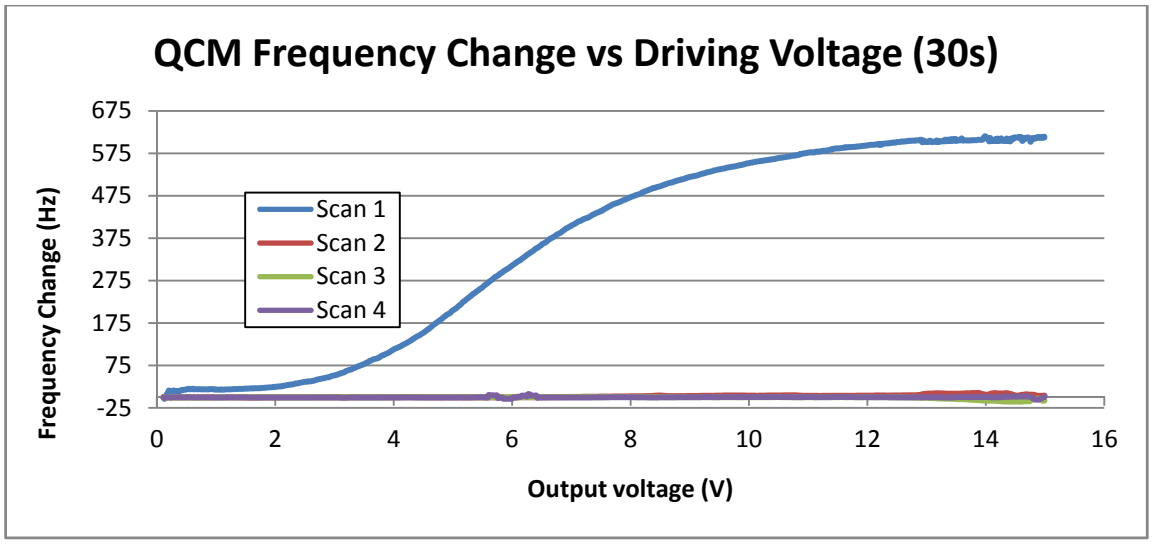


Figure 5.105: Frequency change as the bonds are ruptured over 30s from 0.1 to 15 Volts (921 total data points), 4um beads.

The next experiment was performed using 4µm Streptavidin beads introduced on to a clean uncoated QCM surface. The Bond Rupture was then performed from 0.1 to 8 Volts for 30s over 4 runs creating 3 sets of scan data. This data is shown in Figure 5.106. These experiments were not performed in a stable lab environment which can explain some of the fluctuations present due to temperature and humidity fluctuations. This shows that a significant frequency change occurs at around 0.8 Volts which can be presumed to be due to the weak forces breaking. Subsequent scans showed no significant frequency change.

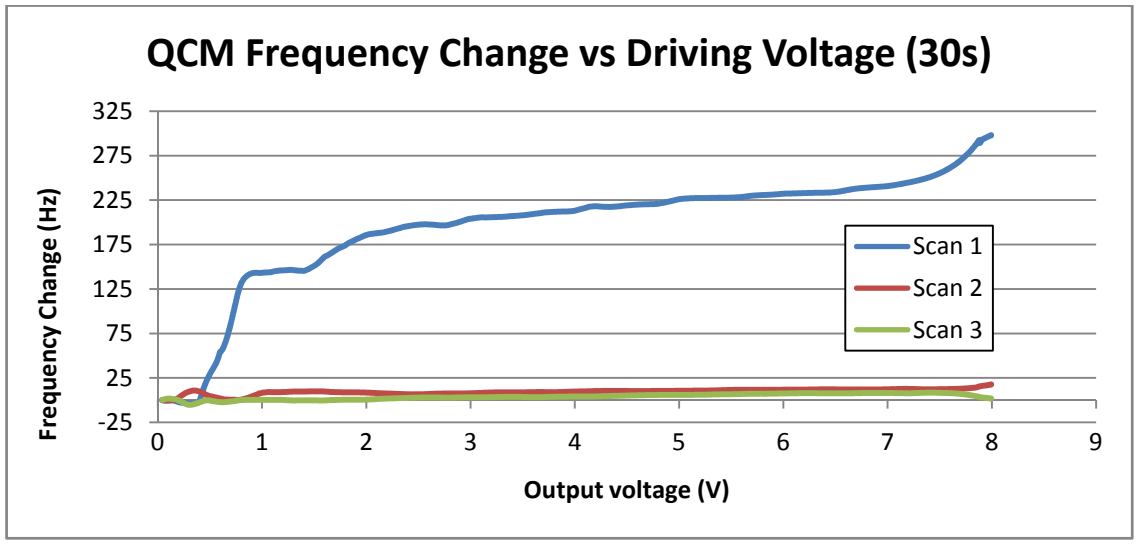


Figure 5.106: Bond Rupture Scan using 4µm beads put on an uncoated QCM surface.

