Development of Microscope Stage BioStretch Device for Mechanobiology

In partial fulfillment of the requirements for the Degree of

Bachelor of Science

Submitted By: Joshua Boynton, Alexander Cruz,

Megan Hendrie, & Elizabeth Quevillon

04/25/2019

Project # KLB1802

Project Advised by: Professor Kristen Billiar

Table of Contents

Authorship

Key – Joshua Boynton (JB), Alexander Cruz (AC), Megan Hendrie (MH), Elizabeth Quevillon (EQ)

Section	Primary author(s)	Primary Editor(s)
Abstract	**MH**	**MH, AC**
Introduction	**MH**	**MH, AC**
Literature Review	**MH**	**MH, EQ**
Project Strategy	**MH, JB, EQ, AC**	**MH, AC**
Design Process	**AC, JB, EQ**	**MH, AC**
Development of Alternative Designs	**MH, AC**	**MH, AC**
Alternative Device Designs	**MH, AC**	**MH, AC**
Motor Device and New Well Design Testing	**MH, EQ, AC, JB**	**MH, AC**
Final Device Design	**AC, EQ, MH, JB**	**MH, JB**
Future Work	**MH, AC, JB**	**MH, AC, JB**
Appendix A: Additional Background Information	**MH**	**MH**
Appendix B: Material Properties & Safety Data Sheets for Resin Stretch Well Mold	**N/A**	**MH**
Appendix C: Motor Specifications	**AC, EQ**	**MH, AC**
Appendix D: Part of Past MQP Code to Run Motors	**JB**	**MH**
Appendix E: Gantt Charts	**EQ**	**MH**
Appendix F: GUI Code	**MH, JB**	**MH, EQ**
Appendix G: Code to Operate Motors	**JB**	**MH, EQ**
Appendix H: CAD Drawings of Parts	**AC, JB**	**MH, EQ**
Appendix I: Pre-Calculated Strain to Power Values for Specific Frequencies	**MH**	**MH**
Appendix J: Final Device Budget and Part Specifications Sheets	**MH, AC**	**MH**

Acknowledgments

We would like to acknowledge:

- Professor Kristen Billiar, our project advisor
- Zachary Goldblatt, our graduate student advisor
- Professor Adriana Hera for providing training for ANSYS Finite Element Analysis
- Stephanie Silvestris for providing instron training
- Lisa Wall, the Biomedical Engineering Lab Manager
- Andrew Gulotta, for helping machine metal parts

Abstract

The Billiar lab performs research in mechanobiology considering both 2D and 3D models to understand mechanical force mechanisms regulating cell and tissue behaviors, as well as ECM biochemistry, biomechanics, and disease pathogenesis. To study mechanobiology, cells are often plated on elastic membranes and deformed uniaxially or biaxially to demonstrate how mechanical signals regulate cell behavior in research. The devices that currently exist on the market for stretching cells come at a high cost, cannot be custom programmed, and typically lack both uniaxial and biaxial functionality. The primary goal of our project was to develop a precise cell stretching device for biaxial strain application at different magnitudes, rates and patterns to observe short and long-term cellular effects. To develop a custom device with both uniaxial and biaxial stretching functionality, our team designed and tested multiple motor-based cell stretching devices. The final design consists of a low profile, four linear actuator uniaxial and biaxial stretching system which successfully stretches cells at a range of strains, frequencies, and strain patterns over a 24-hour period. Validation testing using computer softwares verified the device functionality and reliability.

Table of Figures

Table of Tables

Chapter 1: Introduction

Cells experience several different mechanical forces in vivo. The cellular response to these mechanical stimuli, termed mechanotransduction, directs cell processes such as growth migration, and proliferation. Previous studies have shown that when cells are stretched, the cell responds by reorienting to avoid the stress and strain. By studying cellular responses to imposed forces, researchers can give insight to normal and pathological cellular behavior. Mechanotransduction research can significantly contribute to the design of advanced biomedical interventions.

Several different cell stretching devices exist on the market for mechanotransduction research. Many of these devices are designed to generate a uniaxial or biaxial strain when applied to a silicone membrane. One such device, the StrexCell STB-190-XY, provides a biaxial strain to polydimethylsiloxane (PDMS) stretching wells using motors that drive threaded rods. This device allows for the testing of cell response to stretching on a benchtop. Considered a gold standard stretching device, the STB 190-XY is purchasable on the market for $20,000-$30,000. In this device and similar gold standard marketed devices, cells are stretched using software that controls testing parameters such as strain, frequency, and stretching pattern. A successful stretching device must, therefore, effectively apply strain at a controlled frequency and stretching pattern.

Previous Major Qualifying Project (MQP) teams had developed a cellular stretching device which has been in use in the Billiar lab since 2013. The device consists of four stepper motors fixed to a steel metal plate, connected by linear slides to create both uniaxial and biaxial deformation. These motors collectively move pins to deform a PDMS well (Duoba et al., 2012).

After studying the device and performing validation testing, we found several device qualities that impacted performance. Notably, the device was not suited to reach the desired number of cycles per second at the desired percent strain. The failure to achieve the necessary strain rate occurred even without a PDMS well in place on the device. The improper size of the steel baseplate caused a poor fit on the microscope and prevented the device from sitting level on the microscope stage. The microscope stage itself was obstructed by the protruding metal pin brackets that held the PDMS wells, hindering the cell imaging field. Additionally, the heavy steel plate and large motors made it more difficult to transport the system between benchtop and incubator. Finally, one computer in the lab was equipped to effectively run the necessary code to move the motors. The computer was not easily portable, and the code could not be transferred from the hard-drive to a newer system. The computer also processed the code inefficiently.

The PDMS wells used in the device were known to fail upon mold removal and during testing. Stress concentrators were identified on finished wells from flashing on the edges which caused material failure. The well failure rate was 1 in 7 wells tested (Garcia et al., 2013).With well and manufacturing modifications, we also determined that the PDMS wells used in the stretching device could be improved.

Based on the *in vivo* implications for the cell stretching research, and the required device specifications set by our client, our objective was to design an automated cellular stretching device to fit a Zeiss imaging microscope and operate at 10% strain and 1 Hz for 24 hours. The design was client-centered and specific for mechanobiological research in the Billiar lab with cardiomyocytes at Worcester Polytechnic Institute.

After designing and testing many different device alternatives, our team developed a low-profile, lightweight, linear actuator-driven uniaxial and biaxial cell stretching device operated by an Arduino controller using custom code. The device operates at 1% strain and 1 Hz frequency to stretch cells under real-time observation. Device settings are feasibly controlled by a custom a GUI system and code that are transferable between different computers containing Arduino software. The new device fits in the Billiar lab Zeiss microscope and the storage incubator for 24-hour cell stretching. Validation testing by computer analysis verified that the final device developed can operate with 0.5% error($p<0.02$) of the validated strain and frequency parameters.

Chapter 2: Literature Review

Section 1: Physiological and Mechanical Context In Vivo

Cells in the human body constantly undergo forces of different magnitudes, strains, and strain patterns. Various tissue types respond to these external stimuli differently eliciting unique responses tailored to the tissue-specific function in the body. The ability to study these responses in vitro under controlled conditions paves the way for biomedical advancements in areas such as biomedical materials science, tissue engineering, regenerative medicine, and drug delivery. The following section outlines the variables that impact the study of mechanobiological stretching in vitro including state of the art market standards for biaxial stretching and past approaches to develop a biaxial stretching device.

Various cell functions are significantly affected by mechanical forces including cell migration, differentiation, apoptosis, morphogenesis, and proliferation. Therefore, the effects of mechanical stretching are consequential to pathophysiological functions such as embryogenesis, tumorigenesis, angiogenesis, tissue remodeling, organogenesis, and homeostasis (Abiko et al., 2015). For instance, connective tissues undergo a range of mechanical loads in vivo that regulate development, remodeling and pathogenesis of different tissues (Balestrini et al., 2010). It becomes important then, to consider the effects of cell stretching and understand outcomes for research in the development of medical devices, materials, or artificial cells, tissues, organs, or biomolecules subjected to these stretching conditions. Research forms the foundation of designed interventions by determining known characteristics, morphologies, pathways and corresponding disease states for future replication or other applications.

It is known that when cells are subjected to stretching forces, the cells reorient to avoid the imposed stress and strain. In laboratories, cells are plated on elastic membranes and deformed uniaxially or equi-biaxially to demonstrate how mechanical signals regulate cell behavior (Balestrini et al., 2010). These types of studies have exhibited biological responses including shape changes of epithelial cells, reorientation of fibroblasts and smooth muscle cells, as well as increased proliferation rates, and modified migration behaviors, and expression and synthesis changes for contractile and regulatory proteins (Balestrini et al., 2010). Moreover, strain anisotropy is an important cell activity regulator (Balestrini et al., 2010). For instance, fibroblasts and smooth muscle cells elongate and align in the direction of the principal strain, with a preferred alignment perpendicularly to the direction of maximal uniaxial and strip biaxial stretching (Balestrini et al., 2010). In addition to strain direction, there is strong evidence suggesting cell synthetic and proliferative activities are regulated by strain magnitudes (Balestrini et al., 2010). Durotaxis, or alteration of migration and proliferation in response to stiffness, has been observed. Figure 2.1 below demonstrates the effects of substrate stiffness on cell traction forces in a 2D in vitro model.

