University of Louisville

ThinkIR: The University of Louisville's Institutional Repository

Electronic Theses and Dissertations

8-2018

High dynamic range optical devices and applications.

Elijah Robert Jensen
University of Louisville

Follow this and additional works at: <https://ir.library.louisville.edu/etd>

Part of the [Aviation Safety and Security Commons](#), [Computer-Aided Engineering and Design Commons](#), [Electrical and Electronics Commons](#), [Electromagnetics and Photonics Commons](#), [Electro-Mechanical Systems Commons](#), [Engineering Physics Commons](#), [Instrumentation Commons](#), [Maintenance Technology Commons](#), [Other Astrophysics and Astronomy Commons](#), [Signal Processing Commons](#), [Structural Materials Commons](#), and the [Structures and Materials Commons](#)

Recommended Citation

Jensen, Elijah Robert, "High dynamic range optical devices and applications." (2018). *Electronic Theses and Dissertations*. Paper 3044.
<https://doi.org/10.18297/etd/3044>

This Doctoral Dissertation is brought to you for free and open access by ThinkIR: The University of Louisville's Institutional Repository. It has been accepted for inclusion in Electronic Theses and Dissertations by an authorized administrator of ThinkIR: The University of Louisville's Institutional Repository. This title appears here courtesy of the author, who has retained all other copyrights. For more information, please contact thinkir@louisville.edu.

HIGH DYNAMIC RANGE OPTICAL DEVICES AND APPLICATIONS

By

Elijah Robert Jensen

M.S. in Physics, 2015

A Dissertation

Submitted to the Faculty of the

College of Arts and Sciences of the University of Louisville

in Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

in Physics

Department of Physics and Astronomy

University of Louisville

Louisville, Kentucky

August 2018

Copyright 2018 by Elijah Robert Jensen

All rights reserved

HIGH DYNAMIC RANGE OPTICAL DEVICES AND APPLICATIONS

By

Elijah Robert Jensen

M.S. in Physics, 2015

Dissertation approved on

June 15, 2018

by the following dissertation Committee:

Dissertation Director

John Kielkopf

David N. Brown

Gamini Sumansekera

John Naber

DEDICATION

To the ones who care about the truth, the ones who reach for the stars.

ACKNOWLEDGMENTS

The author would like to thank Jeremy Huber, a former doctoral student, who kindly provided some of the source files for his dissertation as an example. I would also like to thank Geoffrey Lentner for his contributions to L^AT_EX styling as well as the many intriguing conversations we have shared. I would also like to thank Joshua Rimmer of the University of Louisville physics department machine shop, for valuable information, conversations and for going above the call of duty on work done for this project. I would also like to thank my advisor Professor John Kielkopf for his help and support throughout this entire process. I would also like to thank my family for their continued support, my parents, for encouraging my curiosity from an early age, for without this none of this work would have have been completed. I would also like to thank my wife, Clarissa, for her loving support and encouragement during this work and always.

ABSTRACT

HIGH DYNAMIC RANGE OPTICAL DEVICES AND APPLICATIONS

Elijah Robert Jensen

June 15, 2018

Much of what we know about fundamental physical law and the universe derives from observations and measurements using optical methods. The passive use of the electromagnetic spectrum can be the best way of studying physical phenomenon in general with minimal disturbance of the system in the process. While for many applications ambient visible light is sufficient, light outside of the visible range may convey more information. The signals of interest are also often a small fraction of the background, and their changes occur on time scales so quickly that they are visually imperceptible. This thesis reports techniques and technologies developed for sensing and detecting rapid transient phenomenon using ambient light in the infrared (IR) spectrum. Currently, high dynamic range optical sensor technology leveraging low-noise and real-time signal processing is employed for applications to human, animal and structural health monitoring, Earth surface motion and environmental monitoring, material defect analysis and astronomy. This work describes the development and fabrication of devices that are made using a novel 32-bit data acquisition system (DAQ), as well as custom-designed circuits for integrating current optical sensing devices into systems for such applications. This thesis also describes the design, construction, and application of an impulse generator for materials testing and a custom-designed Ethernet-connected automated optical fiber positioning stage with examples of their applications to passive non-contact optical sensing.

TABLE OF CONTENTS

Acknowledgments	iv
Abstract	v
List of Tables	ix
List of Figures	x
INTRODUCTION	1
Overview of Imaging Technologies	2
Applications and Advancements	3
HIGH DYNAMIC RANGE DAQ DESIGN	7
Bits and Dynamic Range	7
Noise Reduction	8
ADC Error	10
Conversion Delay Error	10
24-Bit DAQ Design	11
32-Bit DAQ Design	15
32-Bit Single-Ended input DAQ	21
Comparison and Testing of Developed DAQ Technology	26
Conclusion	28
REMOTE SINGLE FIBER POSITIONING SYSTEM	31
Introduction	31
Design	32
Electronics and Interface	43
Conclusion	47

IMPULSE GENERATOR	48
Introduction	48
Principles of Design	48
Machining and Mechanical Construction	59
Electronics	66
Testing	72
TEST PLATE FABRICATION	73
MECHANICAL AND ELECTRICAL ENGINEERING TECHNIQUES AND PROCEDURES	81
Mechanical CAD Packages	81
Basic Machining Practices	82
Special Tools	82
PCB CAD Packages	86
Printed Circuit Board Design Practices	88
Reflow Soldering	91
Special Tools	93
Conclusion	96
MEASUREMENT AND ANALYSIS	97
Photodiode Noise Measurement	97
Basic Fiber Measurements	99
Impulse Measurements	112
Fiber Positioner with Impulse Experiment	121
CONCLUSIONS	125
Conclusions	125
REFERENCES	130

Appendix A: Acknowledgements	134
Appendix B: Mechanical Drawings and Electrical Schematics	135
Appendix C: Commonly Used Acronyms	147
Appendix D: Code	148
Desktop Code	148
Appendix E: 32-Bit DAQ Code	165
Appendix F: XMEGA Library Code	170
Appendix G: Pinger Controller Firmware	199
Appendix H: Fiber Position Firmware	202
CURRICULUM VITAE	213

LIST OF TABLES

1	Comparison of current off the shelf devices available. The cost of each device is listed as well as noise characteristics.	30
---	--	----

LIST OF FIGURES

1	A representation of the electromagnetic (light) spectrum	2
2	Flow chart of 24-bit DAQ	11
3	Example of via stitching	13
4	Standard deviation of grounded 24-bit DAQ system	15
5	Flowchart 32-bit DAQ	16
6	Isolation power supply circuit	18
7	The 32-Bit DAQ PCB	19
8	The 32-Bit DAQ PCB CAD model	20
9	32 Bit DAQ Schematic.	22
10	Board level design	23
11	Final DAQ device circuit board.	25
12	Final DAQ device case	25
13	Device noise floor comparison.	27
14	Noise floor with differing sample rates	28
15	Comparison of DAQ technologies dynamic range	29
16	An early fiber positioner test device	32
17	Linear rail	34
18	Diagram of a dichroic mirror	35
19	Diagram of the optical setup	36
20	Transmissivity and reflectance of the dichroic.	37

21	Early design CAD of positioning stage	38
22	3D model of fiber positioner	40
23	Fiber positioner base and Y-axis	41
24	Positioner minus fiber attachment	41
25	Optics housing model	42
26	Model of beamsplitter assembly	43
27	Fiber positioner machined and assembled	44
28	Fiber positioner web interface	45
29	Fiber positioner main board schematic	46
30	Surface rebound under impulse test	49
31	Impulse mass motion due to elastic collision	51
32	Impulse mass motion due to inelastic collision	52
33	Impulse mass motion with slow retraction	53
34	FEA of cylindrical bar magnet	56
35	FEA of linear motor stator	57
36	A prototype solenoid-based pinger design	60
37	A prototype fulcrum-style pinger	61
38	CAD drawing of cylindrical pinger design	61
39	Cylindrical pinger mechanism cross section	62
40	Cylindrical pinger test device	63
41	Example of pinger tube destruction	64
42	Side by side CAD visualization and a constructed pinger	65
43	Complete pinger system as constructed.	66
44	Final pinger actuator design	67
45	The final pinger system	68
46	A destroyed MOSFET.	69
47	Schematic of H-bridge circuit.	69
48	Final pinger controller design	70

49	Control electronics enclosure	71
50	Surface plate setup	73
51	Mold release mask.	75
52	Mold release application area	75
53	Mixing epoxy	76
54	Example of application of epoxy	77
55	Alignment of top plate	78
56	Aluminum blocks used for curing pressure.	79
57	Final composite test piece	80
58	First flycutter design	83
59	Final flycutter design.	84
60	CNC mill with mist and flood coolant apparatus	86
61	KiCad software screenshot	87
62	An example of poor footprint design	89
63	An example of better footprint design	89
64	An example of a solder paste stencil.	90
65	Solder reflowing setup	92
66	A board being reflow soldered	93
67	Manual pick and place machine	95
68	Analysis of photodiode noise floor	98
69	InGaAs photodiode measurement of a plate vibrated at 144 Hz	101
70	Analysis of silicon photodiode measurement of a vibrated plate	102
71	FEA analysis without bonding defects	103
72	FEA analysis with bonding defects	104
73	Example of audio signal reproduction from ambient scattered light	106
74	Aerial view of laboratory location	108
75	Signal due to approaching train	108

76	Example of signal from sidewalk	110
77	Seismic signal from floor taps	111
78	Satellite image of seismic field measurement area	112
79	Fourier analysis of seismic signals in scattered ambient light	113
80	Laboratory setup for materials analysis.	114
81	Impulse hammer time domain data.	115
82	Hammer impulse optical signal fitting	116
83	Image of impulse test setup	117
84	High speed camera image 1 frame after impulse	118
85	High speed camera image 2 frames after impulse	119
86	High speed camera image 3 frames after impulse	120
87	High speed camera image 4 frames after impulse	121
88	Fiber positioner materials analysis test	122
89	Optical signal from pinger excitation in the time domain	124
1	Fiber positioner drawing 1	135
2	Fiber positioner drawing 2	136
3	Fiber positioner drawing 3	136
4	Pinger drawing 1	137
5	Pinger drawing 2	138
6	Pinger drawing 3	139
7	Pinger drawing 4	140
8	Pinger drawing 5	140
9	Fiber positioner motherboard PCB layout	141
10	Fiber positioner motherboard schematic	142
11	Pinger power supply schematic	142
12	Pinger power supply PCB layout	143
13	TI based 32-bit DAQ PCB layout	143
14	TI based 32-bit DAQ schematic	144

15	LTC based 32-bit DAQ PCB layout	145
16	LTC based 32-bit DAQ schematic	146

CHAPTER I

INTRODUCTION

Most of what we understand of the world around us is based on what we can observe visually. What humans can see however is a very small part of the electromagnetic spectrum. Our eyes only function in a small range of the spectrum from 400 nm to 700 nm [1]. Figure 1[2] shows a map of the EM spectrum. The region just to the right of the visible region in this figure is the infrared region (IR). Infrared literally means “below red” the name comes from the Latin *infra* meaning below. In frequency terms the IR region extends from 300 GHz to 430 THz (1mm to 700 nm wavelength). With regard to human recorded history, IR radiation is a relatively new discovery; William Herschel noticed its effect on a thermometer in the year 1800. Making further experiments on what Herschel called the “calorific rays” that existed beyond the red part of the spectrum, he found that they were reflected, refracted, absorbed and transmitted just like visible light [3]. It would take Maxwell’s theories to solidify “calorific rays” as just another part of the same electromagnetic spectrum.

Figure 1. A representation of the Electromagnetic (Light) Spectrum showing the relatively small visible spectrum as compared to the infrared spectrum.

1 Overview of Imaging Technologies

The first infrared detectors were based on the Seebeck effect. In this scheme the infrared light thermally heats a detector made of dissimilar metals. The voltage produced is linearly proportional to the temperature of the junction, and thus proportional to the infrared wave power transferred. Many far-infrared detectors are still made using this method. This method of thermal conversion is used in microbolometer arrays. An advantage to microbolometer arrays is that the distribution of detectivity versus wavelength is a constant. That is to say microbolometers have a large bandwidth. There is a disadvantage however due to the fact that the thermal conversion is slow compared to other methods of conversion. The substrate must heat up to provide a signal voltage and therefore the sampling time is on the order of seconds to minutes. It was not until 1957 when W.D. Lawson, et. al. discovered HgCdTe ternary alloy as infrared detector material. The first solid state arrays for imaging were produced in 1961 [4]. Since that time much progress has been made in making CCD and CMOS detectors a viable option for IR detection. Photon capture methods still have some disadvantages however. Because of thermal noise it is

impossible to achieve a very high signal to noise ratio in a photon capture based IR detector without cooling the device. Most mid-IR range devices need to be cooled to at least 77K to operate effectively. Far IR range devices need further cooling, in many cases even down to 4K. Different methods of cooling these devices will be discussed further on in this thesis.

2 Applications and Advancements

There are many applications for imaging, particularly imaging in the non-visible spectral range. The applications I have chosen to focus on in this thesis are

- Human Health Monitoring
- Structural Health Monitoring
- Earth Surface Motion or Seismic Sensing
- Material Reliability Analysis
- Astronomy

All of these topics may seem very different from each other, however in regard to optical sensing applications, each are very similar. In astronomy sensing the fluctuation of light signals from a star can be used to determine if that star has a planet or even multiple planets orbiting it. Seismic sensing focuses on detecting vibration generated on the surface of the earth from natural or artificial disturbances to the earth's crust. This relates to structural health and material reliability analysis since all objects; buildings, machines, automotive, aircraft, just to name a few, vibrate at defined frequencies. In health monitoring for instance, one may want to find the heart or breathing rate of a patient; both signals that cause the body to pulse or vibrate at particular frequencies. All of these applications focus on detecting small vibrations on the surface of an object. It is even possible to investigate the macroscopic and

internal dynamics of an object by gathering information on temporal variations of ambient light scattered off the surface of the material. In many cases signals for this purpose can be acquired by using a single low noise photodiode and a single analog to digital converter (ADC). The modulation in the signal received from the photodiode, while perhaps a small part of the overall scattered light signal, will be proportional to the surface tilts produced by vibrations of the object through dependency on diffuse light scattering bidirectional reflectance distribution function. Modulations may also result from periodic translation of contrasting surface features across the sensor field of view. In either case, small motions produce comparably small fractional modulation on a comparatively large constant background. Sensitivity to modulation is limited by the temporal variation of the background due to other causes, and to noise in the background which, in an ideal system, is determined by random shot noise from photon statistics.

At the high light levels needed to detect small modulations of physical interest, this constant background component will saturate an ADC that does not have a sufficient dynamic range. With a normal Gaussian noise distribution, the signal to noise ratio (SNR) will be $\sqrt{N}$ where N is the number of detected photons. Consequently in an instrument with an ADC digitizing b bits we would design optimally for the least significant bit to be the intrinsic noise in the signal, not a single photon. That is, with an b -bit ADC we could fully utilize a signal of 2^b photons and achieve an SNR of $2^{b/2}$. A typical 16-bit ADC used in this mode will allow detection of modulations exceeding one part in 2^{16} or 1.5×10^{-5} . To improve sensitivity even more, a 24-bit ADC could reach 6.0×10^{-8} , a regime where natural noise sources would be as significant as the signals of interest. With a single photodiode, an optimal amplifier, and a 24-bit ADC, commercial solutions are available at least up to 50 KHz, and speeds in GHz are possible with lowered dynamic range. Typical visible light and near-infrared silicon CCD and CMOS image sensors have well depths of 10^5 electrons. They are limited in dynamic range by their read noise of a few electrons at low light

level, and at near saturation by their signal shot noise which for this well depth is 3.2×10^{-3} of the signal. Virtual pixels created by binning off-chip prior to creating an image can have much larger well depths. For example, a large 2048×2048 sensor could bin 16×16 pixels to still offer a 128×128 image with an effective well depth of 1.6×10^9 and a sensitivity of 2.5×10^{-5} . InGaAs sensors, on the other hand, often are manufactured with well depths exceeding 10^6 . In either case, acquisition and processing of small signal changes in high dynamic range data requires customized hardware and software.

In some applications, for instance when trying to visualize a wave moving across a surface, it becomes necessary to obtain spatial image data in the time domain. With image sensors, the achieved frame rate is lowered from the ADC rate by a factor of the number of pixels per frame. A 1-megapixel image acquired at 1000 frames per second would require an ADC operating at 1 GHz which technically precludes a high bit depth ADC. The compromises to be made are the size of the image device, the desired frame rate, and the sensitivity to small modulations. Commercial high speed camera visible light solutions using CMOS sensors tend to the high speed and large sensor end of the spectrum, while for small signal detection a smaller sensor with deeper well depth matched to a suitable ADC is best.

This necessitates the use of a high speed camera specialized for high dynamic range. Normally these requirements run counter to one another: the higher the dynamic range, the slower the camera. Many design issues must be solved to obtain a viable solution. In addition, much processing power is required to perform algorithms on image data for applications such as surface wave detection. In current technology these algorithms are applied on multi-core computers or on a graphics processing unit (GPU). This requires the all the raw digital data from the camera be offloaded to the computer before the calculations can begin, and that overhead alone makes real-time analysis and interactive visualization impractical, degrading the usefulness of the camera in field applications.

Another solution to this type of measurement is to combine a high speed camera with a photodiode measurement. In this way a user can benefit from the resolution of the camera, while being able to take high dynamic range measurements of a single point in the image field using a photodiode. This is accomplished by positioning a fiber optic in the field of the image and using a beam splitter to allow viewing of the same image by a high speed camera as well. The fiber in this case must be accurately positionable at a given point in the frame. Ideally this position would come from a computer-controlled device, so that the process could be automated, or at the least controlled from afar. This would allow a high dynamic range data acquisition unit to read the voltage/current on the photodiode and run at a high speed. The resolution is diminished to a single point, however a viewing camera positioned in the same optical focal plane allows for the simultaneous capture of full scale resolution data. The design and implementation of this fiber positioning device referred to as a “Remote Fiber Positioner” is detailed later in this work.

For materials defect sensing applications a method of vibrating the surface of the material under test must be obtained. Several methods to achieve this are available. One method is to strike the surface with a hammer or impact device. This impulse creates a surface wave that travels quickly across the material. The wave can then be imaged either by a high speed camera or a photodiode sensor. Various analysis methods are then used to determine the health of the material. Another method of providing a vibration is to physically vibrate the test structure using either a speaker coil or a magnet and coil pair. Both of these methods are discussed later on in this work. To provide an impulse, a device must be used that can provide an accurate controllable “ping” or whack to a surface repeatably. For a vibration device, the device should interfere with the physically structure as little as possible to ensure that the resonance of the structure under test is not changed significantly. A method of impacting a test piece is developed in this work, as well as a comparison to impulse testing versus vibration testing.

CHAPTER II

HIGH DYNAMIC RANGE DAQ DESIGN

In modern scientific disciplines it is often necessary to convert analog voltage and current signals from sensors and other apparatus into digital data for further processing. This process is accomplished by using an Analog to Digital Converter (ADC). This chapter will discuss the design of high dynamic range data acquisition instruments and various necessary aspects of high performance data capture.

1 Bits and Dynamic Range

An ADC works by converting an analog signal into binary data. $Q = V/(2^N - 1)$

Dynamic range in dB is defined in Equation 1. In this case we use 20 dB per decade log scale.

$$DR = 20 \log_{10} \left(\frac{V_{max-input}}{V_{noise}} \right) \quad (1)$$

Since in many cases the non-linearity of the log scale tends to complicate matters and obscure understanding of ADC performance, we may use parts per million (PPM) as a metric. The dynamic range in ppm is expressed by equation 1 where N_{noise} is the ADC output numerical code of the ADC noise level.

$$DR = \frac{N_{noise}}{(2^N_{ADC} - 1) \times 10^{-6}} \quad (2)$$

For example the best dynamic range that an ADC could achieve is where the noise is a output code of 1. This would give a DR of $1/(2^N - 1) = 0.06$ PPM. Another way

to calculate dynamic range is a simple linear scale range: $DR = (\text{Full Range Voltage})/(\text{Noise Voltage})$. Another metric that is sometimes used is the signal-to-noise ratio (SNR). Unlike the dynamic range, the signal-to-noise ratio does not assume the maximum voltage input to be the signal. Instead we use a test signal of a given input voltage (or power) to characterize the noise. The SNR is calculated as follows: $SNR = V_{signal}/V_{noise}$ for comparison to engineering literature, it is useful to use dB notation. The SNR in dB is

$$20 \log\left(\frac{V_{signal}}{V_{noise}}\right) \quad (3)$$

For an ADC the best possible SNR in dB can be calculated by

$$6.02 N + 1.76 \text{ dB} \quad (4)$$

where N is the bit depth of the ADC.

In order to compare different devices an efficiency metric will be used in this work. We define the decibel efficiency as

$$\epsilon = \frac{SNR_{device}}{SNR_{fullbits}} \quad (5)$$

for example a 24 bit device with no noise would have an ϵ of 100%. This metric effectively compares the effective number of bits in a ratio format. In this manner it is easy to compare differing devices even if they are designed for differing bit depths. This value can be easily converted to ENOB by using

$$ENOB = \epsilon \times SNR_{fullbits} \quad (6)$$

Both values will be reported for devices designed in this work.

2 Noise Reduction

To achieve good signal good signal integrity a DAQ must have a high SNR. To achieve a high SNR it is necessary to reduce the noise in the signal to a low enough level to meet that goal. In most devices the resolution of the ADC is low enough that

amplifier noise and system noise is below the bit depth floor of the ADC. However as the bit depth is increased the noise becomes apparent to the ADC and becomes a larger factor in the signal integrity, and thus limits the usefulness of the device.

One technique for noise reduction is averaging the signal data. This is a purely digital technique in that the noise reduction is done *ex post facto*. This technique reduces the data rate of the converter since to achieve one data point, one must average more than one data point. For example, in a regime where the ADC is running at 1000 SPS a signal average of 1000 would give an output data rate of 1 SPS. Many commercial delta-sigma ADC devices have this averaging scheme built in to the chip as a standard function.

Another technique is to use analog filtering. The dominant noise source for an analog signal is Johnson-Nyquist noise. This type of noise was first measured by John B. Johnson at Bell Labs in 1926. The Johnson noise level is given by $v_n^2 = 4k_bTR\Delta f$ where $\sqrt{v_n^2}$ is the RMS noise voltage and T is the effective temperature.[5]

From this equation it is easy to see that limiting the bandwidth (Δf) of the amplifier or ADC will reduce the noise voltage significantly. Simple filter designs or Op-Amp filter designs can be employed to attenuate signals not in the detectable range of the ADC. A good rule of thumb for this is the Nyquist frequency. The Nyquist frequency is defined as $f_{nyquist} = f_{sample}/2$. When an analog signal is discretized, the maximum frequency that the input signal can take without error is the Nyquist frequency. If the input frequency is higher than this limit aliasing will occur. Some AD converter Integrated Circuits (ICs) have digital filtering built in, however analog filtering is faster and will aid the digital filter as well if both schemes are used together. A simple way to implement a low pass filter is by placing a capacitor across the input PCB traces (or input leads) and a resistor in series with the input leads. This creates an RC filter. The cutoff frequency for an RC filter is given by $f_0 = 1/(2\pi RC)$. It is important to note that using two capacitors here will increase the effectiveness of the filter. For example: let's say we have an input resistance of $1\text{ M}\Omega$ and we

want to filter everything above 1000 Hz. We would need a capacitance of: 159.15 pF rounding to the nearest production value of 160 pF. Instead of using one capacitor at 160 pF. we should use two capacitors, one at 100 pF and one at 60 pF. The smaller value capacitor acts to allow high frequency signals to short to ground more easily than a large value capacitor would due to the ESR (Equivalent Series Resistance) of production capacitors. In this regime electrolytic capacitors should be avoided, since high frequencies can cause a DC voltage offset if a large electrolytic cell is used.

3 ADC Error

ADC's suffer from two main error issues: offset, and gain errors. Offset error is an error that causes all ADC reading to be offset by a DC value. In order to account for this offset many systems average data points while the inputs are shorted, and subtract the averaged value from all other data points. This is useful if the ADC can be calibrated frequently, however in a device that will run for days at a time (say a seismic monitoring station) calibrating is not possible after every few thousand conversions. As a result, offset drift cannot be accounted for, and will cause error in very low frequency measurements. To solve this problem a chopping amplifier can be used. A chopping amplifier gets its name from the fact that it “chops” the input signal. This scheme works by making two AD conversions for every data point. The first conversion is made with positive polarity, the amplifier circuitry then reverses the polarity with multiplexer type logic and makes a second conversion. These two conversions are averaged to produce final result. This has the effect of negating any DC offset error that the ADC may have.

4 Conversion Delay Error

In many cases signals from multiple sources must be digitized. Most ADC IC's that are marketed achieve this by an analog multiplexer. In this scheme, a signal ADC is connected to the output of an analog multiplexer that is then connected to multiple

input signals. The ADC logic selects each input signal line, and digitizes the signals in series. The result is that the same ADC can run 4 times faster using only one channel, than it can using 4 channels. For most applications this works well; most signals do not need to be sampled simultaneously. However if an experiment requires that all sensors be sampled at the same time, so that their signals can be correlated, this process will not work. One way of solving this issue is to use multiple ADC's. This however becomes impractical when large numbers of input signals are needed, and also become cost prohibitive when a high bit depth result is necessary. A lower cost, and more efficient method is to use a “track and hold” or “sample and hold” amplifier.