The run time was also experimented with, to see how it affected the Bond Rupture frequency curve. This is shown in Figure 5.107. It was found that the longer the run time used the more spread out the Bond Rupture curve was and alternatively, the shorter the run time the sharper the frequency change curve. So it would seem that the faster the Bond Rupture voltage is ramped the better the results. Unfortunately, changes in the quality of the QCM also have a similar effect. It was found that a low quality factor on a QCM decreases the movement that

occurs on the surface. Further investigation is needed to investigate how much the quality affects the rupture voltage. The amount of frequency change from experiment to experiment has a significant amount of variation due to the bonding preparation. There is much more work that is required to stabilise the QCM's impedance and quality. Also further work is required to reduce the variability in the chemical binding protocols. Also further investigation is required into the impedance matching the QCM's to the equipment and using the QCM's measured quality to calibrate the driving voltages and ramp types required.

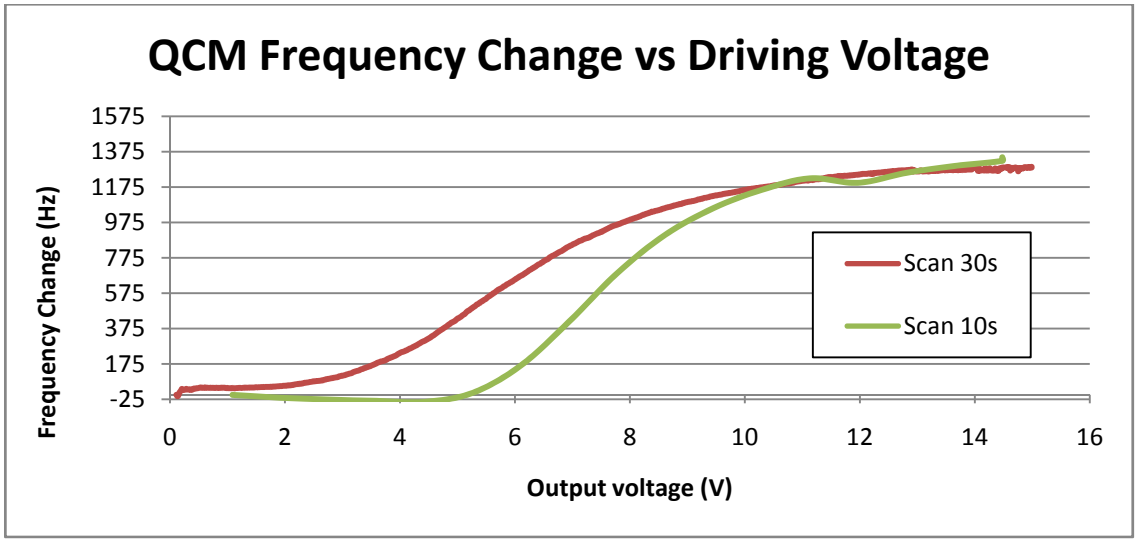


Figure 5.107: Bond Rupture scans from 0.1 to 15 Volts for a variety of run lengths (10s, and 30s).

Another experiment was with the effect of fixed voltage scans. This is shown in Figure 5.108 with two scans at 8V for 30s and then a subsequent scan from 0.1 to 15 Volts shown in Figure 5.109. This shows that the fixed voltage scan ruptured a significant amount of bonds off the surface with the remainder rupturing off in the subsequent scan at above 8V.