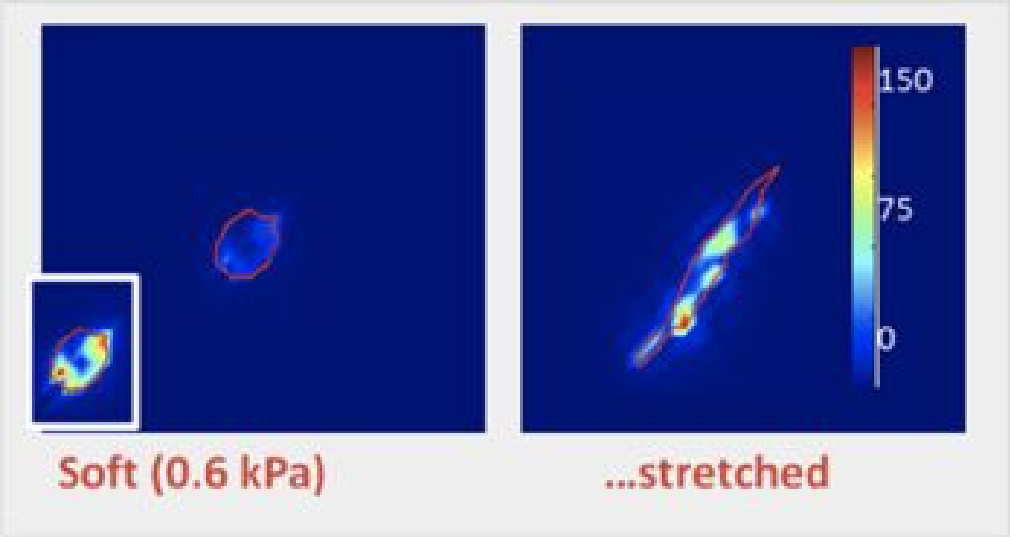


Figure 2.1: Illustration of cell behavior changes with stretching based on cell traction forces on soft substrates in a 2D in vitro model. (Copyright 2016 © Tissue Mechanics & Mechanobiology Lab).

Tenotaxis, or cellular effects due to gradients of stretch anisotropy, is observed by reorientation of fibroblasts with varying degrees of anisotropy related to the degree of global reorientation (Balestrini et al., 2010). In one study by Huang et al., endothelial cells (ECs) were cyclically stretched at a strain of 20% at 0.5 Hz. Stretching at a constant strain for 30-60 minutes, resulted in observed reorientation of the cell's actin filaments increasingly perpendicular to the direction of the stretching force (Huang et al., 2012). In another example, mouse embryonic fibroblasts underwent cyclic uniaxial stretching which resulted in induced strain rate dependent stress fiber alignment which increased with increasing strain rates. Once again, the stretched cells responded to stretching by reorienting their actin filaments and by activating intracellular signaling proteins. This study by Hsu et al. concluded that the frequency of variation in strain rate altered fiber alignment in aortic endothelial cells (Hsu et al., 2010).

Studies by Weston and Yoganathan, 2001; Balachandran et al., 2006; Gupta and Grande-Allen, 2006; Gupta et al., 2007; Merryman et al., 2007; Balachandran et al., 2009 have demonstrated that an altered strain field in atrioventricular tissue can lead to multiple changes at the cellular level and subsequently changes in ECM content causing alternate mechanical responses from physiological loading. Merryman et al. showed in 2007 that 15% static strain acted synergistically with the fibrotic cytokine, TGF-b1 resulting in disease like characteristics. He also mentions that once calcification has been initiated, there are many other signaling factors which are mechano-dependent acting on the heart valves.

The Billiar lab performs research in mechanobiology considering both 2D and 3D models to understand mechanical force mechanisms regulating cell and tissue behaviors, as well as ECM biochemistry, biomechanics, and disease pathogenesis (Tissue Mechanics & Mechanobiology Lab, 2016). The primary goal of our project is to develop a precise cell stretching device for biaxial strain at different magnitudes, rates and patterns with observed cell effects to fulfill the requirements of the Billiar lab research objectives.

Section 2: Stretching Devices on the Market

Several cellular stretching devices exist for purchase on the market today. These devices are conventionally designed for imposing either uniaxial or biaxial strain to cells on silicone rubber membranes. Different designs exist and each one is accompanied by a control system and software. Gold standard cell stretching devices on the market range from $20,000-$30,000 and generally require the purchase of the entire system (stretching platform, controller, and software) along with custom wells.

Although a common stretching device design utilizes motors to generate deformation, another established approach is vacuum based systems. These devices deform a silicone rubber membrane using the force of a vacuum. One popular biaxial stretching device example is Flexcell's StageFlexer®, which employs a silicone rubber membrane stretched over a circular rigid post. Illustrated in Figure 2.2, an open-air pocket under the membrane around the post is emptied by a vacuum. The low pressure pulls the silicone rubber membrane down into the open area, stretching the membrane over the post and providing equi-biaxial strain (FlexCell, 2011). The cost of the complete StageFlexer® system is $29,494. Section A4 of Appendix A provides an additional explanation of this device including the software.

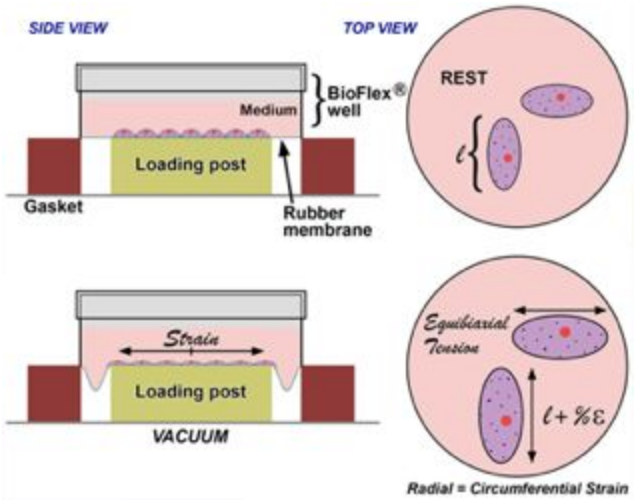


Figure 2.2: StageFlexer® applying equibiaxial strain (FlexCell International Corporation, 2011).

There are currently a few devices on the market that act as a "gold standard" for stretching cells, called StrexCell systems. There are several models with various properties and stretching capabilities. One is the Automated Cell Stretching System STB-140, which is both a uniaxial stretcher and incubator, seen below in Figure 2.3. This hybrid device allows cells to be

kept at proper temperature, humidity, and CO_2 levels while they are being stretched uniaxially. However, the nature of this device does not allow the user to view the cells under a microscope unless the wells housing the cells are removed. The wells can be 4 cm^2 or 10cm^2, allowing 8 or 6 wells respectively to be stretched at the same time, and this device can stretch the wells in 64 different patterns (StrexCell, 2018). The cost for the Strex STB-140 device is $20,000 (StrexCell, 2018).

Figure 2.3: Strex STB-140 with four stretching wells loaded for uniaxial strain(Copyright © 2018 Amuza Inc.).

Another StrexCell system is the Microscope-Mountable Biaxial Stretching System STB-190-XY, seen below in Figure 2.4. This device does not have an incorporated incubator; however it does allow the user to operate the device with a microscope at the same time. The STB-190-XY houses a 4 cm^2 well, which it stretches in 64 different patterns with biaxial stretching and compression capabilities. It runs on two motors that turn threaded rods, moving sliders and pins that pull on the corners of the well. Three of the four corners move while one is fixed. It is compatible with Nikon and Olympus microscopes, but can be ordered to fit Zeiss and Leica microscopes(StrexCell, 2018). The Strex-190-XY device costs $20,000 (StrexCell, 2018).

Figure 2.4: Strex STB-190-XY, red arrows indicate motors and blue indicate sliders(Copyright © 2018 Amuza Inc.).

There are several other cellular stretching devices that exist on the market. The StageFlexer also comes with sister devices that have different design variations. These devices are described in detail in Appendix A Section A4. Another device on the market is called the Cell Stretcher CS-10 Series for uniaxial cell stretching and the Cell Stretcher dcCS10 for biaxial cell stretching both manufactured by Electron Microscopy Sciences. Both of these devices provide microscope stage mounting for real time cell observation stretched using two DC servo motors connected to an external control unit programmed with a custom code. Cells can be stretched at different strain patterns including static phases or repeated patterns ("Computer Controlled Cell Deforming - Cell Stretcher", 2019). A complete description of each device is found in Appendix A, Section 4b.

Section 3: Review of Past MQP Projects

In 2012, the stretching machine that this project will focus on was initially built as a Major Qualifying Project (MQP) (Duoba et al., 2012). The device developed uses four motors to pull on pegs that attach to the corner of the stretch wells. These motors are controlled by an

Arduino program which allows the user to specify the strain and speed achieved in each direction. A portion of this Arduino code is displayed in Appendix D. These motors are linear actuators. The motors, which are size 14 model Q motors that were donated by Haydon-Kerk, no longer seem to be able to handle moving at a rate of 1 Hz at a 10% strain (Duoba et al., 2012). In a discussion with our client Zach Goldblatt, he estimated that at a 10% strain the motors were running at an estimated rate of 0.7 Hz. The motors are run by an Arduino, and our client mentioned this may be limiting the power input to the motors. Therefore, it is one possible explanation for the insufficient speed. The base plate was designed to fit over a Zeiss microscope; unfortunately, due to the distribution of weight on top of this base plate, however, the device is not stable when placed on the microscope. An early prototype of the stretch well was also developed. The current system is pictured in Figure 2.5 below.

