University of Rhode Island

DigitalCommons@URI

Open Access Master's Theses

2014

PORTABLE MICROFLUIDIC FLUID HANDLING SYSTEM FOR THE DETECTION OF PROTEIN BIOMARKERS IN WHOLE HUMAN BLOOD

Michael Franzblau

University of Rhode Island, explosivelements@gmail.com

Follow this and additional works at: <https://digitalcommons.uri.edu/theses>

Recommended Citation

Franzblau, Michael, "PORTABLE MICROFLUIDIC FLUID HANDLING SYSTEM FOR THE DETECTION OF PROTEIN BIOMARKERS IN WHOLE HUMAN BLOOD" (2014). *Open Access Master's Theses*. Paper 458.
<https://digitalcommons.uri.edu/theses/458>

This Thesis is brought to you for free and open access by DigitalCommons@URI. It has been accepted for inclusion in Open Access Master's Theses by an authorized administrator of DigitalCommons@URI. For more information, please contact digitalcommons-group@uri.edu.

PORTABLE MICROFLUIDIC FLUID HANDLING SYSTEM FOR THE
DETECTION OF PROTEIN BIOMARKERS IN WHOLE HUMAN BLOOD

BY

MICHAEL FRANZBLAU

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

IN

MECHANICAL ENGINEERING AND APPLIED MECHANICS

UNIVERSITY OF RHODE ISLAND

2014

MASTER OF SCIENCE THESIS
OF
MICHAEL FRANZBLAU

APPROVED:

Thesis Committee:

Major Professor Mohmmammad Faghri

Constantine Anagnostopoulos

Yana Reshetnyak

Nasser H. Zawia
DEAN OF THE GRADUATE SCHOOL

UNIVERSITY OF RHODE ISLAND
2014

ABSTRACT

The thesis focused on the design and construction of a proof-of-concept polymer based microfluidic Point of Care (POC) device that is portable, requires little user interaction, and is sensitive enough to reliably test for protein biomarkers from a small sample of whole human blood. Specific contributions to the presented system include novel microfluidic channel geometries that support improved fluidic handling, and an improved technique for the fabrication of the poly(dimethylsiloxane) (PDMS) microfluidic chip. Additionally, a commercially available plasma separation membrane was integrated into the microfluidic chip using presented structures and techniques to achieve cellular filtration of whole blood on-chip, yielding a usable volume of plasma to be used as the analyte fluid. A system of valves was used to direct the pressures generated by a new patented syringe-style pump designed for this specific application. The pump was created to more precisely move the collection of fluid volumes through the chip. A previously developed charged couple device (CCD) based vision system was modified and integrated into the presented fluid control system to track the position of the various fluid volumes. Software was developed that used the fluid position sensors data to control the various hardware components of the system. Finally, the above components were combined into a working prototype geared towards ease of use and portability. Efforts were focused on the system's ability to autonomously and reliably control the various microliter sized volumes of fluid within the microfluidic chip.

C-reactive protein (CRP) was used as a benchmark protein for biomarker detection because its normal concentration in human blood is at an approximate

median to other biomarkers, and because the detection of CRP in a laboratory setting has been well established. A limit of detection (LOD) of 1.87 $\mu\text{g/mL}$ of CRP in buffer was achieved utilizing the proposed stop-flow sequential loading technique. An acceptable volume of plasma was extracted and collected in the microfluidic chip from 50 μL of whole blood using the structures and techniques presented in this work. The CCD based fluid position sensor was capable of sensing a change in the contents of a channel over 97% of the time. Totally autonomous stop-flow fluid handling was accomplished at a similar rate of success utilizing a combination of valves, the custom pump, and the fluid position sensor all working in unison.

Results show that protein detection using a previously untested stop-flow procedure still provided an acceptable LOD. The presented portable POC device was able to execute fluid handling in the microfluidic channels at a higher level of precision and reliability than previously possible. The presented prototype gives a viable blueprint for a next-generation protein biomarker detector capable of autonomous complex fluid handling procedures.

ACKNOWLEDGMENTS

I would like to thank my advisor, Dr. Mohammad Faghri, who actively supported me from the time I started working in his microfluidics laboratory, as an undergraduate student, to the completion of this project as a graduate student. His enthusiasm for this work and his students created an atmosphere that was not only enjoyable to work in, but yielded exciting and innovative results. Additionally, I would like to thank Dr. Constantine Anagnostopoulos for his guidance and leadership within the microfluidics laboratory. I would to like to thank my laboratory colleagues Dr. Peng Li, Michael Godfrin, Dr. John D. Jones, Nick DeFilippo, Jeremy Cogswell, Manuel Muller, and David Ferreira, who often provided sound advice and essential expertise. The work I have detailed in this thesis was made possible in large part thanks to the work previously completed by the aforementioned people, their worked served as a foundation upon which I built. Special thanks to the fanatical support provided by the University of Rhode Island Mechanical Engineering department and department chair, Dr. David Taggart, during my time as a graduate student.

Finally, I would like to thank my family for their love, support, and encouragement while I worked towards my degree. Additional thanks to Robert and Sanford Franzblau who ably assisted me during the editing process.

TABLE OF CONTENTS

ABSTRACT	ii
ACKNOWLEDGMENTS	iv
TABLE OF CONTENTS.....	v
LIST OF TABLES	ix
LIST OF FIGURES	x
 CHAPTER 1	
INTRODUCTION.....	1
1.1 Medical Diagnostics and Point-of-Care	1
1.2 Plasma Extraction from Whole Blood	7
1.3 High Abundance Protein Depletion	8
1.4 Fluid Control in Microchannels	9
1.5 Objective	10
 CHAPTER 2	
LITERATURE REVIEW.....	14
2.1 Plasma Separation from Whole Blood	14
2.1.1 Centrifugal Techniques	14
2.1.2 Bifurcation Separation.....	15
2.1.3 Dielectrophoresis Separation.....	16
2.1.3 Membrane Separation.....	17
2.2 High Abundance Protein Depletion	18
2.3 Fluid Control in Microchannels	22
2.3.1 Passive Fluid Control	22
2.3.2 Active Fluid Control.....	24
2.4 Point-of-Care Diagnostics	28

CHAPTER 3	
METHODOLOGY.....	33
3.1 Microfluidic Chip.....	33
3.1.1 Fabrication of Photolithography Molds.....	34
3.1.2 Fabrication of Machined Molds	36
3.1.3 Fabrication of PDMS Microfluidic Chips	40
3.1.4 Microfluidic Chip Design.....	45
3.2 Cell Removal from Whole Blood Sample.....	57
3.2.1 Initial Testing of Plasma Separation Membrane	57
3.2.2 Whole Blood Filtration Integration	59
3.3 Pump.....	63
3.3.1 Piezo Actuated Membrane Pump	63
3.3.2 Dual Chamber Fixed Volume Syringe	66
3.3.3 Dual Chamber Fixed Volume Syringe V2 and Driver	69
3.4 Protein Detection in Microchannels	75
3.4.1 Functionalization of the Detection Site	75
3.4.2 Functionalization of the High Abundance Protein Depletion Sites.....	77
3.4.3 Immunoassay	78
3.4.4 Florescent Signal Detection.....	81
3.5 Fluid Handling.....	82
3.5.1 Fluid Control using External Valves	83
3.5.2 Fluid Position Sensor.....	88
3.5.3 Bench-Top Experimental Setup for Fluid Handling	92
3.6 Portable Unit	94
3.6.1 Device Shell and Internal Configuration	94
3.6.2 Power Supply and Electrical Wiring	109

3.6.3 Programing and User Interface.....	113
CHAPTER 4	
FINDINGS	123
4.1 Protein Detection.....	123
4.2 Fluid Handling.....	128
4.2.1 Blood Filtration	128
4.2.2 Sequential Loading	130
4.3 Portable Unit	136
CHAPTER 5	
CONCLUSION.....	138
5.1 Conclusions	138
5.2 Future Work	139
5.2.1 Fluid Handling.....	139
5.2.2 Portable Unit.....	141
APPENDIXES	144
6.1 Visual Basic GUI Source Code	144
6.2 Arduino Source Code for GUI	147
6.3 Technical Drawings.....	151
6.4 Device Wiring Schematic.....	188
6.5 Device Arduino Source Code.....	190
6.6 Device MATLAB Functions	200
6.6.1 Matlab_Arduino	200
6.6.2 channelFlow	204
6.6.3 binaryMask	205
6.6.4 sumDifferences	205
6.6.5 Inc_Timer	205

6.6.6 detectionFlow	206
6.7 SOP Reagent Formulation.....	208
BIBLIOGRAPHY	212

LIST OF TABLES

Table 3-1	Valve positions for the various fluid handling states.	85
Table 3-2	Functions created in Arduino, and their purpose.	115
Table 4-1	Procedure for Round 1 of the sequentially loaded tests.	124
Table 4-2	Procedure for Round 2 of the sequentially loaded tests.	126
Table 4-4	Example of the data sheet used to collect the pumping volumes and times for each test.	131
Table 4-5	Pumping volumes set for each cycle based on the data in Figure 4-4.	133
Table 4-6	Results of the fluid position sensor tests using a constant flow rate of 200 $\mu\text{L}/\text{min}$. Fluid flow was not interrupted during the course of a test.	135
Table 4-7	Results of the autonomous sequential loading tests.	136

LIST OF FIGURES

Figure 1-1(a) 96 well microtiter plate of an enzyme-linked immunosorbent assay after colorimetric enzymatic reaction. (b) Sandwich ELISA (1) capture antibody bound (2) antigen from sample captured (3) a specific antibody binds to antigen (4) enzyme linked secondary antibody binds to specific antibody (5) substrate added to be converted by the enzyme into a color or fluorescent or electrochemical signal.	6
Figure 2-1(a) Long time exposure of 2.5 μm florescent particles not entering the side (plasma outlet) channel. Re of the inlet flowrate equals 50. (b) 20 μs exposure of whole blood being filtered from plasma at a flow rate of 10 $\mu\text{l h}^{-1}$. (c) Schematic of the blood plasma separation structures utilizing the Zweifach-Fung bifurcation law.	16
Figure 2-2 Photograph of the fabricated filtration device. The gold electrodes were vacuum deposited on glass. The channels are made of PDMS and bonded to the glass plate. Main-channel is 500 μm wide by 200 μm deep, side-channels are 5 μm wide by 2 μm deep, small enough to physically block the passage of red and white blood cells.	17
Figure 2-3(a) Orientation of the valves relative to the center of rotation and corresponding centrifugal force. (b) Geometric capillary valve, (c) hydrophobic valve, (d) centrifuge-pneumatic valve, and (e) hydrophilic siphon valve.	25
Figure 2-4 Main operations of the multilayer PDMS-based Large Scale Integration (LSI) platform using the NanoFlex™ valve. (a) Valve is open until pressure is applied to control valve. (b) Valve is closed. Using the same principle as the valve (c) , pumps (d) and mixers (e) can also be designed.	26
Figure 2-5 Instrument developed by Sauer-Budge <i>et al.</i> to be used in conjunction with their single use lab-on-chip device. Left: Schematic of instrument functionality and photo of chip/instrument interface. Right: Picture of instrument. ..	28
Figure 3-1 Process for making a PDMS microfluidic chip. (a) A master mold was made, patterned with the desired microfluidic features. (b) PDMS was poured over the mold and was cured by heating, (c) the cured PDMS was peeled up from the mold. (d) Ports were punched into the channels and (e) the chip was bonded to a blank sheet of PDMS, thus creating the microfluidic channels.	34
Figure 3-2(a) Screen shot of the CAD software SolidWorks used to design all of the molds. (b) Magnified view of a typical micro end mill used by the CNC to machine the mold. (c) Photograph of a completed Ultem mold.	37
Figure 3-3(a) Shows the maximum channel height possible, limited by the end mill's depth of cut. (b) Shows an interior radii feature and (c) shows the minimum distance between any two features, both limited by the diameter of the end mill.	38

Figure 3-4 Illustration of the reduction in machining, from (a) with large volumes to be removed, to (b) where the design rule limits the maximum wall thickness to 500 μm , resulting in large islands of material no longer removed. Screen captures of the CNC software used; SolidWorks.	39
Figure 3-5(a) Walls to contain the PDMS are created using transparent tape. (b) When using Ultem molds, aluminum is placed around the mold to better transfer heat. (c) Tape is peeled away from the mold and cured PDMS. (d) Cured PDMS is slowly peeled away from the mold. (e) The chip is placed channel side up on the cutting surface. (f) Custom biopsy punch with an integrated level is used to punch holes in the chip.	44
Figure 3-6 Schematic of an early design for an ELISA style microfluidic chip. X's mark where access ports would be punched. Detection site and fiber optic channels are not shown.	46
Figure 3-7(a) V2.0 of the ELISA microfluidic chip, (b) V2.1 brought with it some subtle design changes, the most obvious being the change to the detection site.	47
Figure 3-8(a) Zoomed in isometric view of the mold at the detection site for V2.0 and, (b) V2.1.	48
Figure 3-9 V3.0 of the sequential loading chip. All channels have a width of 500 μm and a depth of 275 μm . Air volumes separating liquid volumes are unlabeled but reside between each liquid volume.	49
Figure 3-10 V4.0 of the sequential loading chip. All channels have a width of 300 μm and a depth of 250 μm . This marks the last major redesign of the microfluidic chip.	51
Figure 3-11 Various changes to V4.0 of the microfluidic chip including, (a) the addition of a PBST wash step. (b) Isometric view of an inlet port found on V3.0 and (c) an isometric view of the Y-section and an inlet port found on V4.0.	54
Figure 3-12(a) Version 4.1 of the microfluidic chip showing the major change to the waste storage section. (b) Scale representation to the change in cross sectional area of the main channels from V3 to V4. (c) Close-up isometric view of a V4.1 port.	55
Figure 3-13 Initial tests of the separation membrane used a negative pressure to pull the plasma through the membrane and into the channel.	58
Figure 3-14 Schematic of the technique used to push plasma into the chip using a positive pressure.	59
Figure 3-15 Photograph of the mold used to make the Filter Holder piece from PDMS. The circle has a diameter of 10 mm and stands 350 μm proud.	60

Figure 3-16 Schematic of how the filter is integrated onto the microfluidic chip and how the double branch feature at the beginning of the plasma channel is utilized.....	62
Figure 3-17 (a) 3D rendering of the integrated filter on V2.0 of the microfluidic chip and (b) a photograph of an integrated filter on V3.0 of the chip containing water with food coloring representing plasma (red), wash1 (brown), detection Ab (blue), wash2 (green), and storage buffer (dark green).....	62
Figure 3-18 Photograph of a Bartels mp6 micropump.....	64
Figure 3-19 Schematic of the two reservoir method, allowing the mp6 to pump water. Here the micropump is pumping water from reservoir 2 to 1 creating a positive pressure in the air pocket of reservoir 1.	65
Figure 3-20 Model for the condition in a microfluidic chip containing many fluid volumes separated by volumes of air with one driving force.	66
Figure 3-21 Model for the condition in a microfluidic chip containing many fluid volumes separated by volumes of air with two driving forces.....	67
Figure 3-22 CAD rendering of the dual chamber fixed volume syringe. The aluminum body has been made transparent for the purpose of showing the inner construction of the device.	68
Figure 3-23 Version 2 of the dual chamber fixed volume syringe. Inset picture shows the seal between the rubber V-washer and the plunger rod.	71
Figure 3-24(a) Photograph of the specified linear actuator with a parting line depicting approximately where it was cut. (b) Linear drive section installed into the housing. (c) View of the back where the stepper motor is connected and (d) a view from the front of the connected pieces.	72
Figure 3-25 Photograph of the drive linkage installed on the stepper motor.....	73
Figure 3-26 CAD rendering of the DCFVS cradle connected to the linear actuator, its housing, the stepper motor, and the DCFVS.....	74
Figure 3-27(a) Photograph of the back side of the stepper and heat sink. (b) Photograph of the assembled pump to be installed into the portable unit.	75
Figure 3-28 Schematic of the functionalization used in the PDMS channels to build a biofunctional layer, the top of which (protein A) has a high affinity for IgG antibodies.	77
Figure 3-29 Functionalization scheme for anti-HSA HAPD channel.....	78
Figure 3-30 V3.0 of the microfluidic chip loaded with colored water representing the various reagents, storage buffers, and washing steps. From top to bottom: wash (blue), PBST wash (yellow), detection antibody (green), wash (blue),	

sample (red), storage buffer for anti-IgG HAPD channel (blue), storage buffer for anti-HSA HAPD channel (brown). 79

Figure 3-31 Schematic of the sandwich assay built on the walls of the detection site channel. 80

Figure 3-32 Image taken by a fluorescent microscope of a detection site after assay. Sample was 10 µg/mL CRP in PBS buffer. 80

Figure 3-33 Sample of the intensity readings taken by the spectrometer during a typical test. **Black** line is the background noise (dark reading). **Red** line is the baseline reading taken with the excitation light on, before the detection antibody is introduced. **Green** line is taken when the detection antibody is in the detection site. **Blue** line is the reading line taken once the assay is complete. The difference between the red and blue lines is the signal strength. 82

Figure 3-34 Schematic of the 6 valves (V1-6) connected to the chip, pump, and each other. Each valve is a 3-way solenoid valve, so the center port is always either connected to the top or the bottom port. 84

Figure 3-35(a) Photograph of the bench-top setup used to test the functionality of the fluid handling using 6 valves. **(b)** Photograph of the same setup from a different angle showing the PEEK tubing installed through the platform to connect into the chip. 87

Figure 3-36 Screen capture of the GUI created in Visual Basic to control the valves for the bench-top experimental setup. Notice the valves can be controlled individually or as States. 88

Figure 3-37 (Left) Drawing of V4.1 of the chip in the orientation that the CCD camera sees it in; dotted box indicates how much of the chip the camera captures. **(Right)** Image taken by the CCD camera. Red boxes mark where the trigger channels are located; from top to bottom they are, detection site 1, detection site 2, HAPD site 2, HAPD site 1, sample (plasma) channel. Washed-out circle centered on the detection site is from the optics underneath the chip used to direct the excitation light. 89

Figure 3-38 (Left) CCD image of microchannels from V4.0 of the chip completely filled with air. **(Right)** Image of the same channels with the bottom 10 channels containing water. 90

Figure 3-39 Images generated by the fluid detection program during a pumping cycle taken in chronological order. **(a)** ROI can be seen as a pixelated rectangle, this is where the program will look for a change in the percentage of white to black pixels indicating a change from air to liquid. **(b)** Liquid has started to fill the ROI channel changing it from black to white until eventually **(c)** the ROI has changed enough to trigger an action, in this case, to look at the **(d)** next ROI for a change. **(e)** The ROI starts to fill with liquid until **(f)** it has filled enough to trigger an action. 91

Figure 3-40 Experimental setup used to test the fluid position sensor and fluid handling system as a whole.....	92
Figure 3-41(a) Position of the light source LED and lens relative to the microfluidic chip. The detection site, center of the lens, and the center of the LED all lie on the same plane. (b) Image captured by the CCD in a dark room apart from the LED.	94
Figure 3-42 Isometric view of the placement of the larger components held within the unit. This rough placement helped determine the required size of the unit.	95
Figure 3-43 Screen capture of the CAD model used to determine the position and angle of the mirrors.	96
Figure 3-44(a) Mock-up CAD drawing with mirror 1, mirror 2, the CCD and lens, and chip platform positioned exactly the same as in Figure 3-42. (b) Holder for mirror 1 was designed around the position of the mirror. (c) Similarly, after the lid was designed, a holder for mirror 2 was added to position it correctly.	97
Figure 3-45 Isometric view of the device body in a transparent state, also showing the placement of three of the valves and the two breadboard holders.....	98
Figure 3-46 Pump mounting bracket viewed through the side of the device shell. ...	99
Figure 3-47(a) Placement of the stepper driver shield mounting bracket on the shell of the device. Larger holes in the center of the bracket accept socket head bolts, fastening it to the shell, smaller holes indicate shield mounting holes. (b) Driver shield installed onto the bracket.	99
Figure 3-48 The Chip platform, three valves, and the CCD/lens viewed from underneath.	101
Figure 3-49 (a) The three pieces that make up the top portion of the device viewed from above. (b) The same three pieces viewed from underneath.	102
Figure 3-50 CAD rendering of the filter interface lid attachment. (a) The assembly with the interface base opaque. (b) The assembly with the interface base made transparent.	103
Figure 3-51 Anchor suspension piece in use, the PDMS anchors are held parallel to, and 1.50 mm above the bottom of the petri dish. PDMS was poured into the dish to an appropriate level and placed on a hot plate to cure.	103
Figure 3-52 CAD rendering of the filter interface lid attachment installed into the device. The lid is set to a transparent state. Also shown are the fiber optic cable guides installed at each opening of the lid.	104
Figure 3-53 CAD rendering of the illumination LED and the holder used to position it at the optimal spot for imaging.	105

Figure 3-54 CAD rendering of the two breadboards mounted on their respective holders in the device.....	105
Figure 3-55 All images are CAD renderings. (a) Shows a square hole that allows a FireWire (IEEE 1394) cable to be plugged into the CCD control board, (b) has the shell set to transparent. (c) Feature that matches the shape of the DC jack shown in (d) installed. (e) Feature that accepts the snap-fit rocker switch used to control the input DC power, (f) shows the switch installed.	106
Figure 3-56(a) CAD rendering of the device and its internal components. (b) Photograph of the assembled and painted device with its lid open.....	108
Figure 3-57 Arduino analog-in readings as a function of the position of the plunger in the pump. The position of the plunger is tied to the position of the linear actuator, at Position=0 the linear actuator is fully in, and at Position=129 the linear actuator is fully out.	111
Figure 3-58 Wiring diagram of the components used in the portable device.	112
Figure 3-59 Photograph of the portable device completely wired and plumbed. The unit is being viewed top down.	113
Figure 3-60 Flowchart showing the broad operation and logic of the device.....	122
Figure 4-1 Data with a dose-response curve fit for CRP in PBS buffer	125
Figure 4-2 Data with a dose-response curve fit for CRP in PBS buffer after various improvements to the immunoassay procedure	127
Figure 4-3 Results of the plasma filtration tests, each point represents 10 tests, and the error bars shown represent the standard deviation. The dotted line marks the ideal amount of filtrate to be inserted into the sample channel.....	129
Figure 4-4 Average volume pumped during each cycle over 15 tests. Table 4-4 gives a description of each cycle.....	133

CHAPTER 1

INTRODUCTION

Point-of-Care (POC) diagnostics is a rapidly growing field of interest not only because of the promise to provide easy to use, disposable, fast, accurate, and reliable readings but also its ability to tap into a massive market. According to the World Health Organization (WHO), the United States spent more on health care per capita (\$8,600), and more as a percentage of gross domestic product (17%), than any other nation in 2011 [1]. In resource limited settings, a POC device could have a dramatic impact on infectious diseases on a global level. Microfluidic diagnostic devices in particular are an attractive concept due to their low cost, small sample consumption, and fast results. Fast and accurate diagnostics are key to quick, less invasive treatments and an overall better chance of survival [2].

1.1 Medical Diagnostics and Point-of-Care

A medical diagnosis is the process of trying to identify a disease or disorder through one or several techniques aimed at reaching a medical diagnostic opinion [3]. There are numerous methods for diagnosing a disease, including clinical diagnosis, radiology diagnosis, and laboratory diagnosis, often used in concert, with each having the potential to be an effective tool in reaching a definitive diagnoses. For example, a doctor may have an idea of what his/her patient's ailment may be on the basis of observable medical signs and patient reported symptoms, and confirm the initial

diagnostic opinion with a laboratory test before proceeding with treatment. Laboratory diagnostics are invaluable in the medical field now more than ever, thanks to the proliferation of biomarker identification and genetic testing, which promise to identify disease before it could otherwise be detected using classical forms of diagnostics [4]. Diseased but asymptomatic patients are unlikely to be diagnosed unless they receive regular blood workups testing for biomarkers linked to that particular disease. Testing of this type is important because early diagnosis often significantly increases a patient's survival rate and decreases the overall cost of the treatment received.

The WHO defines a biomarker as “almost any measurement reflecting an interaction between a biological system and a potential hazard, which may be chemical, physical, or biological. The measured response may be functional and physiological, biochemical at the cellular level, or a molecular interaction.” [5]. The purpose of this investigation is to use proteins present in human blood as the laboratory-measured biomarker for various diseases, biomarkers that have a relationship to relevant clinical endpoints [6]. One such example of using a biomarker as a quantifiable characteristic of a biological process is the diagnosis of Alzheimer's disease by measuring the level of mutant amyloid proteins called prions [7], or possibly a close relative of the beta-amyloid protein itself [8]. An example of a more clinically recognized protein biomarker is human C-Reactive Protein (CRP), produced as part of the nonspecific acute-phase response to most forms of inflammation, infection, and tissue damage [9]. Not only are CRP levels elevated after a cardiac event, but studies show that there is a predictive aspect to elevated CRP levels *before* an atherothrombotic event, including coronary events, stroke, and progression of

peripheral arterial disease [10]. CRP has also proven to be one of the most sensitive and responsive biomarkers present in human blood, thus necessitating a rapid test allowing medical personal to take multiple CRP concentration measurements in the time that would have previously only allowed one measurement, thus allowing a faster response time for treatment. This is one area that a microfluidic POC device could thrive.

Current protein biomarker detection techniques for diagnostic purposes can be divided into two main categories: well established laboratory instruments and procedures, and a fledgling group of POC detectors. The most prevalent of these laboratory techniques include High Performance Liquid Chromatography (HPLC) [11][12], mass spectroscopy [11], western blot, also known as protein immunoblot [13], and Enzyme Linked Immunosorbent Assay (ELISA) [14][15][16]. HPLC can give highly accurate protein concentrations in a biological sample. It works by passing the sample through a column filled with a sorbent (also called the stationary phase), where the different components of the sample will move through the column at different velocities as a function of specific physical interactions with the sorbent, thereby separating the various components of the sample. This separation allows for each analyte (molecular component of the sample to be measured) to emerge (elute) from the column at a specific time, known as the retention time. This elution process is facilitated by a mobile phase consisting of a mixture of water and an organic solvent passing through the column under high pressure (5,000-35,000 KPa). Commonly the ratio of water to solvent during the elution process is decreased, known as gradient elution [17]. The drawbacks of this technique include sample preparation and volume

needed (~mL), analysis time (~hour) and the cost and size of the instrumentation. Additionally, someone who is unfamiliar or untrained using an HPLC must rely on specialized personal to conduct the test.

Mass spectroscopy is a technique in which proteins in a sample are ionized and then introduced into a mass analyzer. This technique has the advantage of being relatively quick, yielding accurate results, and not requiring a large sample size, but it also presents several significant problems when testing biological fluids. Ionization techniques only work well when the mixture contains roughly equal amounts of constituents. This stands in contrast to biological samples in which different proteins tend to be present in widely differing amounts [18]. Also, the mass spectrum from a complex mixture is very difficult to interpret because of the overwhelming number of mixture components and the size and complexity of the molecules present. Often, to contend with the shortcomings associated with mass spectroscopy, HPLC is used to fractionate the sample first. Assuming the analyte protein could then be measured, getting a quantitative amount of the analyte is only possible using a technique called quantitative proteomics like stable isotope labeling [19]. As with HPLC, the instrumentation for conducting mass spectroscopy is large and expensive.

Western blot uses gel electrophoresis to separate native proteins by their structure or denatured proteins by the length of the peptide [20]. This technique has several disadvantages including time consumed, complicated sample preparation, and an additional detection step needed to quantify the protein of interest (*e.g.* mass spectroscopy or HPLC).

The last category of conventional protein detection used in diagnostics is the Enzyme Linked Immunosorbent Assay (ELISA). Conducted in sophisticated clinical or reference laboratories by highly trained personnel, ELISA provides specific, sensitive, and quantitative detection of biological agents [14][15][16]. Like other detection techniques, ELISA is not without its drawbacks. Conducting an ELISA test requires milliliters of sample, expensive and stationary equipment using microtitre plates, highly trained personnel, and at least two hours between sample acquisition and qualitative results. In this technique the capture antibody (which has an affinity for the target protein, also known as the antigen) is immobilized onto the surface of the wells of a microtitre plate (see **Figure 1-1a**). The surface is then blocked, which prevents the non-specific binding of proteins to any area other than the capture antibody. The sufficiently diluted sample is then added to the reservoirs, allowing the antigen in the sample to bind to the capture antibody. Once the appropriate incubation time has passed, the sample is removed, the wells are washed to remove unbound antigen, and a different antibody, called the detection antibody, is added to the reservoirs. This detection antibody, which has an affinity for the antigen, is linked with a certain enzyme. The detection antibody/enzyme complex will bind to the antigen and be immobilized on the surface of the reservoir. Finally, after several washing steps to remove all unbound detection antibodies, a substrate for the enzyme linked to the detection antibody is added to the reservoirs initiating an enzymatic reaction (see **Figure 1-1b**). Depending on the type of detection antibody used, the reaction will result in a particular type of signal, including colorimetric (seen in **Figure 1-1a**), fluorescent, or electrochemical. This signal strength will be proportional to the amount

of antigen bound to the surface, which is proportional to the amount of antigen in the sample. This signal is quantified using a plate reader to determine the presence and quantity of the antigen [21].

Figure 1-1(a) 96 well microtiter plate of an enzyme-linked immunosorbent assay after colorimetric enzymatic reaction. **(b)** Sandwich ELISA (1) capture antibody bound (2) antigen from sample captured (3) a specific antibody binds to antigen (4) enzyme linked secondary antibody binds to specific antibody (5) substrate added to be converted by the enzyme into a color or fluorescent or electrochemical signal.

Unfortunately, a test platform such as this is incompatible with the concept of a POC device, as it requires considerable sample preparation, adequate training, and relatively large and expensive pieces of equipment. The ideal POC device would be a portable and inexpensive device, capable of being used by the layperson and requiring only a small sample size (microliter to nanoliter). One technique for achieving this is to take the science and techniques used in ELISA and adapting them for use in a microfluidic device.

Microfluidics is the science of manipulating small volumes (mL, nL) of fluids typically in sub-millimeter channels called microchannels. Microchannels offer some key advantages over a microtiter plate with respect to a POC device. Because of the small working volumes of microchannels, a smaller sample size is required to conduct

a test. Additionally, the small dimensions of a microchannel means shorter diffusion lengths for the proteins in solution, giving shorter incubation times (minutes) when compared to microtiter plate incubation times. The current commercially available POC market is dominated by products based on Lateral-Flow Assays (LFA) which are relatively fast (<1 hour) but usually only provide a qualitative yes/no diagnosis, giving no insight as to the level of disease, something that can be critical when trying to formulate an appropriate treatment [22].

1.2 Plasma Extraction from Whole Blood

In a traditional ELISA test conducted at a laboratory, the first step in sample preparation, when that sample is whole blood, is to extract all cellular components leaving only blood plasma. In a laboratory this is done on a purpose-built machine called a centrifuge, which exploits the differences in densities of the various components in blood. The blood is spun at a high speed for approximately 15 minutes, separating it into its component parts. This process, called blood fractionation, results in a layered effect where plasma is the top layer; at this point it can easily be removed, diluted, and used in an ELISA test. Clearly this additional external step is not acceptable when a one-step rapid assay conducted wholly on a microfluidic chip is the objective. There have been numerous attempts to find a suitable blood separation technique that could easily be incorporated into a Lab-on-Chip (LOC) microfluidic device.

Two popular techniques for separating the cellular component of blood from plasma on a LOC device are 1.) a separation membrane, and 2.) specialized shape

effects in the microfluidic channels themselves. A separation membrane takes advantage of the size difference between the cellular components of blood and plasma. It works in much the same way an air filter works, by capturing large particulates (red and white blood cells) in a media and allowing air (plasma) to pass through. The plasma separation membrane material looks and feels similar to paper. Specially designed microfluidic channels designed to extract plasma from whole blood generally utilize the Zweifach-Fung (bifurcation) law to separate the relatively large mass of blood cells from plasma.

1.3 High Abundance Protein Depletion

Once the appropriate volume of whole blood has passed through the plasma separation membrane removing its cellular components, further purification of the plasma is necessary to facilitate detection of the target antigen at clinically relevant levels. Human blood contains several proteins at very high concentrations relative to the other proteins. These so called high abundant proteins (HAP) can interfere with the immobilization of the target antigen at the surface of the detection site and adversely affect the limit of detection. An effort must be made to remove as much HAP from the sample plasma as possible before it is used to detect a target antigen. Albumin (~60%) and Immunoglobulin G (~18%) comprise approximately 78% by mass of proteins present in blood plasma. Clearly a reduction in the concentration of these proteins would be beneficial to the detection of other lower concentration proteins like C-reactive protein (0.003% by mass).

1.4 Fluid Control in Microchannels

Fluid control is paramount to ensure the assay is completed with the correct timing and procedure. If two separate assay steps, consisting of two separated fluid volumes, were to mix before reaching the detection site, the entire test would have no diagnostic value. Additionally, during the course of an ELISA style test, reagents are required to be in a particular spot for a predetermined amount of time (incubation time), multiple times in the course of one test.

Fluid control in microfluidics can be separated into two main categories; active or passive control schemes. Active control commonly indicates the use of a pump(s), valve(s) or both to control the movement of fluid volumes. Passive control uses no external force to move the working fluid, and generally indicates the use of capillary forces present in the medium to drive the fluid.

Passive control has the benefit of being inherently simple, inexpensive, and requiring little or no external power to facilitate fluid movement. The ubiquitous home pregnancy test, for example, is a lateral flow assay test which moves the sample fluid from a wetted section of media (often porous paper) to a dry section of media using only capillary force. Despite the obvious benefits of designing an assay around such a simple test platform, there are numerous drawbacks, such as the inability to add multiple washing steps to increase the sensitivity of a test. This shortcoming becomes more important when testing for a protein that is present only at relatively low concentrations.