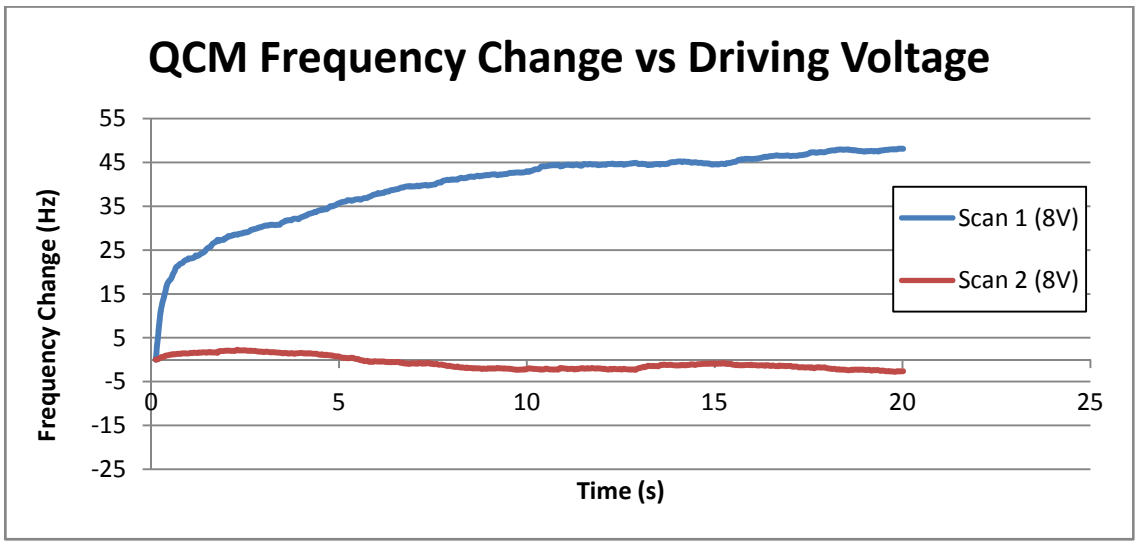


Figure 5.108: Bond Rupture scan at fixed at 8V for 20s.

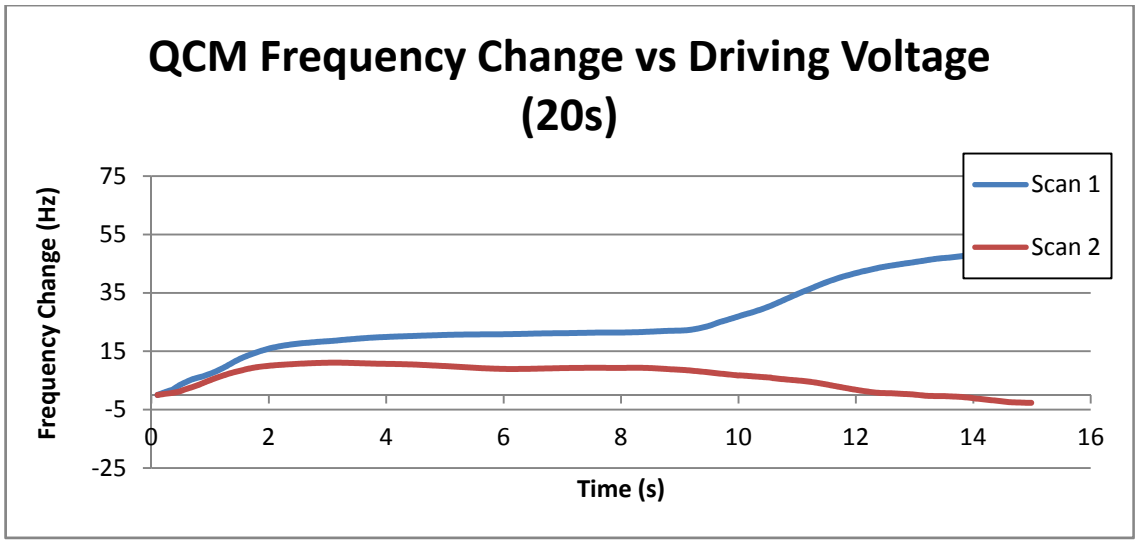


Figure 5.109: Bond Rupture Scan for 30s from 0.1 to 15 Volts following the 8V scan for 20s.

In can be concluded from this section results that Bond Rupture can be observed when the driving voltage is increased via frequency measurement methodology to some degree of accuracy. This could be used for scientific investigation into bond strengths and has the potential to be used to detect viruses and bacteria attached onto the surface though further research is required into the voltage ratio bonding strengths and sensitive of the QCM's used.