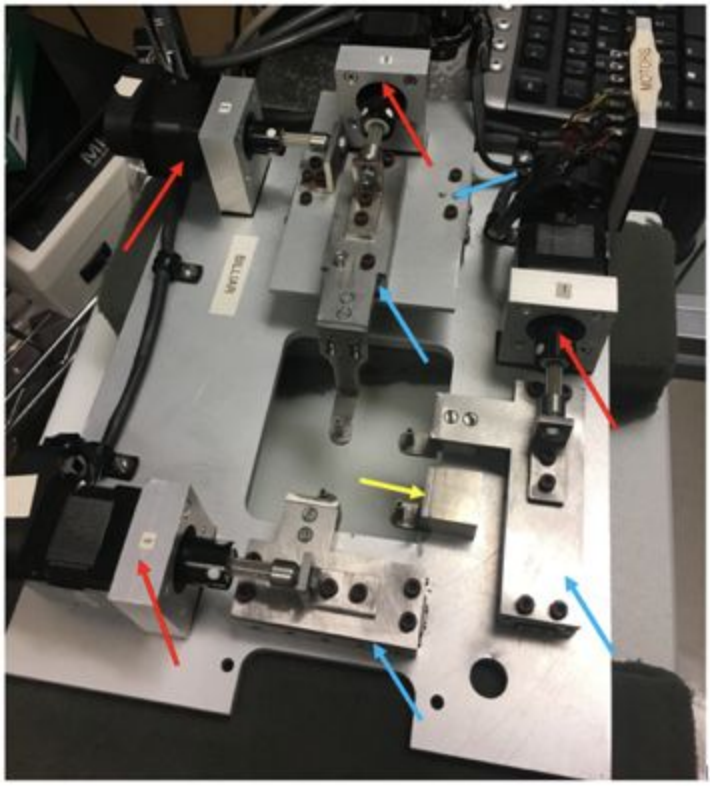


Figure 2.5. 2013 stretching device design configuration. The baseplate of this setup fastens to the stage of the microscope by a thread and screws. The red arrows indicate linear motors. The blue arrows indicate linear slides. The yellow arrow indicates the fixed point.

The 2013 MQP continued where the 2012 one left off (Garcia et al., 2013). Our team redesigned a four linear actuator motor stretching device based on testing results of the 2013 device and analysis of several design alternatives. A new computer program with a built-in graphic user interface (GUI) was developed for in-lab computer to interact with the stretching device. A new mold design was designed to that would achieve a larger percent of uniform strain on the stretching surface.

The 2013 MQP team recommended that in the future the mold be made out of a harder material. The current material, an acrylic, leaves behind scratches when cleaned which has been damaging the PDMS wells currently made with it (Garcia et al., 2013). The damaged on the

wells were manufacturing defects that acted as stress concentrators for crack propagation and well material failure during cyclic stretching.

During testing using a uniaxial or biaxial stretching device, cells are seeded onto a PA substrate within a PDMS stretching well that is fixed on the stretching device with pins. The PDMS and PA substrates with seeded cells are stretched by the stretching device at specific strains and rates. Subsequently, the substrates translate the imposed mechanical forces to strain the cells. For a uniaxial or biaxial stretching pattern, the overall well geometry is square. During uniaxial stretching, the well surface is elongated in one direction according to the imposed strain and compressed in the opposite direction according to the material's poisson's ratio. Figure 2.6 illustrates this effect.

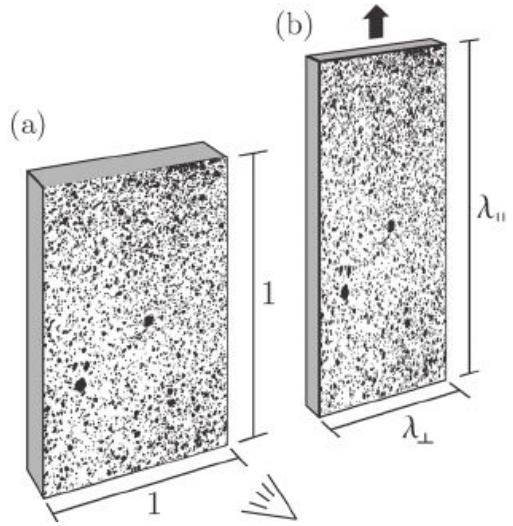


Figure 2.6: Uniaxial elongation of a rectangular specimen with compression in the direction opposite elongation (Pritchard et al., 2013).

PDMS is a commonly used stretching well material with a poisson's ratio of 0.49 for a base to cross-linker ratio of 15:1(Roh et al., 2013). For information on the properties of PDMS and the polyacrylamide gel used to seed cells within the wells, see Appendix A, sections 2 and 3.

The new mold our team proposed helps to eliminate manufacturing defects and the new well design proposal greatly improves the amount of homogeneous strain area. While the existing program and GUI developed to interface with the stretching device is efficient and the GUI is easy to use, both programs can only interact with an in-lab computer. This computer is an older model that runs slowly and limits the code-processing time. Our client specified that the program should be able to run on multiple computers. There were no existing problems with the device in the incubator itself, but the device still remains too bulky and uneven to fit on Professor Billiar's Zeiss microscope.

Chapter 3: Project Strategy

Section 1. Initial Client Statement

Before starting our project, the client discussed several issues with the current device that they would like to see implemented. From this discussion, we formulated our initial client statement:

"To design, construct, and test a system to stretch cells at 10% strain and 1 Hz frequency to simulate the in vivo mechanical and chemical environment of a cell to assess effects of stretch and stiffness on cell alignment, spreading, and cell fate. The system must also accurate fit fixed atop a Zeiss microscope."

This statement was later adapted after further research on the topic and meetings with the client.

Section 2. Objectives and Constraints

Our team created an objective tree to help determine the key objectives, for the project. The objectives were split into five major categories: versatility, safety, cost, functionality, and longevity. To be versatile, the device had to be able to achieve a high variety of strain magnitudes at different frequencies which allow for variable stretching patterns. To be safe, the device could not endanger either the user or the cells during testing. To be cost-effective, prototyping materials and new device materials were obliged to cost less than the MQP maximum budget of $1000. To be functional, the device had to be able to achieve a reliable and accurate equi-biaxial and uniaxial strain. The device also had to be user-friendly and produce a large uniform area of biaxial or uniaxial strain for the cells to achieve true uniform strain. To

operate for the required amount of time, the motors could not overheat or malfunction and the wells on the device had to withstand repeated cycling within the desired operating ranges. Finally, the device required code that could easily be modified for any given testing scenario. Figure 3.1 below is an objective tree that categorizes our objectives.

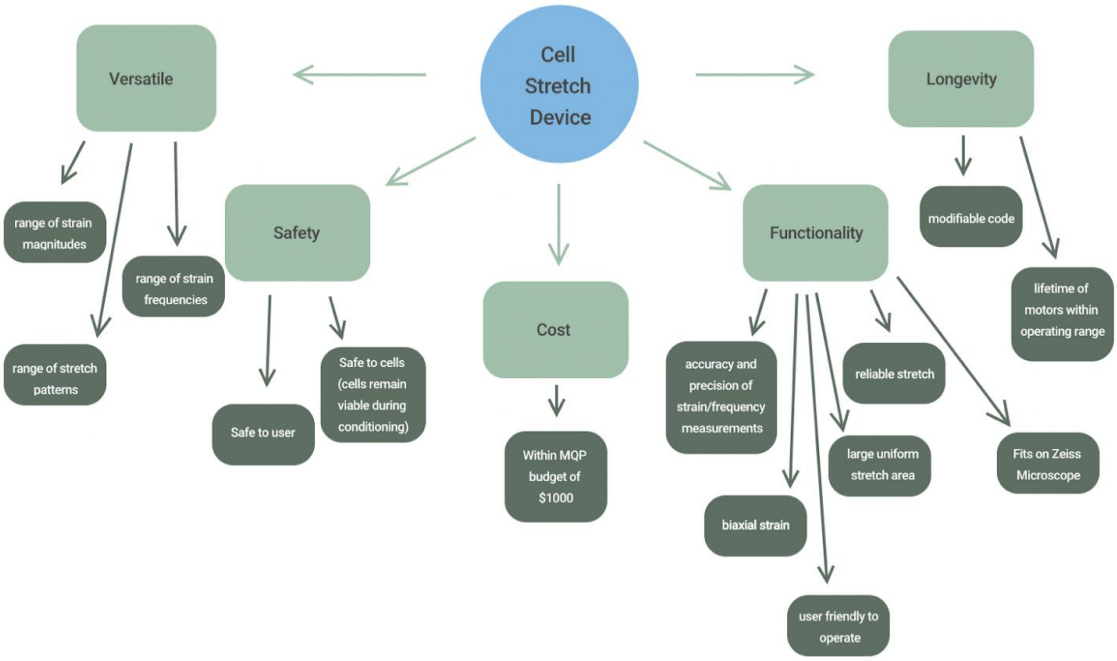


Figure 3.1: The objective tree for our design.

To prioritize the project objectives, a pairwise comparison chart was developed to determine the relative importance of each objective, then rank the objectives, as seen in Table 3.1 below.

Table 3.1: Pairwise comparison chart to rank client and device objectives.