Active control, in general, is a more complicated system, often involving valves, pumps, and triggers in order to move fluid volumes to distinct locations at

distinct times. Active control allows for stop-flow fluid location, more closely mimicking the procedures of a standard bench-top ELISA test wherein a sample is allowed to incubate for a set amount of time before being removed. With appropriate microfluidic chip design, active control allows for an almost limitless supply of washing steps and reagents, allowing for maximum flexibility in maximizing the performance of an assay. This high level of control comes at the cost of simplicity. A microfluidic system containing valves and pumps must have a power source and a secondary control system to control the valves and pumps.

The valves and pumps used in microfluidic systems can be separated into two main categories: on-chip or off-chip. The name ‘off-chip’ refers to the location of the valve or pump, indicating that it is separate and distinct from the microfluidic device. Such valves and pumps must be connected to the microfluidic chip, often using tubing. An off-chip pump creates a pressure differential outside of the chip, with a syringe for example, which is then ported to the chip through tubing.

Alternatively, on-chip mechanisms can generate a pressure differential with a pump and switch between or block off microfluidic channels inside the chip itself. Benefits to this method include a near-zero ‘dead’ volume associated with tubing and connections present in off-chip valves and pumps, and the flexibility to place a valve or pump almost anywhere on the chip.

1.5 Objective

This investigation seeks to develop a portable, microfluidic fluid handling system designed to rapidly detect protein biomarkers in whole human blood. The

detection system is based on a fluorescent heterogeneous assay developed and benchmarked using C-Reactive Protein in buffer. The problem entails integrating and further developing individual methods and components previously tested [23][24][25][26], as well as novel methods and components, in order to yield a functioning Point-of-Care (POC) device.

The first aspect of the problem is to develop an on-chip method for removing the cellular components of whole human blood before analysis. Presently, a suitable filter material has been benchmarked and proven to adequately remove micrometer scale components of human blood, yielding blood plasma, a process that is necessary when detecting proteins that are on the nanometer scale. It is proposed to develop a method to integrate the filter media into the microfluidic chip to produce a precise and accurate volume of blood plasma, to be used as the sample fluid to be analyzed. Additionally, this proposed method must be executed with no human intervention once the appropriate volume (one or two drops) of whole blood is introduced to the microfluidic chip by the user.

The next aspect of the problem involves the integration of a previously established method of blood plasma filtration that consists of removing a large portion of the high abundance proteins (HAP) present in plasma. Removing these high abundance proteins is vital because they interfere with the capture of the target biomarker during the fluorescence sandwich immunoassay conducted in the microfluidic chip. It is proposed that the plasma from the plasma microchannel be sequentially moved to and incubated at two HAP removal microchannels before

entering the detection site. This filtering process must be conducted on-chip and be completely automated.

The next part of the problem entails controlling all fluids within the microfluidic chip; specifically, the need to sequentially deliver multiple biological reagents in solution to the detection site microchannel for the purpose of forming a sandwich type heterogeneous fluorescence immunoassay at that location. The proposed fluid handling system includes the microfluidic chip design itself, as well as off-chip valving and a pump. This diagnostic microfluidic chip must perform two major functions with respect to fluid handling: it must prevent numerous volumes of reagents, separated by a volume of air, to come in contact with each other at any point during the test, and it must allow for a pressure differential to be applied at different points and at different times in the chip. The system must be able to move all individual biological reagents through the microfluidic chip simultaneously. Due to the high pressure drop required to move separate volumes of fluid, a more powerful pump specific to this application will need to be developed. Control systems for the described valves and pump must also be developed through the use of a microcontroller and associated programming, as well as a user interface facilitating the use of the proposed POC device.

The final component of the investigation is the incorporation of a previously benchmarked charged coupled device (CCD) camera to be used for determining the position of fluid volumes [24]. The system, consisting of a CCD camera and a series of MATLAB functions, should provide accurate control of the sequentially loaded biological reagents as they move throughout the chip. All of these listed components

must be integrated into a form factor of reasonable size as to still be considered “portable” and must function in a way that a layperson could use with relative ease.

Over the past twenty years there has been much effort to deliver on the promise of a microfluidic POC diagnostic device. The value of microfluidic systems has been shown rather decisively over that time, demonstrating many attractive characteristics such as fast turnaround time, low consumption of reagents, complex fluid handling, increased automation, capability for multiplexed tests, and miniaturization. This investigation aims to contribute to the pursuit of a fast, accurate, reliable, and user friendly Point-of-Care device.

CHAPTER 2

LITERATURE REVIEW

2.1 Plasma Separation from Whole Blood

2.1.1 Centrifugal Techniques

The first step in many clinical diagnostic analyses is the separation of red blood cells (RBCs) from blood plasma, as certain cellular components of blood can inhibit biological detection [27]. In a laboratory setting, the cellular components of whole blood are separated from plasma using centrifugal force; this principle has been adapted for use in microfluidic devices. Lee *et al.* [28] developed a system of blood plasma separation using a rotating disc shaped microfluidic chip. The chip (made of a hard PMMA plastic) contains microchannels and reservoirs developed to separate cellular components allowing only plasma to move on to the rest of the chip where it is analyzed using a spectrophotometer. Amasia *et al.* [29] show the ability to filter larger volumes of whole blood (2mL) than is generally required by a microfluidic system, with a similar compact disc-based device. The technique proved to be more than twice as fast as standard plasma preparation procedures, while still yielding high purity plasma. Disc/centrifugal microfluidic platforms have an inherent advantage in plasma separation, as it is a platform based on a system that can most closely mimic standard blood separation techniques, exploiting the density differences between the RBCs and plasma. Systems based on a quickly rotating (~ 3500 RPM) disc such as this, unfortunately, immediately increase the power requirements and complexity of

the overall system. Valving, for instance, is usually limited to passive styles utilizing capillary and hydrophobic forces, though Park *et al.* [30] reported an active sacrificial valve controlled by laser irradiation and microbead impregnated wax.

2.1.2 Bifurcation Separation

More relevant to a Point-of-Care LOC device are attempts at using particular geometries in a chip to filter or separate cellular components from plasma. Several studies utilized the Zweifach-Fung effect, or bifurcation law to separate the relatively large mass of blood cells from plasma. At higher Reynold's numbers (~ 80) all cellular components can be removed and a portion of pure plasma can be captured, a technique termed hydrodynamic filtration. Huang *et al.* [31] designed backward facing step geometries (see **Figure 2-1a**) to extract cell-free plasma from whole blood but only at a rate of 8.4% of the total volume passed. Yang *et al.* [32] also conducted experiments (see **Figure 2-1b**) based on the Zweifach-Fung effect, and found plasma separation volume percent between 15% and 25%, and a plasma purity of 100%, rivaling even common centrifugation (99%). Kersaudy-Kerhoas *et al.* [33] conducted studies with a series of bifurcations in microchannels similar to the structures used by Yang (see **Figure 2-1c**), and achieved an improved plasma separation volume percent of 40%, but a significant decrease in plasma purity at only 53%. The most attractive aspects to hydrodynamic filtration are the purity of the extracted plasma and the low risk of cell hemolysis. If red blood cells are exposed to a high of a shear stress, they can rupture releasing hemoglobin proteins that can interfere with further analysis of the blood. There is little to no risk of this happening using hydrodynamic filtration. Despite this,

there are several factors that make this method not ideal for a LOC application. A generally low plasma yield (2.5-40%) dictates using larger initial volumes of whole blood than is ideal for a POC device, low purity plasma should also be a concern due to the possibility of RBCs interfering with downstream tests. Additionally, if hydrodynamic filtration were to be incorporated into a microfluidic chip, further valves and pumps specific to the filtration process would need to be added, significantly increasing the complexity and cost to the system.

Figure 2-1(a) Long time exposure of 2.5 μm florescent particles not entering the side (plasma outlet) channel. Re of the inlet flowrate equals 50. **(b)** 20 μs exposure of whole blood being filtered from plasma at a flow rate of $10 \mu\text{l h}^{-1}$. **(c)** Schematic of the blood plasma separation structures utilizing the Zweifach-Fung bifurcation law.

2.1.3 Dielectrophoresis Separation

Another method for separating cellular components from plasma utilizes dielectrophoresis, a phenomenon in which a force is exerted on a dielectric particle when it is subjected to a non-uniform electric field [34]. Nakashima *et al.* [35] created a small filter that transported 97% pure plasma to a holding chamber from a reservoir of whole blood. The structure consisted of a larger main-channel connected to many smaller side-channels along the wall of the main-channel (see **Figure 2-2**). The side-channels each have a small enough entrance as to not allow red or white blood cells to enter (RBCs and WBCs measure 6-8 μm , and 7-15 μm in diameter, respectively) [36][37]. A voltage applied across electrodes below the whole blood containing main-

channel forced the red blood cells to the left side of the main-channel, allowing plasma to flow through the previously blocked side-channels and into the plasma reservoir. Although the extracted plasma was quite pure, only 300 nL was gathered from 5 μ L of whole blood ($\sim$ 11% of total plasma). This method also requires an AC voltage (10 V) driven at a high frequency (1 MHz), using a function generator. For these reasons, this filtering method would prove difficult to incorporate into a portable POC system.

Figure 2-2 Photograph of the fabricated filtration device. The gold electrodes were vacuum deposited on glass. The channels are made of PDMS and bonded to the glass plate. Main-channel is 500 μ m wide by 200 μ m deep, side-channels are 5 μ m wide by 2 μ m deep, small enough to physically block the passage of red and white blood cells.

2.1.3 Membrane Separation

Plasma can also be separated from cellular components using a physical barrier known as a separation membrane; such membranes promise one step separation without the additional complexity found using previously discussed methods. A separation membrane works by physically blocking and trapping RBCs and WBCs in its appropriately sized pores, while allowing plasma to flow through. The principle of

the membrane is the same as an air filter, allowing air molecules to flow through while trapping larger particulates in the filter media. A problem that plagued early forms of the separation membrane was their tendency to cause hemolysis (the rupturing of red blood cells), thus releasing large amounts of free hemoglobin proteins into the plasma, negatively affecting the ability to detect the target protein [38][39]. Among others, Pall Corporation (Ann Arbor, MI) offers a plasma separation membrane with proprietary chemical post treatments that nearly eradicate hemolysis. The ‘Vivid Plasma Separation’ membrane is specifically engineered and optimized for the generation of high quality plasma for use in downstream diagnostic assays [40]. This membrane has been benchmarked in previous studies [23] and proven to effectively (>99%) remove the cellular components of whole blood quickly (<3 min) with little to no hemolysis while still yielding $\geq 60\%$ of the theoretical plasma available. Additionally, the Vivid brand membrane gives a greater plasma yield than similar plasma separation membranes made of glass fiber media while still providing low analyte binding [41]. A method to integrate this separation membrane onto the POC device must be found that is both effective and user friendly.

2.2 High Abundance Protein Depletion

Once a sample of whole blood has been fractionated, and an adequate volume of plasma is passed downstream, further processing of the plasma is necessary to facilitate detection of the target antigen at clinically relevant levels. Human blood contains several proteins at very high concentrations (>mg/mL in serum) relative to the other proteins present. These high abundant proteins (HAP) can interfere with the

immobilization of the target antigen during the assay, adversely affecting the limit of detection of lower abundance proteins (<ng/mL in serum) [42]. An effort must be made to remove as much HAP from the sample plasma as possible before it is used to detect a target antigen. Albumin (~60%) and Immunoglobulin G (~18%) comprise approximately 78% by mass of proteins present in blood plasma; clearly a reduction in the concentration of these proteins would be beneficial to the detection of other lower concentration proteins like C-reactive protein whose concentration is 7-8 orders of magnitude lower (0.003% by mass) [43].

Protein extraction is typically implemented using chromatography columns. High Performance Liquid Chromatography (HPLC) is a common form of this technology where binding occurs on the surface of beads embedded in the column. A modern self-contained HPLC is not only cost prohibitive for the desired device but it also requires 110 VAC, a computer for analysis, and a relatively large amount of space, making it unsuitable for a POC device. A feasible alternative method for protein removal must therefore be integrated into a microfluidic chip in a section after the whole blood filtration membrane and before the target analyte detection site.

A DNA-based assay is one method growing in popularity due to the high specificity and binding affinity of complimentary DNA strands. The process works by attaching a DNA strand to a substrate, in this case a microchannel, and linking a capture antibody conjugated with the complimentary DNA strand to it. This method allows multiple antibodies to be immobilized on the surface each with its own target antigen, thus making the detection of multiple proteins possible. More important to this application is the strong affinity complimentary pairs of DNA have for each other.

Such an affinity could lead to a greater percentage of the target HAP being removed from the plasma sample. Fan *et al.* [44] developed a DNA-based barcode assay technique in which DNA-linked antibody libraries were used for sensitive detection of up to ten proteins in human blood. Using human plasma as the detection medium, the group was able to achieve a LOD in the low ng/mL range for ten proteins, a testament to the excellent specificity of a DNA based assay. In much the same fashion, Qin *et al.* [45] developed a chemical reaction powered DNA-based system for detecting twelve protein biomarkers in human serum with detection ranges in the low ng/mL range. Stern *et al.* [46] used DNA linked detection antibodies, biotinylated complimentary DNA strands and a nanoribbon detection method for the quantification of two cancer biomarkers in whole human blood. Within twenty minutes and with a 10 μ L whole blood sample, they were able to detect PSA, a prostate cancer biomarker, at a concentration as low as 2 ng/mL.

Despite these significant accomplishments, there are some drawbacks to using a DNA-based barcode system. The level of complexity rises significantly when one is required to develop an array of distinct DNA polymers as well as their compliments and immobilize them on a substrate as well as conjugate them onto proteins. This method of protein separation is well suited for detection, but for the purposes of removing HAP from the sample, a simpler and well-studied binding method, such as the streptavidin/biotin affinity, offers a far less complex system while still offering a binding affinity close to that of complimentary DNA strands.

Another method for protein depletion in sample plasma is with the use of high surface area to volume ratio microbeads functionalized with an antibody having a high

affinity to the target protein. When packed into a microfluidic channel, microbeads can significantly increase the surface area of the channel, thereby increasing the binding rate in the following ways. First, the time it takes for a target protein to find a surface, through diffusion, is much lower due to the fact that it will only have to diffuse a short distance; and second, there is more surface area for target protein attachment. McKenzie *et al.* [47] developed a device for the depletion of IgG antibodies in human serum using Protein G functionalized microbeads and a mixing technique. Using 1:100 diluted human serum as the detection medium, the microbeads removed $70 \pm 19\%$ of IgG from the sample, and on a bench-top unit, utilizing a vortex mixer, the resulting IgG mass depletion reached 99%. Ryu *et al.* [48] developed a system in which sample flow was driven by the capillary force. Cardiac markers in buffer picked up fluorescently labeled antibodies en route to a detection area, in which microbeads with capture antibodies were deposited. Using a CCD, fiber optics and optimally chosen polarized lenses, a detection limit of 1.5 ng/mL in buffer using fluorescent detection was accomplished.

Herrmann *et al.* [49] conducted experiments with 1 μm diameter magnetic beads for the detection of anti-streptavidin in buffer. With an anti-streptavidin sandwich assay built on the surface of the magnetic beads, they were able to induce mixing by introducing an alternating magnetic field, moving the beads back and forth through a buffer containing anti-streptavidin. The study resulted in a LOD of 0.1 pM using a fluorescent microscope.

These studies feature promising results, but most lack an ability to be directly integrated into a POC device where the sole interaction between the device and the

user is an introduction of a drop(s) of blood. Additionally, despite their attributes, using microbeads in a one-step microfluidic detection chip adds a significant level of complexity to the system, requiring a level of containment of the microbeads to one section of the microfluidic chip. In the case of non-magnetic microbeads this is usually accomplished through size discrimination, wherein a mesh structure (for example) holds the beads in position while allowing fluid to pass. Magnetic microbeads have the advantage of being able to be manipulated with a magnetic field, but that control comes at the cost of having to have a way to produce a magnetic field in the system further increasing cost, complexity, and power requirements.

2.3 Fluid Control in Microchannels

A cornerstone in performing biological assays in microfluidics is the ability to control and transport fluid within microchannels. The two modes of fluid control can be split into two groups; fluid transport, consisting of a pressure or force that makes the fluid move in one direction or another, and fluid isolation, where a microchannel is cut off from another, allowing, among other things, each channel to feel a different driving force. Both are important tools used in LOC devices, a domain where reaction times are tightly controlled to achieve optimal detection. In the case of an ELISA style assay, reagents must be allowed to incubate for the correct amount of time, but not so long as to disrupt the binding kinetics at work at the molecular level.

2.3.1 Passive Fluid Control

Despite the great number of valving techniques applicable to microfluidic systems, no standards have emerged so far, so the choice and implementation of on-

chip valves remains difficult [50]. The most common form of the passive fluid transportation is the lateral flow test, also known as the strip test. In these tests (*e.g.* pregnancy test strip), fluid movement is driven by capillary forces, and the rate at which they are driven is a function of the wettability and pore size of the substrate. Fluid will continue to wick through the substrate as long as there is sufficient unwetted substrate and liquid to wet it. In passive liquid transport, incubation times can be defined by adjusting the substrates length and pore size [50]. The passive fluid isolation technique is the check valve, which only opens to forward pressure, similar to the electrical diode. Yang *et al.* [51] developed a pair of one-way passive microvalves made from p+ silicon diaphragms that opened or closed depending on the pressure difference. Experimental results showed a flow rate of 1.6 ml min^{-1} forward pressure and 0.05 ml min^{-1} backward pressure each at 4 kPa. Li *et al.*[52] created a similar check flap made of a thin layer of nickel able to withstand pressures up to 10 MPa. Check valves made of polycrystalline silicon [53], silicon-on-insulator wafer [54], and polymers such as Parylene [55], Kapton [56], Mylar [57], and PDMS [58] have also been produced. Included in this class of mechanical check valves is the check valve that utilizes a spherical ball. The ball is either held in place with gravity [59] or is held in place with an external magnetic field [60].

Non-mechanical passive microvalves also exist, Ahn *et al.*[61] used abrupt passive microvalves to control flow on their LOC device. These valves worked by exploiting abrupt cross-sectional area changes in hydrophobic microchannels, generating pressure drops, resulting in fluidic flow valving. Melin *et al.*[62] showed how a passive valve could be triggered, or opened, by the passing of a second fluid

passing the junction of the stopped fluid by the same abrupt cross-sectional area changes.

2.3.2 Active Fluid Control

Offering the greatest level of control, schemes using active pumps and valves specifically designed for microfluidics are abundant. Starting with active fluid transportation techniques, the following is a brief overview of the main categories.

Centrifugal microfluidics is a class of microfluidic devices where all processes are controlled by the frequency at which the device is rotated, and all forces for liquid transport are centrifugal, Coriolis, and capillary [50]. This limits all fluid movement to move only in one direction; from radially inward positions to radially outward positions. The actual microfluidic devices, or chips on which the assays are run, are disc shaped and made from a rigid transparent material like PMMA plastic or glass. These centrifugal systems have employed many creative valving techniques including using narrowed channel sections, or hydrophobic sections, which only allow fluid to pass when a sufficient back pressure has built up (see **Figure 2-3a**). The geometric capillary valve is adapted for use in a centrifugal device. It consists of a channel that abruptly widens to a larger channel section, and in the context of a centrifugal device, this valve, typically thought of as a passive valve, can be actively ‘switched’ open by simply increasing the rotational velocity of the disc [63]. Other forms of the passive centrifugal microvalve include a hydrophobic valve [64], centrifuge-pneumatic valve [65], and the hydrophobic syphon valve [66] (see **Figure 2-3b-e**).

Figure 2-3(a) Orientation of the valves relative to the center of rotation and corresponding centrifugal force. **(b)** Geometric capillary valve, **(c)** hydrophobic valve, **(d)** centrifuge-pneumatic valve, and **(e)** hydrophilic siphon valve.

A more advanced form of an active valve used in a centrifugal device was developed by the Samsung Advanced Institute of Technology, named LIFM (Laser Irradiated Ferrowax Microvalve) [30]. The valve consists of a ferrowax (paraffin wax dispersed with 10 nm-sized iron oxide nanoparticles) plug used to effectively block off channels at pressures up to ~400 kPa. To open the valve, a laser in the unit targets the ferrowax plug and melts it allowing liquid to flow through the channel. Adaptations to this ferrowax technique allow for the creation of not only normally-closed valves, as just described, but also normally-open valves where a reservoir of ferrowax is melted and allowed to flow into a channel where it re-solidifies and seals it. This technique of ferrowax valving has the advantage of being independent of rotational speed and liquid properties, giving an overall greater level of control. Such control comes at the cost of added complexity of the unit (melting laser, and associated electronics), and the microfluidic device (production now includes ferrowax integration); both of which

will contribute to the total cost of the system. Additionally, due to sacrificial nature of the ferrowax valve, opening and closing a valve multiple times is impossible.

Pressure driven microfluidics encompasses a large verity of both fluidic transportation and valving techniques. Linear actuated devices control liquid movement by mechanical displacement of a flexible pouch containing a reservoir of the liquid. Liquid movement is usually limited to one channel without alternative branches or liquid pathways [50]. Fluid movement can be well controlled by increasing or decreasing the rate at which a plunger is pushed into the flexible pouch. This method of fluid transport is well established, and even commercialized in the form of the i-STAT[®] device produced by Abbott Point of Care Inc. (NJ, USA), which relied on active liquid actuation by displacement [67].

Other pressure sourced used to transport liquid in microfluidic chips include the use of external syringes and syringe pumps, micropumps [68] [69], gas expansion principles [70] [71], and pneumatic displacement of flexible membranes, an example of its working principles is shown in **Figure 2-4** [72].

Figure 2-4 Main operations of the multilayer PDMS-based Large Scale Integration (LSI) platform using the NanoFlex[™] valve. **(a)** Valve is open until pressure is applied to control valve. **(b)** Valve is closed. Using the same principle as the valve **(c)**, pumps **(d)** and mixers **(e)** can also be designed.

A vast number of valving solutions exist for pressure driven fluid flow, well cataloged by Oh *et al.*[73]; with so many competing techniques, no standards have emerged so far. Further complicating the situation is the fact that each valving choice requires a significant knowledge base to successfully implement said valve, possibly taking months to simply duplicate the success of another group. One commonality between all of the ‘on-chip’ valving and pumping options is the reality that while each may have its own superior attributes, they each require exterior triggering mechanisms, and often additional connections to the microfluidic chip (pneumatic supply, electrical supply, physical plunger, etc.). Passive solutions require no such exterior connections or controls, but cannot deliver higher level control of fluid that is sometimes necessary. A reasonable approach, therefore, is to transfer the valving and pumping functionality off-chip. Sauer-Budge *et al.*[74] created a system that conducts bacterial lysis, nucleic acid isolation and concentration, polymerase chain reaction (PCR), and end-point fluorescent detection using remote valves and pumps. The system uses a single-use disposable chip free of any active components, and off-chip valving and pumping, allowing for a truly low cost microfluidic chip. Fluid control is accomplished using 14 valves and two fluid propulsion methodologies; syringe pumps and pneumatic dispensers. The three syringe pumps provide very accurate and metered volumes of fluid while the pneumatic dispensers use compressed air to drive fluid at higher velocities providing better mixing when necessary. With all of this external instrumentation, the resulting unit has a rather large footprint, quickly moving from a portable system to a bench-top system (see **Figure 2-5**).

Figure 2-5 Instrument developed by Sauer-Budge *et al.* to be used in conjunction with their single use lab-on-chip device. **Left:** Schematic of instrument functionality and photo of chip/instrument interface. **Right:** Picture of instrument.

This system succeeds in reducing the complexity of the microfluidic chip and the overall system when compared to others with similar fluid handling capabilities, but fails to keep the final product in the scope of a point-of-care instatement due to its size. Still, building off a remote instrumentation and control architecture, a portable system could be realized.

2.4 Point-of-Care Diagnostics

The concept of a Point-of-Care device that could be utilized directly by the consumer to allow them to take control of their health is more than 20 years in the making. Microfluidics, and the envisioned Lab-on-a-Chip, provided a path to an easy to use, disposable, and informative molecular diagnostic test, inexpensive enough to be ubiquitous in everyday life. Over the years, a large number of technologies and techniques based in microfluidics have been developed, but the market still lacks what amounts to a truly consumer ready POC device, capable of processing a drop of blood and presenting quantitative results. One major factor for this shortcoming seems to be

the fact that obtaining a quantitative result requires a more complex sensor and sample handling techniques, which are typically interpreted by a reusable and often research grade instrument (*e.g.* florescent microscope, spectrometer, mass spectroscopy) [75]. For this reason, the disposable fluidic unit / reusable reader unit, architecture has proven to be the most viable solution to a quantitative analysis requirement, allowing for a higher level of sophistication in the reader unit. The consumer may be willing to spend more on the initial purchase of the reader unit because of its ability to conduct a multitude of tests at a low per cost test of the disposable. Several systems exist that seek to fill this market vacuum; some of them are even currently commercially available.

One such device is the i-STAT[®] for quantitative ‘bedside’ testing, introduced in the 1990s by Abbott Point of Care Inc. (NJ, USA). The reusable reader unit uses test specific disposable cartridges to perform common diagnostic tests including, cardiac markers (troponin), blood gases (pCO₂, pO₂), chemistries (pH, urea, glucose) and electrolytes (Na⁺, K⁺, Cl⁻, Ca²⁺), lactate, coagulation, and hematology (hematocrit) [76]. The unit is battery powered, portable, and easily carried from room to room, simultaneously uploading test results to a patient’s records over a wireless connection. The i-STAT is marketed towards health care professionals and not available to the general public. Additionally, the device is not in a price range that most consumers would find acceptable; costing ~\$15,500 for the reader and anywhere from \$5-90 per test [77].

A device aimed at a similar market is the Alere Triage[®] System (formally Biosite Triage Cardiac Panel) which provides a rapid diagnostic test that a doctor could use to assist in diagnosing a patient’s risk of heart failure and myocardial infarction [78]

The device is a desk-top unit that again uses a single-use microfluidic “Test Device” to detect specific antigens from whole blood through the use of a florescent immunoassay on about 15 minutes [79]. The unit may be connected to a computer to relay test results directly to patient’s digital records, or the results may simply be printed out via the on-board printer. Similar to Abbott’s i-STAT, the Alere Triage is not intended to be purchased or used by the general public but rather as a tool used by healthcare professionals at a private practice or hospitals on-site.

Samsung introduced a device in 2010 based on a centrifugal microfluidic disc platform, coined, lab-on-a-disc. The system consists of a plastic disposable microfluidic disc and a bench-top reader unit. Using the “Samsung Blood Analyzer” (Samsung product number IVD-A10A) Lee *et al.* [28] reported the unit’s ability to simultaneously analyze six blood chemistries (cholesterol (CHOL), high-density lipoprotein cholesterol (HDL), triglycerides (TRIG), alanine aminotransferase (ALT), aspartate aminotransferase (AST), and glucose (GLU)) using biochemical analysis, and CK-MB (cardiac marker phosphocreatine kinase) using an ELISA type immunoassay within 22 minutes. Each disc requires only 350 μ L of whole blood, and testing is fully automated thanks to advanced passive and active (laser irradiated ferrowax) microvalves. Though described as portable, the unit weighs in at 18.3 lbs. and consumes about as much space as a computer tower (~14in x 10in x 10in) [80], moreover, the target market is again healthcare facilities where the cost of the reader unit can be justified. Despite the encouraging results published in 2010 in conjunction with a Samsung press release, no information about the Blood Analyzer or its anticipated commercial availability is exists on the official Samsung website.

Walt *et al.* [81] recently presented an automated system capable of simultaneously testing for multiple protein biomarkers in saliva. The platform uses a disposable test chip, reusable reader box architecture. Using as little as 10 μ L of saliva sample, the system measures the levels of the biomarkers using a microsphere-based array *via* fluorescence sandwich immunoassays. All the necessary reagents for analysis are contained on the disposable microfluidic chip. Fluid handling is carried out using an array of four custom actuator valves. The valves work by mechanically depressing into the elastomeric PDMS chip and closing off a microchannel. Fluid movement is generated by a miniature diaphragm pump connected to the chips outlet. The entire ‘SDReader’ system including the pump, electronics, and valving system, is powered by a 8.5A 12 VDC adapter. The reader box has a relatively small footprint at 15 cm (wide) x 15 cm (tall) x 25 cm (deep) and weighs in at only 2.9 kg. This device represents the state of the art for multiplex protein biomarker detection because it requires very little sample preparation and is accessible by minimally trained personnel.

The previous three microfluidic based Point-of-Care analyzers represent a promising stepping stone to a future unit, available at a price point accessible to the public at large. One possible solution to significantly reducing the cost of future units is to utilize and an extremely powerful piece of technology 46% of American adults already have access to; the smartphone. Of loading all communication, imaging, user interfacing, and computing to a more than capable smartphone may provide a path to significant cost reduction of the reader unit [75]. Zhu *et al.* [82] developed what could be considered a first step in smartphone/bio-analyzer integration with their cytometry

platform. The unit attaches to a smartphone and utilizes its camera and processor to image whole blood and determine a hemoglobin density and a blood cell count.

CHAPTER 3

METHODOLOGY

3.1 Microfluidic Chip

The microfluidic chip is an extremely important aspect of this research; its performance is linked to almost every other aspect of the presented systems ability to function properly. As such, significant effort was expended in developing and evolving the design of a microfluidic chip that would perform its required tasks reliably. New methods for chip fabrication were also developed to better align with design requirements, new design turn-around time, and volume chip production.

All microfluidic chips in this research were made from two pieces of bonded polydimethylsiloxane (PDMS), one of which was a blank (*i.e.* flat and featureless) sheet, while the other contained micron sized channels (microfluidic channels). Once access ports were cut or punched into the channels, and the two pieces were bonded permanently together with the channel features facing the blank surface, a microfluidic chip was created. This process is shown in **Figure 3-1**.

The silicone polymer PDMS (Slygard 184, Dow Corning, Midland, MI) was mixed at the recommended ratio of 10:1 (base:curing agent), poured over the appropriate mold and cured to an elastomeric constancy by placing the mold on a hot plate (HS40A, Torrey Pines Scientific Inc.) for an amount of time dependent on the temperature of the hotplate (different molds can be exposed to different temperatures as detailed later). Fabricating the blank sheets of PDMS is a rather trivial exercise; the

only requirement of the mold is that it be extremely smooth so as not to impart undesired features to the blank. Various substrates were utilized for creating these blank sheets, including petri dishes (booth 95 and 150mm) polished aluminum, and temperature resistant plastic. Creating the mold for the channel side is a somewhat more involved process.

Figure 3-1 Process for making a PDMS microfluidic chip. **(a)** A master mold was made, patterned with the desired microfluidic features. **(b)** PDMS was poured over the mold and was cured by heating, **(c)** the cured PDMS was peeled up from the mold. **(d)** Ports were punched into the channels and **(e)** the chip was bonded to a blank sheet of PDMS, thus creating the microfluidic channels.

3.1.1 Fabrication of Photolithography Molds

Previously, all “master” molds were created through a process called photolithography. This process is quite involved, and due to the fact that most molds in this research were not produced with photolithography, only an abbreviated description of the manufacturing process is described in the following section (a complete description can be found in Li [83]). The desired geometry of the microfluidic chip was created to scale in a drawing software package (Adobe Illustrator) by drawing channels as solid black features. This file was then sent to a

company (CAD/Art Services, Inc., Bandon, OR) that would print said geometries onto highly transparent Mylar film, creating a photomask. A layer of negative photoresist (SU-8) was spin-coated onto a 4 inch silicon wafer (SI-Tech, Inc., Topsfield, MA) with a spin coater (Laurell Technologies, Corp., North Wales, PA) before it was baked on a hot plate to harden this first layer of SU-8. This process was repeated a second time to build up the thickness of the SU-8 to the desired thickness of about 245 μm . The SU-8 layer was then exposed to 365 nm light (i-line, Karl Suss MJB- 3, Suss Microtech, Garching, Germany) for 5 min under the photomask at an intensity of 3.5 mW/cm². After exposure, the wafer was ramp baked and held for 40 min at 85 °C. The wafer was cooled to room temperature and the SU-8 was developed in SU-8 developer solution (MicroChem Corp., Newton, MA) on an Orbit Shaker (Lab-Line Instruments, Melrose Park, IL) at 300 rpm.

Creating molds proved to be a very time consuming process; from designing the chip in Adobe, waiting for the photomask to arrive in the mail, preparing the silicon wafers to be etched, traveling to MIT (Cambridge, Massachusetts) to use the facilities light source and mask aligner, and finally developing the SU-8. Under the best circumstances, the entire process took three weeks from concept to usable mold. This extended turnaround time did not lend itself to innovation and experimentation with channel geometries; a microfluidic chip that worked marginally well was rarely modified for fear of wasting weeks of effort. Other limitations associated with the SU-8/silicon master molds are also prevalent. Channel depth (height of the etched features) was limited to one height (thickness of the SU-8) throughout that entire wafer, limiting design flexibility. The molds themselves were extremely delicate and

demanded great care when handling, especially during the process of separating the cured PDMS from the mold. Finally, because of the thermal expansion of silicon and the SU-8 structures on them, the temperature at which the PDMS could be cured had to be reduced to 65°C (the lowest heat temp/cure time given in the manufactures data sheet is 100°C/35 min) increasing the cure time to 3 hours. The low mold strength necessitating tedious PDMS removal, combined with an increased cure time, culminated in process that could take weeks to make the needed number of microfluidic chips.