5.5.7 LIMITATIONS

There are a few limitations that are present in this system. One which further development is required is the ability to match the impedance actively. This issue was not addressed as the system was originally designed for ultra low noise emission so amplifiers were used directly to the QCM in an open loop configuration. This problem could be addressed using an active impedance matching method such as an operation amplifier with current or voltage feedback to ensure that the impedance does not affect the system. This was however investigated and found that this would affect the noise detection limit so a decision must be made not to use Bond Rupture 'noise' method and rather to use a frequency only Bond Rupture detection method in any later versions. Using the frequency measurement only method would enable a number of optimisations to be made, and unless Bond Rupture 'noise' can be detected, it is believed by this author that any future work should be focused primarily on the frequency detection methods. Using the frequency method only would enable the 3rd harmonic high pass filters to be removed giving significantly better dynamic range, resulting in the quality factor and frequency measurements being much more accurate even at very low signal levels. If we are not as concerned about the noise from the amplifiers getting through to the 3rd harmonic we could reduce the requirements of the low pass filters enabling frequency and quality measurements at multiple higher order harmonics (1st, 3rd, 5th, 7th 9th). This could also be used to increase the information about the Bond Rupture and increase the measurement accuracy. Unfortunately there is but a finite time to complete a PhD as there is much work that could be investigated into and many possibilities for interesting applications for this technology.

5.6 CHAPTER CONCLUSION

It can be concluded that a fully functional Bond Rupture detection system for QCM sensor applications has been developed including hardware and software. This chapter has covered the design of the digital transceiver board and the embedded FPGA software which featured two digital receivers. A system integrated high power low noise RF amplifier board was designed and some work was carried out in the investigation into the impedance matching and filtering for the QCM board. This chapter has also included the design of the embedded software for the DSP and the GUI design to enable this to be a user friendly device. The development of new novel Bond Rupture detection methods has also been achieved. This chapter also contained the testing of the response of the hardware including noise floor measurements, and it was concluded that the noise floor specification of the system was sufficient to detect Bond Rupture 'noise' with the noise within 10dB of the thermal noise limit. The last section contained the testing of the Bond Rupture finding that Bond Rupture can be measured very accurately via the frequency change method. Unfortunately, there was no evidence to suggest that Bond Rupture 'noise' exists with various measurement frequencies analysed with no conclusive results. However, the Bond Rupture frequency measurement method successfully induced Bond Rupture from the surface of the QCM and detected these bonds rupturing. The resonant frequency measurement method enabled sub Hz accuracy, so enabling the Bond Rupture effect to be observed with some precision. This system developed will enable further research to be carried out, into the applications of QCM Bond Rupture.

6 CONCLUSIONS

In the beginning this thesis investigated the various proposed methods for developing a Bond Rupture system. This thesis presents the journey to the development of a Bond Rupture system including detailing initial experimental data leading to a fully developed solution. The system was developed from various technologies including a Digital signal processor (DSP), a Field Programmable Gate Array (FPGA), and high speed digital electronics and various amplifiers. These technologies were combined with various software smarts to enable a fully-fledged Bond Rupture system to be developed which is all combined within a single boxed portable solution.

There has been a significant amount of achievements throughout this thesis. This includes:

- A comprehensive literature critic including the analysis of relevant biosensors, QCM sensing methods, and Bond Rupture background and measurement methods (Chapter 2).
- The initial testing of QCM and Bond Rupture measurement techniques and specifying the requirements for a system that would test all the main QCM Bond Rupture methods (Chapter 3).
- The development of a digital transceiver system that verified the Bond Rupture frequency measurement technique, and laid out the improvements required to verify bond rupture 'noise' and ways to improve on the current measurement method (Chapter 4).
- The development of novel algorithms and techniques for bond rupture 'noise' detection and what has proved to be more important is the Bond Rupture frequency detection. The Bond rupture amplitude frequency tracking method has proved especially useful in tracking the fast frequency changes in low signal and high noise environments (Chapter 4 and Chapter 5).
- The development of a FPGA based digital transceiver system, though unfortunately no conclusive results were found to confirm that Bond Rupture 'noise' exists, the measurement accuracy and reliability of the Bond Rupture frequency method was significantly improved and a significant amount of experiments were able to be carried out to verify Bond Rupture induced frequency change (Chapter 5).

There are still a number of aspects of Bond Rupture that it would be beneficial to investigate further. Among these ideas that require further investigation is the movement of the QCM surface over its area and the impact on the change of the QCM impedance and input voltage on this. This is important as still much is unknown about the actual movement and distribution of the QCM especially at higher voltages inputs and this could enable the force applied to the bonds to be better characterised. It would be also beneficial to perform further processing with the Bond Rupture resonant frequency curve for individual identification of the bond types from this curve as well as being able to differentiate between different bonds on the surface. This could enable multiple receptors to be attached to the sensor for detection of numerous different diseases. There is a considerable amount of research into the chemical binding that is still to be carried out, into the creating of consistent binding to the QCM surface. It would also

be worth investigating further the effect of placing bonds at different places on the surface to increase sensitivity and sharpen the bond rupture frequency curve.