Pairwise Comparison Chart						
	Accurate and easy placement on Microscope(s) (fits 152 x 103 mm stage)	Low cost (i.e., <$1000)	Capable of High Strain (at least 10%) & Rate (at least 1Hz)	Large homogeneous strain (at least 1mm^2) & imaging area with cell tracking accuracy (>95%)	Durable Wells (failure rate <10%)	Total
Accurate and easy placement on Microscope(s)(fits 152 x 103 mm stage)	**X**	1	0.5	1	1	**3.5**
Low cost (i.e., <$1000)	0	**X**	0	0	0	**0**
Capable of High Strain (at least 10%) & Rate (at least 1Hz)	0.5	1	**X**	0.5	1	**3**
Large homogeneous strain (at least 1mm^2) & imaging area with cell tracking accuracy (>95%)	0	1	0.5	**X**	1	**2.5**
Durable Wells (failure rate <10%)	0	1	0	0	**X**	**1**

Based on our pairwise comparison, we determined the two most important objectives were to: 1) ensure the device fits on the microscope and 2) achieve desirable strain magnitudes and frequencies. If the device does not fit on the Zeiss microscope stage, then it cannot properly operate to serve its intended purpose. Without achieving the desirable strain magnitudes and frequencies, the device could not uphold its purpose for research. Our third most important objective was to have a large homogeneous biaxial strain and imaging area within the range for

cell tracking accuracy given by the client. Specific area and accuracy constraints of at least 1 mm^2 strain area and a 10-micron cell tracking accuracy difference (9%-11% error margin) between successive images. A larger uniform area improves the efficacy of analysis by providing a larger area where true equi-biaxial strain is achieved. The error margin allows individual cells to be accurately tracked between images. The fourth objective was that the device should have durable stretching wells defined by a low material failure rate with conditioning. The purpose of a low failure rate is to ensure that the well withstands a 24-hour testing time, maximize the number of tests run by minimizing well manufacturing time. The final objective ranked was the cost. This objective was ranked last because of the pre-existing $1000 budget falling well beneath market standard devices ranging from $20,000-$30,000. Additionally, our team is working with existing materials from the 2013 version of the device and our client expressed a willingness to provide additional necessary funding if the proposed device design worked under ideal conditions.

In addition to the ranked objectives for the device, the following constraints were defined. Most importantly, the stretching device needed to fit within the 152 x 103 mm stage of the Zeiss microscope. The existing device failed to properly fit on the microscope stage, requiring modification to rest flush and subsequently to attain accurate tracking of the stretched cells. Within this constraint, the device needed to clear all objectives on the microscope to prevent lens scratching and allow free movement of the mechanical stage. Next, the strain and frequency parameters were defined such that the device needed to achieve a minimum of 10% strain at 1 Hz within a 10% error margin. Although the desired strain and frequency were specifically defined, the ability to accurately adjust the strain magnitude and frequency for future

experiments was also required. Therefore, the device needs to accurately operate at a range of strains from 1% to 20% and frequencies from 0.1 Hz to 5 Hz as seen in Table 3.2. The PDMS stretch wells used in conjunction with the device required a material failure rate less than 10% of the total wells tested per batch. Furthermore, the production rate of the PDMS wells needed to be in the range of 4 to 10 wells manufactured per cycle. To produce a large homogenous strain area, the PDMS wells had to produce a uniform strain area of at least $1mm^2$. This constraint aimed to maximize the efficacy of each test by creating a larger cell sample size falling between 9%-11% equi-biaxial strain error. Moreover, the device had to run for long periods of time to condition and to observe changes in the cells. The client required at least a 24-hour running time for conditioning. Finally, the device had to cost less than the MQP budget of $1000. Table 3.2 was given to our client to determine the exact requirements and necessary accuracy and error margins of the system.

Table 3.2: Exact requirements and accuracy chart to define specific operating and device specification ranges.

	Min	Max	Unit	Explanation
Strain	1	20	%	i.e. 1-20%
Strain Frequency	0.1	5	Hz	i.e 0-5 Hz
Operating Time	0	24	Hours	i.e. 0-12 hours
Stretch Well Failure Rate	0	10	%	i.e. 1 in 10 wells fail
Manufacturing Rate	4	10	wells/round	i.e. 4 in one round of production
True Biaxial Strain Area on Wells	1	4	mm^2	i.e. 1 mm2 - 4 mm2
Cell Tracking Accuracy	0	10	Micron difference between images	i.e. error: 0- 10%

Based on Table 3.2, the exact desired operating ranges, device specification requirements, and accuracy ranges could be determined. The client's completion of this chart informed our team about the degree of accuracy and precision required for our validation testing as well as guided our hardware specifications by limiting factors such as size, power, accuracy, torque, and method of manufacturing.

Additional considerations for the device included a user-friendly design including a guided-user-interface (GUI), a modifiable code, stretching strains, frequencies, and stretch patterns. The existing device operated on an Arduino-compatible Chip-Kit controller. Although our team was not limited to use only this software, we chose to code the new device using a similar Arduino MultiMoto controller to save money and to customize the code. A custom,

open-source, code allows for a variety of testing options that can be modified within the set coding language. Modification prolongs the ultimate lifetime of the device making it adaptive to future experiments, different research labs, and even changes to the stretching device. Ideally, the device should also attain multiple different stretch patterns in equi-biaxial and uniaxial stretching directions for stretching cells; however, this design feature was considered non-essential.

Section 3. Design Standards

Design standards exist to verify the quality, safety, and efficacy of different devices, products, services, and facilities. Researchers, companies, manufacturers, and distributors use these standards to improve protocols and facility conditions, which directly impact the safety and efficacy of products by mitigating associated risks. Adherence to standards can also increase production or sales by certifying to the consumer that the purchased product is of a certain quality or specification. Different organizations formulate standards for the testing and use of biomedical devices and materials including IEEE, ASTM, and ISO. The *Institute of Electrical and Electronics Engineers*, IEEE, provides standards in areas such as Instrumentation and Measurement, Power and Energy, Software, Batteries, Electronics, and Electromagnets. The American Society for Testing and Materials, ASTM, covers international standards for testing materials. The International Organization for Standardization, ISO develops standards for industrial and commercial requirements for products, services, systems, to ensure safety and efficacy.

Although similar devices exist on the market, and the device for our client is intended for use in research, the standards set by ASTM and ISO must be considered in the materials, design,

construction, and testing of the device. Based on initial discussions with the client, this device will be used for stretching cardiomyocytes. We also know that the PDMS wells used in conjunction with the device require sterilization by autoclaving. Thus, because this device stretches cells on a biocompatible material, it must meet the engineering standard, ISO 10993-5 *Biological Evaluation and Biocompatibility Testing of Medical Devices*(Anonymous Biological evaluation of medical devices, 2009). ISO 10993-5:2009 outlines specific cytotoxicity tests for in vitro mammalian cellular applications. The testing requirement includes the estimation of cell damage based on morphology, and gives requirements for measuring cell damage, and cell growth.

ASTM STP1173 *Biomaterials' Mechanical Properties*, specifies the testing parameters for the mechanical properties of biomaterials(Kambic, 1994). This is especially important for our device because the exact material properties of PDMS directly impacts the resulting strain imposed on the cells.

ASTM F813 - 07(2012) specifies the standard practices for cell culture for cells in direct contact with a medical device and specific evaluation techniques for the device materials(ASTM, 2012). ASTM STP810 specifies standard cell culture testing methods, and ASTM F2739 - 08 specifies protocol for cell counting and determining cell viability on scaffolds made of biomaterials (ASTM, 2012; Brown, 1983). Testing using the device involves cell culture of cardiomyocytes. Therefore, it is essential to determine the cytotoxic effects of any device materials that could contact the cells, mainly the PDMS wells and PA gel that the cells sit atop. Determining the overall cell viability of the cells on the well and the distribution of viable and nonviable cells directly impacts the validation of the device efficacy since stretching directly

impacts cell orientation. If dead cells are present on the slide, the reliability of the results decreases because dead cells could be misinterpreted as live cells in a specific orientation.

ASTM E1837 - 96(2014) *Standard Test Method to Determine Efficacy of Disinfection Processes to Reusable Medical Devices* describes the methods for determining the effectiveness of the disinfection of the device between uses after contact with cells(ASTM, 2012). This is important for disinfecting the device between experiments and disinfecting the stretch wells by autoclaving.

To stretch cells in the cell stretching device, cells were seeded on elastomeric PDMS wells stamped with PA gel. These wells were originally designed and created by another MQP team in 2012. The silicon based elastomer used to created the wells, PDMS, requires a specific stiffness, reproducible in experiments to generate accurate results. These wells required constant stretching over a 24 hour period. The PDMS wells needed to sustain the 24 hour cyclical stretching without failure or undergoing stress relaxation. The stiffness of the PDMS is controlled by the base to cross linker ration prior to polymerization. The PDMS recipe used was a 15:1 ratio of base to cross-linker, following the protocol written by the 2012 MQP team. Standard technical tests exist to create a general consensus for materials, products, systems, and services (ASTM, 2019). The American Society for Testing and Materials (ASTM) provides standards D638-14 and D2990-17 for testing stress relaxation and tensile strength, respectively. The ASTM also put forth a standard for optical properties, D1056-14, considered for the PDMS wells which will be imaged by light microscopy.

IEC 60072-1:1991, Dimensions and output series for rotating electrical machines - Part 1: Frame numbers 56 to 400 and flange numbers 55 to 1080, addresses the tolerances for motor

extension and rotation (International Electrotechnical Commission, 1991). A list of electrical and mechanical safety standards for low voltage motors can be found in Asea Brown Boveri's Low Voltage Motors Motor Guide (2014).