5 24-Bit DAQ Design

Using the design criteria and methods mentioned above, a 24-bit DAQ was designed based on the LTC2440 24-bit ADC from Linear Technologies. The LTC2440 is a monolithic 24-bit Delta-Sigma ADC with a top conversion rate of 3.5 kSPS, and a native output rate of 880 Hz.

Circuit Design

Figure 2. Flow chart of 24-bit custom designed DAQ showing the flow of data from analog through data-link to computer.

The flow chart in Figure 2 shows the basic design of the system. In this design the analog signal is input into a LTC1994 fully differential low noise amplifier. The amplified signal is then digitized by the ADC at a rate of 880 Hz the resulting data is output from the ADC via a SPI bus to a microcontroller which can process the data and send the processed data via USART to an FTDI USB-to-Serial converter. The software on the computer receives and saves the data. Due to various reasons internal to the ADC itself the conversion rate is 840Hz, this was measured and confirmed using a Tektronix TDS-220 Real-Time oscilloscope. The ADC conversion clock runs at 880 Hz but every conversion introduces a time delay causing the output data to be actually 840 Hz. This low conversion rate is more than is necessary for digitizing seismic and vibration data, and this DAQ has been used for data capture for these types of signals successfully (See Measurements Chapter for Data and Analysis). The Input amplifier amplifies the input signal such that the ADC is fully saturated at $\pm 1V$ input. This gives a noise free (theoretical) least significant bit (LSB) voltage of 1.1921×10^{-7} . The theoretical dynamic range is $6.021 \times (24) + 1.763 \text{ dB} = 146.27 \text{ dB}$.

Printed Circuit Board Design

Most of the design challenge for this system is in the design and production of the Printed Circuit Board (PCB). To insure that the maximum performance of the ADC can be obtained, it is necessary to design the PCB in such a way that analog signal input noise and digital noise coupling is reduced. All analog traces are kept at a distance from each other so that analog noise does not couple between lines. Digital traces are kept away from all analog traces so that high frequency data signals do not induce voltages on the input analog signals and cause excess noise. A major noise reducing technique used in this design is known as “via stitching”. A trace that has been via stitched is shown in Figure 3. These traces are routed as normal, making every effort to avoid crossing other traces.

Vias are then placed at close intervals on either side of the trace. These vias are

Figure 3. An example of via stitching. Vias are drilled on either side of each trace to facilitated a ground path. Using this method high speed low noise signals can be routed with little interference from surrounding traces.

connected between top and bottom ground planes. This topology creates a virtual Faraday cage, and shields the input analog signals from outside electromagnetic interference. This technique is heavily used in radio frequency design since the high frequency signals have a tendency to propagate into free space. Generally at low frequencies the induced current would not be high enough for most designers to worry about using via stitching. However since we are digitizing these signals at such high bit depths, even nano-amps of induced current can cause added noise and unwanted error in the signal. For this reason all input analog traces are via stitched. In addition to via stitching, the input traces are kept as short as possible. The analog input filter is such that high frequency noise (in the 100 Mhz to Ghz range) will be filtered out of the input signal before the digitization is achieved. However long wavelength interference can be detected. For this reason the signal traces are kept below 10 mm in total length. This corresponds to 30 Ghz radiation. Using this scheme, the input

traces on the PCB will not resonate well at frequencies lower than 30 GHz. This in combination with the use of via stitching isolates the input signals from outside influence and dramatically increases the SNR and dynamic range.

Noise and Dynamic Range Characterization

In order to characterize the performance of the 24-bit DAQ system a set of data was taken with the input shorted. This will cause the ADC to digitize only the noise inherent in the system and allows for the calculation of useful metrics. With the leads shorted a data set of 2048 samples was taken at 840 SPS. The standard deviation of the data set was 0.0065 mV (6.5×10^{-6} V or $6.5 \mu\text{V}$) with an offset of 1.45870 mV and a variance of 4.24502×10^{-5} V. Using the standard deviation to calculate the RMS noise using a running average of 256 data points the standard deviation fluctuated to a max of 0.0073 mV or $7.3 \mu\text{V}$. Figure 4 shows a plot of both the standard deviation as well as the mean of the standard deviation. Using the maximum value of the standard deviation (0.0073 mV) as our worst case RMS noise level, we calculate the dynamic range to be 102.73 dB. Comparing this with the theoretical dynamic range of 146.27 dB we see that the achieved dynamic range is approximately 70%_{dB} of the theoretical. We call this figure the ADC bit efficiency. We can see also that the ADC has a quite high offset error of 1.25870 mV, this will cause the ADC to not perform as well when a large signal is present and will give a non-symmetrical result. In many cases it is useful to find the ENOB (Effective Number of Bits). To find the ENOB we use the dynamic range calculation from before. We can calculate that for an ADC digitizing a ± 1 V with a dynamic range of 102 dB, the number of bits is 18. This means that the 24-bit DAQ system is functioning as if it were a perfect 18 bit converter.

Figure 4. Standard deviation of a grounded 24-bit DAQ system. To measure the intrinsic noise, the inputs were grounded and a set of data was taken. The standard deviation is shown, and can be used to estimate the intrinsic noise, and thus the dynamic range.

24-bit Conclusion

While the 24 bit DAQ offers very good noise performance, in order to effectively see small signals embedded in a large signal or DC offset greater than 24bit resolution is required.

6 32-Bit DAQ Design

During the design phase of the 24-bit DAQ described above, a new 32-Bit ADC from Texas Instruments became available. This chip known as the ADS1262[6] is a monolithic 32-bit Delta-Sigma ADC with a top conversion rate of 38 kSPS. To increase the dynamic range and signal to noise ratio of measurements, design of a 32-bit DAQ

system was commenced. This design incorporates more advanced noise suppression design techniques and reduces the overall noise of the system. The system contains a processor for both DSP functions as well as clocking and data conversion. In this way the computer receiving the data does not need to process the data as heavily, since much processing can be done in real-time using the embedded processor. This increases speed and reduces bandwidth and RAM requirements for the host machine.

Circuit Design

Figure 5. Flowchart of 32-bit DAQ device. This device utilizes the TI 32-bit ADC. Single ended and differential inputs are both available in this design.

A diagram of the design is presented in Figure 5. This design incorporates a processor for filtering and digital signal processing (DSP). Signals are input as direct differential inputs into the ADC. A preamp is used on one channel to enable the input of non-differential signals as well. The ADC is controlled via a SPI bus

with the processor being the master SPI. The ADC is completely isolated from the processor by the use of a isolation power supply and signal isolators capable of up to $1000 V_{\text{rms}}$ isolation. The data is read out of the ADC by use of the SPI bus. It is then processed by the processor and sent to an FTDI USB2.0 FIFO interface device where it is packaged for USB transmission. A major factor in noise reduction in this design is proper design of the isolation power supply. In order to isolate the power, either a capacitor or transformer must be used to couple the energy from the power input to the isolated side (or output). Sufficient current must be available for the ADC and opto-isolators to function properly, therefore a transformer style design was chosen. For the transformer to produce an EMF on the isolation side of the circuit, an alternating current must be used (Faraday's law of induction). This type of supply is commonly referred to as a switching power supply. In low noise environments switching power supplies are not commonly used since the introduction of current switching causes increased probability of noise in the output. However for this design, because of the basic physics of the situation, a switching design must be used. To reduce the introduction of unwanted noise into the power supply output, the frequency of AC current through the transformer should be chosen properly. In this case a higher frequency is better than a lower frequency. This is because a higher frequency is easier to filter out on the isolation side, and because the transformer can be made smaller for a given power output. The frequency used in this design is a variable frequency that is supplied from a TI SN6505 transformer driver. The average output frequency of this device is 600kHz. Since this frequency is significantly higher than the maximum sample rate of the ADC, any noise coupling is easily filtered out. A schematic of the isolation supply is shown in Figure 6.

The output from the transformer is rectified to a DC output by two low voltage drop Schottky diodes in a full wave rectifier configuration. The output is then smoothed by a $10 \mu\text{F}$ diode. The output at this node is approximately 5.4 V to 5.5 V, this signal has very low noise because of the relatively high value smoothing

Figure 6. Diagram of a custom isolation power supply circuit used to deliver power to the analog front end of the 32-bit DAQ design.

capacitor combined with the high frequency output and full wave configuration. For many applications the power output could be used directly from this node. However to reduce noise further and ensure a steady output voltage a 5 V LDO (Low Dropout Voltage) regulator is used to force the output to 5 V. A final filter consisting of a choke inductor and a 100 μ F electrolytic capacitor. The tested output noise level was negligible being too low to be detectable by lab instruments. The ADC has a 2.5 V reference voltage circuit built in. This requires a capacitor to be placed as a filter capacitor. The 2.5 V reference is powered by the 5 V isolated power supply using high accuracy LDO regulator and an output filter.

Printed Circuit Board Design

As in the 24-bit DAQ, most of the design challenge for this system is in the design of the Printed Circuit Board (PCB). A basic block diagram image of the PCB is shown in Figure 7. The board consists of five main parts: USB interface, digital electronics and processor, low noise power supply, ADC, and signal isolation circuitry. These

Figure 7. The prototype 32-Bit DAQ printed circuit board.

are laid out from left to right on the PCB. The USB interface consists of the FTDI IC and necessary circuitry and a USB micro connector. The processor is an ATMEL XMEGA-A4U in a TQFP package. The input analog preamp is based around a LTC1994 fully differential operational amplifier from Linear Technology. Four fully differential inputs are connected to miniDIN connectors directly to the PGA in the ADS1262 32-bit ADC. Only one input is connected via an SMA port to the preamp and then to the ADC. To insure that the maximum performance of the ADC can be obtained, it is necessary to design the PCB in such a way that noise and digital noise coupling is reduced. All analog traces are kept at a distance from each other so that analog noise does not couple between lines. Digital traces are kept away from all analog traces so that high frequency data signals do not induce voltages on the input analog signals and cause excess noise. All input analog traces are via stitched to insure signal integrity. In addition to via stitching, the input traces are kept as short as possible. The analog input filter is such that high frequency noise (in the 100 Mhz to 1 Ghz range) will be filtered out of the input signal before the digitization is achieved. However long wavelength interference can be detected. For this reason the signal traces are kept below 10 mm in total length. This corresponds to 30 Ghz radiation. Using this scheme, the input traces on the PCB will not resonate well at frequencies lower than 30 Ghz. This in combination with the use of via stitching

isolates the input signals from outside influence and dramatically increases the SNR and dynamic range.

Figure 8. 3D CAD model of TI based 32 Bit DAQ printed circuit board.

Figure 8 shows a 3D model of the board design in progress, and Figure 7 shows the completed board. To complete the isolated power supply, the PCB is divided into two sections. The section on the left in the figure is the digital section; this section is powered off of the USB power supply directly with only a EMI filter connected to the voltage regulator. The section to the right has a separated ground plane. This is the analog isolated ground plane. To enable signals to and from the ADC signal isolators are used. The isolators chosen are low noise switched capacitor isolators capable of 1000 V isolation, as well as high data rates. Originally the optoisolators were chosen for this purpose, but their switching frequencies proved to low for reliable use. The isolators are placed bridging the gap between the digital and analog ground planes. Care was taken to route isolated signals away from input signals to reduce the coupling probability. Here digital signal integrity is less of an issue, we are taking care to isolated these signals so that any noise that may coupling to the digital lines from the power supply or USB connection does not couple to the analog input or inside the ADC itself.

Noise and Dynamic Range Characterization

A set of test data was taken with this device and the Dynamic Range was calculated to be 156dB. Using the previously discussed formulas, this figure corresponds to 82%dB efficient. In effective bit terms, this corresponds to a perfect 25-bit ADC. However the device was not stable, and this data cannot be used as a definite value. Under further inspection it was found that the isolation circuits were causing data loss, such that the design would need to be updated and fixed. However another solution to the problem presented itself which is discussed in the following section.

7 32-Bit Single-Ended input DAQ

During the development of the 32-bit DAQ a new 32-bit ADC device became available from Linear Technology. Some issues with data corruption had been observed with the first 32-bit design, that were attributed to the use of isolation chips. Because of these reasons a new board was developed using a LTC2508-32 ADC chip [7]. Although this device outputs data in high speed serial format (2-wire synchronous) it has the unique capability to allow the master bus interface to choose when to receive data. As a result, it is possible to design a system in which the ADC is never reading back data on the digital lines while an analog acquisition is taking place. This removes the requirement that the ADC be electrically isolated from the digital side of the board and improves noise immunity. In this case the device must have a proper ground plane, however the ground need not be isolated, greatly simplifying both design and construction. For use with other lab-based hardware the input amplifier was selected to convert single ended signals to differential signals to be digitized by the ADC. Extensive use of via stitching was employed, as well as the many previously mentioned noise reduction techniques such as EMI suppression using ferrite beads and decoupling capacitors.

This is done so that the input signal frequency will never be higher than the

Figure 10. 3D CAD model of board level printed circuit board design.

nyquist frequency for a given sample rate. The firmware is written such that the filter selection is automatically changed, for a given sample rate. A high dynamic range ADC needs a precision voltage reference, as such the LTC6655-3.3 low noise voltage reference with internal temperature compensation was selected for the task [9]. Since these devices are dealing with signals whose frequency is also a precise measurement, a stable clock was needed. The master clock is provided by a LTC1799 which can be calibrated with a resistor to the proper speed, in this case the nominal frequency is 1MHz. Before the clock signal is connected to the ADC however it is sent to a AND logic IC so that the microcontroller can choose when to power up the ADC. This is important to allow the ADC boot up time so that no undefined behavior results. A schematic of this design is shown in Figure 9.

A board level 3D model of the design is shown in Figure 10. In the design of this device, the low noise design practices discussed earlier were followed closely. The input analog signal is fully via stitched around each PCB trace. In addition, resistors and

capacitors that are necessary for the preamp operation are placed as close as possible to the amplifier chip itself to prevent unwanted noise from coupling to these passive devices. When routing the analog and digital signals, care was taken to prevent sharp corners. In the analog signals this helps contain the wave, as well as lower the probability of noise coupling to the trace. In digital signals this helps reduce the probability of trace radiation. Filter inductors and capacitors are also placed closely for noise consideration. The ADC chip in this case is a QFN (Quad Flat No leads) package. This prevents unwanted radiation from the leads of the chip as well. As was mentioned, the Serial Peripheral Interface (SPI) bus in this design only communicates with the main processor when the analog input is not being sampled. This prevents cross talk from inducing noise on the analog input while sampling is taking place. As a result, in order to achieve a high sample rate, the SPI bus must operate at a high frequency. As a result the SPI bus traces were routed as close to 90 degrees to each other to prevent cross talk issues at high data rates. Via stitching was used around the SPI bus tracing to further reduced crosstalk noise. The FTDI chip in this design communicates with the processor using a 8-bit parallel bus. Every other line of this bus was alternated from bottom to top side of the board to separate the bus lines. As a result of the spacing, ground plane was able to be placed between each line, improving noise immunity and reducing crosstalk between data lines. This also reduces crosstalk between other components on the board. For connectors, due to previous issues with small SMD connectors, a USB2.0 B type connector was chosen for the USB port, and a SMA input was used for the analog input. These connectors have proved to be robust and can withstand the laboratory environment well.

For containing the device a sturdy case was needed. Electronics case manufacturing companies were contacted, however a suitable case was not found. A case was machined from a standard 4 inch aluminum square tube. This gives good rigidity to the case as well as conductivity for shielding. The PCB was placed inside the case and mounting with 4-40 mounting screws. A picture of the completed design is shown

Figure 11. Image of final device interior.

in Figures 11 and 12.

Figure 12. Image of the final device case.

8 Comparison and Testing of Developed DAQ Technology

To compare the developed technology a simple test was devised. The input was connected to a SMA connector that was shorted at its terminals. Each device was shorted in this way and sampled. In this way the intrinsic noise was able to be measured. The voltage data was then converted to decibels for easy comparison to industry standard values. The 24-bit device has a noise floor of approximately -95 dB. The 32-bit device however has a noise floor of approximately -144 dB and trends down as frequency increases to -160 dB. Using this data it is simple to find the dynamic range which is simply the inverse of the normalized noise floor. So in this case the 32-bit device has a dynamic range of 144dB at DC increasing to 160 dB at the Nyquist frequency. In contrast, the 24-bit device has a constant dynamic range of 95 dB. Using these results we can calculate the bit efficiency of the devices. The 32 bit device is $\epsilon = 74$ %dB efficient whereas the 24-bit device is $\epsilon = 65$ %dB. In terms of effective bits, the 32-bit device is functioning as a perfect 24-bit converter, whereas the 24-bit device is effectively a 16-bit converter.

Figure 13 shows a plot the Fourier transform for both devices with shorted inputs. The 32-bit device is shown in red, while the 24 bit device is shown in blue. Each device has a slightly different sample rate, the 24-bit device samples at 840 SPS while the 32-bit device samples at 976 SPS for this test. As a result the figure is plotted with the x axis as percent of sample rate as frequency, this is to be able to compare the full range of each device to each other, rather than in true frequency terms. The sample rates are close enough that no information is lost in this depiction.

The 32-bit device allows for multiple sample rates. When a new sample rate is chosen the on board selectable filter is changed to its optimal filter setting of that sample rate. This offers improved noise performance across the range of sample rates without interfering with input signal performance detrimentally. As the sample rate is increased however more intrinsic noise is present in the system. This in turn

Figure 13. Comparison of the Fourier space noise floor of the 24-bit vs 32-bit devices engineered in this work.

decreases the dynamic range. A wise user of digital signal processing devices will choose the lowest sample rate possible for the given application to maximize noise performance! To illustrate this effect and characterize the effect for this device the graph in Figure 14 was created. In this figure the noise floor for four different sample rates are compared. The data are taken by shorting the input of the device, and selecting differing sample rates. Each plot is Fourier transformed using 1024 points of data. The colored lines in this plot are 2 point running averages for the differing sample rates. The raw data is shown in the background in green. It can be seen in this graph that the noise varies by almost an order of magnitude between the 60 SPS and 4000 SPS. However even at the highest sample rates the 32-bit DAQ's noise floor is still well below the noise of the 24 bit devices. The dynamic range in this case varies from to 127 dB for the 3906 SPS data to 155dB for the 60 SPS data. To compare this technology to current off-the-shelf products, the dynamic range was measured

Figure 14. Frequency spectrum plot of the noise floor for differing sample rates using the final 32-bit DAQ design.

for LabJack T6 device and a Picoscope device. Figure 15 shows a plot of each devices dynamic range. The LabJack boasts a equivalent bit depth of 21-bits, however this is only for low sample rates at ≈ 6 SPS. For comparable sample rates the equivalent dynamic range is reduced. The 24-bit technology developed in this work is however comparable to currently available hardware. The 32-bit device surpasses all devices tested in this work when it comes to dynamic range at a reasonable sampling rate.

9 Conclusion

In this chapter the design of a 32-bit High Dynamic Range data acquisition unit was discussed. The benefits and challenges were presented and solutions to various design issues solved. The final device is a robust data collection unit that can capture large signals while maintaining low noise and high dynamic range. The comparison of the

Figure 15. Comparison of DAQ technologies dynamic range. Similar devices are measured and dynamic range is plotted vs normalized frequency.

original 24-bit devices constructed with the new higher bit depth devices shows a striking improvement in both the bit efficiency and total dynamic range possible. Further work in this area is needed to fine tune the 32-bit device, including adding the ability to sync multiple devices to a single clock, which would allow for cross correlation studies. Other work in this area could extend to designing faster 32-bit analog to digital conversion technologies to extend the usefulness of these devices to applications that require faster sample rates as well.

To illustrate the usefulness of the 32-bit device Table 1 lists current off the shelf devices available. The cheapest option being the LabJack T7 Pro which is USB or

Device	Speed	Bit Depth	Cost
LabJack T7 Pro	6 SPS	24 ENOB 21	\$529.0
Omega DAQ 2408	1000 SPS	24 ENOB N/L	\$2120.0
NI USB-9229	1000 SPS	24 ENOB 21	\$2120.0
MccDaq 2461	1000 SPS	24 ENOB 22	\$1365.0

Table 1. Comparison of current off the shelf devices available. The cost of each device is listed as well as noise characteristics.

Ethernet capable, but only obtains a 21 ENOB if the sample rate is lowered to 6SPS. National Instruments Devices can obtain 21-bit resolution at 1ksps however the cost is substantially higher. This trend continues over all current high dynamic range DAQ devices. The 32-bit DAQ described in this work can be manufactured for a total parts cost of approximately \$35 in quantity > 100 . A market potential and high ROI exists for this type of data acquisition equipment where precision and speed are necessary. Overall the current 32-bit device outperforms high bit depth devices on the market today and will be a useful laboratory data acquisition unit. Further work can be done with this technology to increase the dynamic range further. For instance using multiple analog to digital converters together to increase dynamic range has been documented and typically results in an appreciable increase in dynamic range and intrinsic signal to noise ratio. Using multiple analog to digital converters together may also be able to increase the effective sample rate by delaying one converter by half the sample frequency. This would give a higher effective sample rate with the same dynamic range and signal to noise ratio figures. Further progress in this area will be pursued beyond this work to implement some of these ideas. The device is used later on in this work to provide a high dynamic range capture device in the applications section.

CHAPTER III

REMOTE SINGLE FIBER POSITIONING SYSTEM

1 Introduction

For testing materials and other optical measurements it is sometimes not necessary to have a large resolution. Rather in these cases it is more important to have high dynamic range and a reasonably high sample rate. Commercially available high speed cameras typically have analog to digital conversion bit depth of 12 bits per pixel. As the signal speed increases so does the noise, and thus sampling many pixels with high bit depth and high speed becomes a formidable if not impossible task. In cases where high dynamic range and speed are required, single photodiode measurements are an optimal choice. Using a single photodiode measurement however sacrifices the visualization of a scene that a camera provides. Another solution to this type of measurement is to combine a high speed camera with a photodiode measurement. In this way a user can benefit from the resolution of the camera, while being able to take high dynamic range measurements of a single point in the image field using a photodidode. This is accomplished by positioning a fiber optic in the field of the image and using a beam splitter to allow viewing of the same image by a high speed camera as well. The fiber in this case must be accurately positioned at a given point in the frame. Ideally this position would come from a computer controlled device, so that the process could be automated, or at the least controlled from afar. This chapter outlines the design and construction of an electronic remote fiber positioning system known as the fiber positioner.

2 Design

Early ideas for the fiber positioner revolved around CD/DVD drive technology. DVD drive stepper motors are rather cheap to purchase and some experimentation was devoted to making a fiber positioning device from a DVD drive stepper motor and small machined stage. A few test devices were constructed from small stepper motors taken from CD/DVD drives. Figure 16

Figure 16. An early fiber positioner test device created from DVD drive motor/shaft with custom designed fiber holder.

shows a simple construction of an FC fiber connector mounted to a block of delrin that rides on a hardened and ground 3mm rod. The motion is provided via a lead-screw with grooves cut into the delrin piece to serve as a kind of “half nut” thus providing the linear motion. Preliminary tests were conducted with this design and some promise was shown to the usefulness of the fiber positioning concept. This

design however was lacking in accuracy, as well as only being a single dimension of motion. For this concept to work properly, the full image plane must be able to be accessed by the fiber. This requires a repeatable accurate two dimensional positioning system. The system should be able to attach easily to a tripod or other mounting mechanism, and therefore should be lightweight and fairly compact in construction. A few more prototype devices were constructed using linear rod and DVD drive stepper motors arranged to form a two dimensional stage, however the machining challenge of creating mounting holes for the linear rod to attach to other components proved too difficult for the shop machine equipment and ability. Another issue with using the DVD motors was found to exist; these small stepper motors typically have 20 or 24 steps per rotation instead of the usual NEMA frame stepper motor designs or 200 or 400 steps per rotation. In addition to having less step resolution, these small steppers are not manufactured to precise tolerances, and can have large variances in the angle of rotation between steps. As a result it was concluded that the new prototype design should have the following attributes:

- Lightweight
- As small as reasonably possible.
- NEMA 8 standard 200 step/rotation motors should be used.
- An accurate positioning stage system.

Finding an accurate way of constraining linear motion was an evolutionary process on my part. Several test setups with linear rod were constructed until the idea of linear rod was abandoned completely. Through a search for a way to hold tolerance with only one degree of freedom, it came to my attention that a product known as linear rail was commercially available. A linear rail along with a linear recirculating ball bearing constrains motion to one axis while providing rigidity and preload such that all other motion is constrained. A 7 mm linear rail is pictured in Figure 17.

Figure 17. Example of commercially available linear rail.

Each linear rail is made with hardened and ground rails. The sides of the rail have precision ground grooves in which bearing balls ride in and provide accurate positioning. Since they use ball bearings, linear rails have very low sliding friction and thus a long life. All of these attributes together make linear rails the superior choice for making devices that must move accurately and freely.

After choosing the design parameters the mechanical design was started. It was decided that in order for the device to be of the greatest use it should include a way to visualize the image plane that the fiber would be traversing and collecting data from. The simplest way of doing this is to use a beamsplitter to send half of the light to a camera and half to the fiber positioning stage. The fiber end and the camera area sensor are thus effectively co-planar and sample the same field of view. This however decreases the amplitude of the light arriving at each sensor. In response to this issue the use of a dichroic mirror was proposed. A dichroic mirror reflects certain

wavelengths and transmits others. Dichroic filters and mirrors use the principle of thin-film interference; they are manufactured by coating a transparent substrate with specific thicknesses of materials that cause a defined transmission/reflection curve to be generated. [10] There are two main types of dichroic mirrors; short-pass and long-pass.

Figure 18. Diagram of a typical short-pass dichroic mirror.