3.1.2 Fabrication of Machined Molds

In an effort to address the limitations of photolithography molds, a new approach to making master molds for PDMS chips was pursued. The main impetus for seeking an alternative mold fabrication technique was in an effort to reduce the turnaround time from channel geometry concept to implementation. The resulting process not only satisfied that key requirement, but it also allowed for faster generation of PDMS chips and reduced the design restrictions of the mold.

Molds were made by machining the inverse of the desired microfluidic chip out of a flat piece of material; aluminum and Ultem[®] plastic were used exclusively for this research. The first step was to create a CAD (Computer Aided Design) drawing of the desired mold (the CAD software SolidWorks was used) at a 1:1 scale (see **Figure 3-2a**). That file was then transferred to a program (MasterCam) that generated G-code (a common numerical control program language) from the CAD file. Finally, the generated G-code was loaded onto a CNC (computer numerical control) milling

machine where the mold was machined out of a flat piece of 0.25'' (Ultem) or 0.5'' (aluminum) stock using micro end mills (**Figure 3-2b**). This process resulted in a highly durable mold (**Figure 3-2c**) from which thousands of microfluidic chips could be cast.

Figure 3-2(a) Screen shot of the CAD software SolidWorks used to design all of the molds. **(b)** Magnified view of a typical micro end mill used by the CNC to machine the mold. **(c)** Photograph of a completed Ultem mold.

When designing an inverse mold to be machined, it was important to follow several key guidelines so as not to design channel geometries impossible to create using the available equipment. The microfluidic chip will be the inverse of the mold; this dictates that a positive volume should be present on the mold where an empty space (channel) is required in the microfluidic chip. The resulting CAD model therefore corresponds exactly to the geometries of the channels (air volumes) in the microfluidic chip (see **Figure 3-3**). The mold was cut using an end mill (McMaster-

Carr part# 8832A008) with a diameter of 203 μm (0.008in) and a depth of cut of 305 μm (0.012in). Due to the ‘depth of cut’ of the end mill, the maximum channel height was set at 300 μm . The end mill diameter limits all interior fillet radii to approximately 100 μm as well as limiting the wall thickness (thickness of PDMS wall between any two channels) to a minimum of 205 μm .

Figure 3-3(a) Shows the maximum channel height possible, limited by the end mill’s depth of cut. **(b)** Shows an interior radii feature and **(c)** shows the minimum distance between any two features, both limited by the diameter of the end mill.

The utilized EMCO CNC (EMCO Concept Mill 55, Austria) has an average positioning variation of 8 μm over its entire work area and a resolution of 500 nm (X, Y, and Z), providing excellent dimensional accuracy and reliably. The CNC proved capable of creating features as small as 15 μm tall. Initially, when the use of a CNC to machine out molds was first explored, the mold was modeled after the silicon molds previously used. This resulted in a mold that took more than 96 hours of continuous machining to complete. Silicon molds have no features on them except the channels themselves, resulting in large areas of negative space (area that will be filled in with

the PDMS). This is fine for silicon molds, but for machined molds, all of this volume must be removed, resulting in severely extended processing times. A new approach was taken in designing the machined mold that left as much material as possible while still providing enough sealing surface area to adequately seal one channel from another. A PDMS wall thickness of 500 μm provided adequate sealing area between the channels. Requiring only a 500 μm machined out area around any channel significantly reduced the total volume to be machined over the entire mold (see **Figure 3-4**). This one change in design philosophy reduced the production time by approximately 400%.

Figure 3-4 Illustration of the reduction in machining, from (a) with large volumes to be removed, to (b) where the design rule limits the maximum wall thickness to 500 μm , resulting in large islands of material no longer removed. Screen captures of the CNC software used; SolidWorks.

Two materials were used to fabricate the molds: aluminum and polyetherimide (Ultem[®], purchased from McMaster-Carr). Each has its advantages and disadvantages, but ultimately, Ultem was chosen as the primary mold material. Initially, molds were machined out of aluminum, providing excellent dimensional stability and even better heat flux, efficiently transporting heat from the hotplate to the curing PDMS. This meant that the PDMS chips could be cured in a mere 20 minutes at a temperature of 120°C, compared to 3 hours at 60°C using silicon molds. Unfortunately, machining

aluminum with such a delicate end mill, at the machines (relatively low) maximum spindle speed of 3400 RPM, allowed for only very low feed rates and still often resulted in broken end mills. The combination of low material removal rates and the regular destruction of expensive end mills led to long turnaround times (1-2 weeks), ultimately necessitating a mold material change. Ultem was chosen because of its high heat resistance (continuous use at 170°C), solvent resistance, and high mechanical properties. Ultem may be hard when compared to other plastics, but its relative softness to aluminum meant that the machine feed rates could be elevated without failure of the end mill. End mills cutting Ultem instead of aluminum lasted 5 to 10 times longer as well.

3.1.3 Fabrication of PDMS Microfluidic Chips

The following section describes the basic process for fabricating the microfluidic chips used in this research using Ultem or aluminum molds. Additional steps to this process, not described here, may exist for specialized chips; such additions are discussed in further sections.

The PDMS layer was produced at a specific thickness by first calculating the volume of PDMS needed to make that thickness. For instance, if a 3 mm thick (t) layer on the channel side was desired, the total area of the mold (A) was 5,250 mm² (70mm x 75mm), the total machined volume of the mold (mv) was 226.1 mm³ (volume of all material removed by the CNC), and the height of the microfluidic channels (h) was 250 μ m, then the volume of PDMS to be poured on the mold would be calculated using Equation 1.

$$Volume = [(t - h) * A] + mv. \quad [1]$$

For this example, the volume of PDMS to be poured in order to get a layer thickness of 3 mm comes out to be 14,663.6 mm³. Knowing the density of PDMS to be 0.001 g/mm³, the calculated volume can be converted to a mass; a 14,663.6 mm³ volume of PDMS has a mass of 14.66 grams. A similar procedure is followed to find the mass of PDMS to be poured in order make the blank side of the chip. For example, if a 2 mm thickness is desired, simply multiply the surface area of the blank mold by 2 and multiply by 0.001 g/mm³.

Once the total mass of PDMS needed was calculated, a disposable mixing cup was placed on a digital scale, and the scale was zeroed. The *base* part of the 2-part PDMS was added to the cup until the total mass needed was reached. The curing agent was added at a ratio of 9 parts base, 1 part curing agent (the instructions for the PDMS kit calls for a ratio of 10:1, but that often left a disproportionate amount of base left over; there was no observable difference between chips made at a 10:1 ratio and a 9:1 ratio) and the two parts were vigorously mixed, incorporating large amounts of air into the mixture. The cup was covered with a piece of paper to keep out any contaminants and placed to the side.

Walls were created around the mold (see **Figure 3-5a**) using 0.75" width Scotch[®] transparent tape (catalog # 600). The taped-up mold was then placed on a digital scale and the scale was zeroed. From the mixed cup, PDMS was slowly poured onto the taped-up mold until the correct mass was added (14.66 grams, using the above example). The mold(s) were then placed in a vacuum chamber until all bubbles were extracted from the viscous PDMS, a process that took anywhere from 5 to 30

minutes. The mold(s) were then removed from the vacuum chamber and placed on a hotplate. If the mold was made of aluminum, a temperature of 140°C and a cook time of 30 minutes was used; if the mold was made of Ultem, a temperature of 100°C and a cook time of 45 minutes was used. Additionally, to counteract the poor heat conductivity of Ultem, aluminum square stock was placed around the Ultem molds to transfer additional heat to the curing PDMS (see **Figure 3-5b**). Prior to using these aluminum ‘heat-transfer-blocks’, a persistent problem existed where the sides of the PDMS molds would not fully cure, even after extensive heating. Once the PDMS was cured, the mold(s) were removed from the hotplate and allowed to cool to the point it could be handled. Next, the tape from the side of the mold was peeled away and the PDMS chip was slowly peeled up from the mold (see **Figure 3-5c and d**). The chip was placed *channel side up* on a 8” x 8” x 5/8” thick piece of borosilicate glass (McMaster-Carr catalog # 8476K18) covered by a 6” x 6” x 1/16” thick piece of silicone rubber (McMaster-Carr catalog # 5787T31) (see **Figure 3-5e**). The chip was then cut following cut lines molded into the chip itself using an extra-wide (4”) razor blade scraper (McMaster-Carr catalog # 26765A31). This brought the chip down to its final length and width. Access ports to the microfluidic channels were punched using a 1.5 mm medical biopsy punch (Miltex[®] catalog # 33-31A) because it created the correct sized hole to seal with 1/16” OD PEEK tubing used throughout this research. The ports were punched perpendicular to the chip using a custom adapted bubble level (McMaster-Carr, catalog # 22395A21) that fit into the back of the Miltex biopsy punch (see **Figure 3-5f**). The two halves of the chip were placed in a plasma asher (Femto, Diener Electronic, Reading, PA) for 3 minutes with the faces to be bonded

face up (channel side up and flat side of the blank piece up). The next step was to take the two pieces out of the plasma asher and within two minutes place one piece on top of the other, pressing them lightly together with a gloved finger. A visual cue was present indicating good contact between the two pieces as all of the air was pushed out between the bonding surfaces. Once all of the air was worked out of the bonding surfaces, the chip was placed on a hotplate at 100°C for 45 minutes to complete the irreversible bonding process. The final step involved simply cutting the excess PDMS on the blank side down to the same size as the channel side using the 4 inch razor-blade scarper.

Figure 3-5(a) Walls to contain the PDMS are created using transparent tape. **(b)** When using Ultem molds, aluminum is placed around the mold to better transfer heat. **(c)** Tape is peeled away from the mold and cured PDMS. **(d)** Cured PDMS is slowly peeled away from the mold. **(e)** The chip is placed channel side up on the cutting surface. **(f)** Custom biopsy punch with an integrated level is used to punch holes in the chip.

3.1.4 Microfluidic Chip Design

Throughout numerous design iterations of the microfluidic chip, several main functions remained constant; the chip must have the volumetric capacity to store all of the needed liquid reagents (including room for an air buffer between each fluidic slug), it must also have the capacity to store all used liquid reagents in a ‘waste chamber’ keeping all liquids on-chip, the chip must have access ports so it can be connected to outside valves and pumps, and it must contain structures facilitating fluorescence detection. In addition to these prime duties, later versions of the chip also limited bubble formation, added a plasma-in section, eliminated problematic dead volumes, and enhanced access port geometries for better chip alignment.

The central idea behind the POC system put forward by Peng Li [83] was a disposable microfluidic chip that had all reagents necessary to complete a fluorescence based ELISA style assay sequentially stored on-chip, as well as room for the reagents to be kept in a “waste chamber” on-chip once they had been used. This technique prevented any reagents from ever leaving the chip and therefore prevented cross contamination between tests by keeping all external valves and pumps dry. Early versions of the ELISA chip utilized large storage chambers connected by smaller channels that acted as air buffer zones separating reagents from one another (see **Figure 3-6**).

Figure 3-6 Schematic of an early design for an ELISA style microfluidic chip. X's mark where access ports would be punched. Detection site and fiber optic channels are not shown.

This design proved to be highly incompatible with any pump except a manual syringe, due to the extremely high pressure drops generated when fluid slugs transitioned from the smaller channel (500 μm width x 275 μm depth) to the larger chambers (1,500 μm width x 275 μm depth). In fact, during initial testing of the above design, several micropumps were discarded due to the belief they were malfunctioning, when in fact they were performing normally but were unable to overcome such a high pressure drop. Additional problems associated with this design included significant and persistent bubble formation, again at the point where the channel geometry jumped in size. Bubbles formed as a result of the liquid slug (usually around 10 μL in volume) breaking apart and allowing a volume of air to enter between the now two pieces of liquid. This process would repeat itself, often leading to one 10 μL slug being broken up into 10-50 individual slugs separated by small volumes of air. Beyond creating problems when trying to incubate liquid reagents in a specific area (and not a volume of air), the pressure drop created by bubble formation often stopped fluidic transport in its tracks. Fuerstman *et al.* [84] showed the pressure drop in a rectangular

microchannel depends predominantly on the number of bubbles present, a relationship clearly observed throughout this research. A substantial redesign was necessary to address these issues.

Version 2.0 (see **Figure 3-7**) of the chip proved to be a significant improvement. The channels throughout the chip, except for the detection site, were kept a uniform size (500 μm width) eliminating excessive pressure drop changes due to changing channel dimensions. V2.0 of the chip introduced the idea of increasing storage volumes by using serpentine channels in place of larger channels; such channels were used for reagent storage as well as waste channels. V2.0 added inlet ports to the sample channel (sample in this case would be plasma) to allow for plasma to be pulled into the chip without disturbing pre-loaded reagents 1 and 2 or the liquid slug of storage buffer in the detection site.

Figure 3-7(a) V2.0 of the ELISA microfluidic chip, **(b)** V2.1 brought with it some subtle design changes, the most obvious being the change to the detection site.

Several issues persisted resulting in the creation of V2.1, a change to the detection site being the most noticeable difference. Often during fluidic testing of V2.0, bubbles would form at the beginning and end of the detection site, despite the somewhat gradual change in channel width. Initially, changing the detection site was not thought of as a viable solution because all previous benchmarking of the sandwich ELISA was done using that detection site design. There was a fear that deviating from that design might yield weaker signal strength, especially when moving to a design that would decrease the overall volume of the detection site. However, upon further study of V2.0 and V2.1 detection sites (see **Figure 3-8**), it was found that although the volume of the detection site would decrease from $\sim 0.62 \text{ mm}^3$ (V2.0) to $\sim 0.49 \text{ mm}^3$ (V2.1), the surface area would actually *increase* from $\sim 5.3 \text{ mm}^2$ to 5.6 mm^2 . Therefore, it was concluded that the signal strength may actually increase switching to V2.1 based on the fact that the sandwich assay is built on the surface of the channel, thus increasing the available binding sites and overall signal strength.

Figure 3-8(a) Zoomed in isometric view of the mold at the detection site for V2.0 and, **(b)** V2.1.

The only other change between V2.0 and V2.1 was the addition of a port between the reagent 1 and 2 channels (see **Figure 3-7**), facilitating easier loading of reagent 2. Besides these two differences, V2.0 and V2.1 were identical; they both had ports

located above and below the detection site for functionalization, both contained ample volume in the waste storage channels to keep all liquids on-chip, and both had fiber optic channel inlet guides leading to the detection site. The utilized detection method using quad-furcated glass fiber optics connected to an Ocean Optics miniature spectrofluorometer (model # USB4000) is the same as described by Peng Li [83].

Version 3.0 of the sequentially loaded immunoassay chip was designed to address serious issues still present in V2.0 and V2.1. For example, all efforts to pump plasma into the chip without significantly disturbing the positions of the other liquid slugs were unsuccessful. This problem led to the first and most significant design change to V3.0: a dedicated plasma channel, connected to the rest of the chip by a Y-section (see **Figure 3-9**).

Figure 3-9 V3.0 of the sequential loading chip. All channels have a width of 500 μm and a depth of 275 μm . Air volumes separating liquid volumes are unlabeled but reside between each liquid volume.

The hope was that a dedicated plasma channel, isolated by an open port at the end of the channel before the Y-section, would prevent movement of any other liquid slug in

the chip. Once the plasma was in the chip, that port (labeled “1” in **Figure 3-9**) would be valved off, allowing the sample fluid to continue to the detection site. The plasma channel had the capacity to hold ~ 5 μL of sample fluid with room for a small air gap between the Y-section and the sample. The two ports at the beginning of the plasma channel is a result of the whole blood filtration system, described in section 3.2.

V3.0 also introduced a new wash step to be run over the detection site immediately following the sample fluid (plasma) to reduce the amount of nonspecific binding at that location. The addition of this wash step required that the channel length before the detection site grow in order to accommodate this new step; **Figure 3-9** shows this new section labeled ‘plasma wash’. Additionally, the volumetric capacity of the “waste storage channel” grew accordingly. The high frequency serpentine channels of V2.0 and V2.1 used for reagent storage were changed in favor of long serpentine channels in an effort to reduce the number of sharp 180° turns the fluid had to pass through. This design ideology formed as a result of numerous observations of fluid separation and bubble formation occurring in these sections of dense 180° turns. From V2.0 to V3.0 the number 180° turns, not including the waste storage channel, were decreased from 42 to 11. The severity of these turns was also changed; increasing the inner radius of the turns from 250 μm to 500 μm . All changes from V2.0 (and V2.1) to V3.0 were accomplished while keeping the overall footprint of the chip almost identical. Some other minor design notes on the V3.0 chip include a port at the beginning and end of every channel section, making the task of loading individual fluid volumes easier. Finally, a small change was made to the fiber optic

inlet geometry (compare **Figure 3-7** and **3-9**), eliminating a plateaued section where the fiber optic cable would often catch while it was being inserted.

Version 4, seen in **Figure 3-10**, introduces the last major redesign of the microfluidic chip, and closely resembles its final form presented in this work. Major changes from V3.0 to V4.0 include the addition of high abundance removal sites, a 55% reduction in channel cross sectional area, a geometry change to the detection site, a 78% increase to the chip's footprint, the addition of a washing step, and modified port geometries for better alignment and reduced dead volumes.

Figure 3-10 V4.0 of the sequential loading chip. All channels have a width of $300\ \mu\text{m}$ and a depth of $250\ \mu\text{m}$. This marks the last major redesign of the microfluidic chip.

V4.0 added channels used to reduce the concentration of high abundance proteins in the plasma sample before it entered the detection site. The necessity of these channels was based on the work done by Godfrin [23], who showed the presence of high abundance proteins in human plasma would interfere with the detection of lower concentration proteins like CRP. A better limit of detection was accomplished

by incubating the plasma sample in channels functionalized to capture these high abundance proteins. Two proteins were selected for reduction, necessitating two sections after the plasma sample channel and before the detection site where the sample could incubate. These channel sections, dubbed high abundance protein depletion sites, had their own inlet ports to facilitate their unique functionalization schemes.

After several runs of experimentation, and out of an abundance of caution, it was decided that a two-step washing scheme, used in the detection site after the detection antibody was incubated, should be used. This decision necessitated the addition of a fourth section (see **Figure 3-11a**) for a wash containing a mild detergent; phosphate buffered saline with Tween[®] 20 (PBST), to run through the detection site following the detection antibody. The PBST would be slightly more aggressive at washing away any detection antibodies not specifically bound to the surface in the assay, thereby reducing the negative control signal strength.

All channels in V4.0 of the chip were changed to a width of 300 μm and a depth of 250 μm (previously 500 and 275 μm respectively) in an effort to magnify capillary forces and reduce fluid separation and subsequent bubble formation. This change resulted in a channel cross sectional area reduction of nearly 55%, dropping from 0.1375 mm^2 to 0.075 mm^2 (see **Figure 3-12b**). The Y-section from V3.0 was carried over to V4.0 keeping the sample plasma channel somewhat isolated from the rest of the chip. The V4.0 Y-section itself is identical in function to V3.0's but changed in form and location (see **Figures 3-10 & 3-11c**) to better utilize space and keep the chips footprint to a reasonable size. The detection site was changed from the

“S” design to a straight channel in favor of simplicity and an extremely reduced chance of bubble formation.

Finally, port geometries were modified in two ways to address two issues present in past chip designs. First was the issue of dead volumes, defined here as a volume not filled with a liquid and outside of the main channel of the chip. This includes the volumes created by the ports, the inlet channels to the ports, and the tubing connected to said ports. This section will discuss just the dead volumes created by the ports and their inlet channels. These volumes created problems during pumping, allowing small pressure differentials to exist within a channel, separating and trapping small volumes of various fluid slugs into these pockets. This not only inevitably led to small amounts of mixing between what should be separated fluid slugs (*e.g.* detection antibody and a washing step), but would also generate what was termed a “bubble factory,” wherein the driving pressure within the chip would build enough to pull off a small portion of the trapped fluid into the main channel. This process could repeat multiple times, creating small slugs ranging in volume from ~ 50 to 250 nL, resulting in severely retarded or even fully stopped flow rates. To prevent liquid volumes from getting trapped in the port inlets in the first place, the channel connecting the port to the main channel and the port itself were severely shrunk (see **Figures 3-11b & c**). This reduction in channel and port size not only reduced the dead volume by more than 26 times (2.72 mm^3 and 0.10 mm^3 , V3.0 and V4.0 respectively) but created a type of capillary valve, where the energy required for a fluid to enter the port channel would be more than required to continue on its normal path. This

prevented fluid from ever entering the port channel and all but eliminated bubble formation at the ports.

The second change to the ports was made in an effort to precisely position the chip by better locating the punch that made the holes in the chip. Previous designs of the port ring allowed play in the location of the biopsy punch, thereby allowing a small variance in the exact position of the punched holes. These holes were used to not only connect the chip to the outside world, but to also locate each chip in the exact same spot every time; a crucial step if the fluid control system was to work properly, discussed in section 3.5.2. The inner hole of the port was enlarged from 1 mm to a 1.2 mm diameter, and the outer diameter was shrunk from 2 mm to 1.75 mm. **Figures 3-11b** and **c** illustrates the difference between the V3.0 and V4.0 port geometries. This effectively eliminated any variance in the location of the punched hole relative to the chips channels, and used in combination with the custom biopsy punch (see **Figure 3-5f**), precision placement of the chip was possible.

Figure 3-11 Various changes to V4.0 of the microfluidic chip including, (a) the addition of a PBST wash step. (b) Isometric view of an inlet port found on V3.0 and (c) an isometric view of the Y-section and an inlet port found on V4.0.

The final version of the microfluidic chip (V4.1) was derived directly from V4.0; it carries with it two minor and one major change to the design. A small change was made to the port geometry to further reduce dead volumes of unused ports (ports not punched out) and all inlet ports above the plasma channel but one were erased. The more important change, in terms of fluid control, was made to the waste storage section of the chip, a section that in V4.0 consisted of 33 closely packed serpentine channels.

Figure 3-12(a) Version 4.1 of the microfluidic chip showing the major change to the waste storage section. **(b)** Scale representation to the change in cross sectional area of the main channels from V3 to V4. **(c)** Close-up isometric view of a V4.1 port.

Conducting tests using V4.0 of the chip proved to be challenging once fluid started to move from the top of the chip to the bottom waste channels because of the increase in pressure drop required to move the fluid. It was found that the pressure required to pump the liquid slugs increased significantly from V3.0 to V4.0, a consequence of decreasing the cross sectional area (see **Figure 3-12b**) and increasing the overall length of the channels from ~ 730 mm to $\sim 3,150$ mm. With the bottom

waste channels significantly contributing to this increase in overall channel length, an alternative method for storing the waste fluid was explored. For V4.1 of the chip, 28 of the serpentine waste channels were replaced with an open volume measuring 20 mm x 15mm (see **Figure 3-12a**) where a piece of filter paper (Whatman Grade 41) of the same size could be inserted before chip assembly. The paper, having a thickness of 220 μm , was cut to fit precisely in the chips rectangular cavity with a CO₂ laser (Epilog Mini 24 Laser). The addition of the filter paper waste chamber greatly reduced the pressures necessary to move fluid throughout the chip, and totally eliminated any opportunity for bubble formation once the fluid had reached the paper. Capillary forces in the paper would actually pull in the entire volume of a fluid slug, unassisted by outside pressures, once it made contact with the paper. The liquid capacity of the 20 x 15 mm piece of filter paper was more than sufficient to absorb all reagents and washing steps.

As seen in **Figure 3-12c**, the port geometry was again changed to reduce the dead volume of un-punched ports. If the port was not punched, the dead volume at each port was only 6 nL, down from 104 nL in V4.0. This change had the added benefit of allowing the circular port geometry to have a depth of 250 μm (compared to 75 μm for V4.0) and not contribute to the dead volume of the un-punched port. A deeper port made for much faster chip fabrication, as locating the correct punch location - a process where one *feels* where the port is located using the biopsy punch - on V4.0 proved to be quite time consuming. The seven inlet ports connected to the top channels of the chip were erased because exact positioning of the various liquids was no longer important thanks to the feedback method used for fluid control. Instead, the

fluids were all loaded sequentially loaded through the top right port (**Figure 3-12a**) of the chip, leaving an air buffer of approximately one channel length (channel length = 46.5 mm $\approx$ 3.5 μ L) between each fluid volume.

3.2 Cell Removal from Whole Blood Sample

Based on previous studies [23], a filter media produced by Pall Corporation was used to create an on-chip plasma separation solution. The first step was to determine how much plasma would be needed for each test and work backwards from there. The plasma channel in the final version of the microfluidic chip (V4.1) had a channel volume of 4.5 μ L, so at least that much plasma should be available. Pall Corp. produces three grades of plasma separation membranes; Vivid™ GF, GX, and GR. Based on previous experimentation [23] and the documentation available from Pall Corp. [41] the GX grade was chosen. Pall Corp. lists the GX as being especially well suited for “small blood volume applications, [found in] microfluidic and lateral flow POC devices”, and also has a proprietary post-treatment that helps minimize hemolysis. Practical tests comparing the performance of these three grades confirmed that the GX grade was the ideal choice.

3.2.1 Initial Testing of Plasma Separation Membrane

Previous benchmarking tests of the membranes were all conducted with a pull-through scheme, where typically a small (10mm x 10mm) piece of filter media was placed over a 1.5mm hole punched in a piece of PDMS. A negative pressure was

applied, effectively pulling the plasma through the membrane and into a piece of tubing inserted in the PDMS (see **Figure 3-13**)

Figure 3-13 Initial tests of the separation membrane used a negative pressure to pull the plasma through the membrane and into the channel.

This technique reliably produced 10 to 15 μL of RBC free plasma from 50 μL of whole blood (~ 2 drops) using a 150 mm^2 piece of filter media at a pumping rate of 150 $\mu\text{L}/\text{min}$ [23].

Implementation of this technique into a sequential loading chip was first experimented with V2.0 of the chip. Mirroring previous successes, an attempt was made to *pull* the plasma through the filter media and into the plasma channel. In **Figure 3-7a**, the port that the filter was placed over is labeled with a 2, and the port where the negative pressure was applied is labeled with a 1. The researcher theorized that if all other ports were blocked off (*i.e.* taped over), all of the other fluid volumes in the chip would stay in place while the plasma filled the plasma channel. This theory proved to be quite wrong; none of the other fluid volumes stayed in place and no plasma was filtered into the plasma channel. This failure resulted in a major change to the plasma channel and its layout, as seen in V3.0 of the chip. **Figure 3-9** shows how the plasma channel was isolated and connected to the rest of the chip with a Y-section. Again, an attempt was made to pull the plasma into the chip. Port 2 was covered with

a piece of filter media and a negative pressure was applied to port 1. The results were no better; all of the other fluids moved and little to no plasma entered the chip. This was a clear indication that a change was still needed in the way the plasma was to get into the chip.

3.2.2 Whole Blood Filtration Integration

Previous methods for integrating the filter relied on creating a pressure differential (a negative pressure) in the plasma channel and thus the chip as a whole, leading to the aforementioned migration of other loaded fluids. If the other fluids were to remain stationary, no pressure differential could exist within the chip. Pulling the plasma into the chip was therefore abandoned in favor of *pushing* the plasma through the filter and into the chip. This would allow for a pressure driven flow of plasma without introducing any pressure to the plasma channel or the chip. **Figure 3-14** illustrates how a positive pressure was used to push whole blood through the filter, yielding plasma.

Figure 3-14 Schematic of the technique used to push plasma into the chip using a positive pressure.

This technique utilizes two additional pieces labeled “Filter Holder” and “PDMS Top” in **Figure 3-14**. The Filter Holder was designed with a ridge in it to accommodate the

thickness of the filter ($\sim 350\ \mu\text{m}$) so as to provide an air tight seal between the filter and the microfluidic chip. Based on previous yield rate studies using this same filter media, a filter diameter of 10 mm was chosen. This size yielded a surface area of $78.5\ \text{mm}^2$, large enough to produce the required $5\ \mu\text{L}$ of plasma. A mold for the Filter Holder piece was designed (Appendix 6.3, Drawing #01) that accommodated the 10 mm diameter filter. The 10 mm diameter filter pieces were cut out of an 8.5" x 11" sheet using a 10 mm biopsy punch (Acuderm[®] catalog # P1025). The mold for the Filter Holder was made (**Figure 3-15**) with a $350\ \mu\text{m}$ relief to account for the filters thickness.

Figure 3-15 Photograph of the mold used to make the Filter Holder piece from PDMS. The circle has a diameter of 10 mm and stands $350\ \mu\text{m}$ proud.

Once the Filter Holder was made (same as the procedure for the microfluidic chips) an 8 mm hole was punched (Acuderm[®] catalog # P0825) through it to allow for whole blood addition. Permanent adhesion between the Filter Holder and the microfluidic chip was accomplished using the same plasma ashing procedure described in section 3.1.3. The “PDMS Top” was simply a piece of PDMS large enough to cover the Filter Holder and has a 1.5mm hole punched in it where the positive pressure is introduced. The Top sealed against the Filter Holder by static adhesion, easily accomplished

between two pieces of smooth clean PDMS. This method allowed plasma to be pushed into the chip without disturbing the other loaded fluid volumes. This can be attributed to the fact that the pressure differential driving the whole blood through the filter and the plasma into the chip is located outside of the microfluidic channels, in a space above the filter created by the Filter Holder. Testing showed no leakage of red blood cells into the chip by traveling around the filter's edge.

Versions 3-4 of the chip (**Figures 3-9, 10, 12**) each contain a feature at the beginning of their plasma channel described here as a double port branch. This feature was specifically designed to accommodate the previously described “push” whole blood filter; one port leads to the bottom of the filter, and one port leads to a fluid control system.

The microfluidic chip is composed of a channel side and a blank side. The channel side is where all of the access ports are punched, leading to the microfluidic channels themselves, while the blank side is featureless. To connect the chip to a device, it is ‘plugged in’ to several pieces of rigid PEEK tubing protruding up from a surface into the ports. This orients the chip channel side down, with all of the access ports facing down. This creates a problem when trying to integrate the filter because it must obviously be oriented facing up to allow the user to introduce a blood sample, hence the need for a second port in the plasma channel. Once the microfluidic chip is fully assembled sans filter, a 1.5 mm hole is punched through the blank side of the chip using the top hole of the double branch as a guide. It was on the blank side of the chip, above the top port hole that the filter was placed (see **Figure 3-16**).

Figure 3-16 Schematic of how the filter is integrated onto the microfluidic chip and how the double branch feature at the beginning of the plasma channel is utilized.

The hole for the top port on the channel side is then closed off using transparent scotch tape. During the whole blood filtration process, the side port was closed via an external valve, and the relief port at the end of the plasma channel was opened until the plasma channel filled with an adequate volume of plasma. At this time the relief port could be closed and the side port opened, allowing a pressure to push/pull the plasma to the detection site. **Figure 3-17** shows what the resulting microfluidic chip with an integrated plasma separation membrane looked like.

Figure 3-17 (a) 3D rendering of the integrated filter on V2.0 of the microfluidic chip and (b) a photograph of an integrated filter on V3.0 of the chip containing water with food coloring representing plasma (red), wash1 (brown), detection Ab (blue), wash2 (green), and storage buffer (dark green).

3.3 Pump

A crucial link in the proposed fluid handling system is the pump, or method by which a pressure differential is created, used here to drive the fluid volumes through the chip and plasma through the filter. Often a major problem in past studies, the pump must be capable of a high enough driving force to move the fluid but also be capable of very low flow rates (100 – 1,000 $\mu\text{L}/\text{min}$), two qualities often mutually exclusive in micropumps. Bench-top tests of the chip and its microfluidic characteristics were often conducted using a 500 μL syringe (Hamilton, item # 81217) and a syringe pump (New Era Pump Systems Inc., NE-1000), a combination capable of extremely low (<1 $\mu\text{L}/\text{min}$) and precise flow rates at practically unlimited pressures (up to 500 psig with the above syringe). This setup worked fairly well for bench-top experiments, but if the system was to shrink to a portable size, an existing micropump needed to be utilized or a unique one needed to be engineered.

3.3.1 Piezo Actuated Membrane Pump

Initially, a piezo (piezoelectric) actuated dual membrane micropump was extensively tested and modified for this application. The micropump (Bartels Micropumps, item # mp6) preformed at varying levels of success in previous studies [24],[25],[83]; often failing to overcome any pressure drop created by bubble formation, driving the fluid at unreliable flow rates (too fast or too slow), and requiring regular fluid flushes to extricate trapped fluid droplets that rendered the micropump useless. **Figure 3-18** shows a version of the mp6 with a clear plastic casing allowing visualization of the piezo actuated diaphragms. These diaphragms

work in concert to produce an oscillating pressure wave (5-300 Hz), driving the liquid or gas from one port to the other.

Figure 3-18 Photograph of a Bartels mp6 micropump.

One of the principles of the disposable microfluidic chip/reusable reader unit architecture, as it is presented here, is the idea that no fluid (reagents/plasma/washes) will leave the chip once they have been inserted. This guideline prevents any cross contamination between tests and requires that only the chip be disposed of after each test. Subsequently the micropumps working fluid was always air, and unfortunately, the Bartels micropump's threshold for pressure generation is more than 4 times less in gas than it is in water. Evident by past failures to reliably control fluid volumes in the chip by pumping air, a method was explored wherein the micropump would pump water from one enclosed reservoir to another, creating a pressure change in the air pockets above the reservoirs; this pressure would be ported to the chip for fluid transport. **Figure 3-19** shows a schematic of the setup used to test this method.

Figure 3-19 Schematic of the two reservoir method, allowing the mp6 to pump water. Here the micropump is pumping water from reservoir 2 to 1 creating a positive pressure in the air pocket of reservoir 1.