Another option worth investigating is if the Bond Rupture 'noise' was to be abandoned totally (as it is now believed it should be) this device could be further optimised to improve the resonant frequency measurements as well as enabling the monitoring of multiple harmonics simultaneously. This could provide a significant improvement to the resolution and information provided during Bond Rupture. Using the higher order harmonics could be also be used to improve the overall frequency resolution and provide other additional useful information. It would also be interesting to drive Bond Rupture at different combinations of higher order harmonics to examine what differences this would make, potential enabling different parts of the surface to be selected for Bond Rupture examination. There are a lot more ideas that could be experimented with surrounding Bond Rupture and now this Bond Rupture system exists it can be used as a base for further scientific discovery.

The development of this Bond Rupture sensing system has opened up opportunities to new techniques for biosensors, and this could one day be used to help improve the sensitivity of current methods to enable better diagnosis of diseases. Though there is much work to be carried out on the comparison of this technique to current biosensor solutions, the initial experiments into Bond Rupture via the resonant frequency method are promising.

REFERENCES

IEEE Standard (1999). "IEEE standard definitions of physical quantities for fundamental frequency and time metrology - random instabilities." IEEE Std 1139-1999.

Agilent (2004). Agilent Network Analyzer Basics, © Agilent Technologies, Inc. 2004.

*Auge, J., P. Hauptmann, et al. (1995). "New Design For Qcm Sensors In Liquids." Sensors And Actuators B-Chemical **24**(1-3): 43-48.*

*Avci, R., M. Schweitzer, et al. (2004). "Comparison of antibody-antigen interactions on collagen measured by conventional immunological techniques and atomic force microscopy." Langmuir **20**(25): 11053-11063.*

*Baborowski, J. (2004). "Microfabrication of piezoelectric MEMS." Journal Of Electroceramics **12**(1-2): 33-51.*

*Barnes, C., C. Dsilva, et al. (1992). "The Theory Of Operation Of Piezoelectric Quartz Crystal Sensors For Biochemical Application." Sensors And Actuators A-Physical **31**(1-3): 159-163.*

*Borovsky, B., B. L. Mason, et al. (2000). "Scanning tunneling microscope measurements of the amplitude of vibration of a Quartz crystal oscillator." Journal Of Applied Physics **88**(7): 4017-4021.*

*Bouzidi, L., S. S. Narine, et al. (2003). "High-stability Quartz-crystal microbalance for investigations in surface science." Review Of Scientific Instruments **74**(6): 3039-3044.*

Bowick, C. (1997). RF circuit design. Boston, Newnes.

*Buck, R. P., E. Lindner, et al. (2004). "Piezoelectric chemical sensors - (IUPAC Technical Report)." Pure And Applied Chemistry **76**(6): 1139-1160.*

*Buttry, D. A. and M. D. Ward (1992). "Measurement Of Interfacial Processes At Electrode Surfaces With The Electrochemical Quartz Crystal Microbalance." Chemical Reviews **92**(6): 1355-1379.*

*Campbell, C. J. and B. A. Grzybowski (2004). "Microfluidic mixers: from microfabricated to self-assembling devices." Philosophical Transactions Of The Royal Society Of London Series A-Mathematical Physical And Engineering Sciences **362**(1818): 1069-1086.*

*Collings, A. F. and F. Caruso (1997). "Biosensors: recent advances." Reports On Progress In Physics **60**(11): 1397-1445.*

*Cooper, M. A. (2003). "Biosensing using rupture event scanning (REVS)(TM)." Measurement Science & Technology **14**(11): 1888-1893.*

*Cooper, M. A. (2003). "Label-free screening of bio-molecular interactions." Analytical And Bioanalytical Chemistry **377**(5): 834-842.*

*Cooper, M. A., F. N. Dultsev, et al. (2001). "Direct and sensitive detection of a human virus by rupture event scanning." Nature Biotechnology **19**(9): 833-837.*

*Cooper, M. A. and V. T. Singleton (2007). "A survey of the 2001 to 2005 Quartz crystal microbalance biosensor literature: applications of acoustic physics to the analysis of biomolecular interactions." Journal of Molecular Recognition **20**: 154-184.*

*Cooper, M. A. and C. Whalen (2005). "Profiling molecular interactions using label-free acoustic screening." Drug Discovery Today: Technologies **2**(3): 241-245.*

*D'Orazio, P. (2003). "Biosensors in clinical chemistry." Clinica Chimica Acta **334**(1-2): 41-69.*