A short-pass mirror will reflect (or pass) short wavelengths while reflecting long wavelengths. To a similar affect, a long-pass mirror will pass long wavelengths while reflecting short wavelengths. In this design a short-pass filter was selected. The light from the lens in this design is sent straight to a standard camera, and the only infrared light is reflected by the dichroic mirror and sent to the fiber image plane. A diagram of the beam splitter setup using a short pass mirror is shown in Figure 18.

A diagram of the basic optical design is shown in Figure 19. In this design a standard C-mount camera is used. This is due to the fact that C-mount cameras are common in machine vision applications and readily available. The optical input for the system is from a standard Nikon F-mount lens. The F-mount was chosen for its larger flange focal distance. The F-mount has a flange focal distance of 46.5 mm

Figure 19. Diagram of the complete optical setup. A beam splitter is used to focus light from an F-mount lens to two separated detectors.

as compared to a C-mount lens flange focal distance of 17.526 mm. This allows 28.974 mm between the lens and the C-mount camera for the beamsplitter to reside. After visible light enters the lens it travels through the dichroic beamsplitter and is imaged by the camera. The infrared light is reflected at a 90° angle to the input lens and sent to the fiber which is mounted in a ferrule mounted on a positionable stage. The fiber is connected to a photodiode and amplifier; the signal is then digitized by the DAQ. This setup is physically not unlike a Single Lens Reflex camera design (SLR).

The particular dichroic used in this design is a Thorlabs DMSP950R. This dichroic has a cutoff wavelength of 950 nm. A graph of the transmission and reflectance is

shown in Figure 20.

Figure 20. The transmission and reflection of the dichroic beam splitter. The solar irradiance at ground level and the quantum efficiency of various detectors show the even division of detectable energy between two devices.

The transmission range is from roughly 400 nm to 900 nm in the visible and the reflection range is roughly 900 nm to 1600 nm for infrared light. Using this setup virtually all available light reaches each sensor in its respective sensitive range.

We now turn our attention to the design of the positioning stage itself. This must be carefully designed and machined to make sure all surfaces are parallel to one another. Figure 21 shows an early design model of the stage with no electronics attached. At this point in the design the lens was going to be mounted directly to

the stage case; this design was abandoned due to constraints with the focal plane distance and logistics of attachment. However the stage design itself did not change drastically. For these purposes we will choose the Y axis as the Left-Right axis in Figure 21.

Figure 21. 3D CAD model of an early design of fiber positioning stage.

The Y-axis consists of two standard 7 mm linear rails with a machined aluminum platform connecting them together. The Y axis platform serves as a mounting plate for the X-axis. The X-axis consists of a single standard 7mm linear rail with an aluminum plate mounted on the guide-way. In Figure 21 the motion control apparatus such as leadscrews had not been designed yet, however this serves as a basic design model for the stage apparatus. In theory the linear rails should be able to support both axis if only one rail per axis was used, however since a motor for the X-axis must be mounted on the Y-axis carriage, the more robust design with two rails was chosen. With the basic design for the stage done we turn our attention to choosing a motion

control method. Since the motors chosen are 200 steps/rev NEMA 8 standard stepper motors, we need to choose the leadscrew based on the smallest step (in X and Y) that is necessary. A 2-56 thread was chosen due to its small size and ease of availability. This corresponds to a X-Y full-step dimension of 8.92×10^{-5} inches or 2.3 microns. This value is obtained by finding the pitch of the thread and dividing that by the number of steps per rotation of the shaft: $(1/56 \text{ tpi})/(200 \text{ steps/rev})$. However many stepper drivers available commercially today offer “microstepping” functions. This allows the stepper motor to be driven in multiple phases at once allowing up to 32 steps per major (normal) step. If we use a $1/16^{th}$ step microstepping driver, then the smallest X-Y step size the stage can move becomes $8.92 \times 10^{-5}/16 = 5.56 \times 10^{-6}$ in which corresponds to 1.4×10^{-7} m or 0.14 microns. Even with the small step size available the linear speed of the carriages is not reduced drastically since the drivers can accept input step signals up to 16 kHz. This corresponds to a rotational speed of $16 \text{ kHz}/3200 \text{ steps/rot} = 5 \text{ rev/s}$. That is 0.089 in/sec or $\approx 5 \text{ in/min}$. Having a high linear speed is not necessary in this design, however a reasonable rate of position change makes the device more useful since the device can slew to a particular part of the focal plane in a decent amount of time.

A CAD model of this design is shown in Figure 22. In overview, the design is a 2 dimensional positioning stage inside a box. An optics housing is connected on top of the positioning stage housing, and allows the attachment of a C-mount camera and an F-mount lens.

This design requires precise machining to ensure accurate positioning ability. As a result, great care was taken in the machining of this device. In fact several custom tools were designed to accomplish this task; these tools are outlined in the Mechanical and Electrical Engineering Techniques and Procedures chapter. To start with the machining of this device the base of the X-Y stage was cut first. The base is the part every other part is built up from, and as such must be square and flat. For this part a 0.5 inch aluminum plate was flycut on both sides down to 0.375 inch. This takes any

Figure 22. 3D model of the assembled fiber positioner.

bow out of the metal and ensures that the base is perfectly flat, within machining tolerance. The piece was then CNC machined to size and drill holes and pockets for electronics were cut. A picture of the base with linear rails mounted is shown in Figure 23. A 1/4-20 hole is machined in the center of the baseplate to accommodate a standard tripod mount.

The sides of the housing were machined from a single block of aluminum. This allowed for the block to be squared before cutting the center section out for the case. This was done to ensure that the top plate and the bottom plate would be connected via flat and parallel sides with no deflection in them. Since the optics will attach to the top part of the case, it is necessary that the plane of fiber motion be parallel with the top plate as well. An image of the assembled device without the top plate or optics is shown in Figure 24. Parts have been labeled for the readers information.

The machining quality of the optics housing is also very important. A CAD model

Figure 23. The fiber positioner base and Y-Axis stage with motor.

Figure 24. Image of positioner without the the fiber attachment fixture.

of the part is shown in Figure 25. This part attaches to the top plate of the fiber positioner and holds the lens mount, camera mount, and beam splitter. The part was machined from an oversized stock of aluminum and flycut down to dimension to ensure squareness and flatness. The critical dimensions in this piece are the distance from the beam splitter to the fiber, and ensuring that the camera and lens mounts

Figure 25. 3D model of the optics housing. This part encloses the beamsplitter and the lens mount.

are 90° to the bottom face that attaches to the top plate of the positioning stage. The beam splitter is held at a 45° angle to the lens mount by a machined aluminum holder. This holder is held in the housing by means of an accurately bored hole. This allows for adjustment of the beam splitter. When the beam splitter is adjusted the final position is set by two 4-40 set screws in the back of the optics housing. The optic is mounted in place by use of RTV silicone. A 3 dimensional model is shown in Figure 26. The 2D lines in this figure correspond to the ray optics of the splitter. The Infrared light is reflected downward while the visible light passes through. The dots denote focus points.

After the optics housing was machined and the optics assembly completed, it was mounted to the fiber positioner frame. The completed device is shown in Figure 27.

Figure 26. 3D CAD model of beamsplitter assembly.

3 Electronics and Interface

The design of the electronics for the fiber positioner is fairly simple. The power electronics such as stepper drivers are separated from the main logic board and connected via wiring. The main logic board consists of a Microchip 328 pb microcontroller connected to a Wiznet 5100 Ethernet card, and supporting electronics. The schematic for the main board can be found in Figure 29.

For the communication interface a browser based system was chosen. On board the microcontroller a web page is stored and served over DHCP utilizing the Wiznet Ethernet chip. This web page allows for the input of X-Y coordinates, and displays the current X-Y position of the fiber. A image of this interface is shown in Figure 28. This scheme allows for the device to be used from virtually any distance from the user, provided the user has access to the network that the device is connected to. The software takes the input coordinates, stores them, and initiates a motion. Each step is controlled by a on-board timer/counter and interrupts. This method ensures

Figure 27. Fiber positioner device completely machined and assembled with a silicon CCD camera and fiber feed attached.

a constant step rate and therefore smooth motion between points.

In order to move the fiber to a new position, the software must know how many steps to send to the motor drives. This is calculated by finding the change in position for each axis and multiplying by a constant to determine the number of steps needed. First we find the change in position:

$$\begin{aligned}\Delta X &= X_{final} - X_{current} \\ \Delta Y &= Y_{final} - Y_{current}\end{aligned}\tag{7}$$

where (X_{final}, Y_{final}) is the new position we want the fiber to move to and $(X_{current}, Y_{current})$ is the current position of the fiber which is stored in memory. At this point the stepper motor direction is calculated. This is achieved by a simple if statement: if $\Delta X < 0$

Figure 28. A screenshot of the fiber positioner web interface.

then set motor to counter clockwise motion; else set to clockwise motion. We can then calculate the number of steps each axis needs to move by

$$\begin{aligned} X_{steps} &= \Delta X \times N \\ Y_{steps} &= \Delta Y \times N \end{aligned} \tag{8}$$

where N is the number of steps per unit (in this case we define these as steps per pixel a defined pixel size).

Once the step direction and number of steps have been calculated, a motion is initiated by writing a logical 1 to a global variable. The timer/counter interrupt is always polling this variable, and is waiting to execute the step motion. Once this variable is true, a step in each axis is executed until each axis reaches its final value.

Figure 29. Fiber positioner main board schematic.

Once each axis is at its final value the interrupt code switches the global variable to false, and another value is allowed to be entered by the user on the screen. In the browser-based environment, this manifests as a loading “wheel” in the browser. Once the motion is complete the web page is updated with the new location of the device and the user is allowed to type in a new location if so desired. In this way the user can never interrupt the device by sending too many commands at one time since they will just be ignored, however at the same time the machine benefits from the timer/counter interrupts stability features. The complete firmware code for this device is included in the appendix.

4 Conclusion

In this chapter the design and implementation of a fiber positioning system was discussed. Various techniques and algorithms were presented and constructed/compiled. A full test procedure along with data from several test setups of this device will be discussed later on in this work in the “testing” chapter. A image of this device with an Allied Vision Manta Camera mounted is shown in Figure 27.

CHAPTER IV

IMPULSE GENERATOR

1 Introduction

For optically testing materials it is sometimes necessary to impact the surface of the material in such a way that a wave will propagate; an optical signal can then be gathered and processed to gain information about the object/material. In this chapter the concept, design and manufacturing of a impulse generator coined the “Pinger” will be discussed.

2 Principles of Design

The design of the pinger went through multiple stages until arriving at a best case design. The design started out as a simple solenoid driven impact hammer. This design relied on a spring to provide the impacting force and used a simple MOSFET switch to activate the solenoid to return the hammer to its retracted position. A similar device was constructed using an analogous approach in which the solenoid provides the impact force, while a spring provides the retraction force. While these designs provided a considerable impact energy to the surface of a test piece, they did not however retract quick enough. As a result the hammer tip would stay on the surface of the material and cause secondary reactions. Sometimes the tip would “ride the wave” and stay on the surface for an extended amount of time; however more commonly the tip would “double bounce” and cause secondary impacts to the surface, which in turn cause secondary waves to propagate through the material, complicating the waveform and thus the analysis of the optical data. Figure 30 shows a diagram

of this process. In Frame 1 the hammer is about to strike a nominal surface. The hammer impacts in Frame 2, a surface deflection is incident and in Frame 3 the wave begins to propagate. The issue arises in Frame 4 when the surface rebounds, the hammer has left the surface a small distance but now touches again causing interference from the hammer in Frame 5. The hammer fully retracts in Frame 6.

Figure 30. Diagram of the qualitative behavior of surface material rebounding after a test impulse.

These issues prompted the conception of a new type of pinger. The main design criteria being that the pinger will both impact and retract under controlled power. In this manner the impact and retract specifications can be tuned to fit a particular application or material.

Physics of Surface Excitation by an Impulsive Impact

In all designs of the pinger the basic premise is the same: A mass is accelerated to a certain velocity until it impacts a surface. The mass is then accelerated the opposite way to clear the object. Ideally this “jerk” acceleration happens before the surface can interact with the mass.

In order to discuss the impact on the test surface we will define the energy imparted to the surface. When the impulsing mass impacts the surface an energy exchange will occur. In the case of a totally elastic collision, the energy will be conserved and no net energy will be imparted to the surface. In the case of a completely inelastic collision, all the impacting masses energy will be transferred to the surface and motion of the impacting mass will cease. To quantify these we define the change in kinetic energy is defined as

$$\Delta E = \frac{1}{2}m(v_f^2 - v_i^2) \quad (9)$$

where v_i is the velocity of the impacting hammer directly before impact and v_f is its velocity after impact. It can therefore be seen directly that the energy imparted to the material depends linearly on the impacting mass and the square of the impacting masses velocity. To model this, we first start by modeling the motion of the impacting hammer assembly.

First we assume a system with one degree of freedom. The device must be constrained to move on the X axis only. From basic kinematics we then have:

$$\vec{F} = m\vec{a} \quad (10)$$

If we assume the force to be constant over a given distance, then the acceleration is constant since the mass does not change giving:

$$\Delta X = Vt + \frac{1}{2}at^2 \quad (11)$$

Using this equation and assuming that 100 % of the mechanical energy is transferred to the struck object (test piece) we make the assumption that energy is conserved i.e. we assume an elastic collision with a rigid body. Thus after the impact the velocity of the impacting device is reversed with approximately the same magnitude

and the test piece velocity is zero. Using all these assumptions the equation of motion is

$$\Delta X = V_{impact}t - \frac{1}{2}at^2 \quad (12)$$

where

$$V_{impact} = t_{impact} \quad (13)$$

$$t_{impact} = \sqrt{\frac{2\Delta X}{a}} \quad (14)$$

Figure 31. Mass motion resulting from an elastic impulse versus time. The model neglects frictional losses and assumes an elastic collision of the impacting object with the surface.

Putting these two equations together the total motion (neglecting frictional losses) can be calculated. A graph of this motion is shown in Figure 31. The mass accelerates for the first part of the motion, impacts the surface of the test object at the point labeled “strike point” and retracts with acceleration. Since we assumed for this model a elastic collision, a sharp peak is observed. This is practice is not however true. There

is no such thing as a perfectly elastic collision, and therefore there is energy loss in the collision. In fact we *need* there to be loss, since we want to transfer energy to the surface that is impacted.

If we instead model the system in which all energy is *transferred* to the surface we get a drastically different system. The change we make to the above equations is simple: the velocity directly after impact is zero. Using this assumption we find the retraction equation of motion to be

$$\Delta X = -\frac{1}{2}at^2 \quad (15)$$

Figure 32. Mass motion resulting from an inelastic impulse versus time. The model neglects frictional losses and assumes an inelastic collision of the impacting object with the surface with total energy transfer to the material.

A graph of the full equation of motion is shown in Figure 32

The real physical system exists in the space between these two extremes. All energy will not be transferred to the surface, yet a perfectly elastic collision will not occur either. In point of fact either of these motion curves are acceptable for

Figure 33. Mass motion resulting from an impulse versus time. The model neglects frictional losses and assumes an nearly partially elastic collision of the impacting object with the surface with with slow retraction.

the operation of the impulse generator. The issue comes when the retract equation acceleration is too low. An example of this is shown in Figure 33. In this figure the retract acceleration shown is $1/4$ of the impulse acceleration, as a result there is a lag time before the impulse mass begins to retract. Depending on the amplitude of the surface waves induced, this can cause the double bounce effect and interfere with measurements.

The major take away in this case is that the retract acceleration should be equal to or higher than the impulse acceleration to ensure that double bounce does not occur. As a result a method of providing a constant yet controllable force that can be reversed quickly is needed. The impulse force needed could be obtained via a compressed air cylinder and was considered, however the retract time lag would be too large. This necessitates the use of electromagnetics as a primary design starting point.

Magnetics

This section will describe the magnetic analysis of motion needed to describe the pinger device. Many of the equations and methods for the following arguments can be found in *Electric Motors and Drives: Fundamentals, Types and Applications*. [11]

To begin with we need to define the force due to a magnetic field. From basic principles we start with the force on a moving charge in a magnetic field.

$$\vec{F} = q\vec{v} \times \vec{B} \quad (16)$$

If we expand this equation and use the fact that the product of charge and velocity is the same as the current through a particular length we can derive the force due to a constant current through a length of wire.

$$\vec{F} = IL\hat{u} \times \vec{B} \quad (17)$$

Where I is the current measured in amperes, L is the length of the wire in the field and $\hat{u}$ is the unit vector along the length of wire. For lengths of wire that are curved, the problem can be broken down into small segments of straight wires and integrated to find the result. In most applications we are only concerned with the force due to a wire that is held perpendicular to the field and in that case this equation can be greatly simplified. In point of fact in most engineering textbooks you will simply find that the cross product is so much a bother that the equation is simply written

$$F = iLB \quad (18)$$

since for engineering purposes we need to know the force from the field at right angles most of the time.

Until this point we discussed the force of the magnetic field, but how do we explain the field part of these equations? The $\vec{B}$ term in these equations is usually called the magnetic field in most first year physics textbooks, however in many cases this concept alone is limiting. A better way to think about $\vec{B}$ is as a density. If $\vec{B}$ is

a large number, than that means there is a lot of flux at that given point in space. That is, more field lines are condensed into that area. This is because there is a finite amount of energy present in the magnetic field. This field extends to infinity in all directions. However the density of the field tends toward zero as $\vec{r}$ approaches infinity. The key to designing magnetic apparatus is to bend and localize most of the field where we want it to be, thus maximizing the density in that part of space, and increasing $\vec{B}$ which in turn increases the force $\vec{F}$ that it can exert.

The above equations help find the force due to the magnetic field, however finding the field and field density is a more complicated problem. In most motor type engineering design cases, some type of permanent magnet is used to provide the field. As can be seen above, these equations assume a constant field, however permanent magnets only exist with dipole fields. This is difficult to solve analytically. A faster and reliable method of solving these types of problems is to use a computational method. In this work the magnetic field densities are solved using the Finite Elements Method Magnetics (FEMM) software.[12] This software is offered under an Aladdin Free Public Licence. Unfortunately this software is offered in a windows executable, as such it was run via wine under linux. [13] This software takes as its input a DXF file and uses a mesh generator to generate a mesh of the design. This mesh is then used to compute the magnetic field density and direction at each point in the meshed region. The user must select the material in each enclosed area. Each area is computed based on the material present in that area. As a result a system comprised of multiple materials can be modeled.

Figure 34 shows an output surface plot of the magnetic field density from a cylindrical rare earth bar magnet of grade N52. This analysis was useful in the design of a cylindrical impulse device which will be discussed later. It can be seen that the field lines are “compressed” inside the permanent magnet and the field density approaches 1 tesla. If we want to move this magnet due to a current through a wire, it is easily seen that a wire coiled around the magnet would produce a force parallel to the Y

Figure 34. Diagram of cylindrical bar magnet FEA analysis.

axis only when the coil is not positioned at the center of the magnet. When the coil is at the center of the magnet the forces on the wire cancel so that the total force is zero. However as we near the poles, the field lines start to include a horizontal (X axis) component. This gives a positive or negative net force on the coil depending on the direction of the current through it. If we imagine multiple coils placed $L/2$ down a shaft where $L/2$ is the length of the magnet, then pulsing each coil when the field lines are horizontal to each respective coil would cause a continuous force on the magnet in a linear manner. In this design however a control circuit would be needed to measure the position of the magnet and calculate the necessary current direction and magnitude needed to sustain linear motion. A device constructed from this approach is discussed later in the Machining and Mechanical Construction section of this chapter. This design was abandoned due to technical issues as well as a simple problem in the theory of the design. As the reader can see in Figure 34 there is no material to constrain the field lines. As a result the energy from the magnet is dispersed from the magnet to infinity. The field density is therefore low, and as

we discussed the force is proportional to the field density. This gives a device that must use a large current to provide enough force to overcome static friction let alone energy to provide an impulse. To fix this problem a “pole piece” cylinder of 1018 steel was constructed to provide a high permeability pathway to constrict the field lines and increase the density in the vicinity of the coil(s). This however proved to have mechanical issues that will be discussed later.

Another issue with the cylindrical pole piece is eddy currents. As a magnet moves at a given velocity close to a conductive material, a current is induced in the conductive material and thus energy is imparted to the material. Since energy in the system is conserved, adding energy to another material has the effect of slowing down the moving magnet, that is, the energy is converted from mechanical energy to electrical current and finally to heat. If the pole piece is stationary with respect to the moving magnet, then eddy currents will be produced in the iron when the magnet is in motion. This means even more electrical energy needs to be provided to overcome the force due to eddy currents. A way of solving this problem with the cylindrical design could not be conceived, so a new design was originated.

Figure 35. Diagram of linear motor stator FEA analysis.

A side view of the new design analysis is shown in Figure 35. This design consists

of two N52 grade rare earth magnets set in plane with each other yet opposing fields. This creates a magnetic circuit between the magnets. Two 1018 steel plates are added on the top and bottom of the magnets and held in place by standoff pieces (not shown in analysis view). As the reader can see, the magnetic field lines are predominantly contained inside the boundary between the two steel plates. This means the field outside the device is very low, nearing to background field densities. As a result more field is “useful” for force production. The X and Dot symbols in Figure 35 denote the winding of the coil. X denotes the current vector inward and dot denotes the current vectors outward. Since the field is also opposite, providing a current to this coil causes a force in the positive or negative X direction depending on the sign of the current. In addition to the increased force per current efficiency this topology allows the magnet motion to be able to be controlled by a single circuit. If the current is positive, the motion is positive, and visa versa. This simplifies the electronics and increases the accuracy of the motion as well. A simple estimate calculation from this model is to take the average field density and use the previous equations to calculate the force due to a particular current through the coils. For the final design of the pinger, the magnets are $5.1 \times 5.1 \times 1.2$ cm ($2 \times 2 \times 0.5$ inch) rare earth N52 magnets mounted in a case to hold them steady and provide a motion carriage. The spacing is 1.5 cm (0.6 inch) from each magnet to its respective pole piece. The field density is approximately 90° to the current at each point and has a average value of 0.3 tesla. If we input this data to 2 we get a value of

$$\frac{F}{I} = LB = 0.05m \times 0.3T = 0.1524 \frac{N}{A} \quad (19)$$

this value is the force per current due to a single wire in the field. We must also include the current traveling in the opposite direction which simply multiplies this by 2. We must also include the number of turns in the coil; in this case $N = 50$. This gives

$$\frac{F}{I} = 0.1524 \frac{N}{A} \times 2 \times 50 = 15N \quad (20)$$

There are two coils in the device so the final value is 30 Newtons per Amp if connected in series. Thus to achieve a target force of 200N the device must sink approximately 7 amps per coil, for a total of 14 amps if connected in parallel.

In this design the pole piece moves with the magnets. This eliminates eddy current loss and hysteresis loss. Eddy currents are caused when a magnetic field changes in a region of conductive material. Hysteresis loss is caused when a magnetic material becomes saturated, and then saturated in the opposing direction. Re-magnetizing the material in the opposite direction takes more energy since the material exhibits hysteresis. If this process takes place over and over, more energy would be needed to magnetize the plates, which exhibits itself as a reduction in force due to the current in the coils. However if the magnets move with the pole piece, as in this design, neither eddy current loss nor hysteresis loss are possible. This creates a much more efficient design, especially useful in applications that require large accelerations or high speeds.

3 Machining and Mechanical Construction

This section chronicles the mechanical construction of the several pinger designs over the course of this project. Both successes and failures are presented for learning and engineering purposes.

For the first versions of the design a very simple case was made to house a solenoid. This case was machined out of Delrin. The solenoid output shaft was connected to a rounded over piece of aluminum rod to serve as a hammer. This design proved to be functional, however the impact energy was low and the previously discussed issues prevented further research. One large issue with this device was the fact that heat would be generated in the solenoid, and since the solenoid was inset into a block of Delrin, there was little heat sinking ability. This caused the device to be able to only run for around 10 minutes. This device is shown deconstructed in Figure 36.

A similar device was constructed using a solenoid and a fulcrum design. This

Figure 36. A solenoid-based pinger prototype design using a standard commercial off-the-shelf solenoid and acetyl material that was designed and built to test composite materials.

device is shown in Figure 37. The main reason for experimenting with this fulcrum design was to remove the body of the pinger from the top of the setup. A large body on the top of the test setup can cause shadowing on the test piece and effect the results of testing. Another reason was increased force from the solenoid would be transferred to the impacting hammer, with the side effect being a reduced hammer travel. This device proved to be too flimsy and under-powered to even use in basic testing, and as a result this design path was abandoned.

The second major version of the pinger design was constructed out of 1" Schedule 40 PVC pipe. This pipe would allow a 1" diameter N52 magnet to freely move up and down inside the pipe. Coils of 28 gauge wire were wound around the pipe using a custom built coil-winder. The entire pipe was held in place by two machined aluminum clam shell clamps. A face-plate was made out of Delrin to allow a low friction sliding surface for the output shaft to rest on. A similar piece was machined from aluminum to serve as a end-plate in which a rubber bump-stop was attached to

Figure 37. A prototype fulcrum-style pinger design designed and tested for excitation of composite materials.

cushion the impact from the magnet on the retraction stroke. The shaft was made from a 1/4" round brass round, 20 TPI threads were turned on the end to allow the shaft to be attached to the magnet coupling block. Since in this design the magnet slides freely up and down the PVC pipe it is coupled to the output shaft using a piece of 1018 magnetic steel turned to 0.9" diameter. An illustration of this design is shown in 39. The coils in this illustration are shown as blue boxes. The force from the current in the coils is shown as the vector F interacting with the coil and the magnetic field (B). Optional backing iron is shown in black, however this was not used in practice.