Section 4. Revised Client Statement

After determining the ranked objectives, device constraints, and considering engineering standards, we developed a revised client statement:

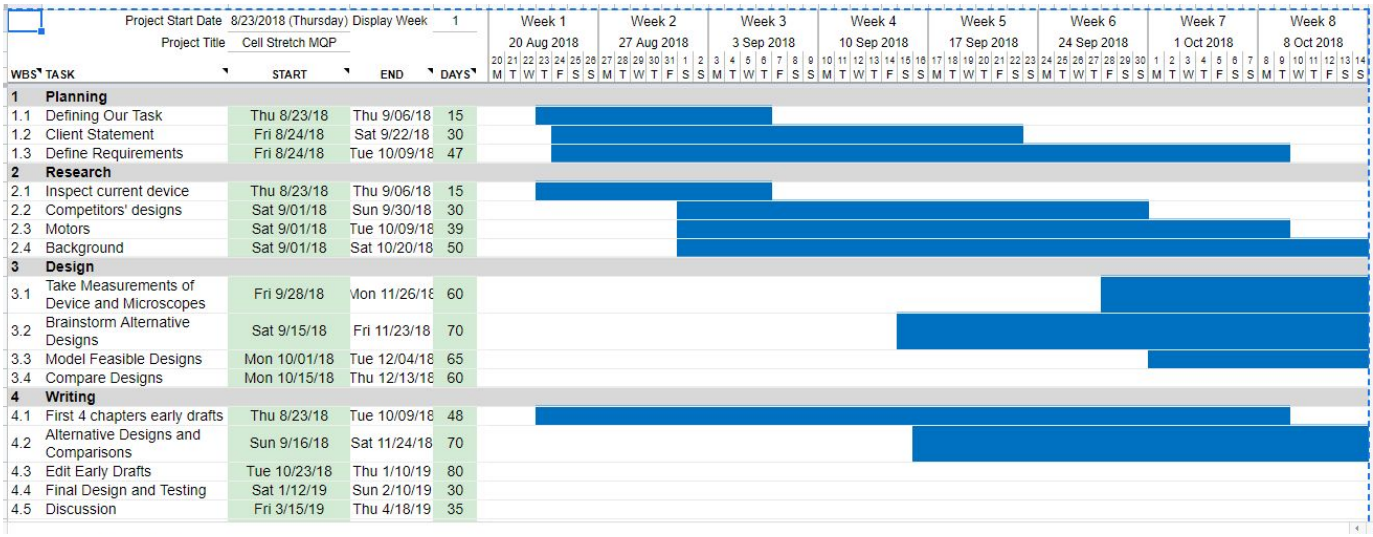
"To design, construct, and test a system to stretch cells at a range of strains (1-20%), but must achieve a 10% strain, and a range of frequencies (minimum 1 to 5 Hz), with an improved fit on the Zeiss microscope. The failure rate of the PDMS stretching wells must also be reduced to less than 10% failure."

This client statement more accurately defined the requirements and limitations of the project within measurable variables.

Section 5. Project Approach

The planning phase of our project occurred from the end of August through early October. In this phase, we identified key goals of the project through multiple client meetings. The primary research phase occurred from late August to early December and involved collecting stakeholder requirements, analyzing market standards, and collecting hardware and other part specifications. The Design portion of the project took place from the end of October to mid-January. This phase consisted of creating and testing prototypes for each component of the device as well as analyzing design models using Finite Element Analysis. The Build and Test phase of our project from mid-January through early April. In this phase, we developed and built our complete device and tested to meet requirements. The Validation phase took place from

mid-March through late April. In this phase, the validity of our test results was examined under the given constraint conditions and error/accuracy ranges. Four Gantt charts, which outline the project management plan in each school term, to accomplish designing, building, testing, and validating a mechanical stretch device are found in Appendix E. Figure 3.2 below shows the A-term Gantt chart.



Project Start Date: 8/23/2018 (Thursday) Display Week: 1
Project Title: Cell Stretch MQP

WBS	TASK	START	END	DAYS
1	**Planning**			
1.1	Defining Our Task	Thu 8/23/18	Thu 9/06/18	15
1.2	Client Statement	Fri 8/24/18	Sat 9/22/18	30
1.3	Define Requirements	Fri 8/24/18	Tue 10/09/18	47
2	**Research**			
2.1	Inspect current device	Thu 8/23/18	Thu 9/06/18	15
2.2	Competitors' designs	Sat 9/01/18	Sun 9/30/18	30
2.3	Motors	Sat 9/01/18	Tue 10/09/18	39
2.4	Background	Sat 9/01/18	Sat 10/20/18	50
3	**Design**			
3.1	Take Measurements of Device and Microscopes	Fri 9/28/18	Mon 11/26/18	60
3.2	Brainstorm Alternative Designs	Sat 9/15/18	Fri 11/23/18	70
3.3	Model Feasible Designs	Mon 10/01/18	Tue 12/04/18	65
3.4	Compare Designs	Mon 10/15/18	Thu 12/13/18	60
4	**Writing**			
4.1	First 4 chapters early drafts	Thu 8/23/18	Tue 10/09/18	48
4.2	Alternative Designs and Comparisons	Sun 9/16/18	Sat 11/24/18	70
4.3	Edit Early Drafts	Tue 10/23/18	Thu 1/10/19	80
4.4	Final Design and Testing	Sat 1/12/19	Sun 2/10/19	30
4.5	Discussion	Fri 3/15/19	Thu 4/18/19	35

Week 1: 20 Aug 2018 | Week 2: 27 Aug 2018 | Week 3: 3 Sep 2018 | Week 4: 10 Sep 2018 | Week 5: 17 Sep 2018 | Week 6: 24 Sep 2018 | Week 7: 1 Oct 2018 | Week 8: 8 Oct 2018

Figure 3.2: Gantt chart to plan project management approach for A-term focused on planning and research followed by initial designing and writing.

Chapter 4: Methods and Alternative Designs

Section 1. Needs Analysis

After determining the device requirements and constraints from client surveying, we determined based on our ranked objectives, which device features were most essential for product completion. The ranked objectives in order were: (1) the device should fit over the microscope, (2) it should be able to achieve a strain of at least 10% and at least a 1 Hz frequency, (3) the wells should produce a large homogeneous strain area for imaging, (4) the wells should be durable (5) the entire product should cost less than the $1000 budget. The device was additionally required to produce both uniaxial and equibiaxial strain on cells, so the coding should be flexible to allow both stretching directions.

The major physical limitations of the device included its size, weight, and orientation. Ideally, the device design was requested to be smaller than the existing device, which was about 11.25 inches by 9 inches. This would facilitate cell- visualization and mounting on the Zeiss microscope. The existing mount was made of stainless steel, which is very heavy. The motors and sliders also added weight and the device configuration caused it to not be placed flush on the microscope stage. This caused the device to tip off the microscope. Additionally, it was requested that the device be able to operate without interfering with the microscope 40x objective. These requirements and limitations were incorporated into the final design of the device.

Section 2: Feasibility Studies: Improving Well Durability

To improve the existing well design, our team evaluated the PDMS stretch well to determine where improvements could be made. The previous well design consisted of an octagonal center area with a thin PDMS base, enclosed by thicker PDMS corner pieces with pin holes. Figure 4.1 illustrates this design.

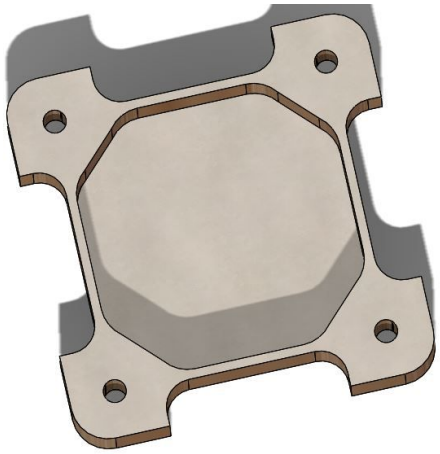


Figure 4.1: Original PDMS well design

The purpose of the larger material thickness and area around the pinholes was to reduce stress at the corners where the motors applied the greatest force. Whereas, the octagonal center area reduced the non-usable strain area. Under 10% uniaxial strain, the Finite Element analysis of this original well design produced a 16.6% uniform strain area. Similarly, under 10% biaxial strain, the well produced a 9% uniform strain area.

Although the primary objective of our project was to develop a biaxial and uniaxial stretching device that fit a Zeiss microscope, we additionally proposed alternative well designs to improve the size of the strain field on the stretched wells and improve the durability of the wells. We initially developed several different well-designs to accomplish these objectives. Our design

ideas focused on making small iterative modifications to the existing well design. We chose this approach because the wells used with the existing device already compare to the market gold standard, Strex well. Furthermore, past MQP teams have focused the whole of their project on the redesign of the existing wells. Finally, the primary focus of this MQP project was to improve the stretching device itself, wherein the client expressed that changing the well design was subordinate compared to fitting the device to the microscope stage and attaining the desired strains and rates. Nonetheless, our team proposed alternative designs for the wells to improve the durability of the wells and increase the uniform strain area for cell imaging.

The first proposed change to the original well design to improve the durability of the wells was to modify the well manufacturing process. We thought that if we could decrease manufacturing defects, we could improve the lifetime of each well. To do so, we considered why the wells were ripping in the first place. Our visual analysis of the wells showed material flashing along the edges of the top surface of the well, as seen in Figure 4.2. Then with analysis by physical stretching of the wells uniaxially, we noticed that the manufacturing defects propagated with cyclic applied force.

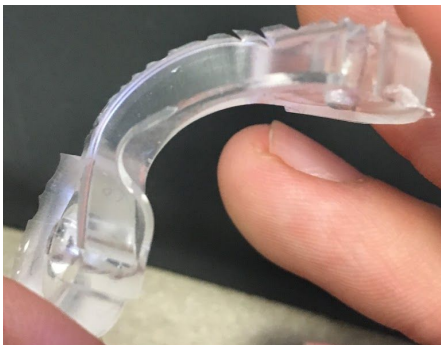
(A)  (B) 

Figure 4.2. (A) Small tears caused by the process of molding the custom PDMS wells cause (B) failure originating near the corners of the wells

The failure pattern with uniaxial stretching was always near the corner of the wells during our analyses. We additionally analyzed the areas on the well most susceptible to failure using SolidWorks to determine the von Mises forces. This analysis showed in Figure 4.3, produced areas of high stress near the pin corners of the wells.