*Donadio, M. P. (2000). "CIC Filter Introduction." Free Publication by Iowegian, July **18**.*

*Dultsev, F. N., V. P. Ostanin, et al. (2000). ""Hearing" bond breakage. Measurement of Bond Rupture forces using a Quartz crystal microbalance." Langmuir **16**(11): 5036-5040.*

*Dultsev, F. N., R. E. Speight, et al. (2001). "Direct and quantitative detection of bacteriophage by "Hearing" surface detachment using a Quartz crystal microbalance." Analytical Chemistry **73**(16): 3935-3939.*

*Dykstra, R. (2006). The Development of Portable NMR Systems. College of Sciences. Palmerston North, Massey University. **PhD:** 214.*

*Edvardsson, M., M. Rodahl, et al. (2006). "Investigation of binding event perturbations caused by elevated QCM-D oscillation amplitude." Analyst **131**(7): 822-828.*

*Edvardsson, M., M. Rodahl, et al. (2005). "A dual-frequency QCM-D setup operating at elevated oscillation amplitudes." Analytical Chemistry **77**(15): 4918-4926.*

*Edvardsson, M., V. P. Zhdanov, et al. (2007). "Controlled radial distribution of nanoscale vesicles during binding to an oscillating QCM surface." Small **3**(4): 585-589.*

*Ehahoun, H., C. Gabrielli, et al. (2002). "Performances and limits of a parallel oscillator for electrochemical Quartz crystal microbalances." Analytical Chemistry **74**(5): 1119-+.*

*Eichelbaum, F., R. Borngraber, et al. (1999). "Interface circuits for Quartz-crystal-microbalance sensors." Review Of Scientific Instruments **70**(5): 2537-2545.*

*Ferrari, V., D. Marioli, et al. (2001). "Improving the accuracy and operating range of Quartz microbalance sensors by a purposely designed oscillator circuit." Ieee Transactions On Instrumentation And Measurement **50**(5): 1119-1122.*

*Godber, B., M. Frogley, et al. (2007). "Profiling of molecular interactions in real time using acoustic detection." Biosensors & Bioelectronics **22**(9-10): 2382-2386.*

*Grandbois, M., M. Beyer, et al. (1999). "How strong is a covalent bond?" Science **283**(5408): 1727-1730.*

*Heitmann, V. and J. Wegener (2007). "Monitoring cell adhesion by piezoresonators: Impact of increasing oscillation amplitudes." Analytical Chemistry **79**(9): 3392-3400.*

*Hogenauer, E. B. (1981). "An economical class of digital filters for decimation and interpolation." IEEE Transactions on Acoustics, Speech and Signal Processing **29**(2): 155-162.*

*Ivnitski, D., I. Abdel-Hamid, et al. (1999). "Biosensors for detection of pathogenic bacteria." Biosensors & Bioelectronics **14**(7): 599-624.*

*Jakoby, B., G. Art, et al. (2005). "Novel analog readout electronics for microacoustic thickness shear-mode sensors." Ieee Sensors Journal **5**(5): 1106-1111.*

*Janshoff, A., H. J. Galla, et al. (2000). "Piezoelectric mass-sensing devices as biosensors - An alternative to optical biosensors?" Angewandte Chemie-International Edition **39**(22): 4004-4032.*

Janshoff, A. and C. Steinem (2001). "Quartz Crystal Microbalance for Bioanalytical Applications." Sensors Update 9: 313-354.

*Kanazawa, K. K. and J. G. Gordon (1985). "Frequency Of A Quartz Microbalance In Contact With Liquid." Analytical Chemistry **57**(8): 1770-1771.*

*Kaur, J., K. V. Singh, et al. (2004). "Atomic force spectroscopy-based study of antibody pesticide interactions for characterization of immunosensor surface." Biosensors & Bioelectronics **20**(2): 284-293.*

*Kulin, S., R. Kishore, et al. (2002). "Real-time measurement of spontaneous antigen-antibody dissociation." Biophysical Journal **83**(4): 1965-1973.*

*Lavrik, N. V. and P. G. Datskos (2003). "Femtogram mass detection using photothermally actuated nanomechanical resonators." Applied Physics Letters **82**(16): 2697-2699.*

*Lavrik, N. V., M. J. Sepaniak, et al. (2004). "Cantilever transducers as a platform for chemical and biological sensors." Review Of Scientific Instruments **75**(7): 2229-2253.*

*Lee, J. H., K. S. Hwang, et al. (2005). "Immunoassay of prostate-specific antigen (PSA) using resonant frequency shift of piezoelectric nanomechanical microcantilever." Biosensors & Bioelectronics **20**(10): 2157-2162.*