Figure 38. CAD drawing of a pinger design using a cylindrical coil with an internal magnet.

Figure 39. Cross sectional diagram of a cylindrical pinger mechanism. The cylindrical magnetic field, coils and force vectors are illustrated.

This “pole piece” provides the retraction force for the shaft, while both magnetic attraction and compression strength provide the impact force. Since the impacting force is much higher than the return force (return force is only the weight of the rod plus friction) this design functions well. The major issue with this design, much like the original design was thermal control. Since the use of metals was precluded, plastics were used, in this case PVC. The coils were wound directly around the PVC pipe and the only place the heat could be conducted to efficiently was the pipe itself! As a result the pipe would heat, subsequently expand, and then sag under the weight of both the pipe and the coils. After a few test runs the pipe was bent enough to render it unusable.

Much consideration was given to different materials while maintaining the same geometry and basic design, however since metals were out of the question not many materials could fit the requirement. Another issue with this design was the fact that there was no magnetic shielding. This was discussed from a physics standpoint in the introduction. However from a purely practical standpoint, the fact that there was no shielding was dangerous. If a steel or magnetizable tool were left near the device, it

Figure 40. A prototype cylindrical pinger test device. The brass actuator extends to the left. There is an optical sensor for the shaft position mounted on this test device as well.

could fly toward the device. This could cause injury to people or objects around the device, and even to the device itself. During assembly the coils had to be rewound 3 times due to accidental tool incursions that caused damage to the enameled copper. A solution to this was to cover the coils with heat resistant electrical tape. However this only added to the thermal conduction problems and caused the device to overheat faster and to even greater temperatures. A method to contain the field was also tried. A steel pipe was used around the pipe, however asymmetries in the attractive force from the magnet to the pipe rendered the magnet unmovable in the pipe.

These downfalls led to the development of a new concept for a pinger which was designed to use a moving pole piece. The theory of this pole piece structure was discussed previously under the theory section. When the pole piece is a moving element, there is no relative motion between the pole piece and the magnets providing the field. This means the only relative motion is between the constant magnetic field and the wire coils. As a result no eddy currents are developed inside the pole piece. Therefore the device does not consume proportionately higher power at higher accelerations. The first model of this design was based on linear rod to provide the motion structure. A three dimensional CAD drawing along with the constructed device is shown in Figure 42. The device contains a carriage with 3 linear bearings riding on 2 linear rods. This provides the mounting structure for the magnets and the

Figure 41. Image of PVC tube destruction due to overheating. The actuator coil heats due to high current consumption, and the dissipation was inadequate to prevent damage to the PVC coil form in this test.

impacting rod. The magnets are mounted in pockets machined into the carriage, and high quality non-magnetic linear sleeve bearings are used. The coils are mounted on 4 more linear rods: 2 rods are used for the top coil, and 2 more for the bottom coil. The pole pieces are mounted on standoffs made from Delrin on the top of the magnet pocket motion carriage. This allows the pole pieces to move with the magnets while at the same time not interfere with the operation of the coils. This new pinger is essentially a powerful single-phase linear motor that allows the shaft to be actuated with a ΔX of ≈ 3.8 cm (1.5 in). The unique pole design for the pinger contains the magnetic field within the coil regions and eliminates hysteresis and eddy current losses.

This new “linear motor” type design functioned as intended, gave a very strong and defined “ping,” retracted to avoid a double strike, and did not overheat. The remaining issues are matters of construction. For one, if there is any misalignment in the 4 rods, the carriage will bind on either the coils or the rods themselves. For another, the design also uses a coil that is wound and then epoxied in place. This choice increases the number of windings possible in each coil by using the maximum

Figure 42. Side by side comparison of a CAD visualization and the constructed rod-based pinger. The coil is static and the flat magnet moves on cylindrical rails to drive the actuator.

depth of the coil form. However in practice it is difficult to produce, and the number of windings per coil are reduced as needed. As a result of these issues a few design changes were made for a second version of the linear motor pinger. The major one was to use linear rails rather than rods to guide the pinger.

The new second generation design of the linear motor pinger uses the same physical principle and electronics as the first generation. Adjustments were made were to the coils, enclosure, hardware mounting, and linear guide ways. The final design uses 7 mm linear ball bearing guide ways. This allows for smooth operation of the device and prevents binding of the linear axis. A single linear bushing is used on the output shaft to limit excessive lateral motion of the shaft during and after striking the test object. The coils are mounted directly to an aluminum plate that serves as the structures case. The coils were redesigned to allow ease of coil winding and prevent need for epoxy. These new coils are more robust and simpler to construct.

This final pinger design was tested and functions as intended. There are no issues

Figure 43. Complete intermediate pinger system. The actuator is separate from the power supply. It is triggered by a manual button-push.

with binding. In addition, the coils are capable of providing more force in this version than in the first one due to the fact that they are rigidly affixed to the case.

4 Electronics

The first versions of electronics to control the pinger were simply fast MOSFET switches that would control current through a solenoid. Since the retraction was handled by a mechanical spring, simply turning off the current would retract the hammer. This simple circuit was constructed on a small PCB and hand soldered. The design of a circuit to control the retraction as well as the impact force was more complex. The first versions of the cylindrical magnet pinger required two separate coils to be controlled with both positive and negative currents. This necessitated the design of a dual H-Bridge circuit capable of high peak currents. Originally high

Figure 44. Final pinger mechanical design. The actuator in this version moves on linear rails.

current MOSFETs were selected for this task. The MOSFETs were arranged in an H-Bridge design and protection zener diodes were added across the coil outputs to clip any over voltage from the high current pulses. The H-Bridge schematic is shown in Figure 47. This circuit uses both P and N type MOSFETs. Issues were present in this design that baffled efforts to fix. In initial testing the MOSFETs would behave fine for a few test impulses however after around five to ten minutes of running, they would explode, smoke, or just catch on fire. In a particular test the current was enough to melt the leads to the MOSFET in a spectacular explosion. See Figure 46.

These failures caused a design change that turned out to not solve the problem. Initially it was thought that the devices were melting because of the die being rated too low. However this turned out to be wrong. The MOSFETs were swapped out for larger package IGBT's and the circuit was made again. Again the IGBT's failed at

Figure 45. The final pinger system setup including power supply, support electronics, and actuator.

around 10 minutes runtime. It was found that the problem was with heat sinking the devices. While the device die could withstand 100 A continuous and 500 A peak, the device package could not. It was found that heat in these types of circuits needs to be conducted away using several pathways. The die in these circumstances produces enough thermal energy that the package pins provide a heat sinking pathway as well as the heat sink pad. This means that heat is being wicked away by the tracing on the printed circuit board as well as the heat sink itself. If the heat sink is undersized or of a low conductivity the PCB traces can get hot enough to melt. This failure mode indeed happened on a couple of the IGBT-based boards. To fix this problem, copper pours were used instead of tracing for the high current traces, this reduces the resistance of the traces, but most importantly gives a greater surface area to conduct heat away. Another issue with the first MOSFET based circuits was that the package

Figure 46. A MOSFET destroyed due to failure to provide proper heat conducting surfaces and large enough PCB traces.

Figure 47. Design schematic of H-bridge MOSFET output circuit.

was rated lower than the die. Only upon close inspection of the data sheet was it found that the package could sink around one quarter the current of the die. The same circuit was redesigned with the larger tracing, bigger heatsinks per MOSFETS, new MOSFETS with high rated packages, and a fan to provide airflow across all heatsinks and MOSFETS. This board was tested and is used as the final power and control board for the pinger.

Figure 48. Diagram of final pinger controller design.

The basic control circuit design for the pinger is an H-bridge controlled via isolation hardware by an microcontroller. Figure 48 shows a diagram of the control circuitry of the pinger. The pinger coils are driven from the high current pulse circuit containing MOSFET transistors capable of switching high current at microsecond timescales. To provide the necessary pulse power (≈ 4000 W) it is necessary to buffer the power input. In this design we use a capacitor to provide the energy to the motor coils during the pulsing. This allows the device to operate off of a single 110 VAC outlet with a standard 15 A circuit breaker. To control the pulses the controller incorporates an ATXmega128a4u microcontroller for timing the phase changes and duration of pulses. (the longer the overall pulse length the greater the impulse) a sequence of timed pulses is stored in firmware on the microcontroller and triggered by an external Input Trigger pulse provided by the user. In this way the device operates as a single electronic input:single mechanical output device.

The power supply for the pinger is contained within a steel enclosure. A large 160 V 100000 μ F capacitor is charged via a custom designed linear regulator circuit. This circuit takes in line voltage (120 VAC) isolates the power via a 1:1 power transformer and rectifies the output. The output is then regulated with a suitable NPN transistor to 100 V. The capacitor is then charged through 100 Ω resistor to limit the current draw on the input circuit. In this way a large output pulse can be trig-

Figure 49. Image of inside pinger control electronics enclosure before complete assembly.

gered without loading the input power line significantly. The circuit contains fuses for safety.

The pinger coils are then connected via military grade Amphenol connectors and wiring. The power supply housing includes a BNC connector for the user to input the ping control pulse. If the user inputs a 5-12 V pulse than the pinger will activate and initiate a ping. Alternatively the BNC port can be wired to use a push button controller to initiate the ping sequence. In this manner a user can initiate a ping by pressing a button. A image of the inside of the power and control box during assembly is presented in Figure 49.

5 Testing

Each of the various pingers that were designed and built as prototypes was put through a series of tests to gauge their abilities. The first few designs had multiple unrecoverable failure modes that prompted major concept and detail design changes as discussed previously. The final design was put through a more rigorous test to ensure that the electronics and mechanics were robust and capable of enduring many hours of use. The device was set to actuate in one second intervals for one hour. The device was monitored to ensure that the impulses were consistent and that heat was being dissipated properly. The heat sinking in the electronics enclosure on this final design was shown to provide adequate thermal sinking and did not rise above 65 °C during the test. The mechanical device was also monitored to ensure that no mechanical failures occurred as well. Upon inspection after the test the linear rails were checked for free motion and no binding was found. This signifies that the device stayed within tolerance during use. Care was taken to ensure that all wiring stayed properly affixed during the test.

Through the process of designing and building the pinger much was learned about magnetism, mechanical design and high power circuitry. In the end, the final device is stable, lightweight and strong. The device is capable of many hours of service.

CHAPTER V

TEST PLATE FABRICATION

In order to investigate the optical detection of material defects in carbon composite materials, a sample of bonded composite sheets was created. This plate was made with a precisely defined defective bond region which was not visible by superficial inspection. The purpose of this study was to see if by applying the technologies developed in this research that region could be identified.

Figure 50. Manufacturing setup using a surface plate to ensure even and level force during cure.

In order to allow the test plate to cure evenly it must be held on a flat surface while it cures. While the floor of the lab was used in early trials, its irregularities could potentially be mirrored in the final product, and for the test described here a machine shop surface plate was chosen. A surface plate by definition is flat to within a few wavelengths of visible light, as well as being a structurally rigid and convenient surface to work on. The massive granite surface plate was covered with a stainless steel sheet to protect it from any epoxy leakage. A picture of the setup is shown in Figure 50

The carbon fiber pieces were purchased from Dragon Plate. One piece of matte finish and one piece of gloss finished were used. Both pieces were 30 cm wide by 60 cm long and 0.16 cm thick ($12 \times 24 \times 1/16$ inch). To create the bad bond area a mold release compound was applied to a defined area so that the epoxy would not adhere to the composite. This type of failed bond is known as a “kissing bond”. It is not visually evident, but it allows the two surfaces to move laterally past one another, or even to slightly separate. Obviously, a failure of this type greatly reduces the strength of the joint between the materials. The area defined by the mask was halfway down the sheet’s length, and halfway across its width, covering an area 5×14 cm (2×5.5 in).

A mask was created to restrict the mold release to this chosen location. Two sheets of 0.076 mm (0.003 in or “3 mil”) stainless steel sheet that had been used for a solder paste masks were cut to the correct size, taped together and laid out on the unfinished or “matte” side of the carbon fiber material. This setup is shown in Figure 51. One corner of each carbon fiber plate was marked with a registration mark, to ensure that the pieces were placed together in the correct configuration. By placing each marked corner together, masks would line up the mold release or bad bond areas for each of the two plates such that after assembly they would be on opposite sides of the epoxy layer.

Frekote 1711 mold release was then applied to the surface of the sheets by spraying the area. Two coats of release agent were applied with a drying period of 60 seconds

Figure 51. The mold release mask setup reuses a stainless steel solder stencil material cut to size.

Figure 52. Mold release application area. The defective bond resulting from this treatment is midway on the long dimension of the material and extends halfway across.

between coats. The mask was then removed after waiting approximately 5 minutes. This same procedure was followed on both the top and bottom pieces of composite material. The work in progress is shown in Figure 52.

Loctite 9340 Hysol epoxy which requires an epoxy-to-hardener ratio recommended

by the manufacturer of 1:1. It is supplied in matched single tubes of each component, and one pair covers the entire bond area. The epoxy and hardener were mixed directly on the carbon fiber surface using a small “Popsicle Stick” style tool as seen in Figure 53.

Figure 53. Epoxy mixing technique.

After a thorough mixing, the epoxy was spread out on the surface of one of the carbon fiber plates. A spatula was used to evenly spread a thin layer of epoxy on the surface. When forming parts from carbon fiber it is important to use as little epoxy as possible. Epoxy itself does not add strength to the material, and is the most dense component of a composite material. Care was taken to not over apply the epoxy, and create a smooth layer of consistent thickness throughout the piece. A photograph of the bottom piece after epoxy application is shown in Figure 54.

Taking care to place the plates together accurately, the registration marks were lined up and a suction cup device was used to lower the top plate perpendicularly onto the bottom plate ensuring accurate alignment of the plates. This is shown in

Figure 54. The application of epoxy to the bottom carbon fiber plate.

Figure 55.

In industry it is common practice to use a vacuum bag to cure the epoxy. This provides even clamping force on the part as well as causes any excess epoxy to be purged from the joint. Since in this project we did not have access to vacuum bag equipment, another system had to be used. In this case a simple matter of using gravity is the best solution.

After aligning the top plate, even pressure was applied to the plates using two large pieces of aluminum stock. This is shown in Figure 56. The aluminum stock was left on the part during curing to ensure constant and even pressure throughout the cure time. In this case the epoxy called for a cure time of 12 to 24 hours. The piece was left to cure with the aluminum blocks on it for 16 hours, checked and then allowed to cure for another 6 hours. After the plate assembly was cured, the edge of the work piece was marked with white-out fluid in the location of the bad bond area,

Figure 55. Suction cup based handle used to align the top plate to the bottom plate of the carbon fiber composite plate.

and the protective film was removed from each finished surface. The finished result is shown in Figure 57 ready for optical testing.

Figure 56. Curing carbon fiber composite. Aluminum blocks provide weight for even pressure during curing.

Figure 57. Final composite test piece. On the surface seen in ambient light there is no indication of a poor bond.

CHAPTER VI

MECHANICAL AND ELECTRICAL ENGINEERING

TECHNIQUES AND PROCEDURES

Much of the work presented in this dissertation was produced in the lab. Because of the bespoke nature of the circuits and mechanical apparatus, custom machined parts and printed circuit boards were designed and manufactured. This chapter outlines the basic methods, tricks and tips that were used and/or discovered while working on these projects.

1 Mechanical CAD Packages

In the beginning of this work, most of my experience had been with 2D drafting tools. For the first parts made to support the pinger work LibreCAD [14] was used for CAD drawing. However as designs became more and more complex the need for a solid modeling package became apparent. Since open source projects were preferred I installed Freecad which is a open source project based in Python. This application is very similar to Autodesk Inventor [15] or Solidworks [16] in its user interface (UI) design. Freecad even includes a CAM package for programming CNC machines. However a major issue with Freecad [17] is that it is still under heavy development. At the time of this writing the version is 0.16 and includes a warning that the software may not be stable. Due to the issues with stability and on the suggestion of a colleague I started using a “cloud-based” application called OnShape [18]. OnShape is not open source software, however for the research purposes only, it offers a free licence. This proved to be a more useful tool than Freecad simply because it is stable and it includes many more standard drawing tools. There is however one flaw:

OnShape does not include CAM software. For parts drawn in OnShape the university licence for MasterCAM [19] was used to create the toolpaths for machining the parts. In responses to these considerations and a desire to streamline the part machining procedure I switched my main development operating system from Linux to Apple OSX and installed Fusion 360 from Autodesk. Since I have need for Linux software I installed a virtual machine to allow the use of both OSX and Linux simultaneously. Fusion 360 also includes a research licence and includes full featured CAD and CAM packages. The CAM software is not separate from the CAD software allowing for changes to be made easily without needing to redo the entire CAM file if a simple change is made to the part. This is essential in a research and development setting. The use of this software increased the productivity, decreased machining errors, and overall contributed to this work positively.

2 Basic Machining Practices

At the beginning of this project my machining experience was limited to manual lathe and manual mill operations. During the course of design and manufacturing of parts for these devices I became skilled with using CNC mills as well as CNC lathes. Most of the numerical control code (G code) for these parts has been programmed via Autodesk's Fusion 360 software [20]. This software has created a great workflow for both design and manufacture of parts. Previous parts were programmed with MasterCAM CAM software. While this software is industry standard and works well, it lacks the ability to quickly make small changes to parts [19].

3 Special Tools

Several special tools were constructed to aid in machining parts for this work. When this research was started the Physics machine shop was equipped with a Haas TM1 Toolroom CNC vertical Milling center, a manual Bridgeport Mill, a Clausing 12" manual lathe, a 14" manual lathe, and other supporting tools.

In order to create very flat surfaces it is preferred to use either a flycutter or a grinding wheel. Most parts in these projects were made from aluminum, which is a poor candidate for grinding. To solve this issue a flycutter was designed and machined from steel.

Figure 58. Image of first flycutter design.

The first flycutter design used a small 7mm insert boring bar set into a steel arbor. This design is shown in Figure 58. This cutter produced a flat surface however the bar had to be held very close to the arbor or modal vibrations would cause surface roughness on the machined surface. A redesign of this tool consisted of a 25mm lathe tool holder attached to a machined steel arbor. This tool is shown in Figure 59. This tool has much higher rigidity and can machine up to 7" diameter cut with no detectable surface roughness.

Other various tools were constructed to produce parts as well. A coil winder unit consisting of a stepper motor to control the position of the wire in order to achieve

Figure 59. Image of final flycutter design.

a consistent winding was designed and used to wind the coils for this project until the acquisition of a CNC lathe was made. Coils were then wound on the CNC lathe. Another tool that was made to complete this work was a slitting saw arbor. Use of a slitting saw was necessary to machine the coilforms for the pinger since they have large undercuts for the wire to rest into. The standard slitting saw arbors did not provide enough clearance for the part to be clamped in the vise, therefore a new slitting saw arbor was machined on the lathe to hold the saw with down to 1/8" clearance available.

During this work various cutting tools were used. At first standard 2 flute end mills were used for cutting most parts. However with experiment it was found that 3 flute solid carbide coated endmills performed much better in both speed and surface quality.

For machining many parts on the fiber positioner (nameley the case) deep cuts

were necessary. To provide a constant clean surface finish a reduced shank endmill was used. This endmill was 3 flute with 1" of flutes but can have a reach up to 3.5" if held in a ER16 collet. The use of insert mills was also used heavily to provide a good surface finish at a high material removal rate. To provide undercuts for the fiber positioner case a T slot cutter was used. This allowed the mounting holes to be a larger size while still providing clearance space for the wires in the case.

For parts that are thin and flat it is important to hold the stock material in the vise with as little force as possible. If the stock is compressed too much the stock will bend, the dimensions will not be on size, and the part will warp. To cut thin plate stock like this there are two main holding methods. One holding method is to use double stick tape. For many parts made from aluminum 3M brand 410M tape was used to mount the part to a flycut flat plate in the mill. This provides hold down force and keeps the part flat while machining. Tape however can lose its effectiveness when wet, as a result flood coolant cannot be used. To fix this problem a mist coolant nozzle was installed on the machine. The use of mist coolant does not effect the holding force of the tape for low runtime parts. For longer runtime parts however such as a part that is made from stainless steel where the feed speed is much lower another method must be used. For these parts a vacuum hold down plate was used. This allowed the use of mist or flood coolant and provided a flat surface for even steel parts.

Through this work it was found that using flood coolant and mist coolant at the same time produces a better machined surface since the coolant can clear the chips much faster with the added air blast. Another coolant nozzle was added at 90° to the main coolant nozzle. This along with mist coolant produced the best results. A picture of the mill running a part using all 3 coolant nozzles is pictured in Figure 60.

Figure 60. CNC mill with mist and flood coolant apparatus.

4 PCB CAD Packages

When this work was started KiCad [21] was still a work in progress. At the start of the pinger work my go-to PCB CAD package was EAGLE-CAD (now owned by AutoDesk) [22]. However the nature of lab funding restricted the use of software to low-cost software that could be used commercially. Since EAGLE's licence was a student licence, it could not be used for lab work. These considerations combined with the lack of professional features and small board sizes/layers in EAGLE student prompted the search for an Open-Source full-featured electronic design suite.

During this search KiCad was downloaded, installed and tested. This software showed much promise, modeled after various commercial tools the interface tried to give the designer useful features. However the user interface and graphics had many bugs that prevented further investigation into the software at that time. The

Figure 61. Screenshot of KiCad PCB editor software.

first software that seemed to have immediate usability was gEDA suite [23]. This is an open source software written primarily for Linux operating systems. It consists of a schematic capture program known as gschem and pcb editing software known simply as “PCB.” Libraries however were lacking. This is not surprising for open source software, and so I wrote a few utilities in C to make footprints and schematic components for the software. The gEDA suite was used to produce the manufacturing files (GERBER files) for the first Pinger and Data Acquisition boards (24 bit). This software was still lacking many features, but for simple things worked well. Major issues for PCB design – creating copper pours, sizing vias, and switching between trace sizes – were all complicated processes in software that would add time to the creation or modification of a project. Design reformatting or tweaking was also very difficult since moving a component would mean retracing the entire ground plane and/or traces.

During this phase of design work I would periodically check the KiCad website for updates. In late 2015 I found that CERN had begun to invest in KiCad and provide resources and mainly manpower to devote to development. This promised more features and a better graphics interface. Because of these changes KiCad became

a viable solution. KiCad's new features are almost as robust as industry standard solutions and thus it became the main CAD software used for the design work for this project. Features such as push routing and 3D modeling with standard CAD output are very useful in designing complex boards. The ability to use up to 16 layers also increases the options for design. Differential tracing also allows easy impedance control for low voltage differential signaling traces. A screencapture of KiCad running under Apple OSX is shown in Figure 61.

5 Printed Circuit Board Design Practices

Much was learned during this project about PCB design practices. Most of the issues encountered during this work centered around soldering. Many boards were ruined in the process of figuring out package footprint sizes and the correct amounts of solder to use. Because noise is such an important part of these designs, originally all designs used surface mount packages on the small end of the spectrum. Most passive surface mount device (SMD) case codes were 0603 with the smallest being 0201. These however are very small and difficult to solder. It was found that when possible it is best to use the largest part or trace available. Unless the design will not electrically be viable, large devices are almost always easier to solder and place. For most of these circuits reflow surface mount components were used. Through trial and error a technique for developing CAD footprints for small pitch packages was developed. Figure 62 shows a poor footprint design. Unfortunately this is the standard footprint for TQFP packages in KiCad and most other programs. However it was found that solder bridges form *under* the pins since the solder is applied too far inward. If we move the pads out farther, to the point where the edge of the pad just touches the inner part of the part leg, we get much better results. Since there is no solder under the leg now, the probability that a solder bridge will occur under the pin is very small. Solder bridges on top of pins are able to be dealt with by using a soldering iron to reheat and flow the solder. Figure 63 shows a better way of designing a footprint for

a TQFP package. Dimensions should be taken from each datasheet when designing a footprint for fine pitch devices.

Figure 62. An example of poor footprint design.

Figure 63. An example of better footprint design.

Preventing bridging in general is simply a task of figuring out the proper amount of solder paste to apply to the pads. When a board is manufactured, a solder paste

stencil is also made. The stencil is placed over the circuit board and solder paste is applied through the stencil holes. CAD packages have the ability to change the size of the solder paste stencil pads to vary the amount of solder paste being applied to the board. Figure 64 shows a footprint where 90% of the pad surface is covered with solder paste. Trial and error has shown that with stainless steel 3mil solder paste stencils 70% to 90% works well. When soldering QFP or other fine pitch packages it is best to avoid using too much solder paste. Less is more with solder paste!

Figure 64. An example of a solder paste stencil with pads filled to 90%.

For producing the best printed circuit board designs it is advisable to use the largest tracing available. In several of the beginning designs, tracing were used that that were too small. In these cases even if the current requirements are not dictating a large trace size, for manufacturing purposes a larger trace is more reliable. It is also best to use 45° angles or arcs for tracing corners. Using a 90° corner creates a sharp

edge on the corner of the trace that is difficult to etch. No etch process can create a sharp corner, as a result the trace will be constricted in that location, causing a higher impedance, and worst for small traces possibly causing a break in the trace. When possible use largest trace possible.

A good gauge to use for figuring trace width is to use the American Wire Gauge chart (AWG). Later designs in this work used an extrapolation of the current ratings of wires to figure the current rating of a trace. For example if we know that a trace is 1 oz copper pour at 6 mil width then we know the depth and width of the trace. It is then possible to figure the cross sectional area of the trace. By matching the cross sectional area of the trace to the corresponding wire gauge, the rating current can be approximated. This method has worked well for both logic and power boards.