According to the manufacturer of the mp6, the maximum back pressure that can be generated when pumping air is 150 mbar, and 650 mbar when pumping water [85]. The increase in driving force using this method was extremely evident during testing, providing reliable pressures and fluid flow in the chip despite occasional bubble formation. Unfortunately, this method proved to not be a viable solution due to excessive flow rates. When pumping air at high back pressures (*i.e.* driving fluid in the chip) with the mp6, there was a large amount of backflow which greatly reduced the effective flow rate, but when pumping water like in **Figure 3-19**, there was no backflow in the pump, resulting in a much higher flow rate. Several methods were experimented with in an effort to reduce the flow rate while maintaining a high driving force, including the use of a micron sized orifice placed between the reservoir's air pocket and the chip, and tuning of the micropump's frequency and amplitude. These efforts yielded little success, leading to the conclusion that an alternate method for pumping must be sought.

3.3.2 Dual Chamber Fixed Volume Syringe

During testing of Version 4 of the microfluidic chip (**Figure 3-10**), it became apparent that using a single pressure to either push or pull all of the fluid volumes through the chip had become less reliable, even when using the syringe/syringe pump combination that set the standard for reliability. Due to the massive increase in chip resistance from V3.0 to V4.0, the pressures required to move the fluidic volumes also increased. This resulted in a phenomenon where a large pressure built up at the bottom of the chip (assuming the fluid was to be pulled through the chip) before any fluid movement occurred, and once it did, the bottom fluid (fluid volume closest to the bottom port) would move first, then the second fluid, then the third, and so on until finally all fluid volumes were moving, each at a different rate; the bottom fluid moving approximately 3 to 4 times faster than the top fluid. The movement of these fluid volumes can be conceptualized by imagining chain of alternating springs and blocks lying on a surface (**Figure 3-20**).

Figure 3-20 Model for the condition in a microfluidic chip containing many fluid volumes separated by volumes of air with one driving force.

The springs represent the compressible nature of air, the blocks represent the incompressible fluid volumes, while the friction between the blocks and the surface correspond to the surface tension between the fluid volumes and the walls of the microchannel. Similar to the chip, the driving force must overcome the friction

between the first block and the surface to initiate movement by loading the first spring sufficiently, then the movement of the first block will begin to load the second spring, eventually moving the second block and so on, until all blocks are in motion. This example also illustrates the difficulty of fluid control, as even if the driving force is stopped, the first block will continue to move until the first spring compresses enough to stop its movement and so on down the line. A quick analysis of this model presents a way to better control the moment of these blocks, and similarly the fluid volumes in the chip.

If an equal driving force was applied at the other end of the chain in the same direction, as seen in **Figure 3-21**, a better response time to changing pressures could be accomplished.

Figure 3-21 Model for the condition in a microfluidic chip containing many fluid volumes separated by volumes of air with two driving forces.

This model is equivalent to having a positive pressure at the top port of the chip and a negative pressure at the bottom port. Practical testing using two syringe pumps in V4.0 of the chip showed significantly improved fluidic control over using a single syringe pump; fluid volumes would start and stop faster and had more equal flow rates to each other. All efforts to find a commercially available pump of a manageable size that could be incorporated into a portable device were unsuccessful, leading to the creation of a custom syringe and syringe pump.

The custom syringe, dubbed the dual chamber fixed volume syringe (patent pending), was designed to create a simultaneous infusion and withdrawal condition, one of each occurring at the ports located at opposite ends of the device (**Figure 3-22**). The device could either be operated by hand or in conjunction with a linear force mechanism, like a syringe pump, and was created to provide equal but opposite pressures, similar to the pressures that could be created using two syringes and two syringe pumps.

Figure 3-22 CAD rendering of the dual chamber fixed volume syringe. The aluminum body has been made transparent for the purpose of showing the inner construction of the device.

When a force was applied to a plunger rod that was sufficient to move the plunger, a pressure differential was created between chamber 1 and chamber 2, one having a positive pressure and the other a negative pressure. Due to the design of incorporating a plunger rod at each end, the volumetric change between chamber 1 and 2 is always the opposite of one another. The volumes of chamber 1 and 2 change proportionally to one another, and thus their pressures also change proportionally. For example, if the

plunger rod is moved in the direction of the force arrow in **Figure 3-22** enough to increase the volume of chamber 1 by 50 mm^3 , chamber 2 will decrease in volume by 50 mm^3 , and if each chamber starts at the same volume and the ports are closed, each chamber will experience an equal but opposite pressure change. The body and plunger are made of aluminum, and the plunger rods are made of polished steel to diminish any pressure leakage. A biopsy punch (Miltex[®] catalog # 33-38) was used to cut the end plugs out from a sheet of silicone rubber (McMaster-Carr, item # 5787T35) and a 1.5 mm biopsy punch was used to cut a through hole for the plunger rod. Rubber O-rings (McMaster-Carr, item # 2418T113) sat in grooves cut into the plunger creating an air tight seal between chambers 1 and 2.

The performance of the DCFVS (dual chamber fixed volume syringe) was evaluated by mounting it in a syringe pump, like any other syringe, and connecting the ‘positive’ port to the top of the chip and the ‘negative’ port to the bottom of the chip via Tygon[®] tubing (Cole-Parmer, item # 95609-32, 0.0512” ID – 0.1192” OD), creating simultaneous push and pull pressures in the chip. Encouraging results from this prototype led to an improved version of the DCFVS as well as a custom miniature system to drive the syringe, capable of being incorporated into a portable device.

3.3.3 Dual Chamber Fixed Volume Syringe V2 and Driver

The second iteration of the DCFVS kept the same basic configuration as the first, but better material selection and construction greatly reduced the driving force required to operate it. This was an important improvement because the lower driving force requires less power, thereby increasing portability and battery life of the device.

Friction between the plunger and the inner wall of the aluminum body (see **Figure 3-22**), as well as friction between the plunger rods and the silicone rubber end plugs was so high that driving the syringe even by hand was difficult. To address this issue while still maintaining air tight seals, the main body of the syringe was changed to glass, while the plunger rods and end caps utilized components from a pneumatic linear actuator.

The body of the syringe was made from 0.5'' OD tubular gauge glass (McMaster-Carr, item # 3723K11) because of its smooth inner surface, allowing for less O-ring interference while still maintaining a positive seal. The O-rings mounted on the plunger were themselves changed to a softer rubber (from A70 to A50) to give better sealability at lower contact pressures. The two plunger rods and rod seals were taken from a pneumatic linear actuator (SMC Pneumatics, part # CXSL6-50). The linear actuator was ideal because its guide rods are specifically designed to be air tight while limiting friction. This is accomplished by using special v-shaped sealing washers riding against the polished steel guide rods. These seals were mounted in aluminum CNC machined end caps (Appendix 6.3, Drawing #02) sized to fit over the glass tubing (see **Figure 3-23**). Holes for PEEK tubing were also drilled into the end caps. The plunger was machined from a 0.375'' polycarbonate rod (Appendix 6.3, Drawing #03) with grooves for the O-rings and holes for the plunger rods. Before final assembly, about 500 μ L of silicone grease (McMaster-Carr, item # 1204K32) was used to coat all of the rubber sealing surfaces. The end caps were glued in place with Dow Corning 700 clear silicone sealant (McMaster-Carr, item # 7425A51).

Figure 3-23 Version 2 of the dual chamber fixed volume syringe. Inset picture shows the seal between the rubber V-washer and the plunger rod.

V2 of the DCFVS required significantly less force to operate than the prototype. Accounting for the plunger rods, the inner cross sectional area of the chambers was calculated to be 39.64 mm^2 , which is equivalent to an inner syringe diameter of 7.1 mm. V2 would therefore pump $\sim 40 \text{ }\mu\text{L}$ of air per mm moved.

Focus was then shifted to designing a device that could accurately drive the DCFVS but still be compact enough to be incorporated into a portable device. Borrowing from the construction of a syringe pump, a small stepper motor was used in conjunction with parts from a miniature linear actuator to create a device capable of precise linear movement and adequate driving force to operate the DCFVS.

The stepper motor (NMB Technologies Corporation, model # PG20L-D20-HHC0) was chosen because of its small size ($\sim 33 \text{ mm}$ length x 20 mm ϕ), low power consumption, and built in planetary gear box with a ratio of 1:91. The gearbox significantly increased the maximum torque output ($3922.8 \times 10^{-4} \text{ Nm}$) and the number of steps per rotation (1,818 with, 20 without) allowing for lower flow rates.

Having chosen a capable motor, the next step was to devise a way to turn this rotational force into a linear force. This was accomplished by using parts from a linear actuator (Firgelli Technologies Inc., model # L12-50-210-06-I). **Figure 3-24a** shows a photograph of the specified linear actuator with a dotted line depicting approximately where the housing was cut, separating the (used) linear drive section from the (unused) motor and gearbox. The linear drive section translated 5 mm per rotation. Once the motor and gearbox were separated, a housing (Appendix 6.3, Drawing #04) was designed and rapid prototyped that accepted the linear drive section, stepper motor, and allowed for wires connected to the position sensor to exit; this housing is shown in **Figures 3-24b-d**.

Figure 3-24(a) Photograph of the specified linear actuator with a parting line depicting approximately where it was cut. **(b)** Linear drive section installed into the housing. **(c)** View of the back where the stepper motor is connected and **(d)** a view from the front of the connected pieces.

Figure 3-24c shows both the back of the housing that mates with the flange of the stepper motor, and, looking through the hole of the housing, shows the connection to the linear drive section.

Designed in conjunction with the above housing, a drive linkage (Appendix 6.3, Drawing #05) connecting the stepper motor's drive shaft to the linear actuator's drive nut was rapid prototyped. The linkage, shown in **Figure 3-25** installed on the stepper motor, is what linked the stepper motor to the linear drive section. It should also be noted that the final version of this linkage was modified by changing the tip from (rapid prototyped) plastic to steel after it sheared off several times during normal operation.

Figure 3-25 Photograph of the drive linkage installed on the stepper motor.

The stepper motor was connected to the linear drive section housing with two 4-40 socket head cap screws; **Figures 3-24c** and **3-25** illustrate how the two pieces fit together.

With the stepper and linear actuator combined, a piece was designed that connected the DCFVS to these two pieces. This piece (Appendix 6.3, Drawing #06 and #07), shown in **Figure 3-26**, cradled and securely clamped in place the DCFVS and the previously designed linear actuator housing. The ears on the cradle piece lined up with the holes on the stepper motor flange and the holes on the linear actuator housing and were tapped to a 4-40 thread, securely connecting all three pieces once the socket head cap screws were installed.

Figure 3-26 CAD rendering of the DCFVS cradle connected to the linear actuator, its housing, the stepper motor, and the DCFVS.

The final step in creating a working pump was to connect the end of the linear actuator to the end of the plunger rod. A piece (Appendix 6.3, Drawing #08) was designed and machined out of aluminum to link the two pieces together. The link was fastened to the plunger rod with a 4-40 set screw while the end of the linear actuator was attached to the link by feeding a 4-40 socket head cap screw through the hole at the end of the linear actuators arm (see **Figure 3-27b**). With this piece, the mechanical aspect of the pump was completed, but during testing of the pump it became clear that excessive heat was being generated in the metal housing of the stepper motor. After just a few minutes of operation, the stepper became too hot to handle and there was a fear that the stepper would become permanently damaged from overheating. To more quickly dissipate the generated heat, a heat sink was designed (Appendix 6.3, Drawing #09) and machined out of aluminum that fit snugly around the metal housing of the motor (see **Figures 3-27a-b**). Additionally, the heat sink had cutouts for the end cap of

the DCFVS, the PCB on the stepper, and flat cut-always on each side allowing it to fit in the portable unit.

Figure 3-27(a) Photograph of the back side of the stepper and heat sink. **(b)** Photograph of the assembled pump to be installed into the portable unit.

At this point the pump was ready to be installed into the portable unit with no further modifications. Tests of the pump outside of the unit showed it to be quite capable, even at very low pumping rates. The pumping rate was calculated multiplying a conversion factor by the stepper speed (pulses/sec), resulting in a volumetric flow rate ($\mu\text{L}/\text{min}$). The conversion factor is 6.541 ($\text{sec } \mu\text{L}/\text{min pulses}$) and was calculated in the following way:

$$\frac{39.64 \mu\text{L}}{\text{mm}} \cdot \frac{5 \text{ mm}}{\text{rev}} \cdot \frac{\text{rev}}{1818 \text{ pulses}} \cdot \frac{60 \text{ sec}}{\text{min}} = 6.541 \frac{\text{sec} \cdot \mu\text{L}}{\text{min} \cdot \text{pulses}}$$

giving the units of $\mu\text{L}/\text{min}$ once multiplied by the stepper speed (pulses/sec).

3.4 Protein Detection in Microchannels

3.4.1 Functionalization of the Detection Site

The functionalization scheme optimized for PDMS microchannels, developed by Li [83], was used because it offered high affinity binding to the constant region of

immunoglobulin G (IgG) antibodies. This functionalization was used to immobilize capture antibodies onto the surface of the microchannels and also to help mitigate non-specific binding of proteins. This portion of the functionalization only concerned the detection site of the chip, the portion where the sandwich assay would be built and eventually the fluorescence would be read; therefore, all other ports of the chip besides the two immediately above and below the detection site (see **Figure 3-12a**) were closed off using transparent tape. This allowed liquid reagents to be inserted into the detection site only, building a functional surface only where it was needed. A detailed standard operating procedure (SOP) for creating the reagents used in the following functionalization method can be found in Appendix 6.7. Unless otherwise specified, all fluid volumes were injected into the chip using a 0.5-10 μL pipette (Eppendorf Reference[®]) and a modified 10 μL pipette tip. Modifications to the tip consisted of simply cutting down the end so it sealed when inserted into the ports of the chip.

The first step in functionalizing the chip required bovine serum albumin (BSA) (Sigma-Aldrich), at a concentration of 1.5 mg/mL in (1X) phosphate buffered saline (PBS) (Fisher-Scientific), to be incubated at room temperature for 4 hours in the channel. The channel was then cleared of the BSA solution, rinsed with 50 μL of the PBS, and aspirated. Glutaraldehyde (Sigma-Aldrich) at a 0.4% (v/v) concentration in ultrapure water (Fisher-Scientific) was then incubated in the channel for 1 hour at room temperature. The channel was then cleared of the Glutaraldehyde solution, rinsed with PBS, and aspirated. Protein A (Sigma-Aldrich) was then incubated in the microchannel at a concentration of 50 $\mu\text{g/mL}$ in PBS buffer at room temperature for 1 hour. The channel was again cleared of the protein A solution, rinsed with PBS, and

aspirated. If the chips were not to be used immediately, a storage buffer was inserted into the channel, capable of preserving the surface chemistry of the functionalization for at least 6 months [83]. The storage buffer consisted of 0.05% (w/v) BSA, 0.05% (v/v) Proclin 300, 5% (v/v) glycerol anhydrous, and 10 mM of Tris in ultrapure water.

Figure 3-28 shows a schematic of the described biofunctional layer built on PDMS.

Figure 3-28 Schematic of the functionalization used in the PDMS channels to build a biofunctional layer, the top of which (protein A) has a high affinity for IgG antibodies.

3.4.2 Functionalization of the High Abundance Protein Depletion Sites

Based on work by Godfrin [23], purification of the blood filtrate, *i.e.* plasma, was conducted on-chip prior to it entering the detection site. Plasma contains many proteins, some at much higher concentrations than the target analyte (CRP in this case), and these proteins often interfere with the assay. Before the immunoassay continued, therefore, an effort was made to deplete the two most abundant proteins in plasma: human IgG and human serum albumin (HAS). The plasma first entered the human IgG depletion channel, the upper section labeled ‘high abundance depletion channels’ in **Figure 3-10**. Because of IgG’s high affinity to protein A, this first high abundance protein depletion (HAPD) site used the same procedure as the detection site described in section 3.4.1 and shown in **Figure 3-28**.

The next HAPD site targeted HSA, the most abundant protein in human plasma. The channel was first functionalized in the same manner as the detection site

and the first HAPD site. Polyclonal rabbit anti-HSA IgG antibodies (Abcam) were then incubated in the channel for 7 minutes, affixing said proteins to the surface of the channel (see **Figure 3-29**). After rinsing with PBS and aspirating, the channel was ready to accept plasma for albumin depletion.

Figure 3-29 Functionalization scheme for anti-HSA HAPD channel.

3.4.3 Immunoassay

Normally, the detection site was pre-functionalized with the CRP capture antibody, but this step could also be done immediately prior to the test. A mouse monoclonal anti-human CRP antibody (Abcam) in a Protein A-IgG binding buffer (Pierce Biotechnology) at a concentration of 200 $\mu\text{g/mL}$ was incubated in the microchannel for 7 minutes. The channel was rinsed with PBS and filled with storage buffer (if a test was not going to be run immediately).

A sequentially loaded test required that all of the reagents be pre-loaded into the microfluidic chip before beginning the test. **Figure 3-30** shows a V3.0 chip loaded with colored water representing the various reagents, storage buffers, and washing steps. All fluid volumes were inserted into the chip using a 0.5-10 μL pipette, each having a volume of 8 μL , except the sample channel (plasma channel) which held 3 μL . The air buffer between each fluid volume was approximately 5 μL . The HAPD and detection site channels were simply loaded to capacity with storage buffer.

Figure 3-30 V3.0 of the microfluidic chip loaded with colored water representing the various reagents, storage buffers, and washing steps. From top to bottom: wash (blue), PBST wash (yellow), detection antibody (green), wash (blue), sample (red), storage buffer for anti-IgG HAPD channel (blue), storage buffer for anti-HSA HAPD channel (brown).

The first step in the testing process was for the CRP doped plasma sample to enter the first HAPD site. The plasma incubated in this channel for 7 minutes, allowing human IgG to be captured by the protein A. Next, the plasma traveled to the next HAPD site where it incubated in the anti-HSA coated channel for 5 minutes. The sample then traveled to the detection site, functionalized with the anti-CRP antibody. Here the sample incubated for 5 minutes, moved a few microliters down, and incubated another 5 minutes. After a washing step washed away any non-specifically bound proteins, the detection antibody, a goat, polyclonal anti-human CRP antibody tagged with FITC (Abcam), was incubated in the channel for 5 minutes. PBS buffer with 0.5 $\mu\text{L/mL}$ Tween 20 (PBST) was brought in and incubated for 5 minutes to flush out any

non-specifically bound detection antibodies. A final washing step of PBS was used to flush out the PBST and fill the detection site for a signal reading. **Figure 3-31** shows a schematic of the sandwich assay built on the walls of the detection site channel.

Figure 3-31 Schematic of the sandwich assay built on the walls of the detection site channel.

The detection site, where the sandwich assay was built, would fluoresce at a wavelength of approximately 520 nm when exposed to a wavelength of approximately 495 nm, proportional to the amount of CRP in the sample. **Figure 3-32** shows an image of a detection site under a fluorescent microscope.

Figure 3-32 Image taken by a fluorescent microscope of a detection site after assay. Sample was 10 $\mu\text{g/mL}$ CRP in PBS buffer.

3.4.4 Florescent Signal Detection

The emission light from FITC was measured using a spectrometer (Ocean Optics, USB4000). The detection site was first exposed to an excitation light source. Like Li [83], a high power deep blue light emitting diode (LED) (Quadica Developments) was used in conjunction with a 470 nm bandpass filter (Edmund Optics, Barrington, NJ). The filter has a bandwidth of 10 nm, a minimum transmission of 45% with an optical density (OD) of 3. The high OD of the filter blocked light outside of the passband, keeping the background noise of the detector low. The emission light from the FITC in the detection site was channeled through a quad-furcated fiber optic cable to the spectrometer.

Figure 3-33 shows a sample of the recorded signals taken during the course a test. Previously [83], [23], only the signal at 520 nm (the peak wavelength for FITC emission) was used to calculate the signal strength, averaging 5 to 15 readings taken every 200 milliseconds. For this study a slightly different approach was taken, instead using the signal read between 510 and 530 nm. The area between the baseline (red) and the reading line (blue) between 510 and 530 nm was used as the signal strength as opposed to just using the difference between the baseline and reading at 520 nm. Using this range of wavelengths to calculate signal strength more closely mimicked the emissions spectra of FITC and provided a clearer signal, particularly for readings with low signal strength. Raw intensity data was exported from the spectrometer software and the difference between the baseline and reading intensity at each wavelength between 510 and 530 nm was calculated. These differences were summed and divided by 1000 for scaling purposes before they were plotted against CRP

concentration. Immunoassay tests were conducted within a range of 0 to 50 $\mu\text{g/mL}$ of CPR.

Figure 3-33 Sample of the intensity readings taken by the spectrometer during a typical test. **Black** line is the background noise (dark reading). **Red** line is the baseline reading taken with the excitation light on, before the detection antibody is introduced. **Green** line is taken when the detection antibody is in the detection site. **Blue** line is the reading line taken once the assay is complete. The difference between the red and blue lines is the signal strength.

3.5 Fluid Handling

A significant portion of this research involved the control of fluid volumes in the microfluidic chip. The previously discussed pump, working in conjunction with numerous microvalves, both of which were controlled by a fluid position sensor, led to a comprehensive fluid handling system. External valves were chosen over integrated microvalves to reduce the complexity of the microfluidic chip, and a fluid position sensor was adapted from work done by DiFilippo [24] that could recognize when fluid had entered a particular channel.

3.5.1 Fluid Control using External Valves

During the run of one complete test, the pressures generated by the pump were directed and exhausted by a system of 3-way miniature solenoid valves (The Lee Corporation, product # LHLA0531111H) that moved the fluid volumes to the right place at the right time. The valves were switched in such a way as to create 6 different states, each one capable of a unique task. State 1 pushed whole blood through the separation membrane into the chip. State 2 was used to transport the sample (plasma) to the HAPD sites and to the detection site. State 3 was a vent function; when activated, it relieved all internal pressure within the chip instantly stopping all fluid movement. State 4 was used to transport all of the reagents to the detection site. State 5 was used whenever the position of the plunger in the pump needed to be changed without building up any pressure within the pump (*e.g.* during the pump homing process). State 6 was used when a chip was being plugged into the unit; the settings of this state prevented fluid from shifting inside the chip during this process. To accomplish all 6 of these states, 6 valves were used. **Figure 3-34** shows a schematic of the 6 valves and how they were connected to the chip, pump, and each other.

Figure 3-34 Schematic of the 6 valves (V1-6) connected to the chip, pump, and each other. Each valve is a 3-way solenoid valve, so the center port is always either connected to the top or the bottom port.

An air-tight seal connecting the tubing to the chip was accomplished using PEEK tubing. This tubing was installed at the locations shown in **Figure 3-34**, through a platform, and about 1mm proud on the top surface (see **Figure 3-35b**). The OD of the PEEK was slightly larger than the hole punched into the PDMS chip, allowing for a positive seal between the PEEK and the chip. The chip would get ‘plugged’ into the platform using a slight pressure at each port. This process also precisely aligned each chip. The only open port into the chip not plugged into PEEK tubing was the filtrate inlet port (connected to the top of valve 3 in **Figure 3-34**). This port led to the

chamber above the filter where a positive pressure was introduced (see **Figure 3-14**) to push the whole blood through the filter.

Table 3-1 shows the positions of each valve for each state. The valves themselves are 3-way solenoid valves that can be in one of two conditions; ‘up’, which refers to the position of the internal connecting plunger and denotes that the common (middle) port is connected to the top port, and ‘down’ indicating that the common port is connected to the bottom port. When a particular port is connected to the common port, the other port is blocked off.

Table 3-1 Valve positions for the various fluid handling states.

	Description	Valve					
		1	2	3	4	5	6
State 1	Blood through filter	down	up	up	down	down	up
State 2	Transport sample	down	up	down	up	down	down
State 3	Vent (stop fluid flow)	up	up	down	up	up	down
State 4	Transport reagents	up	down	down	up	up	down
State 5	Home pump	down	up	up	down	up	down
State 6	Plug in chip	up	down	down	down	up	up

To reduce the dead volume created by the connecting tubing as much as possible, small inner diameter Tygon tubing was used for lengths longer than a few centimeters. The tubing had an OD of 1.50 mm and an ID of 0.50 mm. To further reduce the volume of the tubing, a wire with a diameter of 0.36 mm was inserted into the lengths of tubing. This reduced the internal volume of the tubing from ~2 $\mu\text{L}/10$ mm length to ~0.9 $\mu\text{L}/10$ mm length (see **Figure 3-35b**).

This configuration of valves, working together with the previously described pump, was able to reliably transport the various fluid volumes to their correct locations at defined times. The vent function (State 3) in particular was central to

fluidic control because it produced in an immediate stoppage of all the volumes. This proved valuable, as it meant that larger than necessary pumping volumes could be programmed, allowing for any variation in channel resistance, while still retaining total control over where the fluid would stop.

The valves required a 5 volt pulse lasting no longer than 10 ms to one of its two terminals. A pulse to terminal 1 (T_1) put the valve in its 'down' position, and a pulse to T_2 put the valve in its 'up' position. A quadruple half-H driver IC (Texas Instruments, product # SN754410) was used to send the 5 V pulse and an Arduino MEGA microcontroller was used to send switching signals to the IC. Each valve required 2 drivers (T_1 , T_2), thus each IC could be used to control 2 valves. A total of 3 ICs were used to drive all 6 valves. Accordingly, all 12 drivers (6 valves x 2 terminals) required their own input from the microcontroller. Powering the valves was a 12V 2.5A AC to DC adaptor run through a 5V voltage reducer (Texas Instruments, product # UA7805C). Bench-top testing of the valving system was conducted using a facsimile of the portable device platform (Appendix 6.3, Drawing #10 and #11). This platform, the array of valves, the Arduino microcontroller, and the 3 half-H ICs are shown in **Figure 3-35**. Not shown, is the lid (Appendix 6.3, Drawing #12) used during biological tests that fit over the top of the platform and blocked out all outside light.

Figure 3-35(a) Photograph of the bench-top setup used to test the functionality of the fluid handling using 6 valves. **(b)** Photograph of the same setup from a different angle showing the PEEK tubing installed through the platform to connect into the chip.

In order to easily control the valves during experimentation, a graphical user interface (GUI) was created in Visual Basic that could both control individual valves and activate the four different valve states. Serial communication was first established between the Visual Basic program and the Arduino MEGA over a USB connection. To activate a state or particular valve, the user would click on any one of the buttons in the GUI. This action would prompt the transmission of a particular character (*e.g.* ‘a’), depending on the button pressed, to the Arduino microcontroller. The Arduino would then read this character and activate the appropriate valves by sending switching signals to the driver ICs. The source code for both the Visual Basic GUI and the companion Arduino microcontroller are given in Appendix 6.1 and 6.2 respectively. **Figure 3-36** shows a screen capture of the created GUI.

Figure 3-36 Screen capture of the GUI created in Visual Basic to control the valves for the bench-top experimental setup. Notice the valves can be controlled individually or as States.

3.5.2 Fluid Position Sensor

The logistics for the fluid position sensor can best be described using an analogy to how the fluid volumes were positioned previously by hand. Prior to the fluid position sensor, a user would use their eyes and look at the position of a particular fluid in the microfluidic chip, their brain would tell them when that fluid had reached the correct location, and their hand would flip a switch activating the vent function, stopping the fluid. The fluid position sensor performs the same tasks, replacing the user's eyes and brain with a CCD camera and a MATLAB program, respectively.

The charge-coupled device (CCD) camera (UniBrain, Fire-i, product # 2062) was used in conjunction with a micro video lens (Edmund Optics, product # 85-207) to image the chip and its contents. The lens was selected because of its ability to adequately magnify the chip, but also its ability to capture all of the trigger channels at once. These channels include the plasma channel, the two HAPD sites, and the detection site. These channels represent all the locations in the chip where liquid must

be stopped during the course of a test. A distance (chip to lens) of 216.4 mm was found to be optimal for imaging all trigger channels. **Figure 3-37** illustrates how much of the chip was actually captured with this setup and also highlights the 4 trigger channels.

Figure 3-37 (Left) Drawing of V4.1 of the chip in the orientation that the CCD camera sees it in; dotted box indicates how much of the chip the camera captures. (Right) Image taken by the CCD camera. Red boxes mark where the trigger channels are located; from top to bottom they are, detection site 1, detection site 2, HAPD site 2, HAPD site 1, sample (plasma) channel. Washed-out circle centered on the detection site is from the optics underneath the chip used to direct the excitation light.

As seen in **Figure 3-37 Right** The detection site trigger channel was split into two sections (the smaller rectangles immediately above and below the white circle) in order to avoid the washed-out circle created as a result of the optics located directly below the chip centered on the detection site.

A custom MATLAB program developed by DiFilippo [24] was fed images from the CCD over a FireWire cable at approximately 30 frames per second (fps). The program analyzed the images and determined whether fluid or air occupied a channel. Because the refractive index of PDMS (1.40) is much closer to water (1.33) than it is

to air (1.00), the visual difference between a channel that is occupied by air or liquid is quite stark, as seen in **Figure 3-38**.

Figure 3-38 (Left) CCD image of microchannels from V4.0 of the chip completely filled with air. **(Right)** Image of the same channels with the bottom 10 channels containing water.

This contrast allowed the program, with correct lighting, to easily determine if a channel contained liquid or air. The program worked by comparing a control image, taken before any liquid was present, to images taken at regular time intervals during the pumping process. Each image taken was subtracted from the control image and every element (pixel) of this new difference image was set to a value of 0 (black) or 255 (white). An element was set to black if its value was below a threshold value (normally set between 10 and 16) and set to white if it was above this threshold value. A black element indicated no change between the control image and the n^{th} image, while a white element indicated the opposite. The inclusion of a threshold value in the program prevented (electrical) noise in the CCD from turning any of the elements white. This threshold value must be somewhat honed because if it is set too high, the program may not recognize a change from air to water, but if it is set too low, the noise of the CCD may indicate a change where there is none.

The program also defines a reign of interest (ROI) for each trigger channel; these ROIs are outlined in red in **Figure 3-37**. When the program is checking for a change in a channel, it is actually only looking for a change in the channel's ROI. Once a percentage (typically 50 to 100%) of the elements (pixels) in this ROI have changed from black to white (air to water) or white to black (water to air) the program triggers an action. Depending on the ROI and the tests progress, this action may be a State 3 activation (vent), or a continuation of the current pumping cycle. **Figure 3-39** shows 6 images taken in chronological order during a test of the fluid detection program.

Figure 3-39 Images generated by the fluid detection program during a pumping cycle taken in chronological order. (a) ROI can be seen as a pixelated rectangle, this is where the program will look for a change in the percentage of white to black pixels indicating a change from air to liquid. (b) Liquid has started to fill the ROI channel changing it from black to white until eventually (c) the ROI has changed enough to trigger an action, in this case, to look at the (d) next ROI for a change. (e) The ROI starts to fill with liquid until (f) it has filled enough to trigger an action.

3.5.3 Bench-Top Experimental Setup for Fluid Handling

A setup was designed to test all aspects of the described fluid handling system including the CCD camera, lens, and fluid detection program, as well as determine the correct lens-to-chip distance and placement of the white LED used to illuminate the chip for imaging. **Figure 3-40** shows a photograph of this setup. This setup also incorporates the previously described setup for controlling the six valves.

Figure 3-40 Experimental setup used to test the fluid position sensor and fluid handling system as a whole.

The frame used to position the CCD and its control board over the chip was made from 1" aluminum T-slotted framing (McMaster-Carr, item # 47065T101), and was bolted to a 1/4" thick piece of aluminum, creating a stable base. The 'CCD holder'

(Appendix 6.3, Drawing #13) was designed to accept the remote CCD board and fit into the T-channel of the aluminum framing. Similarly, the ‘control board holder’ (Appendix 6.3, Drawing #14) was designed to accept the control board and fit into the T-channel of the aluminum framing. Both parts were custom designed and printed in-house using a rapid prototyper. This setup allowed for a wide range of adjustment; but more importantly it could be securely fixed, ensuring positional consistency from test to test. The lens-to-chip distance was determined to be 216.4 mm. This height produced the most magnified image of the chip while still capturing all of the trigger channels. At this point the lens could be permanently focused for that working distance. To do this, an aluminum spacer was cut (Appendix 6.3, Drawing #15) to a thickness of 12.6 mm that sandwiched between the head of the lens and the CCD lens mount (see **Figure 3-40**). An important aspect of capturing images with a good liquid/air contrast was the placement of the light source LED. Previous studies [24], [25] used a cool white LED made by VCC (part # 57P7152) for this task, but the placement of the LED, relative to the chip, would obviously be different for this system. A custom LED holder (Appendix 6.3, Drawings #16-19) was designed that facilitated rapid and secure placement of the LED (see **Figure 3-40**). Optimal placement of the LED was found by monitoring the live video feed while adjusting its x, y, z, and pitch coordinates. Additional contrast between channels filled with liquid and those filled with air was obtained by placing a non-reflective black surface behind the chip. Vinyl from the skin of a typical 3-ring binder was used for this background, as can be seen in **Figure 3-40**. The final position of the light source LED is

documented in **Figure 3-41a** where **Figure 3-41b** is the resulting image captured by the CCD.

Figure 3-41(a) Position of the light source LED and lens relative to the microfluidic chip. The detection site, center of the lens, and the center of the LED all lie on the same plane. **(b)** Image captured by the CCD in a dark room apart from the LED.

3.6 Portable Unit

The portable unit was in many respects a culmination of the work described in previous sections. Actualization of the unit required significant 3D modeling beforehand, in an effort to make the unit as compact as possible while still containing all of the necessary components. Apart from the physical form of the device and configuration of its internal components, the Arduino microcontroller programing and MATLAB fluid position sensor functions were modified to work together. The particulars of the software will be discussed after the hardware.