*Lee, J. H., T. S. Kim, et al. (2004). "Effect of mass and stress on resonant frequency shift of functionalized Pb(Zr0.52Ti0.48)O-3 thin film microcantilever for the detection of C-reactive protein." Applied Physics Letters **84**(16): 3187-3189.*

*Li, L., T. Abe, et al. (2003). "- High sensitive, miniaturized plano-convex Quartz crystal microbalance fabricated by reactive ion etching and melting photoresist." **- 1**(-): - 511 vol.1.*

*Lo, Y. S., Y. J. Zhu, et al. (2001). "Loading-rate dependence of individual ligand-receptor bond-rupture forces studied by atomic force microscopy." Langmuir **17**(12): 3741-3748.*

*Lu, F., H. P. Lee, et al. (2005). "Finite element analysis of interference for the laterally coupled Quartz crystal microbalances." Sensors And Actuators A-Physical **119**(1): 90-99.*

*Mecea, V. M. (1989). "A NEW METHOD OF MEASURING THE MASS SENSITIVE AREAS OF QUARTZ CRYSTAL RESONATORS." Journal of Physics E-Scientific Instruments **22**(1): 59-61.*

*Mecea, V. M. (1994). "LOADED VIBRATING QUARTZ SENSORS." Sensors and Actuators a-Physical **40**(1): 1-27.*

*Mecea, V. M. (2005). "From Quartz crystal microbalance to fundamental principles of mass measurements." Analytical Letters **38**(5): 753-767.*

*Merkel, R., P. Nassoy, et al. (1999). "Energy landscapes of receptor-ligand bonds explored with dynamic force spectroscopy." Nature **397**(6714): 50-53.*

Motorola (1998). SEMICONDUCTOR TECHNICAL DATA DSP56309P/D, Rev 0.

*Moulin, A. M., S. J. O'Shea, et al. (2000). "Microcantilever-based biosensors." Ultramicroscopy **82**(1-4): 23-31.*

*Muramatsu, H., J. M. Kim, et al. (2002). "Quartz-crystal sensors for biosensing and chemical analysis." Analytical And Bioanalytical Chemistry **372**(2): 314-321.*

*Nakamura, H. and I. Karube (2003). "Current research activity in biosensors." Analytical And Bioanalytical Chemistry **377**(3): 446 - 468.*

Newton, C. P. P. a. D. J. (1997). Principles and Practice of Immunoassay, Macmillan reference Ltd.

*Nicholls, A., K. A. Sharp, et al. (1991). "Protein Folding And Association - Insights From The Interfacial And Thermodynamic Properties Of Hydrocarbons." Proteins-Structure Function And Genetics **11**(4): 281-296.*

*Oh, B. K., Y. K. Kim, et al. (2004). "Surface plasmon resonance immunosensor for the detection of Salmonella typhimurium." Biosensors & Bioelectronics **19**(11): 1497-1504.*

*Oh, B. K., W. Lee, et al. (2005). "The fabrication of protein chip based on surface plasmon resonance for detection of pathogens." Biosensors & Bioelectronics **20**(9): 1847-1850.*

*Plant, A. L., M. Gray, et al. (1993). "A Fokker-Planck Description Of Multivalent Interactions." Biophysical Chemistry **48**(1): 75-89.*

*Rabe, J., S. Buttgenbach, et al. (2000). "- Design, manufacturing, and characterization of high-frequency." **-**(-): - 112.*

*Rabe, J., V. Seidemann, et al. (2003). "Monolithic fabrication of wireless miniaturized Quartz crystal microbalance (QCM-R) arrays and their application for biochemical sensors." Sensors And Materials **15**(7): 381-391.*

Rodahl, M., F. Hook, et al. (1997). "Simultaneous frequency and dissipation factor QCM measurements of biomolecular adsorption and cell adhesion." Faraday Discussions(107): 229-246.