For high frequency designs it is important to avoid noise coupling. For sensitive analog front ends, elimination of cross talk from the digital side of the board is necessary. There are several ways to avoid issues from cross talk. One way is to run parallel buses at near to 90° to each trace. If the traces do not run parallel then not as much energy is coupled between them. Another way is to make certain that each trace has ground plane pour between them. This provides a ground path for each transmission line and reduces crosstalk by a sort of shielding effect. The best effective way of reducing cross talk and digital noise is to combine all these methods and including a “Faraday shield” . By not running traces parallel to each other, adding ground planes between traces and running traces in the inner layers of the board with via stitched ground planes on top and bottom we have developed designs that eliminate cross talk up to significantly high frequencies. Even at low frequency these design techniques can be used to lower noise and improve the overall system signal to noise ratio.

6 Reflow Soldering

The general method of reflow solder is

Figure 65. Image of solder reflowing setup utilizing a common household toaster oven with a precision temperature controller.

1. Clean the board. Ethanol works best.
2. Place the board on a flat surface
3. Use the solder stencil to apply paste.
4. Place the components on the board using pick and place or tweezers.
5. Place under stereoscope or A USB microscope to inspect placement.
6. Place in oven and heat.

For this work a simple toaster oven connected to a PDI controller was used. The temperature profile from the chip manufactures was used as a guide to set the temperature curve. This setup is shown in Figures 65 and 66. A special device was designed to place large fine pitch parts and will be discussed later in this chapter.

The most important steps in this process is to place the parts properly and to not smear the solder paste. It is important to remove the solder stencil slowly and carefully to allow the solder paste to fall through the stencil openings. If the stencil is removed too quickly some of the paste may stick to the stencil and not be deposited on the board.

Figure 66. A board being reflow soldered in the toaster oven setup.

7 Special Tools

Manual Pick and Place Tool

For placing large parts of pitch over 0.8 mm a simple technique with a stereo microscope and tweezers will suffice. However for placing smaller pitch components the combination of handshake and hand angle causes it to be virtually impossible to place parts perfectly the first time. Normally it is okay to slide a part a bit after

placing it to center the part, however in the case of fine pitch parts this may cause solder paste to be smeared onto adjacent pins, causing solder bridging that cannot be tolerated. As a result a method of placing the components without shaking becomes necessary. Most of the error associated with the hand is caused by the ergonomics of the tweezers. The part is never placed directly downward, but at a slight angle. As a result when the tweezers are removed from the part, the hand moves back at an angle causing the part to skew. A simple fix for this is to position the part from directly overhead. If the part is held at a perfect 90° angle to the board, and released onto the footprint when it is aligned, then the part should not move after placement. This was found to be next to impossible to do using tweezers by hand, and as a result design and construction of a pick and place machine was initiated. The common way of holding a part in commercial pick and place machines is by use of a vacuum pickup head.

A simple vacuum pickup head can be obtained from Ebay [24] sellers for around \$5. One of these kits was purchased and experimented with. The tool works well for large parts however hand-shake is still a dominating factor in placing small pitch devices and the tool did not substantially increase the accuracy of part placement. Some thought and design time was given to the creation of an automated pick and place machine however this proved not to be a time saving option. For an automatic pick and place machine to operate it must be programmed, somewhat like a CNC mill, and loaded with the different parts in known locations. For one-off boards such as research PCB's the setup time would be greater than the manual placement time! As a result the design was downgraded to a manual "Jig" tool that would assist in device placement by constraining the motion of the vacuum head and removing most of the users handshake.

The manual pick and place jig was constructed out of 7 mm linear rail. A base was made from a 12.7 cm (5 in) square tube of aluminum to which pillars of 2.5 cm (1 in) square aluminum rod were attached. A "X" axis linear rail was attached between

Figure 67. The manual pick and place machine designed and built for the completion of this work.

these pillars. A X axis carriage was constructed from PLA plastic on a 3D printer. The Z axis linear rail is attached perpendicular to the X axis onto the X axis carriage. The Z axis linear rail holds another 3D printed carriage with provides an arm to hold the vacuum pick and place head. In order to constrain the z axis to default to a high position, a spring connects the Z axis carriage pick and place head assembly to the top of the Z axis. This provides the lifting force for the pick and place head. A 1/4-20 tapped hole was drilled in the left upright pillar to provide a USB microscope camera mount. The microscope provides visual feedback to the user for accurate placement of parts. A picture of the device can be found in Figure 67.

8 Conclusion

During the course of this work many things were learned and devices made to make prototyping simpler, faster, more accurate and in some instances, possible. Devices were created to allow lab production of high quality multi-layer circuits and assist in pick and place operations as well as the manufacture of one of a kind precision hardware with CNC mills and lathes. A project of this kind cannot be achieved without developing in parallel specialized tools, mechanisms, and methods. For many parts of this project, industry standard tools were either not obtainable or too costly. In other cases the industry could not supply the necessary tool or product needed to achieve the goal on a laboratory scale project. For low rate prototype production, it is cost prohibitive to have a circuit board assembled by a factory. Even when it is taken into account the time spent designing specialized tooling for this project, the cost of outside production would not have allowed for this project to be completed. In many cases such as this it is important to have a small scale prototype machining and manufacturing capability in scientific and engineering research departments.

CHAPTER VII

MEASUREMENT AND ANALYSIS

In the previous chapters the design and implementation of various devices was presented. In this chapter we will build a complete picture of a test setup in which the devices are used for science. Test data will be presented and analyzed comparing and contrasting the effectiveness of the different tools and methods. We will begin by discussing an InGaAs photodiode fiber-coupled sensor that is the detector for basic fiber positioner measurements. From there we will move on to fiber based measurements using the pinger as an impulse device. These measurements will be contrasted to standard measurements made with a commercially available high speed camera system. The advantages and disadvantages of each system will be shown and discussed.

1 Photodiode Noise Measurement

To use a photodiode for measurement purposes it is useful to know the dark current noise characteristics. A simple experiment was performed to analyze the noise floor of the InGaAs photodiode used for these measurements. First the photodiode was capped so that no ambient light could enter the sensor. As a precaution the lab lights were also turned off because ambient room light can weakly couple into a cladded and sheathed optical fiber. A data set was then taken using the previously discussed DAQ system. This measures the response of the device and its analog-to-digital conversion (ADC) components as a single module providing a DC background offset and a noise floor of the photodiode while connected to the amplifier and output wiring. The circuitry for this photodiode amplifier device is described in previous

Figure 68. Fourier analysis of intrinsic noise from an InGaAs photodiode.

work [25]. As a result there is some referred noise from the amplifier in the signal. For these purposes however this is a good thing since we want to see the noise floor of the complete device, not just the photodiode on its own. Figure 68 shows the no-light noise signal in frequency space. From this data we can quickly see that the noise floor is approximately 1×10^{-6} V. This value corresponds to a dynamic range of 133.92 dB when using a 5V peak signal level. This is $(133.92 \text{ dB} - 1.76)/6.02 = 21.9$. Using this result we can see that the 32-bit device which has a noise floor of 1×10^{-7} is more than sufficient to resolve the optical signal. Also of note in the Fourier analysis is that on a linear scale there is a notable increase in noise at low frequencies, but that even below 10 Hz the device is measuring a light signal to a few parts per million. On this scale, a fiber end exposed without lenses to a daytime scene produces a signal of the order of 1 V.

2 Basic Fiber Measurements

The measurements in this section are designed to test the effectiveness of a basic fiber optic single pixel camera. The previously discussed DAQ systems designed in this work are used to make time based optical measurements. Different types of measurements will be presented. The effectiveness of the system for each measurement will be discussed.

Materials Testing

This set of experiments consists of using a single fiber device to measure material defects and properties.

Static Vibration Study

In this test the test piece is vibrated at a specified frequency. The surface is then imaged by the single fiber system and plotted in frequency space as well as in time. The first challenge is to vibrate the test sample without interfering with its natural vibration characteristics. This can be done simply by fixing a magnet to the test sample itself and bringing a coil in close proximity to the magnet. When an oscillating current is applied to the coil, the induced magnetic field will cause the magnet to follow the oscillation, thereby vibrating the test piece. This type of method has been used by T.D.Rossing et. al. [26] for acoustic measurements. The following recipe describes the process of this type of measurement.

1. Setup a frequency generator.
2. Setup magnet on surface of test piece.
3. Setup motion coil. Custom 50 ohm coil.
4. Vibrate the test piece at a particular frequency.
5. Optically observe the test piece.

6. Map the surface with the fiber.
7. Analyze the captured data.

In order to focus the part under test in the camera frame, we use use a light source to send light down the fiber through the lens. When this lens is in focus and dot will appear on the surface that the system is focused on. This allows the user to select the proper alignment and focus of the signal fiber system. Once the lens is properly focused on the target, a selection of measurements is taken. In this test the center of the test piece was chosen for to look for surface deflection due to the vibration. A frequency generator was connected to a standard audio amplifier. This was then connected to a specially constructed coil of wire with impedance of 50 ohms. This coil was attached to the test stand and positioned near a magnet which was double-stick taped to the test piece. In this first test the test piece was held down to the fixture with brackets, however this caused the too much vibration damping and the coil was not able to induce a vibration in the piece. A solution was found by using double stick tape to fix the test piece to the fixture. This allows the piece to freely vibrate while in test. The goal of this test is to show that the system can optically observe a deflection of the surface using an InGaAs photodiode connected to the 32-bit DAQ system. An important fact of making this type of measurement is that the light level be as high as possible for a given scene. In the first attempt at this measurement an low-noise low-power IR DC source was used to provide the light. However, the light level, which is weak visually but peaks in the near-IR, not provide sufficient power to detect the fractionally small oscillation above the noise floor. As a result the setup was changed to use a conventional incandescent desk lamp to illuminate the scene. This adds a 120 Hz power AC component which is then separated in Fourier analysis. The test plate was vibrated using the magnet and coil at 144 Hz. A set of data was then taken from the photodiode device using the 32-bit DAQ without the vibration to set a background baseline signal. Another set of data was taken while

Figure 69. Fourier analysis of InGaAs photodiode measurement of a plate vibrated at 144 Hz. The fundamental and harmonics of the excitation frequency modulate the detected visible scattered light.

vibrating the plate. A fast Fourier transform (FFT) was then performed on the data and plotted in Figure 69. It can be seen in this analysis that the vibration of the plate is clearly captured and well above the noise of the signal. In fact the optical signal noise is above the digitization noise floor of the data acquisition system in general. In the vibration signal (plotted in red) it can be seen that the harmonics are also visible above noise. Even mixing elements are shown above noise in this data (circled in blue). Since the illumination source used was an AC incandescent lamp running at 60 Hz (American wall outlet) we see the lamp power oscillation at 120 Hz and consequent harmonics as well.

Another test of this type was done using a silicon photodiode with a visible spectrum DC LED array lamp as a light source. As with the InGaAs IR sensor, a lens

Figure 70. Fourier analysis of silicon photodiode measurement of a plate vibrated at 53 Hz. The fundamental, second, and fourth harmonics of the excitation frequency modulate the detected visible scattered light.

system imaged the scene onto the end of an optical fiber, and the light passed through the fiber to the photodiode which was closely coupled to a low-noise amplifier, the same amplifier circuit as used with the InGaAs sensor. The signal from the plate vibrated at 53 Hz was digitized by the same 32-bit DAQ.

The LED array used in this test is DC-powered and has some intrinsic optical noise, but there is no major component of lamp power oscillation present in the signal. However the response from the silicon photodiode while above noise is not significantly above the noise floor. An FFT of this data is shown in Figure 70. In fact the 4th harmonic is almost not distinguishable from the noise in this data at a level of 10^{-5} . This is largely because compared to the InGaAs diode this sensor and LED source has a an order of magnitude larger noise component in its background

signal. This observation is consistent with previous work of Hay et al [25] which showed the improved dynamic range and response of working in the infrared region to which the InGaAs sensors are most sensitive and low-noise ambient daylight and thermal sources are available.

Figure 71. FEA analysis of a carbon composite plate without bonding defects.

In order to test a material for defects, this resonance information can be compared to a computer model of the material construction. To illustrate this process a model of the carbon fiber plate was made in Autodesk Fusion 360. A perfect plate was modeled as a 30.5 cm $\times$ 61 cm (12 $\times$ 24 in) solid carbon fiber twill weave plate 0.635 cm (0.25 in) thick. This was then simulated in the modal analysis environment and the resonant modes were studied. Figure 71 shows an output plot of the simulation without a defect present.

In order to simulate a bad bond, a section of the 0.635 cm (0.25 in) plate was removed by boolean cut and replaced by an airgap 5.08 cm (2 in) wide by 15.24 cm

(6 in) long. The airgap has a depth of $25.4\ \mu\text{m}$ (1×10^{-3} in) to simulate the epoxy not adhering in that area. This model was then simulated using a modal simulation environment. The results are shown in Figure 72.

Figure 72. FEA analysis of carbon composite plate with bonding defects in the center of the plate.

Interestingly this analysis showed that the fundamental frequency for a poorly bonded plate was 53 Hz. This frequency corresponds to the resonance observed for the real sample during laboratory tests as shown in in Figure 70. The simulation of a perfectly bonded plate however has a higher fundamental resonance frequency of 97 Hz. In fact for all modes of vibration simulated, the frequencies are higher in the perfectly bonded plate than in the defective plate. This leads to the conclusion that bond defects cause a decrease in fundamental frequency which makes physical sense. Surface and body waves moving through solid material such as carbon fiber typically have considerably high phase velocities. However if a defect (which in this

case is modeled as air) is present, waves in the defect region should travel slower, which changes the fundamental mode structure of the entire material.

This type of test can be used to observe the resonance of structures and materials without the need to place accelerometers or sensors on the surface. While a complete complex FEA analysis of an aged real-world target may not be feasible, long term monitoring and comparisons between similar targets with different histories offer a practical approach to optical non-contact structural health analysis. Since our device operates in ambient light with optical imaging, it can also be used from a distance given a clear sight line. This allows for the testing of materials or structures that are difficult to get to physically, even at long range, with safety and relative ease. In addition to lab-based studies, daytime field tests are possible as well. The large flux of near-infrared light in sunlight, as shown in Fig. 20, makes the InGaAs sensor, or the hybrid InGaAs-silicon fiber positioner, the optimal choices for ambient light field measurements. Combining this optical technique for detecting the surface vibrations with computer simulation of the object or device can detect changes in the material, possible points of bond failure or weakness, and with long term monitoring track the progressive aging of structures and machinery.

Vibration Sensing Tests

The passive detection of surface vibrations can have many uses. To demonstrate the ability of this technology for non-contact detection of vibrations optically from a distance several experiments were conducted. The first of these experiments was to see if a sound wave could be detected optically. Any object that is present in the vicinity of a sound producing device, will vibrate due to the sound wave incident on the surface of the object. If the surface vibration can be detected optically, then the optical signal can be used to recreate the sound wave that is causing the vibration. Of course all objects have resonance features that will create a filtering effect for the output signal, however the object can be chosen such that the range of interesting

frequencies are not damped excessively by the object.

Figure 73. InGaAs photodiode measurement of an audio signal from light scattered off a cookie bag witnessing an acoustic excitation. The signal is clearly seen at 515 Hz.

In this case a package of cookies is used. A speaker is set up facing the cookie bag. A 515 Hz audio signal is played through the speaker via a function generator and an audio amplifier. The photodiode is then focused on the wrapping of the container. This object was chosen for its reflective surface properties and lightweight (low mass) construction that causes it to vibrate freely with air pressure changes. A sample data set was taken without sound to capture a baseline signal. The scene was illuminated using an incandescent light bulb plugged into mains power at 60 Hz. Thus the FFT of the background data shows an expected 120 Hz signal and harmonics. The sound was then turned on and another data set was captured. This data set looks very similar to the background set however a new signal at 515 Hz is obvious. A plot of the FFT

of both background and signal data is shown in Figure 73. This shows that small surface deflections can be observed above background noise using this method. In this application the signal can be isolated, stored in a digital audio file, and recreated for playback if so desired. This test was inspired by previous work conducted by the MIT CSAIL group [27].

Another application of detecting surface vibrations optically is the detection of ground surface vibrations, opening a field of non-contact optical ambient sensing seismography. Several experiments were done to show the ability of these techniques and systems to detect ground motion and seismic activity by optical means. The laboratory location of this research is ideally situated to conduct some of these studies because the University of Louisville Physics & Astronomy department's Natural Sciences Building is approximately 175 m (580 ft) from an active train freight line operated by Norfolk-Southern railroad, and about twice that distance from two CSX main lines. Figure 74 shows a satellite map of the area courtesy of Google Maps.[28] The vibrations from trains moving along the railroad propagate through the ground, possibly from even larger distances, and cause vibrations in the building structure. By focusing the optical fiber onto a wall or stationary object in the lab, the relative motion between the sensor and the wall can be measured. This can be used to detect building shake due to seismic activity, machinery in the building, or in this case a train traveling outside the building. The following data was taken to show this method in practice. The optical fiber was focused onto an aluminum plate hung from the ceiling of the lab. The plate was left to settle for 5 days before the data was taken to ensure that the plate was not moving due to external forces (being touched or moved). The signal was then amplified by the previously described methods and digitized using the 32-bit DAQ.

Figure 75 shows a spectrogram created of the data when a train passed by the building. On inspection it is clear that the bulk of the vibrations due to a train or similar moving object lie in the low frequency range of the spectrum. It is not clear

Figure 74. Aerial image of the Natural Sciences Building as shown on Google Maps.

Figure 75. Optical signal of building shake as a train approaches the campus. A spectrogram is shown below the time domain data.

from this data however if the harmonics seen here are artifacts from the building resonance or if they are inherent in the incident seismic surface wave. The most likely solution to that question is that the resonance pattern seen is in fact from the building and not the train itself. In order to show this, it is necessary to directly

observe the ground surface motion from a traveling train in the vicinity of the rail bed and contrast it to this data. The next experiments do just that. The apparatus is moved outside to see ground vibrations from the train itself. A problem with this experimental setup is seen immediately however. Daylight illumination is not DC. [29] Sunlight contains many variations in the spectrum and the amplitude. On cloudy days this is due to cloud motion and most of the variation is in the very low frequency range. However even ambient light from clear “blue” sky contain significant noise signal amplitudes due to slight, visually imperceptible, variations in aerosols. To illustrate this effect a data set was taken by focusing the optical fiber onto a piece of concrete sidewalk outside the laboratory building. When the data was taken the weather was sunny with few clouds at 1500 m above ground level (5000 ft AGL). Figure 76 shows the signal obtained from this test. It is obvious that the signal varies slowly with a large magnitude; this sub-hertz motion is caused by illumination changes on the concrete. A signal of interest should be above these frequencies or have a distinctive pattern to avoid confusion of seismic signal with the variation of illumination. Similar results have been previously documented. [30, 29]

To illustrate finding a large surface vibration signal in this way, the optical fiber was focused onto a piece of tarmac. In this test however the device was kept indoors. An impulse was then applied to the floor approximately 5 feet away from the sensor tripod. This impulse was provided by stepping down abruptly onto the floor with the experimenter’s foot. The data is presented in Figure 77. It can be easily seen that the floor was tapped 3 times. There is also a decay pattern visible above noise through 2 seconds (0.5 seconds after the event). A more in depth interpretation on this type of measurement will be discussed later for surface wave analysis and detection of defects in materials.

As discussed before, a good way to demonstrate the abilities of these techniques and hardware is to observe the seismic vibrations from a train optically. This was previously done by observing building shake in the lab as a train passed by outside. It

Figure 76. InGaAs photodiode signal from sunlight incident on a section of concrete sidewalk.

would be interesting however to be able to detect the ground relative surface motion directly, by focusing on a point outside near the track itself. This point however should be chosen such that no reflection, shadow, or secondary scattering of light by the train itself is present. If the train shadow or reflection interferes with the optical signal, we cannot be certain that the signal observed is from the ground motion, or the visible train motion. To conduct this experiment a place was chosen by the track, and the fiber positioner camera was focused on a point 30 m (100 ft) away from the camera in the opposing direction of the train's approach on the track. This setup is shown in a satellite photo from Google Maps [28] in Figure 78.

The 32-bit DAQ was connected to the fiber amplifier output and digitized. A sample of data was taken without a train in the vicinity, as well as before, during and after the train passed. The data taken with no train in the vicinity is used

Figure 77. InGaAs photodiode measurement of signal due to a seismic wave caused by tapping on the floor next to the tripod setup.

to gauge the background noise in order to contrast the signal with a train present. The results of this measurement are shown in Figure 79. The background signal is shown as black, whereas the data taken when a train was traveling is shown in red. This allows us to contrast the noise signal (environment signals) with the signals of interest. It can be seen that a large signal above background is present at 8 and 10 Hz as well as higher harmonics and signals. The bulk of the train seismic signal is thought to be in the ≈ 1 Hz regime, however the background noise of surface modulation as well as cloud and sky modulation of the input light causes any signal at those low frequencies to be mixed with noise. This means that if a measurement of a low frequency seismic signal is to be measured outdoors with natural lighting, it is necessary to discriminate the signal pattern from noise, or more simply to look for higher frequency harmonics of the original signal. From this data it can be seen that

Figure 78. Satellite image of seismic field measurement area as seen on Google Maps.

the background noise falls off quite quickly above 5 Hz. As a result, signals above 5 Hz are easily seen. However signals below this threshold may be contaminated by environmental noise. Nevertheless, this experiment however shows that the detection of Earth surface vibrations is possible even in an outdoor environment.

3 Impulse Measurements

In this section we discuss experiments related to the Impulse Device (pinger) and its use in producing surface waves on experimental targets. The pinger is used to create impulses on a test plate, and data is taken with both a signal fiber instrument and a high speed camera.

The basic method of test can be summed up in the following list:

1. Ping the target.

Figure 79. Fourier analysis of ambient scattered light to extract seismic surface motion signals induced by the passing of a freight train. The data shown in red show a significant change from the urban background noise shown in black.

2. Record data with fiber and/or camera.
3. Analyze data.

The act of impacting the target area on the surface creates a surface wave that travels away from the impact site. This wave can then be observed by the instrumentation, and useful information can be extracted about the material from how it responds to an impulse, and how that response decays after the impulse.

Figure 80. The laboratory testing setup used in the analysis of composite materials.

Manual Hammer Experiments

In the first experiment we take a “impulse hammer” and impact the surface of the test plate. This hammer has a calibrated impulse force sensor fixed to the impacting region. Using the factory calibration data, the hammer was connected to a USB oscilloscope (PicoScope)[31] to record the impact made with the surface. The impact is measured in volts. This value is translated to force by a simple scale factor provided by the manufacturer’s calibration. In order to calculate the impulse the force must be integrated over time. In this test the calibrated hammer is used to create an impulse at one end of a test plate. The surface rebound is then measured by use of a photodiode connected to an optical fiber and lens. The lens is focused onto the test piece by shining light onto the fiber and focusing the resulting “dot” of light on the test piece at the desired target location. The fiber is then connected to the

Figure 81. Force and optical signal versus time due to a hammer strike to the surface of the material under test.

photodiode and amplifier circuitry and digitized by the 32-bit DAQ. The signal from the hammer is recorded at the same time via the PicoScope USB oscilloscope. The resulting data is shown in Figure 81.

It can be seen that the impact produces peak forces around 750 newtons. The total impact impulse is measured to be 225 Ns. The resulting deflection data is also shown in Figure 80. This data represents the deflection of the material from the traveling wave produced by the impulse. This traveling wave quickly disassociates into a standing wave with boundaries at the edges of the plate. As discussed before the general characteristics of the plate and mounting change the frequency and damping properties. To illustrate the effect of the standing wave damping the data is fit to a exponentially decaying cosine. This is shown in Figure 82. According to the fit equation the impulse decays to an oscillation with a fundamental frequency of

Figure 82. Plot of optical transient signal due to the impact of an impulse hammer. A fit is also provided to analyze the time dependent signal.

approximately 56 Hz. This is similar to the 53 Hz value obtained previously from vibration analysis and finite elements method computer modeling. A method of test in which a computer model of a material is compared to actual data could be devised. In this method, a material shape and structure would modeled in a finite element analysis (FEA) environment and a modal analysis would be obtained. This modal analysis would then be compared to the fundamental frequencies found by impacting the target test material with an impulse device. This in turn allows the experimenter to glean some information about the interior of the material. More study in this area is needed to show that all aspects of the physics can be predicted accurately by the computer simulation however.

Figure 83. Laboratory test setup utilizing the pinger impulse generator. The pinger can be seen mounted underneath the carbon composite test piece such that the impulse will be applied to the center of its short dimension.

High Speed Camera with Impulse Experiment

To illustrate the effect of an impulse on different sections of the test piece a high speed camera was utilized. In this test the impulse device used is the pinger previously described in this work. The pinger was mounted beneath the test piece using extruded aluminum fixtures and hardware. This configuration ensures that the pinger body and support structure will not interfere with the optical signal by creating extraneous scattered light, shadows, or even blocking the sight-line to the material itself. This setup is shown in Figure 83. The high speed camera is positioned such that the badly bonded (Frekote applied region) area is in focus and the impulse site is visible in the frame.

The camera data were sampled at 10,000 frames per second. At this rate each frame is 0.1 milliseconds exposure and sample cadence. The evolution of the primary wave is on the order of 4 frames in duration, which corresponds to approximately

0.4 milliseconds for the initial wave response. The wave then reaches the boundary of the material and begins to reflect and superposition becomes a factor. The entire impulse event decays in approximately 100 milliseconds.