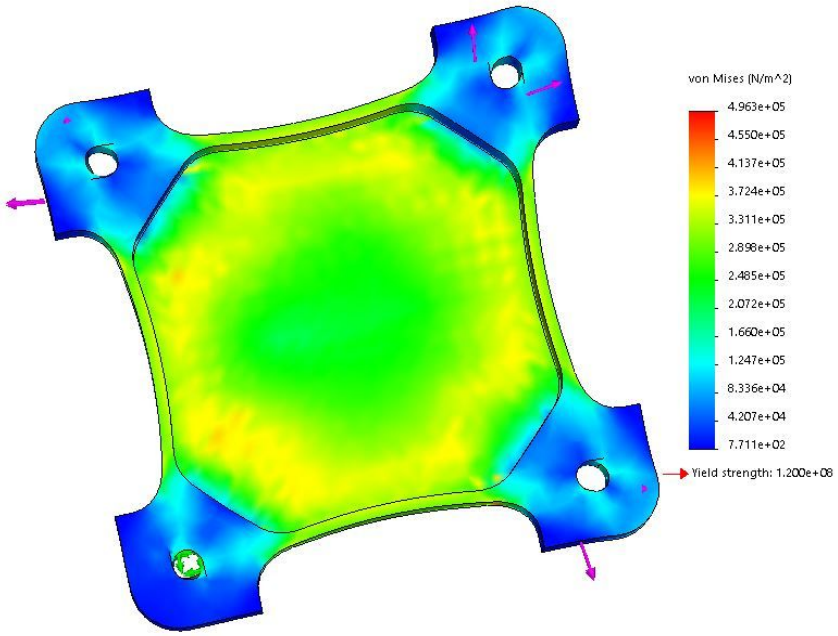


Figure 4.3: Von Mises Stress Analysis of Current Well Design

To improve the well durability, our team modified the manufacturing process for creating the wells by altering the mold design. The mold design incorporates the modified well design as described in section 3, below. A detailed description of the new manufacturing process for the wells is described in section 6 of the final device validation testing chapter of this paper.

Section 3: Designing and Evaluating Alternate Stretching Wells

In addition to evaluating the von Mises stresses, our team used Finite Element to evaluate the amount of homogeneous strain area available under 10% axial strain for the original well design. The resulting uniform strain area produced was reported as 16.6% in the axial direction and 13.8% in the transverse direction. According to Duoba et al., the total uniform strain area for uniaxial strain area was 12% and the total uniform strain area for biaxial stretching was 9%(2012). Figure 4.4 (A) and 4.4 (B) show the FEA analysis of the original wells for uniaxial deformation from Duoba et al.

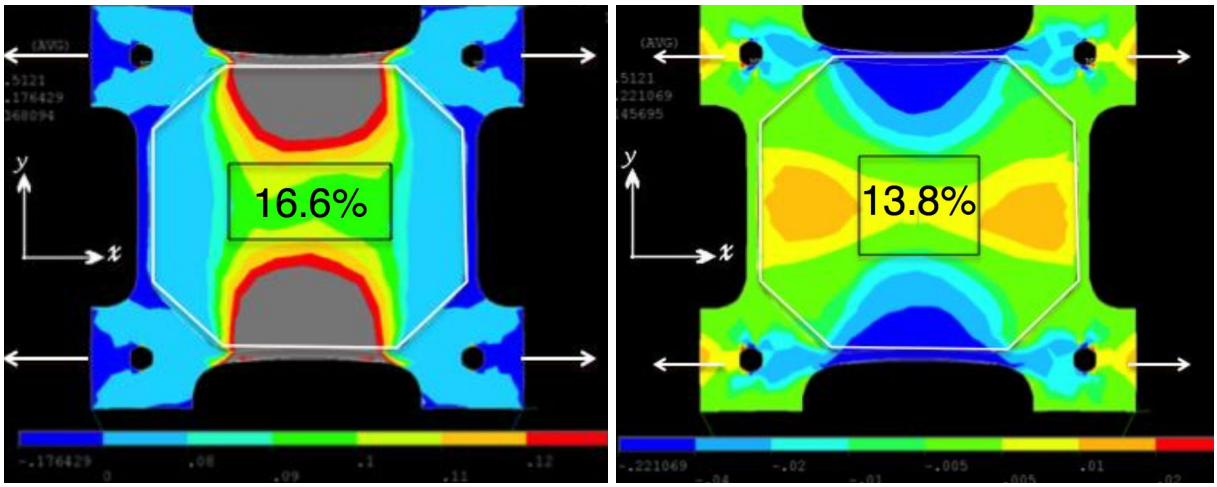


Figure 4.4: (A)FEA Axial strain field of original well, 10% uniaxial, x-direction; (B) Transverse strain field of original well, 10% uniaxial, y-direction.

To improve the amount of uniform strain area on the well, we proposed three alternate well designs and evaluated the designs in Finite Element. Our first design idea added sacrificial

wells along every side of the main structure. These purpose of adding thin wells was to purposely create a weaker area to deformation first along the edges. The edge regions would then undergo the majority of material collapse due to imposed strain so that the central area of the well could achieve more equal, true-biaxial, strain.

The first sacrificial well design contained 23mm long and 3mm wide sacrificial wells along each edge. The initial selection of well length and size was somewhat arbitrary; however subsequent modification allowed for optimization of these values. An illustration of the first alternate well design is represented in Figure 4.5.

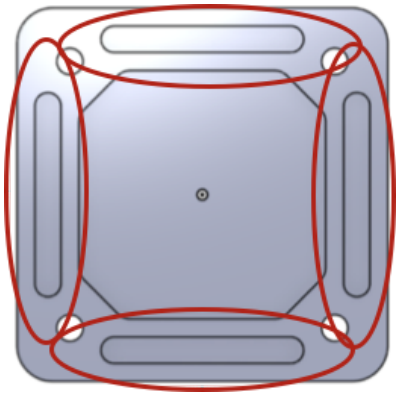


Figure 4.5: First design alternative adding 23x3mm sacrificial wells to each edge of the original well. The red ovals highlight the change from the original design adding the sacrificial wells.

FEA was used to investigate the local strain on the well surface under various mechanical simulations. A solid model of the sacrificial well design was first created using SolidWorks then uploaded to ANSYS. Based on the actual material being used to fabricate the stretch wells in the lab, a new material, PDMS, was created in ANSYS and given an elastic modulus of 2.5 MPa and a Poisson's ratio of .45. A force was applied at 10% uniaxial then equibiaxial strain. To apply a uniaxial strain, two pin holes were set as fixed and the opposite two were pulled in parallel at

10% strain. To determine the usable area, a 10% error was calculated creating an error map. Areas between 9% and 11% strain were noted, then the final percent usable strain area within the 10% error margin was calculated using ImageJ and compared to the total stretching area. Figure 4.6 illustrates the output of the analysis of our first alternative design with a 13.3% uniform axial strain area.

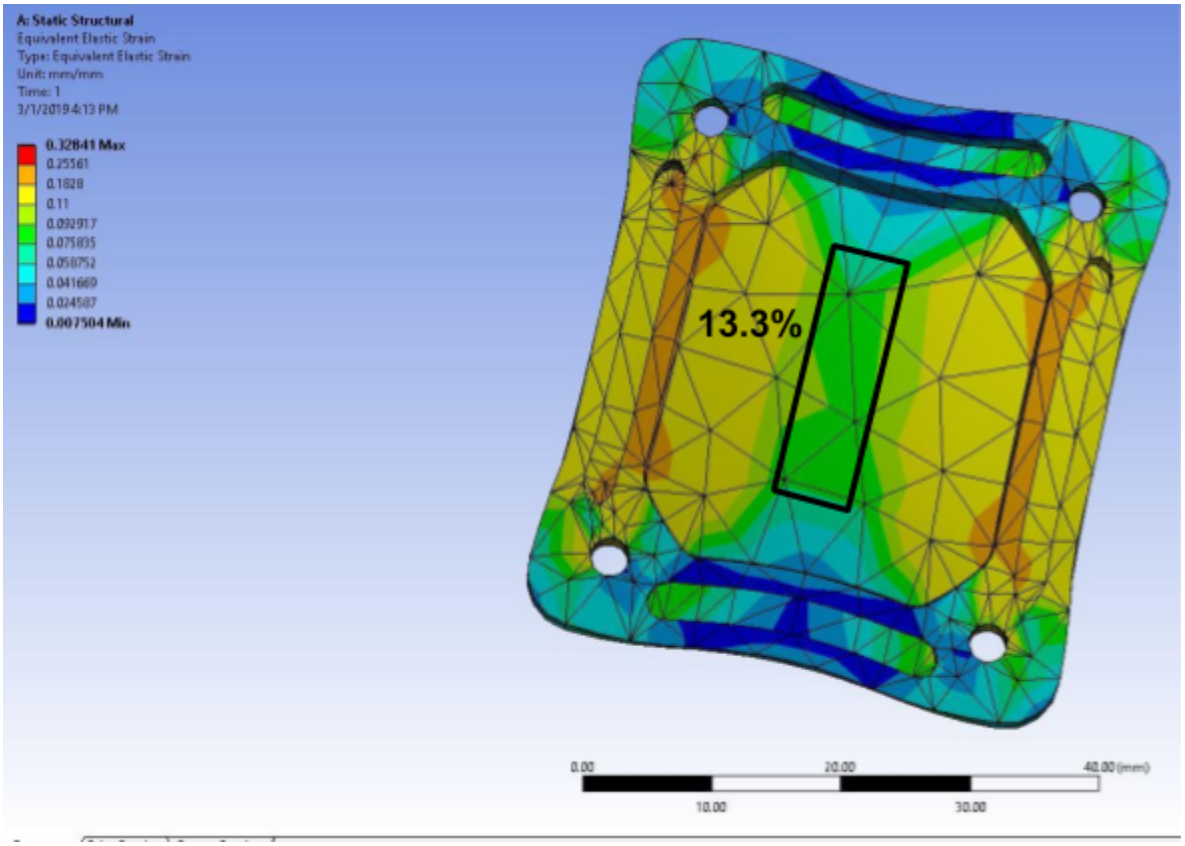


Figure 4.6: Strain field analysis under 10% uniaxial strain of the 23x3mm sacrificial well design. The box labeled 13.3% indicates the area on the stretch well that achieves 10% uniform uniaxial strain +/- 1% strain.