*Rodahl, M. and B. Kasemo (1996). "A simple setup to simultaneously measure the resonant frequency and the absolute dissipation factor of a Quartz crystal microbalance." Review Of Scientific Instruments **67**(9): 3238-3241.*

Rodriguez-Pardo, L., L. Rodriguez-Pardo, et al. (2005). "Sensitivity, noise, and resolution in QCM sensors in liquid media

*Sensitivity, noise, and resolution in QCM sensors in liquid media." Sensors Journal, IEEE **5**(6): 1251-1257.*

*Sauerbrey, G. (1959). "Verwendung von Schwingquarzen zur Wagung dunner Schichten und zur Mikrowagung " Zeitschrift fur Physik **155**(2): 206-222.*

*Schröder, J., R. Borngraber, et al. (2002). "Advanced interface electronics and methods for QCM." Sensors And Actuators A-Physical **97-8**: 543-547.*

*Sota, H., H. Yoshimine, et al. (2002). "A versatile planar QCM-based sensor design for nonlabeling biomolecule detection." Analytical Chemistry **74**(15): 3592-3598.*

*Tsionsky, V., A. Kaverin, et al. (2005). "An experimental verification of the possible influence of gas nano-bubbles on the response of an electrochemical Quartz crystal microbalance." Physical Chemistry Chemical Physics **7**(8): 1830-1835.*

van der Werff, M. J., Y. J. Yuan, et al. (2005). Quartz Crystal Microbalance for Medical Diagnostics. First International Conference on Sensing Technology. S. C. M. G.Sen Gupta, C. H. Messom. Palmerston North, New Zealand. ***1:*** *500-504.*

Vig, J. R., J. R. Vig, et al. (2000). A review of sensor sensitivity and stability

A review of sensor sensitivity and stability. Frequency Control Symposium and Exhibition, 2000. Proceedings of the 2000 IEEE/EIA International.

Wakamatsu, S., S. Wakamatsu, et al. (2007). Oscillation Frequency Stability of QCM-Biosensor in Liquid Based on Electronic Circuit Technology for Telecommunication

Oscillation Frequency Stability of QCM-Biosensor in Liquid Based on Electronic Circuit Technology for Telecommunication. Frequency Control Symposium, 2007 Joint with the 21st European Frequency and Time Forum. IEEE International.

Ward, M. D. and D. A. Buttry (1990). "Insitu Interfacial Mass Detection With Piezoelectric Transducers." Science ***249****(4972): 1000-1007.*

Weisel, J. W., H. Shuman, et al. (2003). "Protein-protein unbinding induced by force: single-molecule studies." Current Opinion In Structural Biology ***13****(2): 227-235.*

Weisstein, E. W. (2007). "Least Squares Fitting." from http://mathworld.wolfram.com/LeastSquaresFitting.html*.*

Wessendorf, K. O. (2001). "The active-bridge oscillator for use with liquid loaded QCM sensors." -(-): - 407.

Wild, D. (2001). The Immunoassay Handbook, nature publishing group.

Wu, D. H., Y. J. Tsai, et al. (2003). "Robust design of Quartz crystal microbalance using finite element and Taguchi method." Sensors And Actuators B-Chemical ***92****(3): 337-344.*

Yuan, Y. J., M. J. van der Werff, et al. (2007). "Bond rupture of biomolecular interactions by resonant Quartz crystal." Analytical Chemistry ***79****(23): 9039-9044.*

APPENDIX A - EXPERIMENTAL PROTOCOL

Protocol for Immobilisation of Streptavidin Beads on Biotinylated SAM

1. Cleaned with hot piranha solution (a mixture of H_2SO_4:H_2O_2 = 3:1).
2. Rinsed thoroughly with Milli-Q water, then acetone, blown dry with nitrogen flow
3. Immediately immersed in 1:9 mixture of 5mM HS-$(CH_2)_{15}$-COOH in ethanol and 5mM HS-$(CH_2)_{11}$-OH in ethanol for overnight.
4. Rinsed with ethanol, then Milli-Q water, dried by nitrogen flow.
5. Biotin incubated with 8μl drop of a 1: 4: 4: mixture of 25mM Biotin and 0.4M EDC and 0.1M NHS in H_2O for 20 minutes (2.8mM Biotin).
6. Rinsed with Milli-Q water.
7. Exposed to 8μl drop of 1M ethanolamine-HCl for 10 minutes.
8. Rinsed with Milli-Q water and dried with nitrogen flow.
9. Streptavidin beads immobilised by exposing the surface in 250μl diluted bead solution (in a 10mM PBS) for 1 hrs, with vibrator induced.

EDC: 1-ethyl-3-(3-dimethylaminopropyl)carbodiimide

NHS: N-hydroxysuccinimide

HS-$(CH_2)_{11}$-OH: 11-mercapto-1-undecanol

HS-$(CH_2)_{15}$-COOH: 16-mercaptohexadecanoic acid

200X diluted for 4μ streptavidin bead solution

400X diluted for 2μ streptavidin bead solution

2000X diluted for 1μ streptavidin bead solution