Figure 84. High speed camera 1 frame after impulse, that is 0.1 millisecond after the tap.

Figures 84, 85, 86, and 87 show the time evolution of an impulse from the pinger. It can be seen in this data that the impulse wave is directly visible utilizing a Photron high speed camera with the use of the impulse device. In Figure 84 the initial impulse

Figure 85. High speed camera 2 frames after the impulse, that is 0.2 millisecond after the tap.

from the pinger is visible in the bottom center of the image. The impulse then travels across the material such that in the 85 the impulse has setup a standing wave in the material. Figures 86 and 87 show the development of these modal vibrations. Using this data, it is possible to make inferences about the material properties and possibly detect badly bonded areas if good material modeling is employed. For instance the speed of the wave as it travels through a badly bonded area may be an indication of the

Figure 86. High speed camera 3 frames after the impulse, that is 0.3 millisecond after the tap.

bond state. The decay rate or damping constant may also be used to detect changes in material properties spatially. The ability of the high speed camera to visualize the entire test piece is useful for bulk analysis since the effects of bulk material can be seen easily. The pinger enables this type of measurement by executing a precisely controlled and time impulse event without a double strike that can be synchronized with high speed image acquisition.

Figure 87. High speed camera 4 frames after the impulse, that is 0.4 millisecond after the tap.

4 Fiber Positioner with Impulse Experiment

The high speed camera allows for the capture of spatial data as well as time dependent data for multi-dimensional analysis of impulse responses. However the dynamic range of this state of the art camera is 12 bits, and that affects the data available. The data rate is high, but the bit depth is low compared to single pixel photodiode measurements. To observe the same test with higher dynamic range, now limited

Figure 88. Plots of four separate regions sampled after impacting a carbon composite test piece with the pinger. The fiber positioner was used to slew to each area.

by temporal response but with spatial resolution, the scene is measured with the fiber input positioner and the 32-bit DAQ coupled to the fiber output. The fiber positioning device is unique in that the sampled point in the image can be moved by simply selecting a new coordinate. In the same way the high speed camera was set up, the fiber positioner was set up and focused onto the badly bonded area of the test material. The fiber was then able to be slewed from the bad bond area to other areas of the test piece. In this way multiple sites were able to be sampled without changing the setup, which improves experimental design by removing variables involving relocating the single fiber lens each time. The fiber device was setup to look at multiple locations on the plate, and a data set was recorded via the 32-bit DAQ at each selected location, after a impulse from the pinger. Nine locations were selected on the test piece. Most of these were outside of the bad bond area. In this work, four

of the most important areas are shown. The data presented in Figure 88. Of interest in this data is the fact that a low frequency decay oscillation is observed in the bad bond area, but is not present elsewhere. This could indicate that the surface tends to resonate at a lower frequency when there is a defect as was suggested previously in this chapter. The data also shows a contrast between the signal form, suggesting that it is possible to detect a poorly bonded area using these methods.

To illustrate the difference between the poorly bonded region and the well bonded region the data were normalized and plotted on the same scale. This analysis is shown in Figure 89. From this it is clear that there is a difference in the response and wave propagation across the poorly bonded area compared to the propagation across the well bonded areas. The initial propagation takes place very quickly and only resolved with the high speed camera. It is thus easier to see the bulk properties of the standing wave than the original pulse. However utilizing a high enough sample rate it is possible to observe the actual propagation differences. In these data it appears that the poorly-bonded (Frekote) region exhibits a larger damping than the well-bonded area, and that there may be a difference in wave velocity. We have already noted that the structural resonances change frequency when a defect is present as a consequence of the dependence of the wave velocity on the subsurface structure. While the combination of methods and aspects could be used to detect a badly bonded region, obviously a more comprehensive study with controlled samples is necessary to explore this hypothesis and to enable these techniques to be put into practice.

Figure 89. Optical signal in the time domain of changes in ambient scattered light resulting from an impulse from the pinger. The data are normalized to show the relative responses of the poorly-bonded and the well-bonded regions.

CHAPTER VIII

CONCLUSIONS

1 Conclusions

In this work we discussed the design and implementation of multiple pieces of equipment, and their impact on different applications. The experimental process and many important lessons learned about design, engineering and experiment design were documented. The use of passive illumination and detection with visible and near-infrared light to investigate materials defects and properties was discussed and experiments were developed to explore these possibilities. An impulse device was created to explore the use of high speed cameras to detect materials defects. A fiber positioning unit was developed to explore similar defects as well as astronomical applications and other uses. A high bit depth 32-bit digital acquisition system was developed for use with photodiodes to make these very small measurements. Each of these devices are still in a prototype stage; further work will be done in the future to enhance and fine tune the abilities of these devices and techniques.

The ability to use passive sensing techniques for experimentation and testing shows promise when given the appropriate hardware. Signal-to-noise ratio and dynamic range will always exist as limiting factors for passive measurements. However, when the devices and experimental methods are designed such to decrease the environmental noise and to select for particular interested signals, as done here, the techniques become advantageous. The main reason for using passive optical sensing in this regime is the fact that the system under test is not subject to interference by the test apparatus. Light is already incident on the target, and therefore any effect that may come from the power in the incident light itself is very small compared to

other environmental influences. The small natural variations in ambient light are only a noise added to the signal from the surface motions. We detect the modulation of the scattered ambient light resulting from the targets translation and change in inclination. The optical system also ideally does not contribute to noise in the signal, or its effects can be mitigated. This is especially true in the case of astronomy where the objects of interest are in most cases many light-years away and not subject to contact measurements of the usual sort. In that application, only the photon statistical noise and the instrumental noise are there to mask the information content of the detected light. Developing methods of materials testing and vibration analysis here on Earth gives important tools that can be used in astronomy observations such as the search for exoplanets and other time and signal sensitive applications. The fiber positioning device has an interesting special role to play in this application. Exo-planets are often detected by sensing a very small change in light level from a star. These measurements can be analyzed to find the size of a planet, for example. However the smaller the planet is, the smaller its effect on the total signal. This means a very high signal-to-noise ratio on a wide dynamic range signal is needed to detect small exoplanets. The fiber positioning device could be used in this application to observe a field of stars with the optical camera as the reference channel, and the positioned fiber device to slew to a particular star in question and sample high dynamic range data with a fast optimized and affordable sensor. These measurements would have much higher dynamic range than the measurements from a conventional camera especially if the sample rates were kept low and co-added to increase the signal to noise ratio. There is potential for use of low-cost single element InGaAs photodiodes, broad range InSb, or even HgCdTe (MCT) sensors, cooled to reduce noise further, in the ground-based detection and verification of exoplanets orbiting cool type M host stars which radiate primarily in the infrared.

Applications to materials reliability analysis are also important aspects of this research. In current aircraft design, it is becoming common place to use carbon com-

posite materials. Many commercial as well as general aviation aircraft are almost entirely manufactured from composite materials bonded with epoxy rather than conventional metal frame bonded with rivets. As a result a method of testing these structures during manufacturing and as they age in use is needed. The general methods used for testing aircraft frames and wings are in their basic nature destructive. If an engineer wants to find the strength of a wing design, typically the structure will be modeled in a computer and simulated using an finite elements analysis package. The structure must be proven to be as strong as simulation however for safety and insurance purposes, so a prototype structure is constructed to test. One spectacular test involves the bending of the aircraft wing until the ultimate failure stress is reached, and the wing breaks. While these destructive tests are useful in proving a design, obviously they cannot be used on every aircraft or in the field. If an operator wants to test an aircraft for structural damage it is often impossible to do with confidence. Structural inspections are performed every 100 hours for commercial use aircraft and on a specialized maintenance schedule for airlines. This takes the aircraft out of service for up to a week, which in commercial aviation costs time which in turn costs money. For older aluminum-based aircraft structures, this was done by performing a visual inspection of the airframe. Special doors were constructed on aircraft wings to allow a mechanic to see inside the aircraft wing and fuselage. These inspection doors are held on by screws. As a result they increase the drag of the aircraft and decrease efficiency. For composite aircraft these inspection panels are few and far between, since the aircraft cannot readily be inspected visually anyway, and composite material is employed mainly to increase strength and decrease weight and drag. In these cases the ability to use a passive test device that would image the aircraft without the need to disassemble the aircraft is advantageous. If a weak point in the wing or fuselage can be detected by simply imaging the aircraft and finding its natural modes of vibration, aircraft could be testing while in run-up on the ramp, even each time the aircraft left the terminal. This scheme would increase safety and also decrease

aircraft downtime, thus being a win-win for the aviation business.

Applications of these kinds of measurements are also useful in machinery and structural health monitoring as well as environment seismic sensing. Structural health monitoring has already been demonstrated in the use of optical sensing for bridge defect analysis.[32] This type of analysis is useful for monitoring railroad and highway bridges in routine inspections since it can be used from a distance without limiting traffic on the bridge as would be required for *in situ* visual inspection or measurements requiring contact. For bridges, some critical areas are also inaccessible without risk to personnel, while standoff optical measurements offer a cost effective safe alternative for the technicians. The current methods also do not account for damage that is not visible to the eye. For instance the tension on a wire or the torque on bolts cannot be detected easily by visual inspection. However the tension and torque differences will cause detectable difference in the vibrational modes of the structure. If the modes are known by modeling or comparison to known healthy structures, the optically measured signals can be compared to models, databases, or a history for the structure in question. In this way a potentially disastrous failure could be avoided long before obvious signs such as cracking and slippage are present.

The hardware and analysis techniques presented in this work have the potential to be used as powerful tools in many industries. As a result some devices in this work could have wide commercial appeal. Vibration analysis is often used in manufacturing and there is a need in industry for simple to use hardware that is robust and of high quality. There are also other applications for designs presented in this work. The impulse device for example can be used as a linear motor. A simple use of this linear motor would be a linear piston air compressor. The ability to use a linear motor coupled to a piston would increase the efficiency of the piston since it would be connected to a crankshaft. Crankshaft friction and angle causes a noticeable loss in efficiency from both piston compressors as well as piston engines. In a similar setup the linear motor could be used as a linear generator coupled to a linear piston engine.

This type of generator would be very efficient since the effect of eddy current and hysteresis loss has been virtually eliminated. The linear piston generator would also have higher mechanical efficiency for the same reasons as the linear piston compressor. The loss of the crankshaft decreases friction and increases efficiency. Other uses for this technology can be found in the construction of a rotary motor using the same principle of pole piece design. In such a motor the back-iron would rotate on both sides causing no eddy current loss. Such a motor could develop the same torque at 10,000 rpm as it did at stall. More work must be done to design these devices, but the concept is similar to that used here and it would contribute to increased efficiency for many different applications.

In conclusion, the devices presented in this work have the ability to improve existing hardware as well as provide better hardware for a large variety of exciting applications. The journey of design exploration has resulted in a gain of information and skills that will hopefully contribute to the scientific community and the world in general in a positive way.

REFERENCES

- [1] C. Starr, C. Evers, and L. Starr, *Biology: Concepts and Applications*, Brooks/Cole biology series (Thomson, Brooks/Cole, 2006).
- [2] Wikipedia, “EM Spectrum Image,” https://en.wikipedia.org/wiki/File:EM_spectrum.svg.
- [3] M. Rowan-Robinson, *Night Vision: Exploring the Infrared Universe* (Cambridge University Press, 2013).
- [4] A. Rogalski, “History of infrared detectors,” *Opto-Electronics Review* **20**, 279–308 (2012).
- [5] J. B. Johnson, “Thermal agitation of electricity in conductors,” *Phys. Rev.* **32**, 97–109 (1928).
- [6] T. Instruments, “ADS1262 ADC,” <http://www.ti.com/product/ADS1262>.
- [7] L. Technology, “Ltc2508 datasheet,” <http://www.analog.com/en/products/analog-to-digital-converters/standard-adc/precision-adc-20msps/single-channel-ad-converters/ltc2508-32.html#product-overview>.
- [8] L. Technology, “Ltc6655 datasheet,” <http://www.analog.com/en/products/amplifiers/ad-drivers/fully-differential-amplifiers/ltc6363.html>.
- [9] L. Technology, “Ltc6655 datasheet,” <http://www.analog.com/media/en/technical-documentation/data-sheets/6655ff.pdf>.
- [10] I. Moreno, J. J. Araiza, and M. Avendano-Alejo, “Thin-film spatial filters,” *Opt. Lett.* **30**, 914–916 (2005).

- [11] A. Hughes and W. Drury, *Electric Motors and Drives: Fundamentals, Types and Applications* (Elsevier Science, 2013).
- [12] D. Meeker, “FEMM Software,” <http://www.femm.info/wiki/HomePage>.
- [13] B. Amstadt, “Wine HQ,” <https://www.winehq.org/>.
- [14] LibreCad, “2d cad software,” <https://librecad.org/>.
- [15] AutoDesk, “Inventor software,” <https://www.autodesk.com/>.
- [16] D. Systems, “Solidworks software,” <https://www.solidworks.com/>.
- [17] FreeCad, “Freecad software,” <https://www.freecadweb.org/>.
- [18] OnShape, “3d cad software,” <https://www.onshape.com/>.
- [19] I. CNC Software, “MasterCAM,” <https://www.mastercam.com/en-us/>.
- [20] Autodesk, “Fusion 360,” <https://a360.autodesk.com/>.
- [21] e. a. Jean-Pierre Charras, “KiCad,” <http://kicad-pcb.org/>.
- [22] Autodesk, “Eagle CAD,” <https://www.autodesk.com/products/eagle/free-download>.
- [23] O. Source, “gEDA Project,” <http://www.geda-project.org/>.
- [24] EBay, “Online marketplace,” <https://www.ebay.com>.
- [25] J. Hay, “High Dynamic Range Imaging for the Detection of Motion,” Ph.D. thesis, University of Louisville, Louisville, Kentucky (2012).
- [26] T. D. Rossing and D. A. Russell, “Laboratory observation of elastic waves in solids,” *American Journal of Physics* **58**, 1153–1162 (1990).

- [27] N. W. G. M. F. D. W. T. F. Abe Davis, Michael Rubinstein, “The Visual Microphone: Passive Recovery of Sound from Video,” *ACM Transactions on Graphics* **33** (2014).
- [28] Google, “Google Maps,” maps.google.com.
- [29] J. Kielkopf, E. Jensen, F. O. Clark, and B. Noyes, “Fractional intensity modulation of diffusely scattered light,” in “Infrared Remote Sensing and Instrumentation XXIII,” , vol. 9608 (2015), vol. 9608, p. 96080I.
- [30] J. Kielkopf, E. Jensen, F. O. Clark, and J. R. Hay, “Observation and analysis of modulation and noise in visible and near infrared diffuse ambient daylight,” in “Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series,” , vol. 10403 (2017), vol. 10403, p. 104030B.
- [31] PicoScope, “Picotech usb devices,” <https://www.picotech.com/products/oscilloscope>.
- [32] J. F. K. Jeffrey R. Hay and F. O. Clark, “Non-contact stand-off optical sensing of cable vibrations for monitoring structural health of the william h. harsha bridge,” 15th International Conference on Experimental Mechanics (ICEM15) (2012).
- [33] J. Huber, “Ph.D Thesis,” Ph.D. thesis, University of Kentucky, Lexington, Kentucky (2017).
- [34] T. Labs, “Photodiode saturation and noise floor,” https://www.thorlabs.com/images/TabImages/Photodetector_Lab.pdf.
- [35] T. Labs, “Thor labs dichroic beamsplitters,” https://www.thorlabs.com/navigation.cfm?guide_id=2354.
- [36] MccDaq, “24-bit DAQ,” <https://www.mccdaq.com/usb-data-acquisition/USB-2416-Series.aspx>.

- [37] A. Vision, “Machine Vision Camera,” <https://www.alliedvision.com/en/digital-industrial-camera-solutions.html>.
- [38] N. Instruments, “24-bit DAQ,” <http://sine.ni.com/nips/cds/view/p/lang/en/nid/205190>.
- [39] M. Young, *The Technical Writer’s Handbook* (University Science Books, 1989).
- [40] G. L. Trigg, “Grammar,” *Physical Review Letters* **42**, 747–748 (1979).
- [41] L. Technology, “Ltc1799 datasheet,” <http://www.analog.com/media/en/technical-documentation/data-sheets/1799fd.pdf>.
- [42] AutoDesk, “Fusion software,” <https://www.autodesk.com/>.

Appendix A: Acknowledgements

The author would like to thank the Office of Naval Research for providing funding to allow for this work to be completed.

Appendix B: Mechanical Drawings and Electrical Schematics

This section is a compilation of mechanical drawings for the devices described in this dissertation.

Figure 1. Fiber positioner drawing 1.

Figure 2. Fiber positioner drawing 2.

Figure 3. Fiber positioner drawing 3.

Figure 4. Pinger drawing 1.

Figure 5. Pinger drawing 2.

Figure 6. Pinger drawing 3.

Figure 7. Pinger drawing 4.

Figure 8. Pinger Drawing 5

Figure 9. Fiber positioner motherboard PCB layout.

Figure 12. Finger power supply PCB layout.

Figure 15. LTC based 32-bit DAQ PCB layout.

Appendix C: Commonly Used Acronyms

FPGA - Field Programmable Gate Array

ISP - In System Programming

ID - Inner Diameter

OD - Outer Diameter

CAM - Computer Aided Manufacturing

CAD - Computer Aided Design

CNC - Computer Numerical Control

DAQ - Data Acquisition unit

ADC - Analog to Digital Converter

DAC - Digital to Analog Converter

ENOB - Effective Number of Bits

SNR - Signal to Noise Ratio

DR - Dynamic Range

SFDR - Spurious Free Dynamic Range

GPU - Graphics Processing Unit

CPU - Central Processing Unit

MCU - Microcontroller Unit

SPS - Samples per Second

SLR - Single Lens Reflex

MOSFET - Metal Oxide Semiconductor Field Effect Transistor

IGBT - Insulated Gate Bipolar Transistor

FEA - Finite Element Analysis

FFT - Fast Fourier Transform

Appendix D: Code

This Appendix includes C code written as part of this work. Both Embedded code and Desktop environment code is included.

Desktop Code

```
1  /*
2   To build use the following gcc statement
3   (assuming you have the static d2xx library in the /usr/local/lib
4   directory
5   and you have created a symbolic link to it in the current dir).
6   gcc -o static_link main.c -ldl -lpthread libftd2xx.a
7  */
8  #include "Include/FTDLUSB.h"
9  #include "Include/ftd2xx.h"
10 #include <stdio.h>
11 #include <stdlib.h>
12 #include <thread>
13 #include <string.h>
14 #include <iostream>
15 #include <fstream>
16 #include <unistd.h>
17
18 using namespace std;
19 using namespace ftdiDevice;
20
21 int change_sample_rate(int rate);
22 int get_data_buffer(char *data_filename, int size); // size is multiple
23           of 2
```

```

23 int get_time();
24 int get_data_stream(int n, char *filename, int file_size);
25
26 // dg32 -c 1024 -o filename.dat      // capture 1024 points to
    filename.dat
27
28 // dg32 -s 4                        //set speed to 4
29
30 // for stream capture use dg32 -n 120 -c 1024 -o filename.dat
31 //.. this will capture 120 files of 1024 points store it in
    filename_N.dat
32
33
34 int main(int argc, char **argv)
35 {
36     int is_s=0;
37     int is_stream = 0;
38     int is_o=0;
39     int is_capture=0;
40     char *cvalue = NULL;
41     char *ovalue = NULL;
42     char *svalue = NULL;
43     char *nvalue = NULL;
44     int c;
45
46     opterr = 0;
47     if(argc>1)
48     {
49         while ((c = getopt (argc, argv, "c:o:s:n:")) != -1)
50             switch (c)
51             {
52                 case 'o':
53                     ovalue = optarg;
54                     is_o = 1;

```

```

55         break;
56     case 's':
57         svalue = optarg;
58         is_s = 1;
59         break;
60     case 'c':
61         cvalue = optarg;
62         is_capture = 1;
63         break;
64     case 'n':
65         nvalue = optarg;
66         is_stream = 1;
67         break;
68     case '?:
69         if (optopt == 'c')
70             fprintf (stderr, "Option -%c requires an argument.\n",
optopt);
71         else if (isprint (optopt))
72             fprintf (stderr, "Unknown option '-%c'.\n", optopt);
73         else
74             fprintf (stderr,
75                     "Unknown option character '\\x%x'.\n",
76                     optopt);
77         return 1;
78     default:
79         abort ();
80 }
81
82 //printf ("ovalue = %s, svalue = %s, cvalue = %s\n",
83 //    ovalue, svalue, cvalue);
84
85 // if statment for capture
86 if(is_o ==1 && is_capture ==1 && is_s == 0 && is_stream ==0)
87 {

```

```

88         printf ("Capturing %s points of data. Saving in %s \n",
cvalue , ovalue);
89         int capture_num_points =0;
90         capture_num_points = atoi(cvalue);
91
92         get_data_buffer( ovalue , capture_num_points);
93     }
94     //if statment for sample speed
95     if( is_s ==1 && is_o ==0 && is_capture ==0 && is_stream ==0 )
96     {
97         printf ("Setting Sample Speed to  %s \n", svalue);
98         int sample_rate_number = 0;
99         sample_rate_number = atoi(svalue);
100        change_sample_rate(sample_rate_number);
101    }
102
103    if( is_o ==1 && is_capture ==1 && is_s == 0 && is_stream ==1)
104    {
105        int capture_num_points =0;
106        capture_num_points = atoi(cvalue);
107
108        int capture_num_files =0;
109        capture_num_files = atoi(nvalue);
110
111        printf ("Capturing  files \n");
112        get_data_stream( capture_num_files , ovalue , capture_num_points);
113    }
114
115
116    } else
117    {
118        printf ("WELCOME TO DATAGRABBER 32 DAQ COMMAND LINE INTERFACE!
\n \n");
119        printf ("Usage: \n\n");

```

```

120     printf ("To set the capture speed:   dg32 -s speedvalue \n\n");
121     printf ("Speed Value           Samples Per Second\n");
122     printf ("1                     61 \n");
123     printf ("2                     244 \n");
124     printf ("3                     976 \n");
125     printf ("4                     3906 \n \n \n");
126
127     printf ("To capture a set of data:   dg32 -c samples -o
filename.dat \n \n");
128
129     printf ("To capture a set of files:   dg32 -c samples -n 12 -o
filename.dat \n \n");
130
131     return 0;
132 }
133
134
135
136 return 0;
137 }
138
139 int get_data_buffer(char *data_filename,int size) // size is multiple
of 2
140 {
141     int32_t dec;
142     double value = 0;
143     int test;
144
145     FILE *datafile = fopen(data_filename,"w");
146
147     size = size*5; //account for 5 bytes of data per sample
148
149     unsigned char buff[size];
150

```

```

151     int n_reps = 0;
152     int n_total_reps = 1;
153
154     double time = 0;
155
156     setup_ftdi("JEN2016DG");
157     purge_device();
158
159     while(n_reps<n_total_reps)
160     {
161         if(read_buffer(buff,size) != 0)
162         {
163             return 0;
164         }
165
166
167         for(int i = 0; i<size-5; i= i+5)
168         {
169             dec = (buff[i]<<24) | (buff[i+1]<<16) | (buff[i+2] <<8) |
170 (buff[i+3]);
171             value = (double(dec)/2147483648.0)*3.3; //convert to decimal
172             printf("%.20f \n",value);
173             //fprintf(datafile,"%.10f      ",time);
174             fprintf(datafile,"%.10f \n",value);
175             time = time + (1.0/3906.0);
176         }
177
178
179         n_reps++;
180     }//while
181
182     ftdi_close();
183     fclose(datafile);

```

```

184     return 0;
185 }
186
187
188 int change_sample_rate(int rate)
189 {
190
191     int32_t dec;
192     int test;
193     int speed_int = rate;
194     int size = 256*5;//320;
195     unsigned char buff[size];
196
197     char *command_array = new char [8];
198
199     printf(" Setting Speed to %i\n ",speed_int);
200
201     if(speed_int == 1)
202     {
203         command_array = "SETFQ{A}"; //229
204     }else if(speed_int == 2)
205     {
206         command_array = "SETFQ{B}"; //197
207     }else if(speed_int == 3)
208     {
209         command_array = "SETFQ{C}"; // 165
210     }else if(speed_int == 4)
211     {
212         command_array = "SETFQ{D}"; //133
213     }else
214     {
215         return 0;
216     }
217

```

```

218
219     setup_ftdi("JEN2016DG");
220     purge_device();
221
222     write_buffer(command_array);
223
224     if(read_buffer(buff, size) != 0)
225     {
226         return 0;
227     }
228     printf(" READ \n ");
229     for(int i = 0; i<size-5; i= i+5)
230     {
231         test = buff[i+4];
232         //printf("%d \n" ,test);
233     }
234     printf("%d \n" ,test);
235
236     ftdi_close();
237     return 0;
238 }
239
240 int get_time()
241 {
242
243 }
244 // n number of file of size file_size and prefix filename
245 int get_data_stream(int n, char *filename, int file_size)
246 {
247     int i = 0;
248
249     char basename[128] = "";
250     strcpy(basename, filename);
251     char filename_with_code[128] = "";

```



```

252     char number_name[8];
253     //filename_with_code = strcat(filename, itoa(0, number_name, 10));
254
255
256
257
258     do{
259         strcpy(filename_with_code, basename);
260         sprintf(number_name, "%i", i);
261         strcat(filename_with_code, number_name);
262         strcat(filename_with_code, ".dat");
263
264         get_data_buffer(filename_with_code, file_size);
265
266         i = i+1;
267     }while(i<n);
268
269     return 0;
270
271 }