3.6.1 Device Shell and Internal Configuration

A determination was first made as to the limiting factor in miniaturization. With the pieces that needed to be integrated, how small could the unit actually be made? Initially, the distance needed between the imaging lens and the chip (216.4

mm) for the fluid position sensor seemed to be the largest required dimension. This would have made the unit quite large (at least 10 in tall) and awkward. Fortunately, it was realized that this working distance could be cumulative, with the image bouncing off a mirror or mirrors totaling 216.4 mm. With the lens-to-chip distance no longer an issue, the next largest piece of hardware was the pump. The required room for the pump to fully extend in both directions was ~ 215 mm, so this became the jumping off point for the internal volume of the unit, having an internal length of about 215 mm. Once the pump's orientation was determined, the other large components were placed in a very rough approximation. **Figure 3-42** shows an isometric view of the placement of most of the larger components.

Figure 3-42 Isometric view of the placement of the larger components held within the unit. This rough placement helped determine the required size of the unit.

The excitation light source in **Figure 3-42** was an assembly of several components stacked one on top of each other. Listed from bottom to top, the assembly was made up of an aluminum heat sink (Fisher Elektronik GmbH, Germany), a high power 470 nm LED (Luxeon V Star blue hp-LED, Quadica Developments), a collimator (Future

Electronics, QC, Canada), a plano-convex lens (20 mm diameter with an effective focal length of 38 mm, IDEX Corp., USA) to further focus the light, and finally a 470 nm bandpass filter.

The next step was to choose a spot for the CCD and to orientate a network of mirrors so the chip could be imaged by the fluid position sensor. A 3D CAD drawing was used to accurately model all of the components of the imaging system (CCD and lens, 2 mirrors, and the chip) where they were positioned and angled to recreate the same spacing as the bench-top setup. **Figure 3-43** shows the relative positioning of the lens, the 2 mirrors, and the chip.

Figure 3-43 Screen capture of the CAD model used to determine the position and angle of the mirrors.

This exact configuration of mirrors created a working distance and image orientation exactly the same as the bench-top setup. The mirrors were positioned in the identical way in the CAD mock-up, and fixtures were created around the mirrors. **Figure 3-44** illustrates how this was done.

Figure 3-44(a) Mock-up CAD drawing with mirror 1, mirror 2, the CCD and lens, and chip platform positioned exactly the same as in Figure 3-42. **(b)** Holder for mirror 1 was designed around the position of the mirror. **(c)** Similarly, after the lid was designed, a holder for mirror 2 was added to position it correctly.

Mirror 1 holder (Appendix 6.3, Drawing #20) was designed to bolt into the floor of the device using the four tabs around its base, and mirror holder 2 was actually the lid of the device. The chip platform (Appendix 6.3, Drawing #21), first shown in **Figure 3-44a**, had many important duties including, holding the CCD in place, holding 3 valves, positioning the excitation light source assembly, and positioning the chip.

Once these components were set in place the main shell of the device (Appendix 6.3, Drawing #22) was designed around the components already in place. The shell incorporated numerous features designed to align and accept other parts. For example, the floor of the shell had tabs that positioned the mirror 1 holder, the LED current driver, the pump bracket, and the excitation light source assembly. It also had cutouts for valve 4, the USB port for the microcontroller, the FireWire port on the CCD control board, the main power switch, and DC power jack. Apart from the cutout for the CCD control board, it also had mounting posts for the board. Another obvious feature of the shell are the 11 tabs standing proud of the top surface. These tabs were

added as points where the top of the shell could be fastened. It was decided that because the entire shell was going to be rapid prototyped, its thickness should be relatively thick compared to a shell that was injection molded. A thickness of 5 mm was chosen based on past experience with the capabilities of this particular rapid prototyper (Stratasys, Dimension SST) and a desire for a fairly sturdy structure.

Figure 3-45 shows the main shell in a transparent state, it also shows where valves 1, 4, and 5 were placed as well as several platforms where breadboards would be mounted in the device (Appendix 6.3, Drawings #23 and #24).

Figure 3-45 Isometric view of the device body in a transparent state, also showing the placement of three of the valves and the two breadboard holders.

The pump was fastened in place with a bracket (Appendix 6.3, Drawing #25) that wrapped around the body of the pump and bolted through the floor of the device shell. This bracket also acted as a mounting point for the Arduino microcontroller (see **Figure 3-46**).

Figure 3-46 Pump mounting bracket viewed through the side of the device shell.

The next component to find a home was the driver shield for the stepper motor (Adafruit, product # 1438). The shield was mounted vertically using a bracket (Appendix 6.3, Drawing #26) fastened next to the small breadboard holder (see **Figure 3-47**). Its odd shape accommodated the mounting holes in the shield and also provided standoff points from the wall and floor of the shell that correctly positioned the bracket.

Figure 3-47(a) Placement of the stepper driver shield mounting bracket on the shell of the device. Larger holes in the center of the bracket accept socket head bolts, fastening it to the shell, smaller holes indicate shield mounting holes. **(b)** Driver shield installed onto the bracket.

The chip platform, introduced earlier, contained many features and did more than just hold the microfluidic chip in place. **Figure 3-48** shows the chip platform from underneath and highlights its other functions. In an effort to reduce the dead

volume created by long runs of tubing, the three valves connected to the ports at the top of the chip (valves 2, 3, and 6, see **Figure 3-34**) were mounted under the chip platform to bring them as close as possible to the chip. Valve 2 was held in place by an appendage (Appendix 6.3, Drawing #27) bolted to the chip platform. Also visible is the ring that positioned the excitation light assembly. The hole through the platform in this ring centers on the detection site of the microfluidic chip. This hole is actually filled in with a cylindrical piece of PMMA (Poly(methyl methacrylate)) plastic wrapped in aluminum tape. This cylinder acted as a light tube, channeling the excitation light directly to the detection site. The chip platform also had a structure, similar to the one used for the bench-top setup, that the remote CCD was fastened to. The circuit board of the CCD fit under two tabs at one end and was bolted through two mounting holes at its other end (not visible in **Figure 3-48**). Because the CCD was fixed to the chip platform, and precise placement of the CCD was crucial to image alignment, the chip platform was securely fastened to the device shell with 5 socket head cap screws, ensuring exact placement every time; these mounting holes can be seen around the perimeter of the chip platform. Lastly, two tubular features at the back of the platform each housed a ¼” cylindrical neodymium magnet (McMaster-Carr, item # 58605K75) for the mag-lid (magnetic lid).

Figure 3-48 The Chip platform, three valves, and the CCD/lens viewed from underneath.

The next major piece to be designed was the top section of the shell (Appendix 6.3, Drawing #28) and the lid (Appendix 6.3, Drawing #29). The top of the device shell was designed to be tall enough to accommodate all of the contained components and wiring, it was lower at the back and gradually sloped up to the same level as the lid. The lid (the piece that covers the chip platform) had a required minimum height in order to hold mirror 2 in its correct place (centroid of the mirror is 30 mm above the chip). As seen in **Figure 3-49**, the entire top portion of the unit is made up of three pieces; the lid, the top of the shell, and the double hinge piece (Appendix 6.3, Drawing #30). A double hinging action was used to allow the lid to open almost a full 180°. Small square magnets (McMaster-Carr, item # 5848K41) (0.125”) were installed in the top shell and double hinge piece to latch the two pieces into position when closed. Looking at **Figure 3-49b**, steel pins fastened to the lid can be seen, these pieces were attracted to the magnets in the chip platform and pulled the lid closed. The steel

cylinders were cut from 1/4" dowel pins and attached to the lid using the shown clamps (Appendix 6.3, Drawing #31).

Figure 3-49 (a) The three pieces that make up the top portion of the device viewed from above. (b) The same three pieces viewed from underneath.

In order for a positive pressure to be introduced above the filter (see **Figure 3-14**) an attachment to the lid was designed that would interface with the PDMS filter holder on the chip. The attachment was designed to place pressure on the filter holder using two small springs, sealing the PDMS surface on the attachment to it. **Figure 3-50** shows the filter interface lid attachment. The part contains 3 custom pieces; the interface base (Appendix 6.3, Drawing #32), the interface slide (Appendix 6.3, Drawing #33), and the PDMS anchor (Appendix 6.3, Drawing #34). Additionally, the assembly contains two springs appropriated from a pair of ballpoint pens, a 4-40 socket head cap screw, and PDMS. The interface slide piece can travel up and down as much as the head of the 4-40 screw will allow. Its travel is such that the bottom PDMS piece places a small pressure on the filter holder piece on the microfluidic chip, sealing the two pieces together. Then, through the hole in the PDMS piece, a pressure can be introduced, pushing the sample through the filter and into the chip.

Figure 3-50 CAD rendering of the filter interface lid attachment. (a) The assembly with the interface base opaque. (b) The assembly with the interface base made transparent.

The PDMS bottom was cured onto the anchor by suspending the anchor in uncured PDMS at height of 1.50 mm above the bottom of a small petri dish. The PDMS was then cured and the PDMS was cut to size. The anchors were suspended two at a time by a rapid prototyped suspension piece (Appendix 6.3, Drawing #35), the operation of which is shown in **Figure 3-51**.

Figure 3-51 Anchor suspension piece in use, the PDMS anchors are held parallel to, and 1.50 mm above the bottom of the petri dish. PDMS was poured into the dish to an appropriate level and placed on a hot plate to cure.

Figure 3-52 shows the filter interface assembly installed on the lid of the device. In this CAD rendering the lid is closed, slightly compressing the springs, sealing the two pieces of PDMS against each other. Also visible in the figure are the two fiber optic cable guides (Appendix 6.3, Drawing #36) at each opening of the lid. These guides kept the cables in place during the course of a test.

Figure 3-52 CAD rendering of the filter interface lid attachment installed into the device. The lid is set to a transparent state. Also shown are the fiber optic cable guides installed at each opening of the lid.

Positioning of the illumination LED for the fluid position sensor was done using a small LED holder (Appendix 6.3, Drawing #37) bolted to the inner structure of the device shell. Its shape positioned the LED in the same optimal spot as in the bench-top setup (see **Figure 3-41a**). **Figure 3-53** shows how and where the illumination LED was mounted inside the device.

Figure 3-53 CAD rendering of the illumination LED and the holder used to position it at the optimal spot for imaging.

The last larger pieces to be positioned in the device were the two breadboards required for all of the wiring of the valves, microcontroller, stepper, stepper driver board, linear actuator position sensor, and the excitation and illumination LEDs. The larger breadboard had 37 columns and 10 rows (5 and 5) and was mounted on the ‘large breadboard holder’, while the smaller breadboard had 17 columns and 10 rows (5 and 5) and was mounted on the ‘small breadboard holder’ (see **Figure 3-43**).

Figure 3-54 shows the two breadboards mounted inside the device.

Figure 3-54 CAD rendering of the two breadboards mounted on their respective holders in the device.

Ancillary features were added to the shell to accommodate a DC power jack, rocker switch, and connectivity to components inside the device. One of these features can be seen in **Figure 3-54**, where a square hole in the back of the shell allows a USB cable to connect to the Arduino microcontroller. **Figure 3-55** highlights the other small features of the shell.

Figure 3-55 All images are CAD renderings. (a) Shows a square hole that allows a FireWire (IEEE 1394) cable to be plugged into the CCD control board, (b) has the shell set to transparent. (c) Feature that matches the shape of the DC jack shown in (d) installed. (e) Feature that accepts the snap-fit rocker switch used to control the input DC power, (f) shows the switch installed.

The shell, along with a majority of the other custom designed parts, were rapid prototyped and assembled. A CAD rendering with the shell, top, and lid set to a

transparent state is shown in **Figure 3-56a**, and a photograph of the assembled and painted device with its lid open is shown in **Figure 3-56b**.

Figure 3-56(a) CAD rendering of the device and its internal components. **(b)** Photograph of the assembled and painted device with its lid open.

3.6.2 Power Supply and Electrical Wiring

A 12V 2.5A, AC to DC transformer (wall wart) was used to power the components inside the device. The power was brought in by way of the DC power jack located on the back of the device; from there power was routed through the main-power rocker switch. If the main-power switch was set to the ‘on’ position, power was routed to the small breadboard where it was routed to various components.

The Arduino MEGA (2560 Rev. 3) required a 12V supply to operate. Digital pins 20 and 21 (SDA, and SCL) were connected to the Adafruit (V2, product ID: 1438) stepper motor shield’s corresponding pins; these pins transferred stepping commands from the Arduino to the Adafruit. The rest of the connections into the Arduino were to digital out pins, these triggered the drivers for the 6 valves and the excitation LED.

The Adafruit motor shield was supplied 5V from a (5V) voltage regulator (Texas Instruments, UA7805C) for its logic circuits and was connected to a common ground. The Power for the stepper motor was connected to the ‘Motor Power’ terminals from a 10V voltage regulator (Texas Instruments, UA7810C). The stepper was connected to the second motor block (M3 &M4). Going from the top of M3 to the bottom of M4 the order of wires from the stepper connected to the shield were; Black, Brown, Orange, Yellow.

Because of its high current draw, power to the excitation LED was sent through an additional quadruple half-h driver (Texas Instruments, SN754410) located on the small breadboard. The position sensor (linear potentiometer) in the pump was placed in series with a 2000 Ω resistor creating a voltage divider referenced to ground.

Doing so allowed the Arduino's analog-in pin (A1), connected between the 2000 Ω resistor and the position sensor, to read the voltage change that resulted from any positional change in the pump. With a circuit potential of 5 volts, the resulting voltage at the analog in pin on the Arduino ranged from 2.18V (with the linear actuator in the pump fully extended), to 0.56V (with the linear actuator fully retracted). The Arduino's analog-in pins (15 total on the MEGA) map input voltages between 0 and 5 volts into integer values between 0 and 1023. This meant that for the voltages of 2.18 and 0.56, the respective integer values given by the Arduino were between 445 and 119. In order to accurately define the pumps position, a curve fit was defined over 130 readings taken between the pumps two extreme positions. The equation for the resulting curve fit was used to calculate the position of the pump based on the analog-in sensor value (445 to 119). **Figure 3-57** shows the 130 data points with a second order polynomial curve fitted to the data.

Figure 3-57 Arduino analog-in readings as a function of the position of the plunger in the pump. The position of the plunger is tied to the position of the linear actuator, at Position=0 the linear actuator is fully in, and at Position=129 the linear actuator is fully out.

The equation for the trend line was used in the code that positioned the pump's plunger. As a result of taking 130 data points to determine the relationship between sensor value and position, the position of the pump was always defined as a value between 0 and 129. Equation 2 finds that positional value of the pump as a function of the read sensor value (x).

$$Position = 0.0008x^2 - 0.0994x + 4.4082 \quad [2]$$

The wiring for the device is shown in **Figure 3-58**. Additionally, a schematic style wiring diagram is given in Appendix 6.4.

Figure 3-58 Wiring diagram of the components used in the portable device.

The use of prototyping breadboards allowed for any subsequent changes to the wiring to be made quickly, and made it easy to trouble shoot any issues with the electronics. The obvious downside to using breadboards is the necessity for a large number of jumper wires and the large volume they consume. The top section of the

shell was designed to accommodate the large nest of wires needed to rout power and triggering lines. **Figure 3-59** shows a top down photograph of the portable device completely wired up and plumbed. Note the 6 wires heading under the chip platform to be connected to the 3 valves mounted at this location.

Figure 3-59 Photograph of the portable device completely wired and plumbed. The unit is being viewed top down.

The use of a custom PCB (printed circuit board) in place of these breadboards could seriously reduce the internal clutter and inherent unreliability of this prototype. Still, for the purposes of a proof of concept, the components used preformed adequately.

3.6.3 Programing and User Interface

The portable unit used two control systems, the Arduino MEGA located inside the device and a MATLAB function that ran on a separate personal computer (PC). The Arduino was connected to the PC through serial communication over a connected

USB cable, and commands were sent between MATLAB and the Arduino during the course of a test. Each control system had their own duties, but ultimately, the MATLAB function held a superior position because it controlled when the Arduino would carry out its duties. For example, a valve configuration would be triggered when the MATLAB function sent a particular character over serial communication to the Arduino, and once received, the Arduino would run a particular subroutine activating the desired valve configuration depending on the character received. The source code used in the device for both Arduino and MATLAB are given in Appendix 6.5 and 6.6, respectively. The following is simply a condensed breakdown of how the two programs worked independently and in concert to make a functioning device.

The Arduino microcontroller had the responsibility of sending the triggering signals to the numerous drivers; changing a valve configuration or turning on and off the excitation LED. In the case of a valve configuration, the Arduino would receive a command from MATLAB to switch to a particular State, the Arduino would then run a function to complete this task. The code below, sampled from the Arduino, shows how the user defined ‘State_One()’ function switched the valves to their correct positions.

```

/***** State One *****/
void State_One(){
    digitalWrite(22, HIGH);    //V1 Down
    delay(timer);
    digitalWrite(22, LOW);
    delay(2);

    digitalWrite(24, HIGH);    //V2 Up
    delay(timer);
    digitalWrite(24, LOW);
    delay(2);

    digitalWrite(51, HIGH);    //V3 Up
    delay(timer);
    digitalWrite(51, LOW);
    delay(2);

    digitalWrite(46, HIGH);    //V4 Up
    delay(timer);

```



```

digitalWrite(46, LOW);
delay(2);

digitalWrite(52, HIGH);    //V5 Up
delay(timer);
digitalWrite(52, LOW);
delay(2);

digitalWrite(49, HIGH);    //V6 Up
delay(timer);
digitalWrite(49, LOW);
delay(2);}
/*****/

```

The variable ‘timer’ was set to 10 ms, defined as the switching pulse for the valves and ‘delay(2)’ was a 2 ms pause added after each switch for stability. Once created, this function provided a way to easily switch to State 1 by simply writing ‘State_One();’ anywhere in the Arduino code. A multitude of user defined functions were used in the Arduino code, all of which are defined in the table below.

Table 3-2 Functions created in Arduino, and their purpose.

Function Name	Purpose
State_One()	Similar to the State_Two and State_Four functions, it configured the 6 valves correctly.
Vent()	Configured the valves to vent all pressure in the chip.
Wake_Up()	Cycled through all 6 valves, switching each Up then Down.
Initialize_Valve_Pins()	Required code in the Arduino coding environment that defined what each used pin would be used for (output in this case) and set every used pin to ‘LOW’.
Spectrometer_Reading_1()	Similar to Spectrometer_Reading_2(), this function allows for a user input before turning on the excitation light for 3 seconds.
s_For_Ready()	Gives instructions to turn on main power and waits for a user input before continuing with the code.
y_For_Chip()	Gives instructions to insert the chip and waits for a user input before continuing with the code.
Filter_Question()	Finds out if that particular test is using a chip with a blood separation membrane.
Start_Test()	Confirms that the user is ready to commence the test before beginning.
First_Step()	Upon receiving the go-ahead from MATLAB this function executes the first step in the test; pumping the sample to the first HAPD site.
Home_Pump()	Conditional loop that runs until the pump is ‘homed’ in its center position.

The pump homing function worked by taking a reading off of the positional sensor and interpolating a position using the coefficients found earlier. If for instance the pump is out too far the function will determine how many steps the stepper should run in order to bring it to its home position. Once it has moved that distance, the function will start again and run over and over until it is within a defined range of the home position. Below is a portion of the ‘Home_Pump()’ function including the ‘while’ statement and the ‘if’ statement executed if the pump was determined to be too far out.

```

/***** Home Pump Function *****/
void Home_Pump() {
  while(LA_Value <= 330 || LA_Value >= 340){
    if (LA_Value >= 340){ // LA is OUT too far
      Serial.print("Pump is too far OUT, with LA value of ");
      Serial.println(LA_Value);
      Position = abs(((65-((coeff_1*(pow(LA_Value,2))-(
      (coeff_2*LA_Value)+coeff_3))*150))); //steps away from center position
      counter = Position + 1;
      Serial.print("Number of steps ");
      Serial.println(counter);
      delay(10);
      myMotor->step(counter, BACKWARD, SINGLE); //steps 'counter' times
      myMotor->release();
      delay(1000);
      LA_Value = analogRead(LA_Position);
      last_pump = 'b'; }
  }
}

```

The ‘while’ statement in the third line of above code defines home position as being any linear actuator sensor reading between 330 and 340. Coefficients 1-3 (coeff_1-3) were set as the coefficients in Equation 2.

After the First_Step() function ran, the remaining Arduino code entered a large loop where it would constantly be looking for key characters being sent from the MATLAB program. The 7 letters (a – g) each triggered a different block of code or ‘case’ to be run. Below is the code run for ‘case a’ from this loop.

```

void loop() {
  if (Serial.available() > 0) { //If a serial com is received, set the-
    action = Serial.read(); //variable 'action'=whatever was received.
    switch (action){ //If action='a' switch will go into case 'a'
      case 'a': //a=signal from MATLAB that plasma is in first HAPD site.

```

```

Vent(); //Stop fluid movement by venting all pressure out of chip.
digitalWrite(light_LED,LOW); //Imaging LED is turned off.
while (timer_over != 'h'){ //MATLAB will send 'h' once-
    //Inc Timer" function has finished.
    if (Serial.available() > 0) { //Checking buffer for signal
        timer_over = Serial.read();}
    delay(10);}

digitalWrite(light_LED,HIGH); //Imaging LED is turned back on.
State_Two(); //continue pumping sample to second HAPD site.
myMotor->step(279, BACKWARD, SINGLE); //Steps needed to pump sample
myMotor->release(); //to second HAPD site.
break; //break exits out of case a, starting the main loop again.

```

If the letter ‘a’ was sent from MATLAB to the Arduino, ‘case a’ would be activated and run through until it reached the ‘break’ command. This command would bring the code out of ‘case a’ and back up to the main loop where it would again be looking for any of the 7 letters. Clearly, the Arduino code was pivotal in the overall function of the device, but it rarely acted without a command from the governing MATLAB code.

MATLAB provided a platform for a user interface, controlled the actions of the Arduino microcontroller, and acquired and analyzed images of the channels taken with the CCD for fluid detection. The user interface consisted of text messages being displayed, and commands being input in the MATLAB Command Window during the course of a test. For example, once serial communication had been established between the Arduino and MATLAB, the MATLAB function waited for a message sent by the Arduino ‘s_For_Ready()’ function over serial com. Once received, MATLAB printed out this message to the Command Window and allowed the user to enter the letter ‘s’ to start the test. If ‘s’ was entered, that character would be sent over serial com to the Arduino, allowing the ‘s_For_Ready()’ function to conclude. What follows is the portion of MATLAB code described above.

```

while (s.BytesAvailable == 0); % Will repeat in this while loop until some data
    pause(0.01)                % is read the serial buffer.
End                             % Pauses 0.01 seconds before it checks again.

while (s.BytesAvailable > 0); % Prints the message sent from the
    fprintf(fscanf(s));       % Arduino's "s_For_Ready()" function.
    pause(0.01);
end

```

```

while (strcmp(z,'s') == 0 && strcmp(z,'end') == 0) % Waits for user to
    z=input('','s'); % enter 's' or 'end'
end

if strcmp(z,'s') == 1 % if user entered 's', send 's' char to
    fprintf(s,z); % Arduino over serial.
elseif strcmp(z,'end') == 1 % If 'end' was entered, break will
    break % exit out of while loop and end com.
end

```

There are several communications between MATLAB and the Arduino similar to the one above that gets the Arduino program through all of its setup functions and ready to start its main loop. The next to last Arduino function before it enters its main loop, 'Start_Test()', prints the following message to the MATLAB Command Window:

```

-----
                Start Test?
            Enter 'y' to confirm
-----

```

And once the user enters the character 'y', the following message is printed:

```

-----
                Test Has Commenced
-----

```

Thus triggering the next function, 'First_Step()', on the Arduino microcontroller. This function contains the first pumping cycle, taking the sample fluid to the first HAPD site. But before the pumping command is executed in the 'First_Step()' function, the CCD needs to be initialized and allowed to settle. Below is the section of the MATLAB code that initializes the CCD, as well as several variables used in the fluid detection functions, and immediately following that is a signal (the character 'y') sent over serial com to the Arduino allowing the 'First_Step()' function to continue.

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%---Initialize camera and associated variables---%
obj = videoinput('winvideo',1,'Y800_640x480');% Connect to the CCD
preview(obj)
disp('10 seconds to settle camera')
pause(10) % allow camera to settle for 10 sec
threshold = 15;
background = 1;
obj = 1;

```

```

background = getsnapshot(obj); %take a background image
imwrite(background, 'images/background.jpg', 'jpg');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

fprintf(s, 'y');           % Signal to Arduino to continue with the
                           % 'First_Step()' Function.

```

The completion of the 'First_Step()' function marks the first time the pump is activated during the course of a test and the beginning of the fluid detection functionality of the device. Immediately following the activation of the pump, the MATLAB function 'channelFlow' (Appendix 6.6.2) begins to run. This function determines if there is a change in state of a defined channel. The section of MATLAB code below shows this function being activated in order to determine when the sample fluid has entered the first HAPD site.

```

% State_Two if filterquestion = 2 (no filter)

disp('Looking for fluid coming into HAPD_1 site')
channelFlow(121,394,400,9,2,threshold,background,obj)%First 4# define area.
% Exits out of channelFlow once fluid has been detected in the defined area

fprintf(s, 'a');
    % Once channelFlow is done, fluid is in first HAPD site. Send 'a' to
    % Arduino to confirm that plasma is in first HAPD site. 'case 'a'' in
    % Arduino code is triggered.

Inc_Timer(300, 'HAPD1') % Function that displays a countdown timer in the
                        % command window in a min:sec format.
                        % 300 seconds = Incubation time of 5 min.

fprintf(s, 'h');           % Signal to Arduino to continue with case 'a'

```

Looking at the previously introduced corresponding block of Arduino code (block starts with the line 'void loop() {'), the cooperation between the two programs is evident. In the above block of code, the function 'Inc_Timer' (Appendix 6.6.5) took the two inputs (time in seconds, reason for incubation) and displayed a countdown timer in the Command Window in a min:sec format for the user. Upon completion of the timer, MATLAB sent the character 'h' to the Arduino allowing it to continue with

‘case a’. Similar sections of code were written for all of the pumping cycles to be completed.

Once the sample had incubated in each HAPD site, it then needed to be transported to, stopped at, and incubated at the detection site. To do this, the MATLAB program switches from using the ‘channelFlow’ function to using the ‘detectionFlow’ (Appendix 6.6.6) function. At their core, these two functions are basically the same, except that the region of interest (where the program is looking for a change) is split into two sections instead of just one (see **Figure 3-37**). From this point on the program will only be looking at the detection site for a change. Because of this, the program needs to know when the previous fluid has left the detection site before it starts looking for the next fluid to enter. This is accomplished by allowing the detectionFlow function to be able to see both a change from air to liquid (fluid entering the detection site), and a change from liquid to air (liquid leaving the detection site). This was accomplished by adding an additional input to the detectionFlow function that defined whether it was to look for a change from air to liquid or liquid to air. Based on this input, the function would either look for a change from a lighter to a darker channel (air to liquid) or a change from a darker to a lighter channel (liquid to air). Accordingly, after every use of the detectionFlow function where liquid is brought into the detection site, the channelFlow function is run until the detection site is again filled with air. Below is a section of MATLAB code that illustrates this point.

```
disp('Look for det_ab to enter the detection site')
detectionFlow(330,292,250,9,60,12,250,9,6,threshold,background,obj,1)
% Looking for det_ab to enter the detection site.
fprintf(s,'e');
% det_ab is in detection site. Send 'e' to Arduino to trigger 'case 'e''
Inc_Timer(300,'det_ab') % Incubation time of 5 min.
```

```

fprintf(s, 'k');           % Signal to Arduino to continue with case 'e'
detectionFlow(330,292,250,9,60,12,250,9,6,threshold,background,obj,0)
    % Will trigger when air has filled the detection site (det_ab is out)

disp('Look for PBST to enter the detection site')
detectionFlow(330,292,250,9,60,12,250,9,7,threshold,background,obj,1)
    % Look for PBST to enter the detection site
fprintf(s, 'f');
    % PBST is in detection site. Send 'f' to Arduino to trigger 'case 'f''

Inc_Timer(10, 'PBST')     % Incubation time of 4 min.

fprintf(s, 'l');           % Signal to Arduino to continue with case 'f'
detectionFlow(330,292,250,9,60,12,250,9,7,threshold,background,obj,0)
    % Will trigger when air has filled the detection site (PBST is out)

```

For reference, the inputs for the detectionFlow function are as follows:

```

detectionFlow(x,y,w,h,x1,y1,w1,h1,cycle,cutoff,background,obj,lookingfor)

```

with the last input ‘lookingfor’ being either a 1 (looking for liquid) or a 0 (looking for air).

The flowchart shown in **Figure 3-60** shows the logic and operation of the overall device.

Figure 3-60 Flowchart showing the broad operation and logic of the device.

CHAPTER 4

FINDINGS

4.1 Protein Detection

Numerous experiments were run in order to ascertain the feasibility of detecting CRP at clinically relevant levels using the previously described sequential loading method. In this method, all reagents are pre-loaded into a microfluidic chip separated by a volume of air in the order that they are to be pumped into the detection site. Once the test has commenced, each reagent is pumped into the detection site sequentially.

This method differs from previous works conducted here [23], [25], [83], in that there is no aspiration of the detection site (channel) between each reagent step. Additionally, the volume, and the volumetric flow rate at which the washing solution it is driven through the channel is vastly different. Previously, the detection site would always be rinsed with between 25-100 μL of washing solution at a flow rate between 2000-5000 $\mu\text{L}/\text{min}$. This washing step was conducted between each reagent and preceded an aspiration procedure wherein the channel would be exposed to extremely high pressures to blow out all remaining liquid. Aspiration of the channel was accomplished using a 3 or 5 cc syringe; usually over 10 cc of air was blasted through the channel (at an approximate flow rate of 300,000 $\mu\text{L}/\text{min}$) between each reagent.

With a truly sequential test, these steps were no longer possible as such pressures required for aspiration would certainly disrupt every other loaded liquid, and

washing volumes of that size would require a substantially larger chip. For these sequentially loaded experiments, the pre-loaded washing steps (post sample, post detection antibody, final wash/reading) were set at 8 μL and the aspiration step was totally eliminated. The tests conducted used V4.1 of the microfluidic chip preloaded with all reagents. CRP doped PBS buffer was used as the sample fluid. All reagents were pumped into the detection site at a rate comparable to flow rates capable by the DCFVS pump used in the system ($\sim 300\text{-}1000\text{ }\mu\text{L}/\text{min}$). **Table 4-1** lists the procedure followed for the first round of tests.

Table 4-1 Procedure for Round 1 of the sequentially loaded tests.

Substance in Detection Site	Action
CRP doped sample	Incubate 10 minutes
Wash	Take reading, Negative Control
Detection Antibody	Incubate 5 minutes
PBST Wash	Incubate 5 minutes
Wash	Take reading, Signal

Figure 4-1 shows the results of the above procedure using the signal processing method described in chapter 3.

Figure 4-1 Data with a dose-response curve fit for CRP in PBS buffer

Each data point represents 5 separate tests; the error bars on each represent the standard error. The data was fit to a sigmoidal dose-response (or four-parameter logistic) curve, which most accurately models a biological system such as this. Equation 3 shows the form of the dose-response curve used.

$$y = min + \frac{max - min}{1 + \left(\frac{EC_{50}}{x}\right)^{\alpha}} \quad [3]$$

Where x is the concentration of the analyte (dose), y is the expected fluorescence signal (response), max is the signal for infinite analyte concentration, min corresponds to the signal with a zero dose of the analyte (negative control), EC_{50} is the concentration of the analyte with an expected signal exactly halfway between max and

min, and α is the slope factor. The values of these parameters used for the curve fit shown in **Figure 4-1** are listed in **Table 4-3**.

The measured limit of detection of 4.37 $\mu\text{g/mL}$ was higher than the cut-off value for a clinical diagnosis (3 $\mu\text{g/mL}$). This necessitated a change to the immunoassay procedure to reduce the LOD. For the second round of tests, the CRP doped sample was incubated for two 5 minute stops. That is, the sample fluid was stopped in the detection site, allowed to incubate for 5 minutes, pumped forward slightly bringing in ‘fresh’ sample fluid, and allowed to incubate an additional 5 minutes. This simple change allowed the detection site to be exposed to the highest possible concentration of sample for two 5 minute periods as opposed to one 10 minute period; this revised procedure is shown below.

Table 4-2 Procedure for Round 2 of the sequentially loaded tests.

Substance in Detection Site	Action
CRP doped sample	Incubate 5 + 5 minutes
Wash	Take reading, Negative Control
Detection Antibody	Incubate 5 minutes
PBST Wash	Incubate 5 minutes
Wash	Take reading, Signal

Additionally, the signal processing procedure was changed slightly from using readings between 510 and 530 nm to using readings between 505 and 540 nm. Finally, the integration time (exposure time) of the spectrometer was changed from 500 ms to 250 ms, which provided a less noisy signal. **Figure 4-2** shows the results from the modified immunoassay test.

Figure 4-2 Data with a dose-response curve fit for CRP in PBS buffer after various improvements to the immunoassay procedure

Again, each data point is the average of 5 tests, with the error bars showing the standard error. The values of the parameters for the dose-response curve fit are listed in the following table.

Table 4-3 The value of the LOD, Negative Control, and various parameters of the dose-response curve fit for round 1 (Figure 4-1) and 2 (Figure 4-2) of the CRP detection tests.

Round	max (signal)	min (signal)	EC50 (µg/mL)	α	NC (signal)	LOD (µg/mL)
1	875	220	8	2.3	194	4.37
2	570	40	7	1.4	43	1.87

The second round of tests yielded a much improved LOD over round one, justifying the changes made to the immunoassay procedure. These results proved that despite having significantly changed the way in which the detection site was washed

between each reagent, a LOD of clinical relevance could still be attained, a critical benchmark if a sequentially loaded assay were to be employed.

4.2 Fluid Handling

4.2.1 Blood Filtration

The method for separating plasma from whole blood described in section 3.2.2 was tested for plasma yield repeatability and to determine the time required to pump in the necessary volume of plasma. V4.1 chips with separation membrane filters permanently attached were made in the fashion described in section 3.2.2. A filter diameter of 10 mm (GX line) was used. The filter holder PDMS piece was made 3 mm thick, and in combination with the 8 mm hole through its center, the open volume above the filter (volume where positive pressure would be introduced) came to 150 μL . Pressure was generated using a 3 ml syringe and syringe pump. The syringe pump was set to a volumetric flow rate of 447 $\mu\text{L}/\text{min}$, equivalent to the flow rate created by the DCFVS pump at 67 pulses per second (pps). The total volume pumped into the chamber above the filter was 1500 μL , a volume reached approximately 3 minutes and 21 seconds into each test. 50 μL (equal to about two drops) of whole blood containing the anticoagulant heparin was pipetted onto the filter media before covering the filter holder with a piece of PDMS (see **Figure 3-16**) and beginning the test. Elapsed time was recorded for every 0.375 μL of plasma infused into the channel. A volume of 0.375 μL was chosen because it corresponded to a 5 mm channel length (0.250 mm height x 0.300 mm width x 5 mm length = 0.375 μL), hence a 5 mm grid was placed behind the microfluidic chip and the elapsed time was recorded each time the leading

edge of the filtrate crossed a grid line. It is important to recognize that the volume of filtrate (plasma) recorded at a particular time is not equal to the total plasma yield as some of this plasma must saturate the bottom surface of the filter and fill the inlet port leading to the plasma channel before it can start to fill the actual channel. The filtrate volume lost to these factors has been estimated to be about 3 μL . **Figure 4-3** shows the compiled results of the plasma pumping tests.

Figure 4-3 Results of the plasma filtration tests, each point represents 10 tests, and the error bars shown represent the standard deviation. The dotted line marks the ideal amount of filtrate to be inserted into the sample channel.

The dotted line in the above figure marks where the ideal infusion volume lies. A volume of 3 μL allows enough of a buffer zone as to not unintentionally fill the sample channel and pump plasma into the Y-section of the chip (see **Figure 3-11c**), while still having a large enough volume to be easily visualized by the fluid position sensor. As seen in **Figure 4-3**, a filtrate volume of 3 μL was reached approximately 4

minutes and 70 seconds into the test. The time required to reach any filtrate volume is highly dependent on the set pumping rate, which for these tests was held at 447 $\mu\text{L}/\text{min}$. Looking at the first 9 points in the figure there is a clear linear relation, one that corresponds to a volumetric flow rate of 0.85 $\mu\text{L}/\text{min}$. Additionally, as the supply of whole blood diminishes the data trends towards an asymptotic maximum, the total volume of filtrate capable of being extracted.

The system itself will not have to rely totally on what can sometimes be unreliable timing schemes thanks to the fluid position sensor. Traditionally, the pump would run for a predetermined amount of time (3:21 or 1500 μL) and a pressure vent at 4:70 would stop the flow of filtrate (hopefully) allowing the correct volume into the channel. Illustrating this point, the error bars in **Figure 4-3** show it is possible that only 2.5 μL of filtrate could be present at the 4:70 mark. With the fluid position sensor on the other hand, the pump can be set to run for the same amount of time, but instead of relying on timing to stop the fluid, the position sensor looking at the sample channel will know exactly when to activate the vent and stop all fluid movement.

4.2.2 Sequential Loading

An experiment was devised to define the correct volumes to be pumped for each step of a complete sequential loading test. For these tests, V4.1 of the chip was loaded with colored water volumes (similar to **Figure 3-30**) representing each reagent of a test and plugged into a testing platform (see **Figure 3-35**). The test platform used V2 of the DCFVS and an identical valving scheme as in the portable unit. For these pumping tests it was assumed that the filtration had already taken place and 3 μL of

filtrate was present in the sample channel. The chip was loaded first with the stand-in sample fluid, then loaded bottom to top with the remaining volumes using a 10 μL pipette. An air buffer of $\sim 5 \mu\text{L}$ was placed between each volume. Incubation times were also included in each test to truly simulate an actual test and to also account for any small air leakages. Due to the way the valves were set up and the speed at which they switch, pressures built up during a pumping cycle are partially stored and reintroduced to the chip once that State has been reactivated from a vent State. This meant that often (particularly in the last pumping cycle) the pump did not need to be turned on to get the last fluid volume (final wash) to the detection site. **Table 4-4** gives an example of the data collected for each test. This particular test did not require the activation of the pump for cycle 6, evident by the static volume value from start to stop.

Table 4-4 Example of the data sheet used to collect the pumping volumes and times for each test.

Discription of Cycle	Cycle	Action	Volume (μL)	Time (min.sec)
Pump the sample fluid to the first HAPD site.	1	State 2		
		Start	0.0	0.00
		Stop	93.0	0.58
		Vent		1.13
		Incubate		5.00
Pump the sample fluid to the second HAPD site.	2	State 2		6.13
		Start	93.0	6.55
		Stop	104.0	7.03
		Vent		7.22
		Incubate		5.00
Pump the sample fluid to the detection site.	3	State 2		12.22
		Start	104.4	12.40
		Stop	118.9	12.49
		Vent		13.15
		Incubate		5.00
Pump the sample Wash to	4	State 4		23.15

the detection site.		Start	118.9	23.30
		Stop	242.7	24.30
		Vent		28.00
		Incubate		2.00
Pump the Detection Antibody to the detection site.	5	State 4		30.00
		Start	242.7	30.21
		Stop	309.8	31.00
		Vent		31.11
		Incubate		5.00
Pump the PBST Wash to the detection site.	6	State 4		36.11
		Start	309.8	
		Stop	309.8	
		Vent		37.30
		Incubate		2.00
Pump the Final Wash to the detection site.	7	State 4		39.30
		Start	309.8	39.43
		Stop	332.3	40.00
		Vent		40.30

A run of 15 tests was completed and the volumes pumped for each cycle was plotted, the results of which are shown below in **Figure 4-4**.

Figure 4-4 Average volume pumped during each cycle over 15 tests. Table 4-4 gives a description of each cycle.

Using this data as a guide, pumping volumes were set for each cycle.

Table 4-5 Pumping volumes set for each cycle based on the data in Figure 4-4.

Cycle	Volume pumped (μL)
1	140
2	35
3	23
4	150
5	70
6	45
7	30

These volumes were decided upon by summing the average and standard deviation at each cycle and rounding to the closest multiple of 5. Again, because of the fluid position sensor, it is better to have the pumping volumes be on the high side because of the system's ability to completely stop all fluid movement using the vent State.

These volumes were subsequently programmed into the Arduino program and an experiment testing the capability of the fluid position sensor during the course of a complete test was conducted.

The test platform shown in **Figure 3-40** was utilized in this final round of fluid handling experiments. It incorporated all the elements of the fluid handling system found in the portable unit in a more flexible bench-top set up. The first experiment consisted of 20 tests in which a sequentially loaded chip was loaded onto the test platform and a pressure from a syringe loaded into a syringe pump drove all of the fluids through the chip at a flow rate of 200 $\mu\text{L}/\text{min}$. During the course of a test the fluid position sensor was to write a unique text message to the Command Window of MATLAB acknowledging that it had indeed sensed the following actions: sample fluid entered first HAPD site, sample fluid entered second HAPD, sample fluid entered the detection site, sample fluid *left* the detection site, Wash one entered the detection site, Wash one left the detection site, Detection Antibody entered the detection site, Detection Antibody left the detection site, PBST Wash entered the detection site, PBST Wash left the detection site, Final Wash entered the detection site, and Final Wash left the detection site. A successful trigger was defined by the appropriate message (or pair of messages in the case of fluid in the detection site) being displayed in the Command Window as the volumes were pumped through the chip. This test was run 25 times, the results of which are shown in **Table 4-6**

Table 4-6 Results of the fluid position sensor tests using a constant flow rate of 200 $\mu\text{L}/\text{min}$. Fluid flow was not interrupted during the course of a test.

Cycle	Successful Triggers (out of 25)
1	24 (96%)
2	24
3	23 (92%)
4	25
5	25
6	25
7	25

Results from this test were encouraging and proved that the fluid position sensor was highly sensitive. Less than perfect results for cycles 1-3 were usually a result of the program not acting quite fast enough due to the significantly smaller volume of sample fluid compared to the rest of the volumes (3 and 8 μL respectively), a quirk that could be mitigated with fine-tuning of the fluid detection algorithm.

Next, an experiment was run similar to the last, only this time the valve configuration and pump used in the portable system would be incorporated. Additionally, they would be controlled by the MATLAB software in the exact manor of the portable system. In a sense, this was a test of the portable units' fluid control system in that its exact configuration of parts, electronic components, and software was used for the bench-top set up. For these 20 tests, a sequentially loaded V4.1 chip was plugged into the test bed and the system was required to autonomously pump the correct volume for each cycle, stop the fluid at the appropriate time, then continue on to the next cycle. Volumes for each cycle were taken from **Table 4-5**. Like the previous test, a unique message was printed to the Command Window after each action; a cycle was deemed successful if the message matched what was actually happening in the chip. The results of this run of tests are compiled in **Table 4-7** below.

Table 4-7 Results of the autonomous sequential loading tests.

Cycle	Successful Triggers (out of 20)
1	20
2	19 (95%)
3	18 (90%)
4	20
5	20
6	20
7	19

These results simultaneously show that the fluid position is quite capable and that the volumes set for each pumping cycle are effective. The less than perfect results in cycles 2, 3, and 7 occurred in 3 separate tests (fault in cycle 2 and 3 test 6, fault in cycle 3 test 14, fault in cycle 7 test 15). As with the previous results, fine tuning of the MATLAB algorithm should lead to even better results.

4.3 Portable Unit

The construction of the portable unit was discussed extensively in the methodology section, but a brief discussion of what was accomplished with this unit is warranted. As was mentioned in the previous section, all of the electronics and many of the parts from the bench-top set up were incorporated into the much smaller portable unit. In addition, the shell was designed to block out all ambient light once the lid was closed, thus replicating the dark room conditions that the bench-top set up was tested in. A system of two reflecting mirrors was designed into the unit that allowed the CCD imaging lens with a working distance of 216.4 mm to be mounted ~ 10 mm away from the chip. The Portable unit also contained the excitation light

source and imaging light source in the same exact positions as in the bench-top, relative to the chip.

Extensive programing was conducted on both the MATLAB and Arduino platforms to enhance the functionality and usability of the device. The fluid position sensor functions created previously by DiFilippo [24] were modified and incorporated to suit the needs of this work.

Finally, the blood filtration step was incorporated into the portable unit with the filter interface lid attachment allowing pressure to be applied above the filter holder on the chip, an arraignment identical to the one used during the blood filtration experiments, discussed in section 4.2.1.

CHAPTER 5

CONCLUSION

5.1 Conclusions

A technique for the quantification of CRP in a PBS buffer, introduced in previous studies, was adapted for use in this work and proved capable in the presented sequential loading scheme. The limit of detection of the presented system was shown (see **Figure 4-2**) to be sensitive to clinically relevant levels.

A custom syringe was created that was specifically designed to effectively and accurately move multiple volumes of liquid simultaneously through microfluidic channels. A custom linear actuator capable of position sensing and extremely slow and high torque activation was engineered. Combined, the two parts were capable of precise high pressure pumping, similar to the capabilities of a bench-top syringe pump.

A fluid handling system was developed for a sequential loading immunoassay scheme. The ability of the system to control the various fluid volumes was a result of specific geometries in the microfluidic channels, the off-chip valving system utilizing 6 micro solenoid valves, the custom pump, and the fluid position sensor all working in concert. The results of fluid handling tests, shown in **Table 4-6** and **4-7**, shows the effectiveness of the presented system.

An on-chip technique was developed that provided the means to separate plasma from whole blood. A separation membrane specifically designed for small

volumes was permanently affixed to the chip and a positive pressure applied above the filter prevented any movement of the pre-loaded reagents. As shown in **Figure 4-3**, the plasma yield using this technique proved to be more than sufficient with the required volume of filtrate being reached at around the 4 minute and 70 second mark.

The portable unit developed for this work successfully integrated all of the fluid handling techniques developed on the bench-top set up, into a compact and self-contained package. The technique for on-chip blood filtration was also added to the unit's capabilities. Additionally, the unit's interface, hosted by MATLAB, was designed to be user friendly and accessible by someone without a technical background.

5.2 Future Work

5.2.1 Fluid Handling

Performance of the fluid handling system could be improved relatively easily by optimizing all of the parameters of the MATLAB fluid detection function. While the function was significantly modified from its original version to accommodate the goals of this work, further optimization should yield even better reliability. The fluid detection's 'threshold' value, percentage of ROI to be filled with air/liquid before triggering, and the time delay between a change of state in an ROI and a vent action are all examples of parameters that could be fine-tuned to optimize performance.

Illumination of the chip for imaging the fluid volumes flowing through the chip is extremely crucial to the performance of the fluid sensor. Additional optimization of the illumination light would produce a better image (*i.e.* more contrast

between liquid and air) and thus, a better sensor. The LED used for illumination in the presented system worked well, but the single point source of light created an uneven illumination that may have hampered the sensors performance.

Further reduction of dead volumes created by tubing could be completely eliminated in certain instances if ‘face mount’ style microvalves were used in place of the ‘soft tube ported’ style currently used. Using a face mount style, the microvalve could be mounted directly to the underside of a specially designed chip plate and the need for any connecting tubing would be eliminated.

Currently, the way in which the Arduino sends stepping commands to the stepper driver shield, all other actions on the Arduino microcontroller are blocked. This means that if a particular fluid volume were to reach its destination before the unit had finished its pumping cycle, the signal sent from the MATLAB function telling the Arduino to vent could not be executed until after the cycle had finished. Often this was not an issue and the pumping cycle had finished before a vent was required, but due to the variation in flow characteristics from test to test, occasionally a vent was delayed enough that the volume was pumped past its destination. A simple fix would be to allow the vent signal to interrupt the pump cycle. This change to the programing architecture would prevent any such overlapping and even allow for a more aggressive pumping scheme.

Because of the large encapsulated volume of air above the integrated filter (~ 150 μ L), there was a possibility of back flow through the sample channel during the course of a test. This problem was somewhat mitigated by the coagulation of the blood in the filter which often created a strong enough seal to prevent such back flow. Still,

if the test is to be totally reliable, a permanent solution to this intermittent problem must be found. All of the filtration tests were conducted using 50 μL of *drawn* whole blood. When blood is drawn into vials, the anticoagulant heparin is added to prevent the blood from clotting. Conceivably, heparin free blood (finger prick to filter) could clot the filter more reliably than drawn blood. The issue of back flow could also be eliminated by integrating a micro check valve into the microchannel at a point immediately following the inlet port for the plasma. One of the many micro check valves that exist for microfluidic channels [86] could be utilized for such a task.

5.2.2 Portable Unit

The complex arrangement of mirrors in the unit was necessary to create the correct working distance between the imaging lens and the microfluidic chip. This lens was chosen because it provided the required magnification and field of vision. Most importantly, it was chosen because it was available for purchase. A lens with custom specifications could provide the same magnification and field of view while reducing the working distance to a few cm. Such a lens would allow the imaging CD to be mounted directly above the chip, possibly in the lid, reducing the overall size of the unit significantly.

Using a CCD for fluorescence detection instead of a spectrometer, similar to what Jones [25] did, could further reduce the size of the unit, no longer requiring the excitation light assembly or the current driver for the high power LED. Using a CCD in direct contact with the chip for detection could improve the LOD and eliminate the

need for outside software for fluorescence data acquisition, allowing the data to be interpreted and displayed by the device instantaneously.

An improved version of the custom pump designed for this work could be further reduced in size, both in the volumetric capacity and overall size. These changes would allow the unit to shrink in size and may further improve the performance of the pump. By reducing the size of the chambers in the syringe, the pump could act on the volumes of fluid loaded in the chip more rapidly, effectively reducing the lag time between syringe movement and fluid movement. The current design is capable of ~ 2000 μL of displacement, far more than is necessary.

One of the greatest and most exciting leaps forward that this unit could take includes the integration of the now ubiquitous ‘smartphone’. Today’s phones are more than just tools to communicate; they are in fact extremely powerful mobile computing devices. A common iPhone has more computational power than all of NASA back in 1969, when it placed two astronauts on the moon, harnessing this power would allow for a total transformation of the unit and its capabilities. The unit itself could be little more than a shell containing the hardware needed to control the fluid volumes and detect fluorescence. Once docked with the unit, the smartphone could take the place of the Arduino, MATLAB, and even the imaging CCD, utilizing the smartphones onboard camera(s). Every aspect of the device could easily be handled by the smartphone while providing a touch screen user interface that has become second nature to most in the United States and many around the world. Off-loading the bulk of electronics to the smart phone could drastically reduce the size of the unit, down to a size that would now only need to house a pump, valves, excitation light source,

detection CCD, and the microfluidic chip. Utilization of a smartphone could provide a way to vastly improve almost all aspects of the portable unit.

APPENDIXES

6.1 Visual Basic GUI Source Code

```
Imports System
Imports System.IO.Ports
Imports System.Threading
Imports System.ComponentModel

Public Class Form1

    Dim myPort As Array

    Private Property a As String

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

        myPort = IO.Ports.SerialPort.GetPortNames()
        PortComboBox.Items.AddRange(myPort)

        SerialPort1.PortName = PortComboBox.Text
        SerialPort1.BaudRate = BaudComboBox.Text
        SerialPort1.Open()

    End Sub

    Private Sub GroupBox1_Enter(ByVal sender As System.Object, ByVal e As System.EventArgs)

    End Sub

    Private Sub SplitContainer1_Panel2_Paint(ByVal sender As System.Object, ByVal e As System.Windows.Forms.PaintEventArgs)

    End Sub

    Private Sub State2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles State2.Click
        SerialPort1.Write("n")
        State2.Enabled = False
        State1.Enabled = True
        State3.Enabled = True
        State4.Enabled = True

        Button2.Enabled = False 'V1 Down
        Button1.Enabled = True 'V1 Down
        Button4.Enabled = False 'V2 Up
        Button3.Enabled = True 'V2 Up
        Button9.Enabled = False 'V3 Down
        Button10.Enabled = True 'V3 Down
        Button8.Enabled = False 'V4 Up
        Button7.Enabled = True 'V4 Up
        Button13.Enabled = False 'V5 Down
        Button14.Enabled = True 'V5 Down
        Button11.Enabled = False 'V6 Down
        Button12.Enabled = True 'V6 Down

    End Sub

    Private Sub PortComboBox_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PortComboBox.SelectedIndexChanged

    End Sub

    Private Sub BaudComboBox_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BaudComboBox.SelectedIndexChanged

    End Sub

    Private Sub Init_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Init.Click

        SerialPort1.PortName = PortComboBox.Text

    End Sub

End Class
```

```

        SerialPort1.BaudRate = BaudComboBox.Text
        SerialPort1.Open()
        Init.Enabled = False
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button1.Click
        SerialPort1.Write("a")
        Button1.Enabled = False
        Button2.Enabled = True
    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button2.Click
        SerialPort1.Write("b")
        Button2.Enabled = False
        Button1.Enabled = True
    End Sub

    Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button4.Click
        SerialPort1.Write("c")
        Button4.Enabled = False
        Button3.Enabled = True
    End Sub

    Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button3.Click
        SerialPort1.Write("d")
        Button3.Enabled = False
        Button4.Enabled = True
    End Sub

    Private Sub Button10_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button10.Click
        SerialPort1.Write("e")
        Button10.Enabled = False
        Button9.Enabled = True
    End Sub

    Private Sub Button9_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button9.Click
        SerialPort1.Write("f")
        Button9.Enabled = False
        Button10.Enabled = True
    End Sub

    Private Sub Button8_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button8.Click
        SerialPort1.Write("g")
        Button8.Enabled = False
        Button7.Enabled = True
    End Sub

    Private Sub Button7_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button7.Click
        SerialPort1.Write("h")
        Button7.Enabled = False
        Button8.Enabled = True
    End Sub

    Private Sub Button14_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button14.Click
        SerialPort1.Write("i")
        Button14.Enabled = False
        Button13.Enabled = True
    End Sub

    Private Sub Button13_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button13.Click
        SerialPort1.Write("j")
        Button13.Enabled = False
        Button14.Enabled = True
    End Sub

    Private Sub Button12_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button12.Click
        SerialPort1.Write("k")
        Button12.Enabled = False

```

```

        Button11.Enabled = True
    End Sub

    Private Sub Button11_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button11.Click
        SerialPort1.Write("1")
        Button11.Enabled = False
        Button12.Enabled = True
    End Sub

    Private Sub State1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
State1.Click
        SerialPort1.Write("m")
        State1.Enabled = False
        State2.Enabled = True
        State3.Enabled = True
        State4.Enabled = True

        Button2.Enabled = False 'V1 Down
        Button1.Enabled = True 'V1 Down
        Button4.Enabled = False 'V2 Up
        Button3.Enabled = True 'V2 Up
        Button10.Enabled = False 'V3 Up
        Button9.Enabled = True 'V3 Up
        Button7.Enabled = False 'V4 Down
        Button8.Enabled = True 'V4 Down
        Button13.Enabled = False 'V5 Down
        Button14.Enabled = True 'V5 Down
        Button12.Enabled = False 'V6 Up
        Button11.Enabled = True 'V6 Up

    End Sub

    Private Sub State3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
State3.Click
        SerialPort1.Write("o")
        State3.Enabled = False
        State2.Enabled = True
        State1.Enabled = True
        State4.Enabled = True

        Button1.Enabled = False 'V1 Up
        Button2.Enabled = True 'V1 Up
        Button14.Enabled = False 'V5 Up
        Button13.Enabled = True 'V5 Up

    End Sub

    Private Sub State4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
State4.Click
        SerialPort1.Write("p")
        State4.Enabled = False
        State2.Enabled = True
        State3.Enabled = True
        State1.Enabled = True

        Button2.Enabled = False 'V1 Down
        Button1.Enabled = True 'V1 Down
        Button3.Enabled = False 'V2 Down
        Button4.Enabled = True 'V2 Down
        Button9.Enabled = False 'V3 Down
        Button10.Enabled = True 'V3 Down
        Button8.Enabled = False 'V4 Up
        Button7.Enabled = True 'V4 Up
        Button13.Enabled = False 'V5 Down
        Button14.Enabled = True 'V5 Down
        Button11.Enabled = False 'V6 Down
        Button12.Enabled = True 'V6 Down

    End Sub
End Class

```

6.2 Arduino Source Code for GUI

```
int OneA1 =      22;           // Pins 22&23 Control Valve 1
int TwoA1 =      23;
int ThreeA1 =    24;           // Pins 24&25 Control Valve 2
int FourA1 =     25;

int OneA2 =      50;           // Pins 50&51 Control Valve 3
int TwoA2 =      51;
int ThreeA2 =    46;           // Pins 46&47 Control Valve 4
int FourA2 =     47;

int OneA3 =      52;           // Pins 52&53 Control Valve 5
int TwoA3 =      53;
int ThreeA3 =    48;           // Pins 48&49 Control Valve 6
int FourA3 =     49;

int timer = 10;
int inByte = 0;

void setup() {
  // initialize serial communication:
  Serial.begin(9600);
  // initialize the LED pins:
  pinMode(OneA1, OUTPUT);      // initialize Valve pins as outputs
  pinMode(TwoA1, OUTPUT);
  pinMode(ThreeA1, OUTPUT);
  pinMode(FourA1, OUTPUT);

  pinMode(OneA2, OUTPUT);
  pinMode(TwoA2, OUTPUT);
  pinMode(ThreeA2, OUTPUT);
  pinMode(FourA2, OUTPUT);

  pinMode(OneA3, OUTPUT);
  pinMode(TwoA3, OUTPUT);
  pinMode(ThreeA3, OUTPUT);
  pinMode(FourA3, OUTPUT);
}

void loop() {
  // read the sensor:
  if (Serial.available() > 0) {
    int inByte = Serial.read();
    // do something different depending on the character received.
    // The switch statement expects single number values for each case;
    // in this exmaple, though, you're using single quotes to tell
    // the controller to get the ASCII value for the character. For
    // example 'a' = 97, 'b' = 98, and so forth:

    switch (inByte) {

      case 'a':    // Valve 1 Up
        digitalWrite(23, HIGH);
        delay(timer);
        digitalWrite(23, LOW);
        break;

      case 'b':    //Valve 1 Down
        digitalWrite(22, HIGH);
        delay(timer);
        digitalWrite(22, LOW);
        break;

      case 'c':
        digitalWrite(24, HIGH);
        delay(timer);
        digitalWrite(24, LOW);
```



```

        break;

    case 'd':
        digitalWrite(25, HIGH);
        delay(timer);
        digitalWrite(25, LOW);
        break;

    case 'e':
        digitalWrite(51, HIGH);
        delay(timer);
        digitalWrite(51, LOW);
        break;

    case 'f':
        digitalWrite(50, HIGH);
        delay(timer);
        digitalWrite(50, LOW);
        break;

    case 'g':
        digitalWrite(47, HIGH);
        delay(timer);
        digitalWrite(47, LOW);
        break;

    case 'h':
        digitalWrite(46, HIGH);
        delay(timer);
        digitalWrite(46, LOW);
        break;

    case 'i':
        digitalWrite(53, HIGH);
        delay(timer);
        digitalWrite(53, LOW);
        break;

    case 'j':
        digitalWrite(52, HIGH);
        delay(timer);
        digitalWrite(52, LOW);
        break;

    case 'k':
        digitalWrite(49, HIGH);
        delay(timer);
        digitalWrite(49, LOW);
        break;

    case 'l':
        digitalWrite(48, HIGH);
        delay(timer);
        digitalWrite(48, LOW);
        break;

    case 'm': // State 1: pump/pull blood through filter
        digitalWrite(22, HIGH); //V1 Down
        delay(timer);
        digitalWrite(22, LOW);
        delay(1);
        digitalWrite(24, HIGH); //V2 Up
        delay(timer);
        digitalWrite(24, LOW);
        delay(1);
        digitalWrite(51, HIGH); //V3 Up
        delay(timer);
        digitalWrite(51, LOW);
        delay(1);
        digitalWrite(46, HIGH); //V4 Down
        delay(timer);

```

```

digitalWrite(46, LOW);
delay(1);
digitalWrite(52, HIGH);    //V5 Down
delay(timer);
digitalWrite(52, LOW);
delay(1);
digitalWrite(49, HIGH);    //V6 Up
delay(timer);
digitalWrite(49, LOW);
break;

case 'n':    // State 2: push/pull blood to HAP channels
digitalWrite(22, HIGH);    //V1 Down
delay(timer);
digitalWrite(22, LOW);
delay(1);
//digitalWrite(24, HIGH);    //V2 Up    Does not change from State 1
//digitalWrite(24, LOW);
digitalWrite(50, HIGH);    //V3 Down
delay(timer);
digitalWrite(50, LOW);
delay(1);
digitalWrite(47, HIGH);    //V4 Up
delay(timer);
digitalWrite(47, LOW);
delay(1);
digitalWrite(52, HIGH);    //V5 Down
delay(timer);
digitalWrite(52, LOW);
delay(1);
digitalWrite(48, HIGH);    //V6 Down
delay(timer);
digitalWrite(48, LOW);
break;

case 'o':    // State 3: Vent/stop fluid movement
digitalWrite(23, HIGH);    //V1 Up
delay(timer);
digitalWrite(23, LOW);
delay(1);
//digitalWrite(24, HIGH);    //V2 Up    Does not change from State 2
//digitalWrite(24, LOW);
//digitalWrite(50, HIGH);    //V3 Down    Does not change from State 2
//digitalWrite(50, LOW);
//digitalWrite(47, HIGH);    //V4 Up    Does not change from State 2
//digitalWrite(47, LOW);
digitalWrite(53, HIGH);    //V5 Up
delay(timer);
digitalWrite(53, LOW);
//digitalWrite(48, HIGH);    //V6 Down    Does not change from State 2
//digitalWrite(48, LOW);
break;

case 'p':    // State 4: push/pull reagents through detection site
digitalWrite(22, HIGH);    //V1 Down
delay(timer);
digitalWrite(22, LOW);
delay(1);
digitalWrite(25, HIGH);    //V2 Down
delay(timer);
digitalWrite(25, LOW);
delay(1);
//digitalWrite(50, HIGH);    //V3 Down-Position of V3 is inconsequential w/V2 Down
//digitalWrite(50, LOW);
//digitalWrite(47, HIGH);    //V4 Up    Only State 1 has V4 Down
//digitalWrite(47, LOW);
digitalWrite(52, HIGH);    //V5 Down
delay(timer);
digitalWrite(52, LOW);
//digitalWrite(48, HIGH);    //V6 Down    Only State 1 has V6 Up
//digitalWrite(48, LOW);

```

```
        break;  
    }  
}
```

6.3 Technical Drawings

Name: End Cap for DCFVS V2	Scale: 2:1	Units: mm
	Material: aluminum	

Drawing # 02

Name: DCFVS V2 plunger	Scale: 5:1	Units: mm
Drawing # 03	Material: clear polycarbonate

Name: Linear Actuator Housing	Scale: 1:1	Units: mm
	Material: Rapid Prototyped ABS	

Drawing # 04

Name: Drive Linkage, stepper to liner actuator	Scale: 5:1	Units: mm
	Material: Rapid Prototyped ABS	

Drawing # 05

Name: DCFVS Cradle	Scale: 1:1	Units: mm
Drawing # 06	Material: Rapid Prototyped ABS

Name:
Chamber Clamp

Scale: 2:1

Units: mm

Drawing # 07

Material: Rapid Prototyped ABS

Name: Stepper Motor Heatsink	Scale: 2:1	Units: mm
Drawing # 09	Material: aluminum

Name: Test bed optical housing and chip platform	Scale: 1:2	Units: mm
	Material: Rapid Prototyped ABS	

Drawing # 10

Name: Excitation light source and heat sink holder	Scale: 2:1	Units: mm
	Material: Rapid Prototyped ABS	

Drawing # 11

Name: Test bed cover	Scale: 1:2	Units: mm
	Material: Rapid Prototyped ABS	

Drawing # 12

Name: CCD holder	Scale: 1:1	Units: mm
	Material: Rapid Prototyped ABS	

Drawing # 13

Name: CCD Control Board Holder	Scale: 1:1	Units: mm
	Material: Rapid Prototyped ABS	

Drawing # 14

Name: LED positioner, base	Scale: 1:2	Units: mm
	Material: Rapid Prototyped ABS	

Drawing # 16

Name: LED Positioner, secondary rail	Scale: 1:2	Units: mm
	Material: Rapid Prototyped ABS	

Drawing # 17

Name: LED Positioner, cart	Scale: 1:2	Units: mm
	Material: Rapid Prototyped ABS	

Drawing # 18

Name: LED Positioner, LED holder	Scale: 2:1	Units: mm
Drawing # 19	Material: Rapid Prototyped ABS

Name: Mirror 1 Holder	Scale: 1:1	Units: mm
	Material: Rapid Prototyped ABS	

Drawing # 20

Name:
Chip Platform

Scale: 2:3

Units: mm

Drawing # 21

Material: Rapid Prototyped ABS

Name: Device Shell	Scale: 1:3	Units: mm
Drawing # 22	Material: Rapid Prototyped ABS

Name: Small Breadboard Holder	Scale: 2:1	Units: mm
	Material: Rapid Prototyped ABS	

Drawing # 23

Name: Large Breadboard Holder	Scale: 1:1	Units: mm
	Material: Rapid Prototyped ABS	

Drawing # 24

Name: Pump Mounting Bracket	Scale: 1:1	Units: mm
	Material: Rapid Prototyped ABS	

Drawing # 25

Name: Stepper Driver Mounting Bracket	Scale: 1:1	Units: mm
	Material: Rapid Prototyped ABS	

Drawing # 26

Name:
Valve 2 Holder

Drawing # 27

Scale: 2:1

Units: mm

Material: Rapid Prototyped ABS

Name: Top Section of Shell	Scale: 1:2	Units: mm
	Material: Rapid Prototyped ABS	

Drawing # 28

Name:
Chip Plate Cover

Scale: 2:3

Units: mm

Drawing # 29

Material: Rapid Prototyped ABS

Name: Dbl. Hinge Connector for Lid	Scale: 2:1	Units: mm
	Material: Rapid Prototyped ABS	

Drawing # 30

Name: Steel Pin Clamp	Scale: 5:1	Units: mm
	Material: Rapid Prototyped ABS	

Drawing # 31

Name: Filter Interface Base	Scale: 1:1	Units: mm
	Material: Rapid Prototyped ABS	

Drawing # 32

Name:
Filter Interface Slider

Scale: 2:1

Units: mm

Drawing # 33

Material: Rapid Prototyped ABS

Name: Filter Interface PDMS Anchor	Scale: 3:1	Units: mm
	Material: Rapid Prototyped ABS	

Drawing # 34

Name: PDMS Anchor Suspension Piece	Scale: 1:1	Units: mm
Drawing # 35	Material: Rapid Prototyped ABS

Name: Fiber Optics Guide	Scale: 5:1	Units: mm
	Material: Rapid Prototyped ABS	

Drawing # 36

6.4 Device Wiring Schematic

6.5 Device Arduino Source Code

```
/*
  Code Loaded Onto the Arduino MEGA
*/

#include <Wire.h>
#include <Adafruit_MotorShield.h>
#include "utility/Adafruit_PWMServoDriver.h"

// Create the motor shield object with the default I2C address
Adafruit_MotorShield AFMS = Adafruit_MotorShield();