A second alternate well design was created by shortening the sacrificial wells to 17.5x3mm, as seen in figure 4.7. The shorter wells put more space between the pinholes which

was intended to increased the amount that the sacrificial wells would deform. By doing so, we hoped to increase the amount of uniform area in the middle of the well.

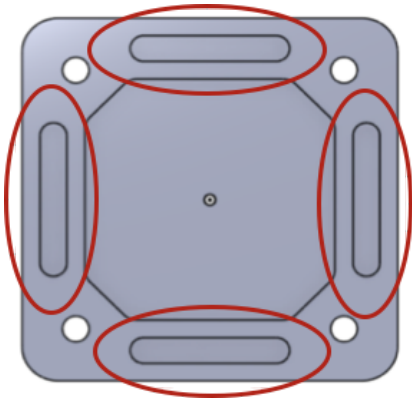


Figure 4.7: Second design alternative shortening the sacrificial wells to 17.5x3 mm on each edge. The red ovals highlight the change from the previous alternate design shortening the sacrificial wells.

The same type of FEA analysis was performed on this well design which produced a 21% uniform strain area, as seen in figure 4.8.

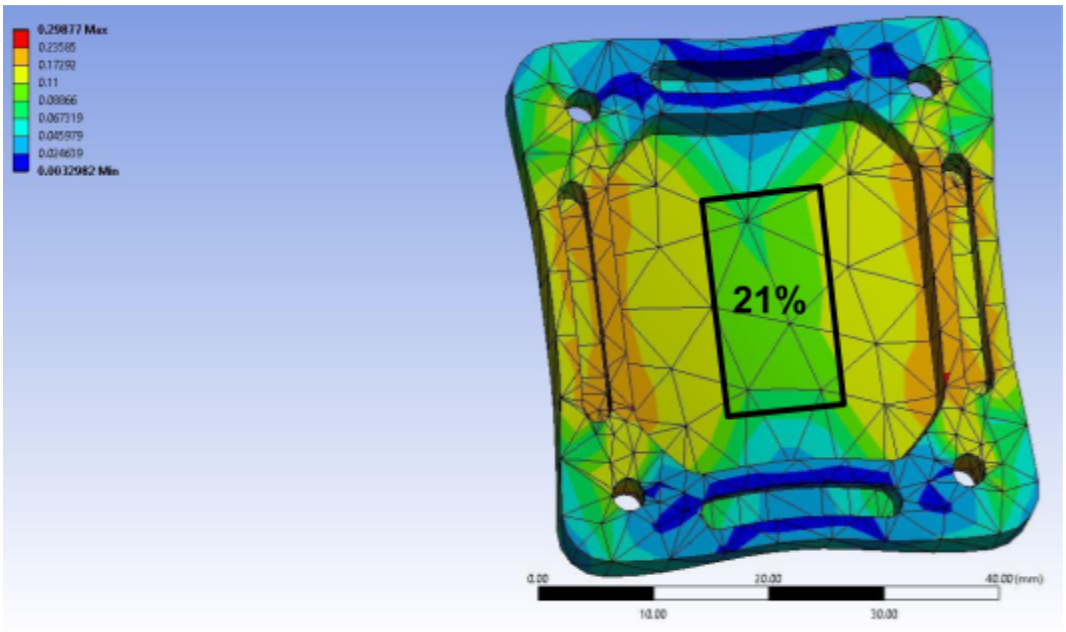


Figure 4.8: Strain field analysis under 10% uniaxial strain of the second alternate design with 17.5x3mm sacrificial wells. The box labeled 21% indicates the area on the stretch well that achieves 10% uniform uniaxial strain +/- 1% strain.

Finally, a third alternate design was created wherein the 17.5x3mm sacrificial wells along the edges were maintained; however, the inner corners of the well were rounded, as seen in Figure 4.9.

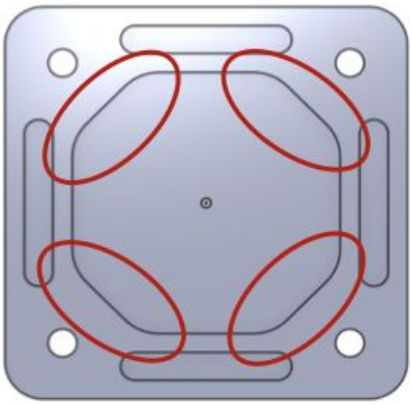


Figure 4.9: Third design alternative maintaining the sacrificial wells and rounding the inner corners of the well. The red ovals highlight the change from the previous alternate design adding rounded internal corners.

After FEA analysis, it was found that this design produced a 16.6% uniform strain area. Figure 4.10 represents the FEA analysis output.

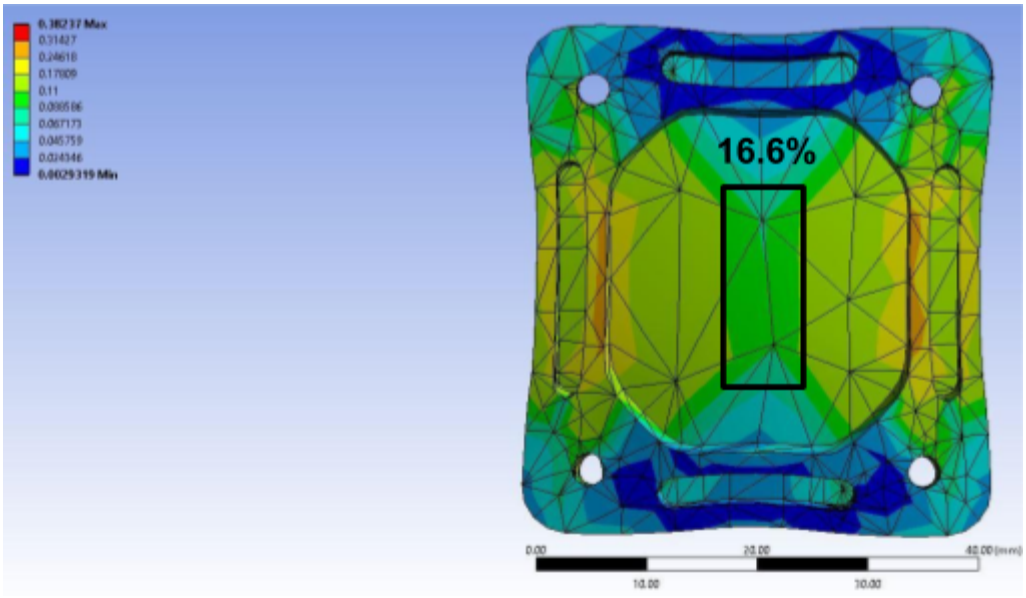


Figure 4.10: Strain field analysis under 10% uniaxial strain of the third alternate design idea with rounded center corners and 17.5x3mm sacrificial wells. The box labeled 16.6% indicates the area on the stretch well that achieves 10% uniform uniaxial strain +/- 1% strain.

Based on these FEA analysis, our team concluded that the second design alternative produced the largest uniform uniaxial strain area.

Similarly, to test the new well designs for biaxial strain, we used FEA to apply a force at 10% equibiaxial strain. To apply a biaxial strain, we first attempted to deform the entire well by pulling on each corner of the model in FE. This method produced several errors and the model could not be computed because there was no fixed point. When we attempted to try this model again with one fixed point and pulling from three corners at equivalent 10% strain amounts, the program again failed. We tried several other methods including adding nodes to the CAD drawing; however, we were still unsuccessful. Finally, we tried modifying the CAD drawing to be one corner of the total well. This simplified the calculation in FEA an allowed us to pull only from one corner then multiply the results to reflect the whole well deformation. Nonetheless, this

method worked accurately because our wells were symmetrical when reflected across the x-axis and the y-axis (if Z is the height). We tested both the original well design and the new 17.5x3mm sacrificial well design using this method. Our results showed that the original well design analysis from the previous MQP team included too much error in their calculation for uniform biaxial strain area. When we performed the 10% equibiaxial FEA analysis on the original well, only a 4.6%[1] uniform strain area was within the 9-11% strain field. This differed from the reported 9% biaxial strain field by almost half the area. Based on the images in the previous MQP paper, the "uniform" strain area included area outside the 9-11% uniform range which is likely the result of their error. Figure 4.11 shows the solved FEA strain area for 10% equibiaxial strain for the corner piece of the original well design.

[1] Even though one quarter of the total well was analyzed, the strain area was calculated as uniform area/total area*100 on the same quarter. If we were to multiply by four to account for the whole well, the percent would not change because the total area would increase by the same factor as the uniform area.