```

```

1
2 #include <stdio.h>
3 #include <stdlib.h>
4 #include <unistd.h>
5 #include "ftd2xx.h"
6 #include <stdint.h>
7 #include <string.h>
8
9 #define MAX_DEVICES 24
10 #define BUF_SIZE 128
11
12
13 #define VID 0x0403

```

```

14 #define PID 0x6014
15
16 FT_STATUS ftStatus;
17 FT_HANDLE ftHandle;
18 int device_number = 0;
19
20 namespace ftdiDevice{
21
22 int purge_device()
23 {
24
25     ftStatus = FT_Purge (ftHandle,  FT_PURGE_RX | FT_PURGE_TX);
26     if (ftStatus == FT_OK) {
27         return 0;
28     }
29     else {
30         return -1;
31     }
32 }
33
34
35
36 int setup_ftdi(const char *device_serial_number)
37 {
38     int found = 0;
39
40     char * pcBufLD [MAX_DEVICES + 1];
41     char cBufLD [MAX_DEVICES][64];
42
43
44     int iNumDevs = 0;
45     int i;
46
47     int device_number =0;

```

```

48
49
50 ftStatus = FT_SetVIDPID(VID,PID);
51 if(ftStatus != FT_OK)
52 {
53     printf("Could not set VID or PID");
54     return -1;
55 }
56
57
58 // FIND device number
59 for(i = 0; i < MAX_DEVICES; i++) {
60     pcBufLD[i] = cBufLD[i];
61 }
62 pcBufLD[MAX_DEVICES] = NULL;
63
64 ftStatus = FT_ListDevices(pcBufLD, &iNumDevs, FT_LIST_ALL |
FT_OPEN_BY_SERIALNUMBER);
65
66 if(ftStatus != FT_OK) {
67     printf("Error: FT_ListDevices returned %d\n", (int)ftStatus);
68     return 1;
69 }
70
71 for(i = 0; ( (i < MAX_DEVICES) && (i < iNumDevs) ); i++) {
72
73     if(strcmp(device_serial_number, (const char *) &cBufLD[i]) == 0)
74     {
75         device_number = i;
76         printf("Device %d Serial Number - %s\n", i, cBufLD[i]);
77         found = 1;
78         break;
79     }
80 }

```

```

81
82
83     if(found == 1)
84     {
85         ftStatus = FT_Open(device_number,&ftHandle);
86     }
87     else
88     {
89         printf("DID NOT Open device %s\n", cBufLD[device_number]);
90         return 1;
91     }
92     /* Setup */
93     if((ftStatus != FT_OK)){
94         /*
95          This can fail if the ftdi_sio driver is loaded
96          use lsmod to check this and rmmod ftdi_sio to remove
97          also rmmod usbserial
98          */
99         printf("Error: FT_Open returned %d for device %d\n",
(int)ftStatus, device_number);
100         return 1;
101     }
102
103
104     UCHAR Mask = 0xff;
105     UCHAR Mode = 0x40;
106     ftStatus = FT_SetBitMode(ftHandle, Mask, Mode);
107     if (ftStatus != FT_OK) {
108         printf("Could not set FIFO mode! ");
109         return -1;
110     }
111
112     ftStatus= FT_SetLatencyTimer (ftHandle, 200);
113     if (ftStatus != FT_OK) {

```

```

114         printf("Error: Latecncy Timer returned %d\n", (int)ftStatus);
115         return 1;
116     }
117     printf("Opened device %s\n", cBufLD[device_number]);
118
119     return 0;
120 }
121
122
123 int ftdi_close()
124 {
125     FT_Close(ftHandle);
126
127     return 0;
128 }
129
130
131 int write_buffer(char *tx_buffer)
132 {
133     DWORD BytesWritten;
134
135     ftStatus = FT_Write(ftHandle, tx_buffer, sizeof(tx_buffer),
136         &BytesWritten);
137     if (ftStatus != FT_OK) {
138         return -1;
139     }
140
141     return 0;
142 }
143
144
145 int read_buffer(uint8_t *data, int n)
146 {

```

```

147 //uint8_t *   pcBufRead = NULL;
148 DWORD dwRxSize = 0;
149 DWORD dwBytesRead =0;
150
151 /* Read */
152 //int num_of_buffers = 12;
153 //int count = 0;
154 //uint16_t data_out[32768];
155
156
157
158 dwRxSize = n; //32768;
159
160 while ((dwRxSize < BUF_SIZE) && (ftStatus == FT_OK)) {
161     ftStatus = FT_GetQueueStatus(ftHandle , &dwRxSize);
162
163     if(ftStatus != FT_OK)
164     {
165         return -1;
166     }
167
168 }
169
170
171 if(ftStatus == FT_OK) {
172     //pcBufRead = (uint8_t*)realloc(pcBufRead , dwRxSize);
173     if((ftStatus = FT_Read(ftHandle , data , dwRxSize , &dwBytesRead)) !=
FT_OK){
174         printf("Error: FT_Read returned %d\n", (int)ftStatus);
175     }
176     else {
177
178
179

```

```

180     //     printf("FT_Read read %d bytes\n", (int)dwBytesRead);
181     }
182
183     }
184     else {
185
186         printf("Error: FT_GetQueueStatus returned %d\n", (int)ftStatus);
187         return -1;
188     }
189
190
191     return 0;
192 }
193
194
195
196
197 int read_line(uint8_t data[128][128], int n)
198 {
199
200     int count =0;
201     int trigger = 0;
202     uint8_t data_buffer[16384];
203
204
205     int frame = 0;
206     printf("FRAME START\n" );
207
208
209     while (frame < 128)
210     {
211         read_buffer (data_buffer, 16384);
212         for (int i =0; i < 16384; i++)
213         {

```

```

214         if (data_buffer[i] == 0)
215         {
216             trigger = 1;
217             count = 0;
218         }
219
220         if (trigger == 1)
221         {
222             data[count][frame] = data_buffer[i];
223
224             if (data[count][frame] != count)
225             {
226                 printf(" ERROR DATA NOT VALID\n" );
227
228                 printf(" frame = %u    count = %u    data = %u\n", frame,
count, data[count][frame] );
229             }
230
231             printf(" frame = %u    count = %u    data = %u\n", frame,
count, data[count][frame] );
232             count++;
233         }
234
235
236         if (count > (n-1))
237         {
238             frame++;
239             trigger = 0;
240             count = 0;
241             break;
242         }
243     }
244 }
245 printf("FRAME END\n" );

```



```
246 return 0;
```

```
247 }
```

```
248
```

```
249 }
```

Appendix E: 32-Bit DAQ Code

```
1
2 #include "pin_numbers.h"
3 #include <avr/io.h>
4 #include <avr/interrupt.h>
5 #include <inttypes.h>
6 #include <util/delay.h>
7 #include <avr/sleep.h>
8 #include "easy_avr.h"
9 #include <string.h>
10 #include <stdio.h>
11 #include <stdlib.h>
12 #include <math.h>
13
14 volatile uint8_t config_byte = 0x0F;
15
16 volatile uint8_t counter = 0;
17 volatile uint8_t data_aval = 0;
18 volatile uint8_t data_byte[6];
19
20 volatile uint8_t data;
21
22
23 void send_data_ftdi(uint8_t data)
24 {
25     PORTA.OUT = data;
26     PORTB.OUTCLR = PIN3_bm;
27     //delay here
28     delay_microseconds(2);
29     PORTB.OUTSET = PIN3_bm;
```

```

30 }
31
32 interrupt(PORTD_INT0_vect) // if DRL goes low then data available
33 {
34     data_aval = 1;
35     counter = 0;
36 }
37
38 interrupt(PORTD_INT1_vect) // if BUSY goes low then send data
39 {
40     if(data_aval && counter < 5)
41     {
42         data = spi_transmit(&SPID, 0x00);
43         send_data_ftdi(data);
44         counter++;
45     } else
46     {
47         data_aval = 0;
48     }
49
50 }
51
52
53
54 int main()
55 {
56     setup_clock(OSC_RC32MEN_bm, CLK_SCLKSEL_RC32M_gc);
57
58     //setup FTDI pins
59     pinMode(&PORTB, RXF, INPUT); //RXF INPUT
60     pinMode(&PORTB, TXE, INPUT); //TXE input
61     pinMode(&PORTB, PIN2_bm, OUTPUT); //RD out
62     pinMode(&PORTB, PIN3_bm, OUTPUT); //WR out
63     PORTA_DIR = 0xFF; // set as output

```

```

64
65
66     pinMode(&PORTE,RDLA1,OUTPUT) ;
67     pinMode(&PORTE,RDLA2,OUTPUT) ;
68     pinMode(&PORTE,SEL1,OUTPUT) ;
69     pinMode(&PORTE,SEL0,OUTPUT) ;
70     pinMode(&PORTD,CLK_ON,OUTPUT) ;
71     digitalWrite(&PORTD,CLK_ON,LOW) ;
72
73
74     //setup SPI bus on portD as master mode 0
75     spi_setup(&SPID,&PORTD,SPI_MASTER_bm | SPI_MODE_0_gc) ;
76     spi_enable(&SPID) ;
77
78     pinMode(&PORTD,BUSY,INPUT) ;
79     pinMode(&PORTD,DRL,INPUT) ;
80
81     delay(1000) ;
82
83     //setup input filter
84     pinMode(&PORTC,F1,OUTPUT) ;
85     pinMode(&PORTC,F2,OUTPUT) ;
86     pinMode(&PORTC,F3,OUTPUT) ;
87     pinMode(&PORTC,F4,OUTPUT) ;
88
89     digitalWrite(&PORTC,F1,LOW) ;
90     digitalWrite(&PORTC,F2,LOW) ;
91     digitalWrite(&PORTC,F3,LOW) ;
92     digitalWrite(&PORTC,F4,LOW) ;
93
94     //setup ADC
95
96     /*
97     Down-Sampling Factor Select

```

```

98 Input 0, Down-Sampling Factor Select Input 1. Selects the
99 down-sampling factor for the digital filter. Down-sampling
100 factors of 256, 1024, 4096 and 16384 are selected for
101 [SEL0 SEL1] combinations of 00, 01, 10 and 11 respectively.
102 Logic levels are determined by OVDD
103 */
104 digitalWrite(&PORTE,SEL0,LOW);
105 digitalWrite(&PORTE,SEL1,HIGH); //down sample 1024
106 digitalWrite(&PORTE,RDLA1,LOW);
107
108 delay(100);
109 digitalWrite(&PORTD,CLK_ON,HIGH); // turn on clock to ADC
110 delay(10);
111
112
113 // External interrupt 0 on PD2, enable pullup, sence falling edge of
DRL
114 PORTD.PIN2CTRL = PORT_OPC_PULLUP_gc | PORT_ISC_FALLING_gc;
115 PORTD.INT0MASK = PIN2_bm;
116 PORTD.INTCTRL = PORT_INT0LVL_LO_gc;
117
118 // External interrupt 1 on PD0, enable pullup, sence falling edge of
BUSY
119 PORTD.PIN0CTRL = PORT_OPC_PULLUP_gc | PORT_ISC_FALLING_gc;
120 PORTD.INT1MASK = PIN0_bm;
121 PORTD.INTCTRL = PORT_INT1LVL_LO_gc | PORT_INT0LVL_LO_gc;
122
123 // Enable low level interrupts
124 PMIC.CTRL |= PMIC_LOLVLEN_bm;
125 sei();
126
127
128
129

```

```
130
131  while(1)
132  {
133      //loop let the interrupts do their thing
134  }
135
136  return 0;
137 }
```

Appendix F: XMEGA Library Code

```
1
2 #ifndef _EASY_AVR_2015_H
3 #define _EASY_AVR_2015_H
4 #ifdef AVR8
5 #include <avr/io.h>
6 #include <avr/interrupt.h>
7 #include <inttypes.h>
8 #include <util/delay.h>
9 #include <avr/sleep.h>
10
11 #define sbi(ADDRESS,BIT)  ADDRESS |= (1<<BIT) // set Bit
12 #define cbi(ADDRESS,BIT)  ADDRESS &= ~(1<<BIT) // clear Bit
13 #define toggle(ADDRESS,BIT) ADDRESS ^= (1<<BIT)
14
15 #define OUTPUT 1
16 #define INPUT 0
17 #define HIGH 1
18 #define LOW 0
19 #define interrupt ISR
20 #endif
21
22 #ifdef AVR_XMEGA
23 #include <avr/io.h>
24 #include <avr/interrupt.h>
25 #include <inttypes.h>
26 #include <util/delay.h>
27 #include <avr/sleep.h>
28
29 #ifndef F_OSC
```

```

30 #define F_OSC 32000000
31 #endif
32
33 ///GPIO FUNCTIONS
34 #define sbi(ADDRESS,BIT) ADDRESS.OUTSET = BIT // set Bit
35 #define cbi(ADDRESS,BIT) ADDRESS.OUTCLR = BIT // clear Bit
36 #define toggle(ADDRESS,BIT) ADDRESS.OUTTGL = BIT // toggle bit
37 //TODO FIX THIS
38 #define vsbi(ADDRESS,BIT) ADDRESS.OUTSET = BIT // set virtual port
    bit
39 #define vcbi(ADDRESS,BIT) ADDRESS.OUTCLR = BIT // clear virtual port
    bit
40 #define vtoggle(ADDRESS,BIT) ADDRESS.OUTTGL = BIT // toggle virtual
    port bit
41
42 #define interrupt ISR
43 #define OUTPUT 1
44 #define INPUT 0
45 #define HIGH 1
46 #define LOW 0
47
48
49 // GPIO CODE
50
51
52 void pinMode(PORT_t *port, uint8_t pin, uint8_t mode)
53 {
54
55     if(mode == 1)
56     {
57         port->DIRSET = pin;
58     }
59     else if (mode ==0)
60     {

```



```

61     port->DIRCLR = pin;
62 }
63 }
64
65
66 inline void digitalWrite(PORT_t *port, uint8_t pin, uint8_t value)
67 {
68     switch (value)
69     {
70
71         case LOW:
72             port->OUTCLR = pin;
73             break;
74
75         case HIGH:
76             port->OUTSET = pin;
77
78
79     }
80 }
81
82
83 inline uint8_t digitalRead(PORT_t *port, uint8_t pin)
84 {
85
86     return port->IN & pin;
87
88 }
89
90
91
92 void invert_pin(PORT_t *port, uint8_t pin)
93 {
94

```

```

95     switch (pin)
96     {
97
98         case 0:
99             port->PIN0CTRL |= PORT_INVEN_bm;
100            break;
101         case 1:
102             port->PIN1CTRL |= PORT_INVEN_bm;
103            break;
104         case 2:
105             port->PIN2CTRL |= PORT_INVEN_bm;
106            break;
107         case 3:
108             port->PIN3CTRL |= PORT_INVEN_bm;
109            break;
110         case 4:
111             port->PIN4CTRL |= PORT_INVEN_bm;
112            break;
113         case 5:
114             port->PIN5CTRL |= PORT_INVEN_bm;
115            break;
116         case 6:
117             port->PIN6CTRL |= PORT_INVEN_bm;
118            break;
119         case 7:
120             port->PIN7CTRL |= PORT_INVEN_bm;
121            break;
122     }
123
124
125 }
126
127 //          VIRTUAL PORTS CODE
128

```

```

129 #define map_to_vport0(port) PORTCFG.VPCTRLA =
    PORTCFG.VP02MAP##port##_gc;
130
131 #define map_to_vport1(port) PORTCFG.VPCTRLA =
    PORTCFG.VP13MAP##port##_gc;
132
133 #define map_to_vport2(port) PORTCFG.VPCTRLB =
    PORTCFG.VP02MAP##port##_gc;
134
135 #define map_to_vport3(port) PORTCFG.VPCTRLB =
    PORTCFG.VP13MAP##port##_gc;
136
137
138 //                SYSTEM CLOCK CODE
139
140
141 void setup_clock(uint8_t osc, uint8_t system_clk)
142 {
143
144
145
146     OSC.CTRL |= osc; //enable 32MHz oscillator
147     while (!(OSC.STATUS & osc)); //wait for stability
148     CCP = CCP_IOREG_gc; //secured access
149     CLK.CTRL = system_clk; //0x01; //choose this osc source as clk
150
151 }
152
153 void setup_osc(uint8_t osc)
154 {
155     OSC.CTRL |= osc;
156
157 }
158

```

```

159 //          SPIC CODE
160
161 #ifndef SPIC
162
163 void spic_enable_interrupt(uint8_t LVL)
164 {
165     SPIC.INTCTRL = LVL;
166
167 }
168
169
170
171 void spic_setup(uint8_t flags )
172 {
173
174     SPIC.CTRL = flags;
175     PORTC.DIRSET = (PIN5_bm); // set MOSI as output
176     PORTC.DIRSET = (PIN7_bm); //set clock as output
177     PORTC.DIRSET = (PIN4_bm); // SS out
178     PORTC.DIRCLR = (PIN6_bm);
179
180 }
181
182 void spic_enable()
183 {
184     SPIC.CTRL |= SPI_ENABLE_bm;
185 }
186
187 uint8_t spic_transmit(uint8_t data)
188 {
189
190     SPIC.DATA = data;
191     while (!(SPIC.STATUS & (1<<7)));
192     return SPIC.DATA;

```

```

193
194 }
195
196 uint32_t spic_transmit32_lsb(uint32_t data)
197 {
198
199
200
201 }
202
203 uint32_t spic_transmit32_msb(uint32_t data)
204 {
205     // big endian
206     uint32_t buffer32;
207     uint8_t    buffer;
208     cli();
209     //MSB first !
210     buffer = data>>24;
211     buffer=spic_transmit(buffer);
212     buffer32 = (uint32_t)buffer<<24;
213
214     buffer = data>>16;
215     buffer =spic_transmit(buffer);
216     buffer32 |= (uint32_t)buffer<<16;
217
218     buffer = data>>8;
219     buffer = spic_transmit(buffer);
220     buffer32 |= (uint32_t)buffer<<8;
221
222     buffer = data;
223     buffer = spic_transmit(buffer);
224     buffer32 |= buffer;
225
226     sei();

```

```

227
228         return buffer32;
229
230
231     }
232
233
234 #endif
235
236
237 //             Generic SPI CODE
238
239
240 void spi_enable_interrupt( SPI_t *SPI, uint8_t LVL)
241 {
242     SPI->INTCTRL = LVL;
243
244 }
245
246
247 void spi_setup(SPI_t * SPI,  PORT_t *PORT, uint8_t flags )
248 {
249
250     SPI->CTRL = flags;
251
252     PORT->DIRSET = (PIN5_bm); // set MOSI as output
253     PORT->DIRSET = (PIN7_bm); //set clock as output
254     PORT->DIRSET = (PIN4_bm); // SS out
255     PORT->DIRCLR =  (PIN6_bm);
256
257
258 }
259
260 void spi_enable( SPI_t *SPI)

```

```

261 {
262     SPI->CTRL |= SPI_ENABLE_bm;
263 }
264
265 inline uint8_t spi_transmit(SPI_t *SPI, uint8_t data)
266 {
267
268     SPI->DATA = data;
269     while (!(SPI->STATUS & (1<<7)));
270     return SPI->DATA;
271 }
272 }
273
274 uint32_t spid_transmit32_lsb(uint32_t data)
275 {
276
277
278
279 }
280
281 uint32_t spid_transmit32_msb(uint32_t data)
282 {
283     // big endian
284     uint32_t buffer32;
285     uint8_t    buffer;
286     cli();
287     //MSB first !
288     buffer = data>>24;
289     buffer=spic_transmit(buffer);
290     buffer32 = (uint32_t) buffer<<24;
291
292     buffer = data>>16;
293     buffer =spic_transmit(buffer);
294     buffer32 |= (uint32_t) buffer<<16;

```

```

295
296         buffer = data>>8;
297         buffer = spic_transmit(buffer);
298         buffer32 |= (uint32_t)buffer<<8;
299
300         buffer = data;
301         buffer = spic_transmit(buffer);
302         buffer32 |= buffer;
303
304         sei();
305
306         return buffer32;
307
308
309     }
310     //          Real Time Counter Code
311
312     typedef struct {
313     volatile   uint16_t year;
314     volatile   uint16_t days;
315     volatile   uint8_t  hours;
316     volatile   uint8_t  minutes;
317     volatile   uint8_t  seconds;
318
319     }   realtime_t;
320
321
322
323     void rtc_clock_init(uint8_t flags , uint16_t timeout) // timeout in 1sec
324     {
325
326     CLK.RTCCTRL = flags | CLK.RTCEN_bm;
327

```



```

328 RTC_INTCTRL |= RTC_OVFINTLVL_LO_gc;
329
330 while( ( RTC_STATUS & 0x01 ) );
331
332 RTC_PER = timeout - 1;
333 RTC_CTRL = RTC_PRESCALER_DIV1_gc;
334
335
336 }
337
338 realtime_t set_time(uint8_t days, uint8_t hours, uint8_t minutes,
    uint8_t seconds)
339 {
340
341     //rtc_clock_init(CLK_RTCSRC_RCOSC_gc, 1024);
342
343
344     realtime_t _time;
345
346     _time.seconds = seconds;
347     _time.minutes = minutes;
348     _time.hours = hours;
349     _time.days = days;
350
351 return _time;
352
353 }
354
355
356 void timekeeper_init()
357 {
358
359 rtc_clock_init(CLK_RTCSRC_RCOSC_gc, 1024);
360

```

```

361 }
362 #define timekeeper() realtime_t real_time; \
363 ISR(RTC_OVF_vect) \
364 { \
365     real_time.seconds++; \
366     \
367     if(real_time.seconds > 59)\
368     {\
369         real_time.seconds = 0;\
370         real_time.minutes++; \
371     }\
372     \
373     \
374     if(real_time.minutes>59)\
375     {\
376         real_time.minutes = 0;\
377         real_time.hours++; \
378     }\
379     else {} \
380     \
381     if(real_time.hours>24)\
382     {\
383         real_time.hours = 0;\
384         real_time.days++; \
385     }\
386     else {} \
387     \
388     if(real_time.days>365)\
389     {\
390         real_time.days = 0;\
391         real_time.year++; \
392     }\
393     else {} \
394     \

```

```

395 PORTB.OUTTGL = PIN0_bm;\
396 }
397
398 //                Interrupt Controller Code
399
400
401
402
403 void enable_all_interrupts()
404 {
405
406 PMIC_CTRL |= PMIC_LOLVLEN_bm | PMIC_MEDLVLEN_bm | PMIC_HILVLEN_bm;
407 sei();
408
409 }
410
411 void round_robin_interrupt()
412 {
413
414 PMIC_CTRL |= PMIC_RREN_bm;
415
416
417 }
418
419 void enable_interrupts(uint8_t flags)
420 {
421
422 PMIC_CTRL = flags;    // careful! this resets the PMIC reg.
423 sei();
424
425 }
426
427
428

```

```

429
430 void enable_external_interrupt0(PORT_t *port, uint8_t LVL, uint8_t pins)
431 {
432
433     port->INTCTRL = LVL;
434
435     port->INT0MASK = pins;
436
437 }
438
439
440 // example    enable_external_interrupt0(&PORTC, PORT_INT0LVL_LO_gc,
441     PIN0_bm | PIN1_bm)
442 // this setup interrupt on PC0 and PC1
443
444
445 void enable_external_interrupt1(PORT_t *port, uint8_t LVL, uint8_t pins)
446 {
447
448     port->INTCTRL = LVL;
449
450     port->INT1MASK = pins;
451
452 }
453
454
455 void set_pin_config(PORT_t *port, uint8_t config_output, uint8_t sense,
456     uint8_t pin)
457 {
458     switch(pin)
459     {
460         case 0:

```

```

461     port->PIN0CTRL = config_output | sense;
462     break;
463
464     case 1:
465     port->PIN1CTRL = config_output | sense;
466     break;
467
468     case 2:
469     port->PIN2CTRL = config_output | sense;
470     break;
471
472     case 3:
473     port->PIN3CTRL = config_output | sense;
474     break;
475
476     case 4:
477     port->PIN4CTRL = config_output | sense;
478     break;
479
480     case 5:
481     port->PIN5CTRL = config_output | sense;
482     break;
483
484     case 6:
485     port->PIN6CTRL = config_output | sense;
486     break;
487
488     case 7:
489     port->PIN7CTRL = config_output | sense;
490     break;
491
492 }
493
494

```

```

495
496 }
497
498
499 //                      Delay   Code
500
501 #define delay(time) _delay_ms(time)
502
503
504 #define delay_microseconds(time) _delay_us(time)
505
506
507
508 //                      Timer0 Controller Code
509
510 #ifdef AVR_XMEGA
511 #define delay_ms(ms) _delay_ms(ms);
512 #endif
513
514
515
516 void timer0_setup( uint8_t clkssel, uint8_t waveformflag, uint8_t event,
                    uint8_t bytemode )
517 {
518     TCC0.CTRLA = 0;
519     TCC0.CTRLB = event;
520     TCC0.CTRLC = bytemode;
521     TCC0.CTRLD = waveformflag;
522     TCC0.CTRLE = clkssel;
523
524 }
525
526
527

```

```

528 inline void timer0_set_period(uint16_t period)
529 {
530     TCC0.PER=period;
531 }
532
533 inline void timer0_set_OCA(uint16_t oc)
534 {
535     TCC0.CCA=oc;
536 }
537
538 inline void timer0_set_OCB(uint16_t oc)
539 {
540     TCC0.CCB=oc;
541 }
542 inline void timer0_set_OCC(uint16_t oc)
543 {
544     TCC0.CCC=oc;
545 }
546 inline void timer0_set_OCD(uint16_t oc)
547 {
548     TCC0.CCD=oc;
549 }
550
551
552
553
554
555
556 // timer0 PWM code
557
558
559 void timer0_setup_pwm(uint8_t divisor, uint8_t output_ocr)
560 {
561     timer0_setup(divisor, output_ocr | TC_WGMODE_SINGLESLOPE_gc,