// Connect a stepper motor with 20 steps per revolution to motor port #1 (M1 and M2)
Adafruit_StepperMotor *myMotor = AFMS.getStepper(20, 1);

// These constants will not change:
const int LA_Position = A1;    // Pin that the Linear Actuator's position sensor is
                                // attached to (potentiometer).
const int ex_LED = 45;         // Pin connected to excitation LED H-bridge switch.
const int light_LED = 27;      // Pin connected to light source LED.
const int timer = 10;          // Valve timer, 10ms (@ 5V) of power to valve.

const int OneA1 = 22;           // Pins 22&23 Control Valve 1
const int TwoA1 = 23;
const int ThreeA1 = 24;         // Pins 24&25 Control Valve 2
const int FourA1 = 25;
const int OneA2 = 50;           // Pins 50&51 Control Valve 3
const int TwoA2 = 51;
const int ThreeA2 = 46;         // Pins 46&47 Control Valve 4
const int FourA2 = 47;
const int OneA3 = 52;           // Pins 52&53 Control Valve 5
const int TwoA3 = 53;
const int ThreeA3 = 48;         // Pins 48&49 Control Valve 6
const int FourA3 = 49;

// These constants will change:
int LA_Value = 0;
int Position = 0;
int counter = 0;
int last_pump = 0;
int inByte = 0;
int filter_chip = 0;
int action = 0;
int question = 0;
int timer_over = 0;
float coeff_1 = 0.0008; //0.00005 = old value
float coeff_2 = 0.0994; //0.01120 = old value
float coeff_3 = 4.4082; //0.93350 = old value

void setup() {

  Initialize_Valve_Pins(); // Does what it says, along with LED pins, sets all to LOW

  Serial.begin(9600);      // Initialize serial communication

  s_For_Ready();           // Waits for 's' character to be received before continuing

  Wake_Up();               // Wakes up valves (flip each valve UP then DOWN)

  LA_Value = analogRead(LA_Position); // Initialize value from Linear Actuator

  /*----- Adafruit Motor Shield Stuff -----*/
  AFMS.begin();             // create with the default frequency of 1.6KHz
  myMotor->setSpeed(1500);   // 1200 rpm (gives 400pps)

  Valve_Config_Home_Pump(); // Valve config for homing pump
}
```

```

Home_Pump();    //Pump homing loop

Valve_Config_Chip_Ins();  // Valve config for chip insertion

y_For_Chip();    // Waits for user to input 'y' once chip has been inserted and lid
has been closed

Filter_Question();    // filter chip being used? state one if yes, state two if no

Start_Test();

First_Step();
}

void loop() {

  if (Serial.available() > 0) {
    action = Serial.read();

    switch (action){

      case 'a': //Signal from MATLAB indicating plasma is in first HAPD site.
        Vent(); //Stop fluid movement.
        digitalWrite(light_LED,LOW);    //LED light for fluid imaging, turned on in
                                         First_Step function is turned off.
        while (timer_over != 'h'){      // MATLAB will send 'h' once "Inc_Timer"
          if (Serial.available() > 0) {  // function has finished.
            timer_over = Serial.read();}
          delay(10);}

        digitalWrite(light_LED,HIGH);
        State_Two();    // Continue pumping plasma to second HAPD site.
        myMotor->step(279, BACKWARD, SINGLE);    // Turn on pump.
        myMotor->release();
        break;

      case 'b': // Signal from MATLAB indicating plasma is in second HAPR site.
        Vent(); // Stop fluid movement.
        digitalWrite(light_LED,LOW);

        while (timer_over != 'i'){      // MATLAB will send 'i' once "Inc_Timer"
          if (Serial.available() > 0) {  // function has finished.
            timer_over = Serial.read();}
          delay(10);}

        digitalWrite(light_LED,HIGH);
        State_Two();    // Continue pumping plasma to detection site.
        myMotor->step(186, BACKWARD, SINGLE);    // Turn on pump.
        myMotor->release();
        break;

      case 'c': // Signal from MATLAB indicating plasma is in the detection site.
        Vent(); // Stop fluid movement.
        digitalWrite(light_LED,LOW);

        while (timer_over != 'j'){      // MATLAB will send 'j' once "Inc_Timer"
          if (Serial.available() > 0) {  // function has finished.
            timer_over = Serial.read();}
          delay(10);}

        digitalWrite(light_LED,HIGH);
        State_Four();    // Pump wash_1 to detection site.
        myMotor->step(1209, BACKWARD, SINGLE);    // Turn on pump.
        myMotor->release();
        break;

      case 'd': // Signal from MATLAB indicating first wash is in the detection site.
        Vent(); // Stop fluid movement.
        digitalWrite(light_LED,LOW);
        Spectrometer_Reading_1(); // Allows user to take a spectrometer reading.
        digitalWrite(light_LED,HIGH);

```

```

        State_Four(); // Pump det_ab to detection site.
        myMotor->step(465, BACKWARD, SINGLE); // Turn on pump.
        myMotor->release();
        break;

case 'e': // Signal from MATLAB indicating det_ab is in the detection site.
    Vent(); // Stop fluid movement.
    digitalWrite(light_LED, LOW);

    while (timer_over != 'k'){ // MATLAB will send k once "Inc_Timer"
        if (Serial.available() > 0) { // function has finished.
            timer_over = Serial.read();
            delay(10);}

    digitalWrite(light_LED, HIGH);
    State_Four(); // Pump PBST to detection site.
    myMotor->step(372, BACKWARD, SINGLE); // Turn on pump.
    myMotor->release();
    break;

case 'f': // Signal from matlab indicating PBST is in the detection site.
    Vent(); // Stop fluid movement.
    digitalWrite(light_LED, LOW);

    while (timer_over != 'l'){ // MATLAB will send lower case L once
        if (Serial.available() > 0) { // "Inc_Timer" function has finished.
            timer_over = Serial.read();
            delay(10);}

    digitalWrite(light_LED, HIGH);
    State_Four(); // Pump wash_2 to detection site.
    myMotor->step(186, BACKWARD, SINGLE); // Turn on pump.
    myMotor->release();
    break;

case 'g': // Signal from MATLAB indicating wash_2 is in the detection site.
    Vent(); // Stop fluid movement.
    digitalWrite(light_LED, LOW);
    Spectrometer_Reading_2();
    Serial.println(" ");
    Serial.println("-----");
    Serial.println("                Test is");
    Serial.println("                Complete");
    Serial.println("-----");
    Serial.println(" ");
    break;
    }
}

/*****
/***** FUNCTIONS *****/
/*****/

/***** Home Pump Function *****/

void Home_Pump() {

while(LA_Value <= 330 || LA_Value >= 340){
    if (LA_Value >= 340){ // LA is OUT too far
        Serial.print("Pump is too far OUT, with LA value of ");
        Serial.println(LA_Value);
        Position = abs(((65-((coeff_1*(pow(LA_Value,2)))- // =steps away from
            (coeff_2*LA_Value)+coeff_3))*150)); // center position
        counter = Position + 1;
        Serial.print("Number of steps ");
        Serial.println(counter);
        delay(10);
        myMotor->step(counter, BACKWARD, SINGLE);
        myMotor->release();
        delay(1000);
        LA_Value = analogRead(LA_Position);
    }
}

```

```

        last_pump = 'b';
    }
    if (LA_Value <= 330){          // LA is IN too far
        Serial.print("Pump is too far IN, with LA value of ");
        Serial.println(LA_Value);
        Position = abs(((65-((coeff_1*(pow(LA_Value,2)))-
            (coeff_2*LA_Value)+coeff_3))*150));
        counter = Position + 1;
        Serial.print("Number of steps ");
        Serial.println(counter);
        delay(10);
        myMotor->step(counter, FORWARD, SINGLE);
        myMotor->release();
        delay(1000);
        LA_Value = analogRead(LA_Position);
        last_pump = 'f';
    }
}
if (last_pump == 'f'){
    myMotor->step(900, BACKWARD, SINGLE);
    myMotor->release();
}
if (last_pump == 'b'){
    //myMotor->step(900, FORWARD, SINGLE);
    //myMotor->release();
}
delay(50);
Serial.println("Pump Has Found HOME");
}
/*****/

/***** Wait for 's' Function *****/
void s_For_Ready(){
    inByte = 0;
    Serial.println(" ");
    Serial.println("-----");
    Serial.println("          Turn on main power to unit, then ");
    Serial.println("          Enter 's' to continue setup");
    Serial.println("-----");
    Serial.println(" ");
    while (inByte != 's'){
        if (Serial.available() > 0) {
            inByte = Serial.read();
        }
        delay(10);
    }
}
/*****/

/***** Wait for 'y' Function (chip insertion) *****/
void y_For_Chip(){
    inByte = 0;
    Serial.println(" ");
    Serial.println("-----");
    Serial.println("Insert Chip into device and Close the Lid");
    Serial.println("          Enter 'y' when finished");
    Serial.println("-----");
    Serial.println(" ");
    while (inByte != 'y'){
        if (Serial.available() > 0) {
            inByte = Serial.read();
        }
        delay(10);
    }
}
/*****/

/***** Filter Chip Question *****/
void Filter_Question(){
    filter_chip = 0;
    Serial.println(" ");
    Serial.println("-----");
    Serial.println("          Does this chip use a filter?");
    Serial.println("          Enter 'y' for Yes or 'n' for No");
    Serial.println("-----");
    Serial.println(" ");

```

```

while (filter_chip != 'n' && filter_chip != 'y'){
    if (Serial.available() > 0) {
        filter_chip = Serial.read();
    }
    delay(10);}
if (filter_chip == 'y'){
    question = 1;} // Start with state one.
if (filter_chip == 'n'){
    question = 2;} // Start with state two.
delay(10);}
/*****/

/***** Start Test ? Function *****/
void Start_Test(){
    inByte = 0;
    Serial.println(" ");
    Serial.println("-----");
    Serial.println("                Start Test?");
    Serial.println("                Enter 'y' to confirm");
    Serial.println("-----");
    Serial.println(" ");
    while (inByte != 'y'){
        if (Serial.available() > 0) {
            inByte = Serial.read();
        }
        delay(10);}
    Serial.println(" ");
    Serial.println("-----");
    Serial.println("                Test Has Commenced");
    Serial.println("-----");
    Serial.println(" ");
}
/*****/

/***** First Step Function *****/
void First_Step(){
    inByte = 0;
    while (inByte != 'y'){ // MATLAB will send 'y' once the camera
        if (Serial.available() > 0) { // has settled.
            inByte = Serial.read();}
        delay(10);}
    if (question == 1){ // Yes filter.
        State_One();}
    else if (question == 2){ // No filter.
        State_Two();
        digitalWrite(light_LED, HIGH); //Light source LED for CCD on.
        myMotor->setSpeed(200);
        myMotor->step(1302, BACKWARD, SINGLE);
        myMotor->release();
    }
}
/*****/

/***** Spectrometer Reading_1 Function *****/
void Spectrometer_Reading_1(){
    inByte = 0;
    Serial.println(" ");
    Serial.println("-----");
    Serial.println("                Baseline Reading");
    Serial.println("Enter 'y' to turn on excitation light for 3 seconds");
    Serial.println("-----");
    Serial.println(" ");
    while (inByte != 'y'){
        if (Serial.available() > 0) {
            inByte = Serial.read();
        }
        delay(10);}
    delay(3000);
    digitalWrite(ex_LED, HIGH);
    delay(3000);
    digitalWrite(ex_LED, LOW);
}

```

```

    delay(10);
}
/*****

/***** Spectrometer Reading_2 Function *****/
void Spectrometer_Reading_2(){
    inByte = 0;
    Serial.println(" ");
    Serial.println("-----");
    Serial.println("                Intensity Reading");
    Serial.println("Enter 'y' to turn on excitation light for 3 seconds");
    Serial.println("-----");
    Serial.println(" ");
    while (inByte != 'y'){
        if (Serial.available() > 0) {
            inByte = Serial.read();
        }
        delay(10);}
    delay(3000);
    digitalWrite(ex_LED,HIGH);
    delay(3000);
    digitalWrite(ex_LED,LOW);
    delay(10);
}
/*****

/***** Initialize Valve Pins *****/
void Initialize_Valve_Pins(){
    pinMode(OneA1, OUTPUT);    // Initialize Valve pins as outputs.
    pinMode(TwoA1, OUTPUT);
    pinMode(ThreeA1, OUTPUT);
    pinMode(FourA1, OUTPUT);
    pinMode(OneA2, OUTPUT);
    pinMode(TwoA2, OUTPUT);
    pinMode(ThreeA2, OUTPUT);
    pinMode(FourA2, OUTPUT);
    pinMode(OneA3, OUTPUT);
    pinMode(TwoA3, OUTPUT);
    pinMode(ThreeA3, OUTPUT);
    pinMode(FourA3, OUTPUT);

    digitalWrite(OneA1, LOW);
    digitalWrite(TwoA1, LOW);
    digitalWrite(ThreeA1, LOW);
    digitalWrite(FourA1, LOW);
    digitalWrite(OneA2, LOW);
    digitalWrite(TwoA2, LOW);
    digitalWrite(ThreeA2, LOW);
    digitalWrite(FourA2, LOW);
    digitalWrite(OneA3, LOW);
    digitalWrite(TwoA3, LOW);
    digitalWrite(ThreeA3, LOW);
    digitalWrite(FourA3, LOW);

    pinMode(light_LED,OUTPUT);
    digitalWrite(light_LED, LOW);

    pinMode(ex_LED,OUTPUT);
    digitalWrite(ex_LED, LOW);

    delay(2);}
/*****

/***** Wake Up Valves *****/
void Wake_Up(){
    inByte = 0;
    Serial.print("Cycling Valves ");
    Serial.print("1, ");
    digitalWrite(23, HIGH);    // Valve 1 UP then DOWN
    delay(timer);              // in that order.
    digitalWrite(23, LOW);

```



```

delay(250);
digitalWrite(22, HIGH);
delay(timer);
digitalWrite(22, LOW);
delay(250);

Serial.print("2, ");
digitalWrite(24, HIGH);      // Valve 2 UP and DOWN
delay(timer);               // in that order.
digitalWrite(24, LOW);
delay(250);
digitalWrite(25, HIGH);
delay(timer);
digitalWrite(25, LOW);
delay(250);

Serial.print("3, ");
digitalWrite(51, HIGH);     // Valve 3 UP and DOWN
delay(timer);               // in that order.
digitalWrite(51, LOW);
delay(250);
digitalWrite(50, HIGH);
delay(timer);
digitalWrite(50, LOW);
delay(250);

Serial.print("4, ");
digitalWrite(47, HIGH);     // Valve 4 UP and DOWN
delay(timer);               // in that order.
digitalWrite(47, LOW);
delay(250);
digitalWrite(46, HIGH);
delay(timer);
digitalWrite(46, LOW);
delay(250);

Serial.print("5, ");
digitalWrite(53, HIGH);     // Valve 5 UP and DOWN
delay(timer);               // in that order.
digitalWrite(53, LOW);
delay(250);
digitalWrite(52, HIGH);
delay(timer);
digitalWrite(52, LOW);
delay(250);

Serial.println("and 6");
digitalWrite(49, HIGH);     // Valve 6 UP and DOWN
delay(timer);               // in that order.
digitalWrite(49, LOW);
delay(250);
digitalWrite(48, HIGH);
delay(timer);
digitalWrite(48, LOW);
delay(250);
Serial.println("Valves are ready to valve!");
Serial.println(" ");
Serial.println("Enter 'c' to Continue");
while (inByte != 'c'){
    if (Serial.available() > 0) {
        inByte = Serial.read();
    }
    delay(10);}
}
/*****/

/***** State One Vent Function *****/
void State_One_Vent(){
    digitalWrite(23, HIGH);  // V1 Up
    delay(timer);

```

```

digitalWrite(23, LOW);
delay(2);

//digitalWrite(24, HIGH);    //V2 Up      Should be unchanged from state 1, 2, or 4
//delay(timer);
//digitalWrite(24, LOW);
//delay(2);

//digitalWrite(50, HIGH);    //V3 Down    Should be unchanged from state 1, or 2
//delay(timer);
//digitalWrite(50, LOW);
//delay(2);

//digitalWrite(47, HIGH);    //V4 Up      Should be unchanged from any state
//delay(timer);
//digitalWrite(47, LOW);
//delay(2);

//digitalWrite(53, HIGH);    //V5 Up
//delay(timer);
//digitalWrite(53, LOW);
//delay(2);

//digitalWrite(48, HIGH);    //V6 Down    Does not chance from State 2
//delay(timer);
//digitalWrite(48, LOW);
//delay(2);
}
/*****/

/***** Vent Function *****/
void Vent(){
    // This vent function is for state 2 & 4
    digitalWrite(23, HIGH);    // V1 Up
    delay(timer);
    digitalWrite(23, LOW);
    delay(2);

    //digitalWrite(24, HIGH);    // V2 Up      Should be unchanged from state 2 or 4
    //delay(timer);
    //digitalWrite(24, LOW);
    //delay(2);

    //digitalWrite(50, HIGH);    // V3 Down    Should be unchanged from state 2 or 4
    //delay(timer);
    //digitalWrite(50, LOW);
    //delay(2);

    //digitalWrite(47, HIGH);    // V4 Up      Should be unchanged from any state
    //delay(timer);
    //digitalWrite(47, LOW);
    //delay(2);

    digitalWrite(53, HIGH);    // V5 Up
    delay(timer);
    digitalWrite(53, LOW);
    delay(2);

    //digitalWrite(48, HIGH);    //V6 Down    Does not chance from State 2
    //delay(timer);
    //digitalWrite(48, LOW);
    //delay(2);
}
/*****/

/***** Valve Configuration for Home Pump *****/
void Valve_Config_Home_Pump(){
    digitalWrite(22, HIGH);    //V1 Down
    delay(timer);
    digitalWrite(22, LOW);
    delay(2);
}

```

```

digitalWrite(24, HIGH); //V2 Up
delay(timer);
digitalWrite(24, LOW);
delay(2);

digitalWrite(51, HIGH); //V3 Up Lid should be open
delay(timer);
digitalWrite(51, LOW);
delay(2);

digitalWrite(46, HIGH); //V4 Down
delay(timer);
digitalWrite(46, LOW);
delay(2);}
/*****

/***** Valve Configuration for Chip Insertion *****/
void Valve_Config_Chip_Ins(){
    digitalWrite(23, HIGH); //V1 Up
    delay(timer);
    digitalWrite(23, LOW);
    delay(2);

    digitalWrite(25, HIGH); //V2 Down
    delay(timer);
    digitalWrite(25, LOW);
    delay(2);

    digitalWrite(50, HIGH); //V3 Down
    delay(timer);
    digitalWrite(50, LOW);
    delay(2);

    digitalWrite(53, HIGH); //V5 Up
    delay(timer);
    digitalWrite(53, LOW);
    delay(2);

    digitalWrite(49, HIGH); //V6 Up
    delay(timer);
    digitalWrite(49, LOW);
    delay(2);}
/*****

/***** State One *****/
void State_One(){
    digitalWrite(22, HIGH); //V1 Down
    delay(timer);
    digitalWrite(22, LOW);
    delay(2);

    digitalWrite(24, HIGH); //V2 Up
    delay(timer);
    digitalWrite(24, LOW);
    delay(2);

    digitalWrite(51, HIGH); //V3 Up
    delay(timer);
    digitalWrite(51, LOW);
    delay(2);

    digitalWrite(46, HIGH); //V4 Down
    delay(timer);
    digitalWrite(46, LOW);
    delay(2);

    digitalWrite(52, HIGH); //V5 Down
    delay(timer);
    digitalWrite(52, LOW);
    delay(2);

```

```

    digitalWrite(49, HIGH);    //V6 Up
    delay(timer);
    digitalWrite(49, LOW);
    delay(2);}
/*****

/***** State Two *****/
void State_Two(){
    digitalWrite(22, HIGH);    //V1 Down    Changes from 'Vent'
    delay(timer);
    digitalWrite(22, LOW);
    delay(2);

    digitalWrite(24, HIGH);    //V2 Up
    delay(timer);
    digitalWrite(24, LOW);
    delay(2);

    digitalWrite(50, HIGH);    //V3 Down
    delay(timer);
    digitalWrite(50, LOW);
    delay(2);

    digitalWrite(47, HIGH);    //V4 Up
    delay(timer);
    digitalWrite(47, LOW);
    delay(2);

    digitalWrite(52, HIGH);    //V5 Down    Changes from 'Vent'
    delay(timer);
    digitalWrite(52, LOW);
    delay(2);

    digitalWrite(48, HIGH);    //V6 Down
    delay(timer);
    digitalWrite(48, LOW);
    delay(2);}
/*****

/***** State Four *****/
void State_Four(){
    digitalWrite(22, HIGH);    //V1 Down
    delay(timer);
    digitalWrite(22, LOW);
    delay(2);

    digitalWrite(25, HIGH);    //V2 Down
    delay(timer);
    digitalWrite(25, LOW);
    delay(2);

    digitalWrite(51, HIGH);    //V3 Up    Creates smaller back volume
    delay(timer);
    digitalWrite(51, LOW);
    delay(2);

    digitalWrite(47, HIGH);    //V4 Up
    delay(timer);
    digitalWrite(47, LOW);
    delay(2);

    digitalWrite(52, HIGH);    //V5 Down
    delay(timer);
    digitalWrite(52, LOW);
    delay(2);
}
/*****

```

6.6 Device MATLAB Functions

6.6.1 Matlab_Arduino

```
function Matlab_Arduino
clear all
clc
s=serial('COM3');
set(s,'BaudRate',9600);
fopen(s); %opens the serial port
pause(1); %allow time for the arduino to send text lines
z = 1;
k = 1;
filterquestion = 0; % 1=yes filter, 2=no filter,
check = 0; % used in Home_Pump Arduino function.

while (strcmp(k,'end') == 0);
    while (s.BytesAvailable == 0); % wait for message from Arduino to be
        pause(0.01) % sent
    end
    while (s.BytesAvailable > 0); % Prints the message sent from the
        fprintf(fscanf(s)); % Arduino's "s_For_Ready()"
        pause(0.01); % function.
    end
    while (strcmp(z,'s') == 0 && strcmp(z,'end') == 0) % Waits for user to
        z=input('','s'); % enter 's' or 'end'.
    end
    if strcmp(z,'s') == 1 % If user entered 's', send 's' char to
        fprintf(s,z); % Arduino over serial.
    elseif strcmp(z,'end') == 1 % If 'end' was entered, break will
        break % exit out of while loop and stop com.
    end
    while (s.BytesAvailable == 0); % Wait for 'Wake up" function to start
        pause(0.01) % printing messages.
    end
    fprintf(fscanf(s,'%c',17)); % Prints "Cycling Valves 1,".
    pause(0.6)
    for x = 1:4 % Prints 2 through 5.
        fprintf(fscanf(s,'%c',3));
        pause(0.55)
    end
    fprintf(fscanf(s,'%c',6)); % Prints "and 6".
    pause(0.6)
    while (s.BytesAvailable > 0); % Loops and prints until buffer
        fprintf(fscanf(s)); % is empty.
        pause(0.1);
    end
    z=1;
    while (strcmp(z,'c') == 0 && strcmp(z,'end') == 0) % Waits for user to
        z=input('','s'); % enter 'c' or 'end'.
    end
    if strcmp(z,'c') == 1 % If user entered 'c', send 'c' char to
        fprintf(s,z); % Arduino over serial.
    elseif strcmp(z,'end') == 1 % If 'end' was entered, break will
        break % exit out of while loop and stop com.
    end
end

%%% Home_Pump section %%%
while strcmp(check,'Pump Has Found HOME')==0 % Loops until final message in Home_Pump
    % function has been printed.
    while (s.BytesAvailable == 0); % Wait for message from Arduino
        pause(0.01) % to be sent.
    end
    vv=fscanf(s);
    check=regexprep(vv,'\r\n|\n|\r',''); %Replaces newline with blank in string 'vv'.
    if strcmp(check,'Pump Has Found HOME')==1
        break
    else
end
```

```

        fprintf(vv)
    end
end

%%% y_For_Chip %%%
pause(0.5)
while (s.BytesAvailable > 0);    % Loops and prints until buffer
    fprintf(fscanf(s));        % is empty.
    pause(0.01);
end
z=1;
while (strcmp(z,'y') == 0 && strcmp(z,'end') == 0)    % Waits for user to
    z=input('','s');                                % enter 'y' or 'end'.
end
if strcmp(z,'y') == 1                                % If user entered 'y', send 'y' char to
    fprintf(s,z);                                    % Arduino over serial.
elseif strcmp(z,'end') == 1                          % If 'end' was entered, break will
    break                                             % exit out of while loop and stop com.
end
%%% Filter_Question
while (s.BytesAvailable == 0);    % Wait for message from Arduino
    pause(0.01)                  % to be sent.
end
while (s.BytesAvailable > 0);    % Loops and prints until buffer
    fprintf(fscanf(s));        % is empty.
    pause(0.01);
end
z=1;
while (strcmp(z,'y') == 0 && strcmp(z,'n') == 0 && strcmp(z,'end') == 0)
    z=input('','s');            % Waits for user to enter 'y', 'n', or 'end'.
end
if strcmp(z,'y') == 1
    fprintf(s,z);              % If user entered 'y' send char to Arduino.
    filterquestion = 1;        % Yes filter.
elseif strcmp(z,'n') == 1
    fprintf(s,z);              % If user entered 'n' send char to Arduino.
    filterquestion = 2;        % No filter.
elseif strcmp(z,'end') == 1
    break                      % If 'end' was entered, break will
                                % exit out of while loop and stop com.
end
%%% Start_Test
while (s.BytesAvailable == 0);    % Wait for message from Arduino to be
    pause(0.01)                  % to be sent.
end
while (s.BytesAvailable > 0);    % Loops and prints until buffer
    fprintf(fscanf(s));        % is empty.
    pause(0.01);
end
z=1;
while (strcmp(z,'y') == 0 && strcmp(z,'end') == 0)
    z=input('','s');            % Waits for user to enter 'y' or 'end'.
end
if strcmp(z,'y') == 1
    fprintf(s,z);
elseif strcmp(z,'end') == 1
    break                      % If 'end' was entered, break will
                                % exit out of while loop and stop com.
end
while (s.BytesAvailable == 0);    % Wait for message from Arduino to be
    pause(0.01)                  % to be sent.
end
%x = 1;
for x = 1:5;
    fprintf(fscanf(s));          % Loops 5 times to print last message in
    pause(0.01);                % "Start_Test" function, "Test Has Commenced".
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%---Initialize camera and associated variables---%
obj = videoinput('winvideo',1,'Y800_640x480');    % Connect to camera.

```

```

    preview(obj)
    disp('10 seconds to settle camera')
    pause(10) % Allow camera to settle.
    threshold = 15;
    background = 1;
    obj = 1;
    background = getsnapshot(obj); %Take background image.
    imwrite(background, 'images/background.jpg', 'jpg');
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

fprintf(s,'y'); % Signal to Arduino to continue with "First Step Function".

if filterquestion == 1
    %% "First_Step" function
    % State_One if filterquestion = 1 (filter).
    channelFlow(120,441,400,9,1,threshold,background,obj)
    % The first step would be to look for fluid in the plasma channel
    % coming in from the separation membrane.
end

    % State_Two if filterquestion = 2 (no filter)
    disp('Look for fluid coming into HAPR_1 site')
    channelFlow(121,394,400,9,2,threshold,background,obj)
    % Look for fluid coming into the first HAPD site.
    fprintf(s,'a');
    % Once channelFlow is done, fluid is in first HAPR site. Send 'a' to
    % Arduino to confirm that plasma is in first HAPR site. 'case 'a' in
    % Arduino code is triggered.

Inc_Timer(300,'HAPR1') % Incubation time of 5 min.

fprintf(s,'h'); % Signal to Arduino to continue with case 'a'.

disp('Look for fluid coming into HAPR_2 site')
channelFlow(121,329,400,9,3,threshold,background,obj)
    % Look for fluid coming into the second HAPR site
    fprintf(s,'b');
    % Once channelFlow is done, fluid is in second HAPR site. Send 'b' to
    % Arduino to trigger 'case 'b''.
Inc_Timer(300,'HAPR2') % Incubation time of 5 min.

fprintf(s,'i'); % Signal to Arduino to continue with case 'b'.

disp('Look for plasma coming into detection site')
detectionFlow(330,292,250,9,60,12,250,9,4,threshold,background,obj,1)
    % Look for plasma coming into the detection site
    fprintf(s,'c');
    % plasma is in detection site. Send 'c' to Arduino to trigger 'case 'c''
Inc_Timer(420,'det')
    % Incubation time of 7 min.
    fprintf(s,'j');
    % Signal to Arduino to continue with case 'c'
    disp('trigger when air has filled the detection site')
    detectionFlow(330,292,250,9,60,12,250,9,4,threshold,background,obj,0)
    % Will trigger when air has filled the detection site (plasma is out)

disp('Look for wash_1 coming into detection site')
detectionFlow(330,292,250,9,60,12,250,9,5,threshold,background,obj,1)
    % Look for wash 1 to enter the detection site
    fprintf(s,'d');
    % wash 1 is in detection site. Send 'd' to Arduino to trigger 'case 'd''
%Spec reading 1
while (s.BytesAvailable == 0); % wait for message from Arduino to be
    pause(0.01) % sent
end
while (s.BytesAvailable > 0); % Loops and prints until buffer
    fprintf(fscanf(s)); % is empty
    pause(0.01);

```

```

end
    %6 line message from "Spectrometer Reading_1" Function, then it waits
    %for y to be received before turning on the ex LED for 3 seconds
z=1;
while (strcmp(z,'y') == 0 && strcmp(z,'end') == 0)
    z=input('','s');          % waits for user to enter 'y' or 'end'
end
if strcmp(z,'y') == 1
    fprintf(s,z);            % if user entered 'y', send char to Arduino
elseif strcmp(z,'end') == 1 % 'end' was entered, break will
    break                    % exit out of while loop and stop com.
end
pause(6); %time for ex LED to pause for 3 and be on for 3 sec
disp('trigger when air has filled the detection site (wash 1 is out)')
detectionFlow(330,292,250,9,60,12,250,9,5,threshold,background,obj,0)
    % Will trigger when air has filled the detection site (wash 1 is out)

disp('Look for det_ab to enter the detection site')
detectionFlow(330,292,250,9,60,12,250,9,6,threshold,background,obj,1)
    % Look for det_ab to enter the detection site
fprintf(s,'e');
    % det_ab is in detection site. Send 'e' to Arduino to trigger 'case 'e''
Inc_Timer(10,'det_ab')
    % Incubation time of 5 min.
fprintf(s,'k');
    % Signal to Arduino to continue with case 'e'
disp('trigger when air has filled the detection site(det_ab is out)')
detectionFlow(330,292,250,9,60,12,250,9,6,threshold,background,obj,0)
    % Will trigger when air has filled the detection site (det_ab is out)

disp('Look for PBST to enter the detection site')
detectionFlow(330,292,250,9,60,12,250,9,7,threshold,background,obj,1)
    % Look for PBST to enter the detection site
fprintf(s,'f');
    % PBST is in detection site. Send 'f' to Arduino to trigger 'case 'f''
Inc_Timer(10,'PBST')
    % Incubation time of 4 min.
fprintf(s,'l'); % lower case L
    % Signal to arduino to continue with case 'f'
disp('trigger when air has filled the detection site (PBST is out)')
detectionFlow(330,292,250,9,60,12,250,9,7,threshold,background,obj,0)
    % Will trigger when air has filled the detection site (PBST is out)

disp('Look for wash_2 to enter the detection site')
detectionFlow(330,292,250,9,60,12,250,9,8,threshold,background,obj,1)
    % Look for wash 2 to enter the detection site
fprintf(s,'g');
    % wash 2 is in detection site. Send 'g' to Arduino to trigger 'case 'g''
%Spec reading 2
while (s.BytesAvailable == 0); % wait for message from Arduino to be
    pause(0.01)                % sent
end
while (s.BytesAvailable > 0); % Loops and prints until buffer
    fprintf(fscanf(s));        % is empty
    pause(0.01);
end
    %6 line message from "Spectrometer Reading_2" Function, then it waits
    %for y to be received before turning on the ex LED for 3 seconds.
z=1;
while (strcmp(z,'y') == 0 && strcmp(z,'end') == 0)
    z=input('','s');          % waits for user to enter 'y' or 'end'
end
if strcmp(z,'y') == 1
    fprintf(s,z);            % if user entered 'y', send char to Arduino
elseif strcmp(z,'end') == 1 % 'end' was entered, break will
    break                    % exit out of while loop and stop com.
end
pause(6); %time for ex LED to pause for 3 and be on for 3 sec

```



```

while (s.BytesAvailable == 0);    % wait for message from Arduino to be
    pause(0.01)                  % sent
end
while (s.BytesAvailable > 0);    % Loops and prints until buffer
    fprintf(fscanf(s));          % is empty
    pause(0.01);
end

k = 'end';
end
fclose(s); %close the serial port
delete(s);
clear s;
disp('Communication with Unit has been Severed');

```

6.6.2 channelFlow