```

```

    0,TC_BYTEM_NORMAL_gc);
562 }
563
564
565
566
567
568 // calculate closest frequency to a float value of freq:
569 void set_timer0_pwm_freq(float freq, uint8_t speed) // percent is
570 {
571
572     uint16_t divisor = 1;
573
574     switch (speed)
575     {
576         case TC_CLKSEL_DIV1_gc:
577             divisor = 1;
578             break;
579         case TC_CLKSEL_DIV2_gc:
580             divisor = 2;
581             break;
582
583         case TC_CLKSEL_DIV4_gc:
584             divisor = 4;
585             break;
586
587         case TC_CLKSEL_DIV8_gc:
588             divisor = 8;
589             break;
590
591         case TC_CLKSEL_DIV64_gc:
592             divisor = 64;
593             break;
594

```



```

595     case TC_CLKSEL_DIV256_gc:
596         divisor = 256;
597         break;
598
599     case TC_CLKSEL_DIV1024_gc:
600         divisor = 1024;
601         break;
602 }
603
604
605 TCC0.PER = (uint16_t) (F_OSC/(divisor*(freq-0.03*freq)))+1; // see
        data sheet for equation
606 TCC0.CCA = TCC0.PER/2;
607
608 }
609
610
611
612
613
614 void set_timer0_pwm_percent_OCA(float percent)
615 {
616
617 TCC0.CCA = (uint16_t) TCC0.PER*(percent/99);
618
619 }
620
621 void set_timer0_pwm_percent_OCB(float percent)
622 {
623
624 TCC0.CCB = (uint16_t) TCC0.PER*(percent/99);
625
626 }
627

```

```

628 void set_timer0_pwm_percent_OCC(float percent)
629 {
630
631 TCC0.CCC = (uint16_t) TCC0.PER*(percent/99);
632
633 }
634
635 void set_timer0_pwm_percent_OCD(float percent)
636 {
637
638 TCC0.CCD = (uint16_t) TCC0.PER*(percent/99);
639
640 }
641
642 //                      ADC Controller Code
643
644 //TODO                set up ADC
645
646 uint8_t ReadSignatureByte(uint16_t Address)
647 {
648     NVMCMD = NVMCMD_READ_CALIB_ROW_gc;
649     uint8_t Result;
650     __asm__ ("lpm %0, Z\n" : "=r" (Result) : "z" (Address));
651     NVMCMD = NVMCMD_NO_OPERATION_gc;
652     return Result;
653 }
654
655
656
657
658 void adc_setup(uint8_t mode, uint8_t prescaler )
659 {
660 ADCA.CTRLA = ADC_ENABLE_bm ; // Enable the ADC
661

```

```

662 }
663
664 void adc_control_ch0();
665
666 /*
667
668 if ((ADCA.CTRLA & ADC_ENABLE_bm) == 0)
669     {
670         ADCA.CTRLA = ADC_ENABLE_bm ; // Enable the ADC
671         ADCA.CTRLB = (1<<4); // Signed Mode
672         ADCA.REFCTRL = 0; // Internal 1v ref
673         ADCA.EVCTRL = 0 ; // no events
674         ADCA.PRESCALER = ADC_PRESCALER_DIV32_gc ;
675         ADCA.CALL = ReadSignatureByte(0x20) ; //ADC Calibration Byte 0
676         ADCA.CALH = ReadSignatureByte(0x21) ; //ADC Calibration Byte 1
677         //ADCA.SAMPCTRL = This register does not exist
678         // _delay_us(400); // Wait at least 25 clocks
679     }
680     ADCA.CH0.CTRL = ADC_CH_GAIN_1X_gc | ADCMode ; // Gain = 1, Single
        Ended
681     ADCA.CH0.MUXCTRL = (Channel<<3);
682     ADCA.CH0.INTCTRL = 0 ; // No interrupt
683     //ADCA.CH0.SCAN Another bogus register
684
685
686 */
687
688
689 uint16_t ReadADC(uint8_t Channel, uint8_t ADCMode) // Mode = 1 for
        single ended, 0 for internal
690 {
691     if ((ADCA.CTRLA & ADC_ENABLE_bm) == 0)
692     {
693         ADCA.CTRLA = ADC_ENABLE_bm ; // Enable the ADC

```

```

694     ADCA.CTRLB = (1<<4); // Signed Mode
695     ADCA.REFCTRL = 0; // Internal 1v ref
696     ADCA.EVCTRL = 0 ; // no events
697     ADCA.PRESCALER = ADC_PRESCALER_DIV32_gc ;
698     ADCA.CALL = ReadSignatureByte(0x20) ; //ADC Calibration Byte 0
699     ADCA.CALH = ReadSignatureByte(0x21) ; //ADC Calibration Byte 1
700     //ADCA.SAMPCTRL = This register does not exist
701     _delay_us(400); // Wait at least 25 clocks
702 }
703 ADCA.CH0.CTRL = ADC_CH_GAIN_1X_gc | ADCMode ; // Gain = 1, Single
    Ended
704 ADCA.CH0.MUXCTRL = (Channel<<3);
705 ADCA.CH0.INTCTRL = 0 ; // No interrupt
706 //ADCA.CH0.SCAN Another bogus register
707 for(uint8_t Waste = 0; Waste<2; Waste++)
708 {
709     ADCA.CH0.CTRL |= ADC_CH_START_bm; // Start conversion
710     while (ADCA.INTFLAGS==0) ; // Wait for complete
711     ADCA.INTFLAGS = ADCA.INTFLAGS ;
712 }
713 return ADCA.CH0RES;
714 }
715
716 //          USART Controller Code
717
718 //&USARTC0
719
720 void usart_setup(USART_t *uart, uint8_t ctrlb_flags, uint8_t mode,
    uint16_t baud)
721 {
722
723
724     //Disable interrupts, just for safety
725     uart->CTRLA = 0;

```

```

726
727     uart->BAUDCTRLB = 0; //Just to be sure that BSCALE is 0
728     uart->BAUDCTRLA = 0x00ff & baud; // 207
729
730     uart->CTRLC = mode;
731     //Enable receive and transmit
732     uart->CTRLB |= USART_TXEN_bm | USART_RXEN_bm | ctrlb_flags; // And
    enable high speed mode
733
734
735 }
736
737
738
739 #define USART_CTRLC_UDORD_bm 4
740 #define USART_SPI_MSB_bm 0
741 #define USART_SPI_LSB_bm 1
742
743
744 void usart_set_as_spi(USART_t *uart, PORT_t *port, uint8_t clock_pin,
    uint8_t mode, uint8_t dord, uint8_t div)
745 { // baud was 159
746     usart_setup(uart, 0, USART_CMODE_MSPI_gc, div);
747     //uart->CTRLC |= 4; // set UDORD // DORD bit // LSB first
748
749     switch (dord)
750     {
751     case 0:
752         break;
753     case 1:
754         uart->CTRLC |= 4; // set UDORD // DORD bit // LSB first
755         break;
756     }
757

```

```

758
759
760     switch (mode)
761     {
762
763     case 0:
764         // INVEN bit = 0 UCPHA = 0
765         break;
766     case 1:
767         // INVEN = 0    UCPHA = 1
768         uart->CTRLC |= 2; // set UCPHA // CPHA bit
769         break;
770     case 2:
771         invert_pin(port, clock_pin);
772         break;
773
774     case 3:
775         uart->CTRLC |= 2; // set UCPHA // CPHA bit
776         invert_pin(port, clock_pin);
777         break;
778
779     }
780
781
782 }
783
784
785 char usart_transmit(USART_t *MYUSART, char data)
786 {
787
788     while( !(MYUSART->STATUS & USART_DREIF_bm) ); //Wait until DATA
789         buffer is empty
790         MYUSART->DATA = data;

```

```

791     while( !(MYUSART->STATUS & USART_RXCIF_bm) ); //Interesting DRIF
       didn't work.
792     return MYUSART->DATA;
793
794
795
796
797 }
798
799
800 void usart_send_byte(USART_t *MYUSART, char c )
801 {
802
803     while( !(MYUSART->STATUS & USART_DREIF_bm) ); //Wait until DATA
       buffer is empty
804
805     MYUSART->DATA = c;
806
807 }
808
809 char usart_receiveByte(USART_t *MYUSART)
810 {
811     while( !(MYUSART->STATUS & USART_RXCIF_bm) ); //Interesting DRIF
       didn't work.
812     return MYUSART->DATA;
813 }
814
815
816 void usart_send_string(USART_t *MYUSART, char *text)
817 {
818     while(*text)
819     {
820         usart_send_byte(MYUSART,*text++);
821     }

```

```

822 }
823
824
825 //          DMA CODE
826
827 #ifdef DMA
828
829 void dma_init()
830 {
831
832 DMA_CTRL = DMA_ENABLE_bm;
833 //set the burst length to 1 byte
834 DMA_CH0_CTRLA = ( DMA_CH_SINGLE_bm | DMA_CH_BURSTLEN_1BYTE_gc );
835 //set the following: source address incremented, reload after each
    block; destination address fixed (reload after each block)
836 DMA_CH0_ADDRCTRL = ( DMA_CH_SRCRELOAD_TRANSACTION_gc |
    DMA_CH_SRCDIR_INC_gc | DMA_CH_DESTRELOAD_TRANSACTION_gc |
    DMA_CH_DESTDIR_FIXED_gc );
837
838 }
839
840 #define dma_set_block_size(ch,size) ch.TRFCNT = size;
841
842 #define dma_set_trigger_source(ch,source) ch.TRIGSRC = source;
843
844 #define dma_set_source_address(ch,address) \
845 ch.SRCADDR0 = ( (uint16_t) address >> 0 ) & 0xFF; \
846 ch.SRCADDR1 = ( (uint16_t) address >> 8 ) & 0xFF; \
847 ch.SRCADDR2 = 0x00; \
848
849
850 #define dma_set_dest_address(ch,address) ch.DESTADDR0 = ( (uint16_t)
    address >> 0 ) & 0xFF; \
851 ch.DESTADDR1 = ( (uint16_t) address >> 8 ) & 0xFF; \

```



```

852 ch.DESTADDR2 = 0x00; \
853
854
855
856
857
858 void dma_start_transfer(DMA_CH_t *ch) //usage: start_transfer(&DMA.CH0)
859 {
860
861 if (!(ch->CTRLB & DMA.CH.CHBUSY_bm))
862 {
863 //DMA.CH0.CTRLA |= 0b10000000;
864 ch->CTRLA |= DMA.CH.ENABLE_bm;
865
866 }
867
868 }
869
870
871
872
873
874
875 #endif
876
877 // #define dma_setup(channel, flags) DMA.channel.CTRLA = flags;
878
879 // address is a 16bit variable
880
881
882 //DMA.channel.SRCADDR0 = ((uint16_t)(address) >> 0U) & 0xFFU; \
883 //DMA.channel.SRCADDR1 = ((uint16_t)(address) >> 8U) & 0xFFU; \
884 //DMA.channel.SRCADDR2 = ((uint32_t)(address) >> 16) & 0xFFU; \
885

```

```

886 // #define dma_set_dest_addr(channel, address) DMA.channel.DESTADDR0 =
      ((uint16_t)(address) >> 0U) & 0xFF;\
887 DMA.CH0.DESTADDR1 = ((uint16_t)(address) >> 8U) & 0xFF;\
888 DMA.CH0.DESTADDR2 = ((uint32_t)(address) >> 16) & 0xFF;
889
890
891
892
893 // #define dma_enable() DMA.CTRL = DMA_ENABLE_bm;
894
895
896
897 #endif
898 // NOW FOR 32BIT AVR DEVICES
899 #ifdef AVR_32
900 //          GPIO CODE
901
902
903
904 void pinMode(PORT_t *port, uint8_t pin, uint8_t mode)
905 {
906
907     if(mode == 1)
908     {
909         port->ovrs = pin;
910     }
911     else if (mode == 0)
912     {
913         port->ovrc = pin;
914     }
915 }
916
917
918 inline void digitalWrite(PORT_t *port, uint8_t pin, uint8_t value)

```

```

919 {
920     switch (value)
921     {
922
923         case LOW:
924             port->ovrs = pin;
925             break;
926
927         case HIGH:
928             port->ovrc = pin;
929
930
931     }
932 }
933
934
935 inline uint8_t digitalRead(PORT_t *port, uint8_t pin)
936 {
937
938     return port->IN & pin;
939
940 }
941 #endif
942 #endif

```

Appendix G: Pinger Controller Firmware

The Following Code is used to control the pinger power circuits. This code relies on the "Easy AVR" library also included in this work.

```
1
2 #include <avr/io.h>
3 #include <avr/interrupt.h>
4 #include <inttypes.h>
5 #include <util/delay.h>
6 #include <avr/sleep.h>
7
8 #include "easy_avr.h"
9
10
11 #define PING_SIG PD0
12 #define DIG_POS_SIG PA1
13 #define ANA_POS_SIG PA0
14
15
16
17 void ping();
18
19 int main()
20 {
21
22     setup_clock(OSC_RC32MEN_bm, CLK_SCLKSEL_RC32M_gc);
23
24     uint8_t input_ping; // input signal toggle
25
26     pinMode(&PORTA, PIN1_bm | PIN0_bm , INPUT);
```

```

27  pinMode(&PORTD, PIN0_bm, INPUT);
28  pinMode(&PORTC, PIN0_bm | PIN1_bm | PIN2_bm | PIN3_bm, OUTPUT);
29
30  digitalWrite(&PORTC, PIN0_bm, LOW);    //channel A off
31  digitalWrite(&PORTC, PIN1_bm, LOW);    //channel A off
32
33  digitalWrite(&PORTC, PIN2_bm, LOW);    //channel B off
34  digitalWrite(&PORTC, PIN3_bm, LOW);    //channel B off
35
36  //digitalWrite(&PORTA, PIN3_bm, HIGH);
37
38  while(1)
39  {
40      //read signal input
41      uint8_t input_ping = digitalRead(&PORTD, PIN0_bm);
42      //while(input_ping == LOW);
43
44      if(input_ping==HIGH)
45      {
46          ping();
47          delay_ms(1000);
48      }
49
50  }
51
52
53
54
55  return 0;
56  }
57
58  void ping()
59  {
60

```

```

61 #define ontime 100
62 #define delaytime 50
63 /*#define A1 PC0
64 #define A2 PC1
65 #define B1 PC2
66 #define B2 PC3
67 */
68 digitalWrite(&PORTC,PIN0_bm,HIGH); //channel A + polarity
69
70 //while(digitalRead(&PORTA,PIN1_bm,LOW)); // wait for pinger posistion
    signal
71 delay(ontime);
72
73 digitalWrite(&PORTC,PIN0_bm,LOW); //channel A off
74 digitalWrite(&PORTC,PIN1_bm,LOW); //channel A off
75
76 delay(delaytime); // program this delay based on test material
77
78 digitalWrite(&PORTC,PIN1_bm,HIGH); //channel A- polarity
79
80 delay(ontime);
81 digitalWrite(&PORTC,PIN1_bm,LOW); // shutdown device
82 digitalWrite(&PORTC,PIN0_bm,LOW);
83
84
85 }

```

Appendix H: Fiber Position Firmware

The Following Code is used to control the fiber positioner ethernet interface and motion control software. This code relies on the the "arduino ethernet" library.

```
1 #include <SPI.h>
2 #include <Ethernet.h>
3 #include <stdlib.h>
4 #include "fiberwebserver.h"
5
6 void setup(){
7     //enable serial monitor
8     // Serial.begin(9600);
9     //start Ethernet
10    setup_motion();
11    Ethernet.begin(mac);
12
13    //Ethernet.begin(mac, ip, gateway, subnet);
14    //initialize variables
15    HttpHeader="";
16
17 }
18
19
20
21 void loop()
22 {
23     // Create a client connection
24     EthernetClient client = server.available();
25     if (client) {
26         while (client.connected()) {
```

```

27     if (client.available()) {
28         char c = client.read();
29         //read MaxHeaderLength number of characters in the HTTP header
30         //discard the rest until \n
31         if (HttpHeader.length() < MaxHeaderLength)
32         {
33             //store characters to string
34             HttpHeader = HttpHeader + c;
35         }
36
37         // Serial.print(HttpHeader);
38
39         //if HTTP request has ended
40         if (c == '\n') {
41             // show the string on the monitor
42             // Serial.println(HttpHeader);
43
44             //parse http header to get X and Y values
45             parse_http_header(HttpHeader.c_str());
46             // delay(1000);
47             if(HttpHeader[5] != 'f') // ignore favicon request from
browser
48             {
49                 print_webpage();
50             }
51
52             //clearing string for next read
53             HttpHeader="";
54             //stopping client
55             client.stop();
56         }
57     }
58 }
59 }

```



```

60 Ethernet.maintain();
61 }

1
2 #ifndef FIBER_MOTION_H
3 #define FIBER_MOTION_H
4
5 #define STEP_X_PIN 7
6 #define STEP_Y_PIN 5
7
8 #define DIR_Y_PIN 4
9 #define DIR_X_PIN 6
10
11 #define STEPS_PER_PIXEL 100
12
13
14
15 volatile bool do_motion = false;
16
17 volatile uint32_t Xsteps;
18 volatile uint32_t Ysteps;
19
20 volatile uint32_t currentXsteps;
21 volatile uint32_t currentYsteps;
22
23
24 volatile int currentX =0;
25 volatile int currentY =0;
26
27 volatile int currentX_dir;
28 volatile int currentY_dir;
29
30
31

```

```

32 void setup_motion()
33 {
34     pinMode(STEP_X_PIN, OUTPUT);
35     pinMode(STEP_Y_PIN, OUTPUT);
36     pinMode(DIR_X_PIN, OUTPUT);
37     pinMode(DIR_Y_PIN, OUTPUT);
38
39     digitalWrite(DIR_Y_PIN, LOW);
40     digitalWrite(DIR_X_PIN, LOW);
41
42
43
44
45
46     noInterrupts();           // disable all interrupts
47     TCCR1A = 0;
48     TCCR1B = 0;
49     TCNT1  = 0;
50
51     OCR1A = 2048;             // preload timer 65536-16MHz/256/2Hz
52     TCCR1B |= (1<<CS10);
53     TCCR1B |= (1 << WGM12);   // 256 prescaler
54     TIMSK1 |= (1 << OCIE1A);  // enable timer overflow interrupt
55     interrupts();             // enable all interrupts
56
57
58
59
60
61 }
62
63 volatile void set_x_dir_neg()
64 {
65     digitalWrite(DIR_X_PIN, LOW);

```

```

66 }
67 volatile void set_x_dir_pos()
68 {
69     digitalWrite(DIR_X_PIN,HIGH);
70 }
71 volatile void set_y_dir_neg()
72 {
73     digitalWrite(DIR_Y_PIN,LOW);
74 }
75 volatile void set_y_dir_pos()
76 {
77     digitalWrite(DIR_Y_PIN,HIGH);
78 }
79
80
81 volatile void step_x()
82 {
83     digitalWrite(STEP_X_PIN,HIGH);
84     delayMicroseconds(10);
85     digitalWrite(STEP_X_PIN,LOW);
86     delayMicroseconds(10);
87 }
88
89 volatile void step_y()
90 {
91     digitalWrite(STEP_Y_PIN,HIGH);
92     delayMicroseconds(10);
93     digitalWrite(STEP_Y_PIN,LOW);
94     delayMicroseconds(10);
95 }
96
97
98 volatile void timer_isr()
99 {

```

```

100
101 if( (currentXsteps<Xsteps || currentYsteps<Ysteps) && do_motion)
102 {
103
104 if(currentXsteps<Xsteps)
105 {
106     step_x();
107     currentXsteps++;
108 }
109
110 if(currentYsteps<Ysteps)
111 {
112     step_y();
113     currentYsteps++;
114 }
115
116     do_motion = true;
117 }else
118 {
119     do_motion = false;
120
121 }
122
123 }
124
125 ISR(TIMER1_COMPA_vect)
126 // interrupt service routine that wraps a
127 user defined function supplied by attachInterrupt
128 {
129     timer_isr();
130 }
131
132
133

```

```

134 void compute_line(int16_t xstart, int16_t ystart, int16_t x, int16_t y)
135 {
136     int32_t delX = (x-xstart); // *STEPS_PER_PIXEL;
137     int32_t delY = (y-ystart); // *STEPS_PER_PIXEL;
138     // Serial.println(xstart);
139     // Serial.println(x);
140     // Serial.println(delX);
141     // Serial.println(delX*100);
142
143     if (delX < 0)
144     {
145         set_x_dir_neg();
146         // Serial.println("X neg");
147     } else {
148         set_x_dir_pos();
149     }
150
151     if (delY < 0)
152     {
153         set_y_dir_neg();
154         // Serial.println("Y neg");
155     } else {
156         set_y_dir_pos();
157     }
158     Xsteps = abs(delX)*STEPS_PER_PIXEL;
159     Ysteps = abs(delY)*STEPS_PER_PIXEL;
160
161     // Serial.println(Xsteps);
162     // Serial.println(Ysteps);
163     currentXsteps = 0;
164     currentYsteps = 0;
165     do_motion = true;
166
167 }

```

```

168
169 #endif

1 #ifndef FIBER_WEBSERVER_H
2 #define FIBER_WEBSERVER_H
3 #include "fibermotion.h"
4
5 #define MaxHeaderLength 32 //maximum length of http header required
6
7 byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
8
9 EthernetServer server(80); //web server port
10 String HttpHeader = String(MaxHeaderLength);
11
12 int currentX_display;
13 int currentY_display;
14
15 void print_webpage();
16 void parse_http_header(char *http_header);
17
18
19 void print_webpage()
20 {
21     // start of web page
22     server.println("<html lang='en-US'> <head> <title> Fiber Positioner
    Interface </title> <meta http-equiv='cache-control '
    content='no-cache ' />    </head><body>");
23     server.println();
24     server.print("<style>input [ type=text ] , select
    { width:100%;padding:12px20px;margin:8px0;");
25     server.print(" display :
    inline-block;border:1pxsolid#ccc;border-radius:4px;
26     box-sizing:border-box;}");
27     server.print("input [ type=submit ]

```

```

    {width:100%;background-color:#4CAF50;color:white;padding:14px20px;");
28     server.print("margin:8px0;border:none;border-radius:4px;
29     cursor:pointer;}");
30     server.print("input[type=submit]:hover{background-color:#45a049;}");
31     server.print("div {border-radius:5px;background-color:
    #f2f2f2;padding:20px;}</style>");
32     server.println("<div>");
33     server.println("<p> Enter the X and Y coordinate to move fiber.</p>");
34     server.print("<form
    style='color:blue;margin-left:10px;font-size:20px' method=get>");
35     server.print(" X coordinate: <input type='text' name='X' value='');
    server.print(currentX); server.print("><br>");
36     server.print("Y coordinate: <input type='text' name='Y' value='');
    server.print(currentY); server.print("><br>");
37     server.print("<input size type=submit value='Move Fiber'></form>");
38     server.println("</div>");
39     server.print("</body></html>");
40     //end of web page
41 }
42
43
44 void parse_http_header(char * http_header)
45 {
46     uint8_t length = strlen(http_header);
47
48     uint8_t begin_data = 0;
49     uint8_t x_location = 0;
50     uint8_t y_location = 0;
51     uint8_t end_data = 0;
52
53     bool found_data = false;
54
55     for(int i = 0; i<length-1; i++)
56     {

```

```

57     if(http_header[i] == 'X')
58     {
59         x_location = i+2;
60         found_data = true;
61         break;
62     }
63 }
64
65 for(int i = 0; i<length-1; i++)
66 {
67     if(http_header[i] == 'Y')
68     {
69         y_location = i+2;
70         break;
71     }
72 }
73
74 // at this point all locations of X and Y
75 //have been found except the full length
76
77
78 for(int i = 0; i<length-1; i++)
79 {
80     if(http_header[i] == 'H')
81     {
82         end_data = i;
83         break;
84     }
85 }
86
87
88
89 char x[10]= "";
90 char y[10] = "";

```



```

91
92 //int y_location-3
93 if(found_data)
94 {
95     strncpy(x,&http_header[x_location],((y_location-3)-x_location));
96     strncpy(y,&http_header[y_location],(end_data-y_location));
97
98     int16_t x_int;
99     int16_t y_int;
100
101     //set x y values.
102     x_int = atoi(x);
103     y_int = atoi(y);
104
105     compute_line(currentX,currentY,x_int,y_int);
106     while(do_motion){} //wait while in motion
107     currentX = x_int;
108     currentY = y_int;
109
110     //Serial.println(currentX);
111     //Serial.println(currentY);
112 }
113 }
114
115 #endif

```

CURRICULUM VITAE

Elijah Robert Jensen

Education

Ph.D., University of Louisville, Louisville, KY, August 2018

M.S. , University of Louisville, Louisville, KY, May 2015

B.S. Austin Peay State University, Clarksville, TN, May 2012

Professional Positions

Graduate Research Assistant 2013-2018

Graduate Teaching Assistant UofL, 2014-2015

Graduate Teaching Assistant UVM, 2012-2013

Thin Film Lab Assistant, 2012-2013 Dr. Walter Varhue

Honors and Awards

2015 Recipient of Iyad Khair Award for Excellence in Physics

2011 Inducted into Pi Mu Epsilon Math Society

2010 Inducted into Phi Kappa Phi, Honorary Society