```

function channelFlow(x,y,w,h,cycle,cutoff,background,obj)
%channelFlow(120,441,400,9,1,threshold,background,obj)
%x is row, y is column, w is width, h is height

check = 1; %Start loop

if cycle==1
    location = 'plasma';
elseif cycle==2
    location = 'HAPR_1';
elseif cycle==3
    location = 'HAPR_2';
end

counter = 1;          %counter for images
msg_length = 0;

while(check==1)

    ssl = getsnapshot(obj);
    diff = background-ssl;
    bm = binaryMask(x,y,w,h,diff,cutoff);
    imwrite(bm, ['images/cf_' num2str(location) num2str(counter) '.jpg'], 'jpg');

    differenceCalculations = sumDifferences(x,y,w,h,bm);
    %differenceCalculations(1)=white (2)=black

    if differenceCalculations(1)>differenceCalculations(2) %more whitepixels
        fprintf(repmat('\b',1,msg_length)); %delete line before
        fprintf('There is a change in the state of the ROI for %s\n',location)
        check = 0; %Exit while loop
    else
        fprintf(repmat('\b',1,msg_length)); %delete line before
        msg = sprintf('There is no change at %s',location);
        fprintf(msg)
        msg_length = numel(msg);
    end

    counter = counter+1;

end
fprintf('\n')
end

```

6.6.3 binaryMask

```
function d = binaryMask(x,y,w,h,diff,cutoff)
%row,column,width,height,image,threshold

for i =y:(y+h) %row start y +column
    for j=x:(x+w) %column starting x + width
        if diff(i,j)>cutoff
            diff(i,j)=255;
        else
            diff(i,j)=0;
        end
    end
end
imshow(diff)
%imwrite(diff,'filename','bmp')
d=diff;

end
```

6.6.4 sumDifferences

```
function difference = sumDifferences(x,y,w,h,img)

whiteCounter=0;
blackCounter=0;
%Top of the detection site
for i = y:(y+h)
    for j=x:(x+w)
        if img(i,j)==255
            whiteCounter=whiteCounter+1;
        else
            blackCounter=blackCounter+1;
        end
    end
end

difference(1)=whiteCounter;
difference(2)=blackCounter;
end
```

6.6.5 Inc_Timer

```
function Inc_Timer(seconds,step)
time = seconds;
msg_length = 0;
switch step
    case 'HAPR1'
        fprintf('Plasma incubating in first HAPR site\r')
    case 'HAPR2'
        fprintf('Plasma incubating in second HAPR site\r')
    case 'det'
        fprintf('Plasma incubating in Detection site\r')
    case 'wash1'
```

```

        fprintf('First Wash in Detection site\r')
    case 'det_ab'
        fprintf('Detection Antibody incubating in Detection site\r')
    case 'PBST'
        fprintf('PBST incubating in Detection site\r')
    case 'wash2'
        fprintf('Second Wash in Detection site\r')
    otherwise
        fprintf('Error, enter correct step\r')
    end
    for i = 1:seconds+1
        Min = floor(time/60);
        Sec = rem(time,60);
        fprintf(repmat('\b',1,msg_length)); %delete line before printing new line
        msg = sprintf('Time Remaining = %0.0f Mins %0.0f Secs',Min,Sec);
        fprintf(msg)
        msg_length = numel(msg);
        pause(1)
        time = time - 1;
    end
    fprintf('\n')
    fprintf('Incubation time is over\n')
end

```

6.6.6 detectionFlow

```

function detectionFlow(x,y,w,h,x1,y1,w1,h1,cycle,cutoff,background,obj,lookingfor)
% "lookingfor" - 1 = looking for liquid, 0 = looking for air
if cycle==4
    location='Plasma';
elseif cycle==5
    location = 'Wash 1';
elseif cycle==6
    location = 'det_ab';
elseif cycle==7
    location = 'PBST';
elseif cycle==8
    location = 'Wash 2';
end

counter = 1;
msg_length = 0;
check = 1;
fprintf('Looking at the Detection Site for %s\n',location)

while(check == 1)
    detectionsite = 0;
    ssl=getsnapshot(obj);
    diff=background-ssl;
    bm=binaryMask(x,y,w,h,diff,cutoff);
    differenceCalculations = sumDifferences(x,y,w,h,bm);
    bm=binaryMask(x1,y1,w1,h1,diff,cutoff);
    differenceCalculations1=sumDifferences(x1,y1,w1,h1,bm);
    imwrite(bm, ['images/cf_' num2str(location) num2str(counter) '.jpg'], 'jpg');

    if lookingfor == 1 % Looking for liquid to enter
        if differenceCalculations(1)>differenceCalculations(2) %more whitepixels
            detectionsite = detectionsite+1;
        end

        if differenceCalculations1(1)>differenceCalculations1(2)
            detectionsite=detectionsite+1;
        end

        if detectionsite == 2

```

```

        fprintf(repmat('\b',1,msg_length)); %delete line before
        fprintf('There is a change at ROI to liquid')
        check = 0; % Exit while loop
    elseif detectionsite == 1
        fprintf(repmat('\b',1,msg_length)); %delete line before
        msg = sprintf('Only one ROI registered a change to liquid');
        fprintf(msg)
        msg_length = numel(msg);
    else
        fprintf(repmat('\b',1,msg_length)); %delete line before
        msg = sprintf('No change in ROI to liquid');
        fprintf(msg)
        msg_length = numel(msg);
    end

elseif lookingfor == 0 % Looking for air (liquid to leave)
    if differenceCalculations(1)<differenceCalculations(2)
        detectionsite=detectionsite+1;
    end

    if differenceCalculations1(1)<differenceCalculations1(2)
        detectionsite=detectionsite+1;
    end

    if detectionsite == 2
        fprintf(repmat('\b',1,msg_length)); %delete line before
        fprintf('There is a change at ROI to air')
        check = 0; % Exit while loop
    elseif detectionsite == 1
        fprintf(repmat('\b',1,msg_length)); %delete line before
        msg = sprintf('Only one ROI registered a change to air');
        fprintf(msg)
        msg_length = numel(msg);
    else
        fprintf(repmat('\b',1,msg_length)); %delete line before
        msg = sprintf('No change in ROI to air');
        fprintf(msg)
        msg_length = numel(msg);
    end
end
counter=counter+1;
end
fprintf('\n')
end

```

6.7 SOP Reagent Formulation

Making 1X PBS buffer

PBS comes in 10X solution.

To make One Liter of 1X PBS mix:		
Value	Units	Substance
900	mL	Ultrapure Water
100	mL	10X PBS

Surface Functionalization

Step 1:

BSA comes as a pure powder that must be diluted in 1X PBS to 30 mg/mL and measured out into small aliquots each containing 50 μ L. Make 5 mL of the 30 mg/mL BSA in 1X PBS so that 100 aliquots, each containing 50 μ L, can be made.

To make 5 mL of 30 mg/mL of BSA in 1X PBS mix:		
Value	Units	Substance
150	mg	BSA powder
5	mL	1X PBS

With this solution make 100 (or close to it) aliquots, each containing 50 μ L at 30 mg/mL.

To make 1.5 mg/mL BSA in 1X PBS, take one BSA aliquot vile (50 μ L at 30 mg/mL) and add 1X PBS to create the desired concentration.

To make 1.5 mg/mL of BSA in 1X PBS mix:		
Value	Units	Substance
50	μ L	BSA aliquot (at 30 mg/mL)
950	μ L	1X PBS

Step 2:

Glutaraldehyde comes in 25% concentration. Mix with ultrapure water to get a 0.4% (v/v) glutaraldehyde in ultrapure water. Assume we want to make 10 mL of 0.4% (v/v) glutaraldehyde in ultrapure water.

$$V_1C_1=V_2C_2$$

$$V_1(25\%)=(10\text{ mL})(0.4\%) \rightarrow V_1=0.16\text{ mL of }25\%\text{ glutaraldehyde}$$

To make 10 mL of 0.4% glutaraldehyde in ultrapure water		
Value	Units	Substance
0.16	mL	25% glutaraldehyde
9.84	mL	Ultrapure water

Step 3:

To get Protein A in a concentration of 50 µg/mL, mix 100 µL aliquot at concentration of 500 µg/mL with 1X PBS.

$$V_1C_1=V_2C_2$$

$$(100 \mu\text{L}) (500 \mu\text{g/ml}) = V_2 (50 \mu\text{g/mL}) \rightarrow V_2 = 1000 \mu\text{L}$$

To make 1 mL of 50 µg/mL of Protein A in 1X PBS		
Value	Units	Substance
100	µL	Protein A aliquot (concentration of 500 µg/mL)
900	µL	1X PBS

Storage Buffer

We wish to make 500 mL of storage buffer.

10 mM Tris

0.05% BSA

0.05% Proclin 300

5% glycerol anhydrous

Ultrapure water

Tris comes 1.5 M (M=mole/L). We want to get 10 mM

$$V_1C_1=V_2C_2$$

$$V_1 (1.5 \text{ mole/L}) = (0.5 \text{ L}) (0.01 \text{ mole/L}) \rightarrow V_1 = 3.333 \text{ mL of Tris}$$

Proclin 300 comes 100%

$$V_1C_1=V_2C_2$$

$$V_1 (100\%) = (0.5 \text{ L}) (0.05\%) \rightarrow V_1 = 0.25 \text{ mL of Proclin 300}$$

Glycerol anhydrous

$$V_1C_1=V_2C_2$$

$$V_1 (100\%) = (0.5 \text{ L}) (5\%) \rightarrow V_1 = 25.0 \text{ mL of Glycerol anhydrous}$$

BSA, assume that we will use a 1% solution of BSA in ultrapure water.

$$V_1C_1=V_2C_2$$

$$V_1 (1\%) = (0.5 \text{ L}) (0.05\%) \rightarrow V_1 = 25.0 \text{ mL of 1\% solution of BSA in ultrapure water.}$$

To make 25 mL of 1% BSA in ultrapure water		
Value	Units	Substance
250	mg	BSA powder
25	mL	Ultrapure water

Immunoassay

Step 1:

rIgG must be at a concentration of 200 µg/mL in Protein A/IgG binding buffer. Aliquots of rabbit polyclonal (rIgG) to C-reactive protein contain 20 µL at 1.00 mg/mL

$$V_1C_1=V_2C_2$$

$$(20\mu\text{L}) (1000 \mu\text{g/mL}) = V_2 (200 \mu\text{g/mL}) \rightarrow V_2 = 100 \mu\text{L}$$

To make 100 µL of rIgG at 200 µg/mL in Protein A/IgG binding buffer		
Value	Units	Substance
20	µL	rIgG at 1.00 mg/mL (one aliquot)
80	µL	Protein A/IgG binding buffer

Step 2:

Flush channels with 1X PBS

Step 3:

CRP comes in aliquots at a concentration of 2.13 mg/mL and contain 10 µL. Run the test using different levels of concentration of CRP in PBS. Start by making 200 µg/mL of CRP in PBS and make more diluted samples from that.

$$V_1C_1=V_2C_2$$

$$(10 \mu\text{L}) (2130 \mu\text{g/mL}) = V_2 (200 \mu\text{g/mL}) \rightarrow V_2 = 106.5 \mu\text{L}$$

To make 106.5 µL of CRP at 200 µg/mL in PBS		
Value	Units	Substance
10	µL	CRP at 2.13 mg/mL (one aliquot)
96.5	µL	1X PBS

Step 4:

Flush channels with 1X PBS

Step 5:

gIgG must be at a concentration of 120 µg/mL in PBS. Aliquots of FITC labeled goat polyclonal (gIgG) to C-reactive protein contain 10 µL at 1.00 mg/mL

$$V_1C_1=V_2C_2$$

$$(10\mu\text{L}) (1000 \mu\text{g/mL}) = V_2 (120 \mu\text{g/mL}) \rightarrow V_2 = 83.3333 \mu\text{L}$$

To make 83.33 μ L of gIgG at 120 μ g/mL in PBS		
Value	Units	Substance
10	μ L	gIgG at 1.00 mg/mL (one aliquot)
73.33	μ L	PBS

Step 6:

PBST

We want 0.5 μ L/mL of Tween 20 in 1x PBS. Say we want to make 40 mL of PBST

$$V_1 C_1 = V_2 C_2$$

$$V_1 (1000 \mu\text{L/mL}) = (40 \text{ mL}) (0.5 \mu\text{L/mL}) \rightarrow V_1 = 0.02 \text{ mL} = 20 \mu\text{L of Tween 20}$$

To make 40 mL of PBST Mix:		
Value	Units	Substance
0.02	mL	Tween 20
39.98	mL	1X PBS

BIBLIOGRAPHY

- [1] Apps.who.int, 'WHO Region of the Americas: United States of America - statistics summary (2002 - present)', 2014. [Online]. Available:
<http://apps.who.int/gho/data/?theme=country&vid=20800>. [Accessed: 25-Jun- 2014].
- [2] S. Moon, H. Keles, A. Ozcan, A. Khademhosseini, E. Hæggstrom, D. Kuritzkes and U. Demirci, 'Integrating microfluidics and lensless imaging for point-of-care testing', *Biosensors and Bioelectronics*, vol 24, iss 11, pp. 3208-3214, 2009.
- [3] Wikipedia, 'Medical diagnosis', 2014. [Online]. Available:
http://en.wikipedia.org/wiki/Medical_diagnosis. [Accessed: 25- Jun- 2014].
- [4] N. Holtzman, 'Predictive Genetic Testing: From Basic Research to Clinical Practice', *Science*, vol 278, iss 5338, pp. 602-605, 1997.
- [5] Inchem.org, 'Biomarkers In Risk Assessment: Validity And Validation (EHC 222, 2001)', 2014. [Online]. Available:
<http://www.inchem.org/documents/ehc/ehc/ehc222.htm>. [Accessed: 25- Jun- 2014].
- [6] K. Strimbu and J. Tavel, 'What are biomarkers?', *Current Opinion in HIV and AIDS*, vol 5, iss 6, pp. 463-466, 2010.
- [7] J. Laurén, D. Gimbel, H. Nygaard, J. Gilbert and S. Strittmatter, 'Cellular prion protein mediates impairment of synaptic plasticity by amyloid- β oligomers', *Nature*, vol 457, iss 7233, pp. 1128-1132, 2009.
- [8] A. Nikolaev, T. McLaughlin, D. O'Leary and M. Tessier-Lavigne, 'APP binds DR6 to trigger axon pruning and neuron death via distinct caspases', *Nature*, vol 457, iss 7232, pp. 981-989, 2009.
- [9] M. Pepys and G. Hirschfield, 'C-reactive protein: a critical update', *Journal of Clinical Investigation*, vol 111, iss 12, pp. 1805-1812, 2003.

- [10] W. Koenig, M. Sund, M. Frohlich, H. Fischer, H. Lowel, A. Doring, W. Hutchinson and M. Pepys, 'C-Reactive Protein, a Sensitive Marker of Inflammation, Predicts Future Risk of Coronary Heart Disease in Initially Healthy Middle-Aged Men : Results From the MONICA (Monitoring Trends and Determinants in Cardiovascular Disease) Augsburg Cohort Study, 1984 to 1992', *Circulation*, vol 99, iss 2, pp. 237-242, 1999.
- [11] H. Liao, J. Wu, E. Kuhn, W. Chin, B. Chang, M. Jones, S. O'Neil, K. Clauser, J. Karl, F. Hasler, R. Roubenoff, W. Zolg and B. Guild, 'Use of mass spectrometry to identify protein biomarkers of disease severity in the synovial fluid and serum of patients with rheumatoid arthritis', *Arthritis & Rheumatism*, vol 50, iss 12, pp. 3792-3803, 2004.
- [12] F. Gerber, M. Krummen, H. Potgeter, A. Roth, C. Siffrin and C. Spöndlin, 'Practical aspects of fast reversed-phase high-performance liquid chromatography using 3µm particle packed columns and monolithic columns in pharmaceutical development and production working under current good manufacturing practice', *Journal of Chromatography A*, vol 1036, iss 2, pp. 127-133, 2004.
- [13] C. Yao, A. Williams, A. Ottens, X. May Lu, R. Chen, K. Wang, R. Hayes, F. Tortella and J. Dave, 'Detection of protein biomarkers using high-throughput immunoblotting following focal ischemic or penetrating ballistic-like brain injuries in rats', *Brain Injury*, vol 22, iss 10, pp. 723-732, 2008.
- [14] W. Xia, T. Yang, G. Shankar, I. Smith, Y. Shen, D. Walsh and D. Selkoe, 'A Specific Enzyme-Linked Immunosorbent Assay for Measuring β -Amyloid Protein Oligomers in Human Plasma and Brain Tissue of Patients With Alzheimer Disease', *Archives of Neurology*, vol 66, iss 2, 2009.
- [15] D. Alamdari, E. Kostidou, K. Paletas, M. Sarigianni, A. Konatas, A. Karapiperidou and G. Koliakos, 'High sensitivity enzyme-linked immunosorbent assay (ELISA) method for measuring protein carbonyl in

samples with low amounts of protein', *Free Radical Biology and Medicine*, vol 39, iss 10, pp. 1362-1367, 2005.

- [16] R. Lequin, 'Enzyme Immunoassay (EIA)/Enzyme-Linked Immunosorbent Assay (ELISA)', *Clinical Chemistry*, vol 51, iss 12, pp. 2415-2418, 2005.
- [17] Wikipedia, 'High-performance liquid chromatography', 2014. [Online]. Available: <http://en.wikipedia.org/wiki/HPLC>. [Accessed: 26- Jun- 2014].
- [18] Wikipedia, 'Protein mass spectrometry', 2014. [Online]. Available: http://en.wikipedia.org/wiki/Protein_mass_spectrometry. [Accessed: 26- Jun- 2014].
- [19] A. Snijders, M. de Vos and P. Wright, 'Novel Approach for Peptide Quantitation and Sequencing Based on 15 N and 13 C Metabolic Labeling', *J. Proteome Res.*, vol 4, iss 2, pp. 578-585, 2005.
- [20] Wikipedia, 'Western blot', 2014. [Online]. Available: http://en.wikipedia.org/wiki/Western_blot. [Accessed: 26- Jun- 2014].
- [21] Wikipedia, 'ELISA', 2014. [Online]. Available: <http://en.wikipedia.org/wiki/ELISA>. [Accessed: 26- Jun- 2014].
- [22] A. Yetisen, M. Akram and C. Lowe, 'Paper-based microfluidic point-of-care diagnostic devices', *Lab on a Chip*, vol 13, iss 12, p. 2210, 2013.
- [23] M. Godfrin, "Development of a System for the Rapid Detection of Protein Biomarkers in Whole Human Blood," M.S. thesis, Dept. Mech. Eng., Univ. of Rhode Island, Kingston, 2011.
- [24] N. M. Difilippo, "Smartphone Integration and Fluid Flow Control in a Hand-Held Microfluidic Lab-on-Chip Biosensor," M.S. thesis, Dept. Mech. Eng., Univ. of Rhode Island, Kingston, 2012.

- [25] J. D. Jones, "A Hand-Held Microfluidic Fluorescence Lab-on-Chip Biological Detection System," Ph.D. dissertation, Dept. Mech. Eng., Univ. of Rhode Island, Kingston, 2012.
- [26] P. Li, A. Abolmaaty, C. D'Amore, S. Demming, C. Anagnostopoulos, M. Faghri. "Development of an ultrafast quantitative heterogeneous immunoassay on pre-functionalized poly(dimethylsiloxane) microfluidic chips for the next-generation immunosensors." *Microfluid Nanofluid* 7 (2009): 593-598. Web.
- [27] R. Gorkin, J. Park, J. Siegrist, M. Amasia, B. Lee, J. Park, J. Kim, H. Kim, M. Madou and Y. Cho, 'Centrifugal microfluidics for biomedical applications', *Lab on a Chip*, vol 10, iss 14, p. 1758, 2010.
- [28] B. Lee, Y. Lee, H. Kim, T. Kim, J. Park, J. Lee, J. Kim, H. Kim, W. Lee and Y. Cho, 'Fully integrated lab-on-a-disc for simultaneous analysis of biochemistry and immunoassay from whole blood', *Lab on a Chip*, vol 11, iss 1, p. 70, 2011.
- [29] M. Amasia and M. Madou, 'Large-volume centrifugal microfluidic device for blood plasma separation', *Bioanalysis*, vol 2, iss 10, pp. 1701-1710, 2010.
- [30] J. Park, Y. Cho, B. Lee, J. Lee and C. Ko, 'Multifunctional microvalves control by optical illumination on nanoheaters and its application in centrifugal microfluidic devices', *Lab on a Chip*, vol 7, iss 5, p. 557, 2007.
- [31] C. Huang, P. Li, C. Pai, T. Leu and C. Jen, 'Design and simulation of a microfluidic blood-plasma separation chip using microchannel structures', *Separation Science and Technology*, vol 45, iss 1, pp. 42--49, 2009.
- [32] S. Yang, A. Undar and J. Zahn, 'A microfluidic device for continuous, real time blood plasma separation', *Lab on a Chip*, vol 6, iss 7, pp. 871--880, 2006.
- [33] M. Kersaudy-Kerhoas, R. Dhariwal, M. Desmulliez and L. Jouvet, 'Hydrodynamic blood plasma separation in microfluidic channels', *Microfluidics and nanofluidics*, vol 8, iss 1, pp. 105--114, 2010.

- [34] Wikipedia, 'Dielectrophoresis', 2014. [Online]. Available:
http://en.wikipedia.org/wiki/Dielectrophoresis#Dielectrophoresis_Field-Flow_Fractionation_.28DEP-FFF.29. [Accessed: 02- Jul- 2014].
- [35] Y. Nakashima, S. Hata and T. Yasuda, 'Blood plasma separation and extraction from a minute amount of blood using dielectrophoretic and capillary forces', *Sensors and Actuators B: Chemical*, vol 145, iss 1, pp. 561-569, 2010.
- [36] Wikipedia, 'Red blood cell', 2014. [Online]. Available:
http://en.wikipedia.org/wiki/Red_blood_cell. [Accessed: 02- Jul- 2014].
- [37] Wikipedia, 'White blood cell', 2014. [Online]. Available:
http://en.wikipedia.org/wiki/White_blood_cell. [Accessed: 02- Jul- 2014].
- [38] M. Onishi, K. Shimura, Y. Seita and S. Yamashita, 'Design of a new plasma separation membrane by graft copolymerization', *Radiation Physics and Chemistry*, vol 46, iss 2, pp. 219-223, 1995.
- [39] T. Yamaguchi, S. Nakao and S. Kimura, 'Design of pervaporation membrane for organic-liquid separation based on solubility control by plasma-graft filling polymerization technique', *Industrial & Engineering Chemistry Research*, vol 32, iss 5, pp. 848-853, 1993.
- [40] Pall Corporation, 'Plasma Separation', US8440085 B2013.
- [41] Pall Corporation, 'Vivid Plasma Separation Membrane, Product Data', 2009.
[Online]. Available: http://www.pall.com/pdfs/OEM-Materials-and-Devices/09.2730_VividPlasma_DS_6pg.pdf. [Accessed: 02- Jul- 2014].
- [42] J. Adkins, 'Toward a Human Blood Serum Proteome: Analysis By Multidimensional Separation Coupled With Mass Spectrometry', *Molecular & Cellular Proteomics*, vol 1, iss 12, pp. 947-955, 2002.
- [43] C. Burtis, E. Ashwood, B. Border and N. Tietz, *Tietz fundamentals of clinical chemistry*, 1st ed. Philadelphia: W.B. Saunders, 2001.

- [44] R. Fan, O. Vermesh, A. Srivastava, B. Yen, L. Qin, H. Ahmad, G. Kwong, C. Liu, J. Gould, L. Hood and J. Heath, 'Integrated barcode chips for rapid, multiplexed analysis of proteins in microliter quantities of blood', *Nat Biotechnol*, vol 26, iss 12, pp. 1373-1378, 2008.
- [45] L. Qin, O. Vermesh, Q. Shi and J. Heath, 'Self-powered microfluidic chips for multiplexed protein assays from whole blood', *Lab Chip*, vol 9, iss 14, pp. 2016--2020, 2009.
- [46] E. Stern, A. Vacic, N. Rajan, J. Criscione, J. Park, B. Ilic, D. Mooney, M. Reed and T. Fahmy, 'Label-free biomarker detection from whole blood', *Nature nanotechnology*, vol 5, iss 2, pp. 138--142, 2009.
- [47] K. McKenzie, L. Lafleur, B. Lutz and P. Yager, 'Rapid protein depletion from complex samples using a bead-based microfluidic device for the point of care', *Lab Chip*, vol 9, iss 24, pp. 3543--3548, 2009.
- [48] G. Ryu, J. Huang, O. Hofmann, C. Walshe, J. Sze, G. McClean, A. Mosley, S. Rattle, D. Bradley and others, 'Highly sensitive fluorescence detection system for microfluidic lab-on-a-chip', *Lab on a Chip*, vol 11, iss 9, pp. 1664--1670, 2011.
- [49] M. Herrmann, T. Veres and M. Tabrizian, 'Enzymatically-generated fluorescent detection in micro-channels with internal magnetic mixing for the development of parallel microfluidic ELISA', *Lab on a Chip*, vol 6, iss 4, pp. 555--560, 2006.
- [50] D. Mark, S. Haeberle, G. Roth, F. von Stetten and R. Zengerle, 'Microfluidic lab-on-a-chip platforms: requirements, characteristics and applications', *Chem. Soc. Rev.*, vol 39, iss 3, p. 1153, 2010.
- [51] E. Yang, S. Han and S. Yang, 'Fabrication and testing of a pair of passive bivalvular microvalves composed of p+ silicon diaphragms', *Sensors and Actuators A: Physical*, vol 57, iss 1, pp. 75-78, 1996.

- [52] B. Li, Q. Chen, D. Lee, J. Woolman and G. Carman, 'Development of large flow rate, robust, passive micro check valves for compact piezoelectrically actuated pumps', *Sensors and Actuators A: Physical*, vol 117, iss 2, pp. 325-330, 2005.
- [53] D. Bien, S. Mitchell and H. Gamble, 'Fabrication and characterization of a micromachined passive valve', *Journal of Micromechanics and Microengineering*, vol 13, iss 5, pp. 557-562, 2003.
- [54] M. Hu, H. Du, S. Ling, Y. Fu, Q. Chen, L. Chow and B. Li, 'A silicon-on-insulator based micro check valve', *Journal of Micromechanics and Microengineering*, vol 14, iss 3, pp. 382-387, 2004.
- [55] G. Feng and E. Kim, 'Micropump based on PZT unimorph and one-way parylene valves', *Journal of Micromechanics and Microengineering*, vol 14, iss 4, pp. 429-435, 2004.
- [56] A. Wego, S. Richter and L. Pagel, 'Fluidic microsystems based on printed circuit board technology', *Journal of Micromechanics and Microengineering*, vol 11, iss 5, pp. 528-531, 2001.
- [57] S. Böhm, W. Olthuis and P. Bergveld, 'A plastic micropump constructed with conventional techniques and materials', *Sensors and Actuators A: Physical*, vol 77, iss 3, pp. 223-228, 1999.
- [58] S. Santra, P. Holloway and C. Batich, 'Fabrication and testing of a magnetically actuated micropump', *Sensors and Actuators B: Chemical*, vol 87, iss 2, pp. 358-364, 2002.
- [59] M. Carrozza, N. Croce, B. Magnani and P. Dario, 'A piezoelectric-driven stereolithography-fabricated micropump', *Journal of Micromechanics and Microengineering*, vol 5, iss 2, pp. 177-179, 1995.

- [60] T. Pan, S. McDonald, E. Kai and B. Ziaie, 'A magnetically driven PDMS micropump with ball check-valves', *Journal of Micromechanics and Microengineering*, vol 15, iss 5, pp. 1021-1026, 2005.
- [61] C. Ahn, 'Disposable polymer "smart" lab-on-a-chip for point-of-care testing (POCT) in clinical diagnostics', vol 1, pp. 437--440, 2005.
- [62] J. Melin, N. Roxhed, G. Gimenez, P. Griss, W. van der Wijngaart and G. Stemme, 'A liquid-triggered liquid microvalve for on-chip flow control', *Sensors and Actuators B: Chemical*, vol 100, iss 3, pp. 463--468, 2004.
- [63] D. Duffy, H. Gillis, J. Lin, N. Sheppard and G. Kellogg, 'Microfabricated Centrifugal Microfluidic Systems: Characterization and Multiple Enzymatic Assays', *Anal. Chem.*, vol 71, iss 20, pp. 4669-4678, 1999.
- [64] N. Honda, U. Lindberg, P. Andersson, S. Hoffmann and H. Takei, 'Simultaneous multiple immunoassays in a compact disc--shaped microfluidic device based on centrifugal force', *Clinical Chemistry*, vol 51, iss 10, pp. 1955--1961, 2005.
- [65] D. Mark, T. Metz, S. Haeberle, S. Lutz, J. Ducrée, R. Zengerle and F. von Stetten, 'Centrifugo-pneumatic valve for metering of highly wetting liquids on centrifugal microfluidic platforms', *Lab on a Chip*, vol 9, iss 24, p. 3599, 2009.
- [66] J. Ducrée, S. Haeberle, S. Lutz, S. Pausch, F. Stetten and R. Zengerle, 'The centrifugal microfluidic Bio-Disk platform', *Journal of Micromechanics and Microengineering*, vol 17, iss 7, pp. S103-S115, 2007.
- [67] K. Erickson and P. Wilding, 'Evaluation of a novel point-of-care system, the i-STAT portable clinical analyzer.', *Clinical Chemistry*, vol 39, iss 2, pp. 283--287, 1993.
- [68] S. Shoji and M. Esashi, 'Microflow devices and systems', *Journal of Micromechanics and Microengineering*, vol 4, iss 4, pp. 157-171, 1994.

- [69] D. Laser and J. Santiago, 'A review of micropumps', *Journal of Micromechanics and Microengineering*, vol 14, iss 6, pp. R35-R64, 2004.
- [70] D. Strickland, J. Ramunas, J. Gonzalez and J. Santiago, 'Electrolytic Micropump Actuator with Controlled Cyclic Bubble Growth and Recombination', iss 45, pp. 2050--2050, 2011.
- [71] J. Portilla and U. Kim, 'Low-power, self-contained, reciprocating micropump through electrolysis and catalyst-driven recombination toward drug delivery applications', pp. 1077--1080, 2013.
- [72] W. Grover, A. Skelley, C. Liu, E. Lagally and R. Mathies, 'Monolithic membrane valves and diaphragm pumps for practical large-scale integration into glass microfluidic devices', *Sensors and Actuators B: Chemical*, vol 89, iss 3, pp. 315--323, 2003.
- [73] S. Haeberle and R. Zengerle, 'Microfluidic platforms for lab-on-a-chip applications', *Lab on a Chip*, vol 7, iss 9, pp. 1094--1110, 2007.
- [74] A. Sauer-Budge, P. Mirer, A. Chatterjee, C. Klapperich, D. Chargin and A. Sharon, 'Low cost and manufacturable complete microTAS for detecting bacteria', *Lab on a Chip*, vol 9, iss 19, pp. 2803--2810, 2009.
- [75] D. Erickson, D. O'Dell, L. Jiang, V. Oncescu, A. Gumus, S. Lee, M. Mancuso and S. Mehta, 'Smartphone technology can be transformative to the deployment of lab-on-chip diagnostics', *Lab on a Chip*, 2014.
- [76] Abbottpointofcare.com, 'CTI Sheets – www.abbottpointofcare.com', 2014. [Online]. Available: <http://www.abbottpointofcare.com/Customer-Info-Center/Cartridge-and-Test-Info-Sheets.aspx>. [Accessed: 09- Jul- 2014].
- [77] Fishersci.com, 'Abbott Point of Care i-STAT 1 Wireless Handheld Point-of-Care Testing', 2014. [Online]. Available: <http://www.fishersci.com/ecom/servlet/fsproductdetail?storeId=10652&productId=14473720>. [Accessed: 09- Jul- 2014].

- [78] Alere.com, 'Medical Testing Meters and Diagnostic Testing Devices from Alere.', 2014. [Online]. Available: <http://www.alere.com/us/en/products/brands/triage.html>. [Accessed: 10- Jul- 2014].
- [79] cliawaived.com, 'Alere Triage BNP Test Product Insert', 2014. [Online]. Available: http://www.cliawaived.com/web/items/pdf/BSTE-98000XR_AlereTriageBNPTest_Insert~3194file1.pdf. [Accessed: 08- Jul- 2014].
- [80] Samsung Village, 'Samsung Technology Digitalizes Medical Care, Delivers Innovations', 2012. [Online]. Available: <http://www.samsungvillage.com/blog/2012/03/15/samsungblog-samsung-technology-digitalizes-medical-care-delivers-innovations/>. [Accessed: 11- Jul- 2014].
- [81] S. Nie, W. Henley, S. Miller, H. Zhang, K. Mayer, P. Dennis, E. Oblath, J. Alarie, Y. Wu, F. Oppenheim, F. Little, A. Uluer, P. Wang, J. Ramsey and D. Walt, 'An automated integrated platform for rapid and sensitive multiplexed protein profiling using human saliva samples', *Lab on a Chip*, vol. 14, no. 6, p. 1087, 2014.
- [82] H. Zhu, I. Sencan, J. Wong, S. Dimitrov, D. Tseng, K. Nagashima and A. Ozcan, 'Cost-effective and rapid blood analysis on a cell-phone', *Lab on a Chip*, vol 13, iss 7, p. 1282, 2013.
- [83] P. Li , "Development of a microfluidic fluorescence immunosensor for point-of-care diagnostics", Ph.D. dissertation, Dept. Mech. Eng., Univ. of Rhode Island, Kingston, 2010.
- [84] M. Fuerstman, A. Lai, M. Thurlow, S. Shevkoplyas, H. Stone and G. Whitesides, 'The pressure drop along rectangular microchannels containing bubbles', *Lab on a Chip*, vol 7, iss 11, p. 1479, 2007.

- [85] Bartels-mikrotechnik.de, 'Bartels Mikrotechnik GmbH - mp6-pp', 2014. [Online].
Available: <http://www.bartels-mikrotechnik.de/content/view/69/118/lang,english/>. [Accessed: 09- Aug- 2014].
- [86] K. Oh and C. Ahn, 'A review of microvalves', Journal of Micromechanics and Microengineering, Vol. 16, no. 5, pág R13-R39, 2006.