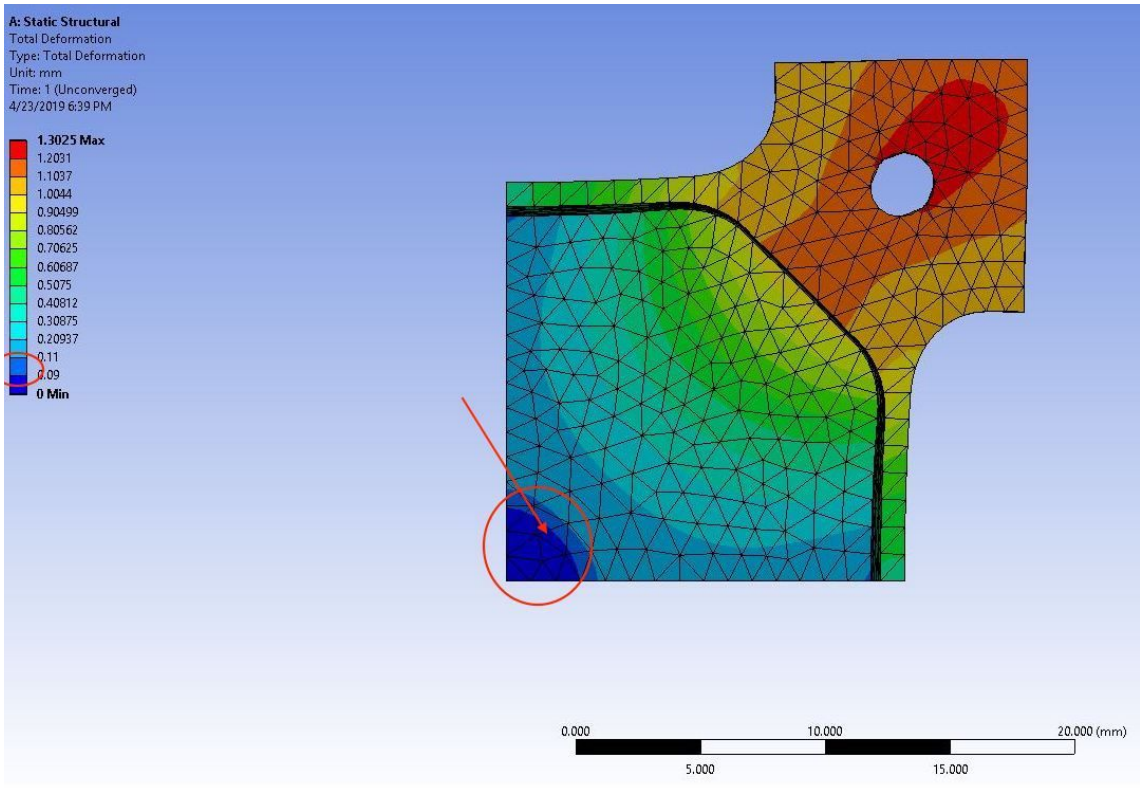


Figure 4.11: FEA 10% Equi-biaxial strain analysis for the original well design. The amount of uniform area is represented by the two darkest shades of blue, circled in red, which accounted for 4.6% of the total strain field.

This process was repeated for the new 17.5x3mm sacrificial well design. Again, one quarter of the total well was loaded into ANSYS under the original (uniaxial) material conditions. A displacement was set for the corner to move at 10% strain. The resulting strain field showed a 30.2% uniform equibiaxial strain area. This was a 556% increase in total amount of homogeneous strain area from the original well design. Figure 4.12 represents the solved FEA for the new sacrificial well design under 10% equi-biaxial testing conditions.

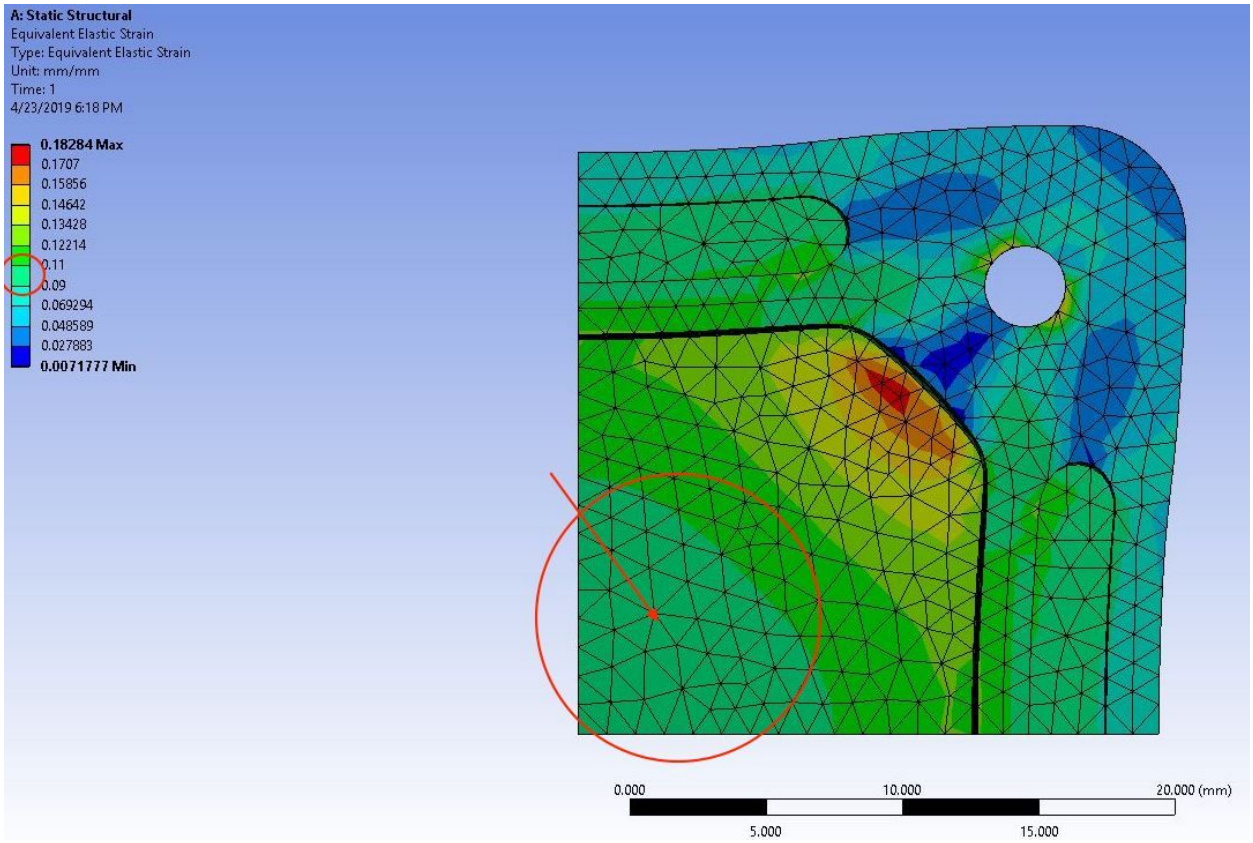


Figure 4.12: FEA 10% Equi-biaxial strain analysis for the new sacrificial well design. The amount of uniform area is represented by the teal (blue-green) shade, circled in red, nearest the bottom left corner which accounted for 30.2% of the total strain field.

This equi-biaxial analysis in FEA showed a large improvement in the amount of homogeneous strain area on the stretching surface of the well with the inclusion of the sacrificial wells to the perimeter.

To further evaluate the effectiveness of our well designs, we performed a von Mises stress analysis on each of the wells in SolidWorks. Figure 4.13 shows the von Mises stress analysis for the first alternate well design with 23x3mm sacrificial wells.

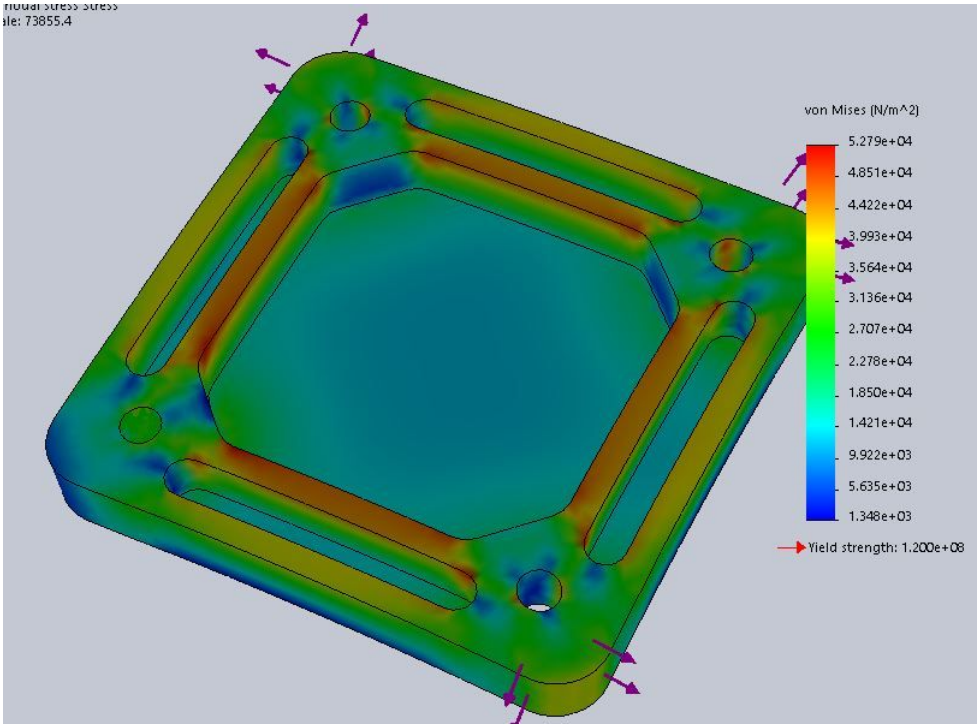


Figure 4.13: Design alternative 1: Analysis of von Mises stresses. The figure shows a large area of low stress in the center of the well and high stresses at the edges of the sacrificial wells.

This analysis showed that there is a large area of low uniform stress in the middle (blue), with a few spots of high stress on the corners and edges of the sacrificial wells. These high stress areas could mark potential spots of failure; however, the purpose of our design was to generate high stresses at the sacrificial wells, so the concept of sacrificial wells was validated. The same analysis was performed for the second alternate design with 17.5x3mm sacrificial wells, as seen in figure 4.14.

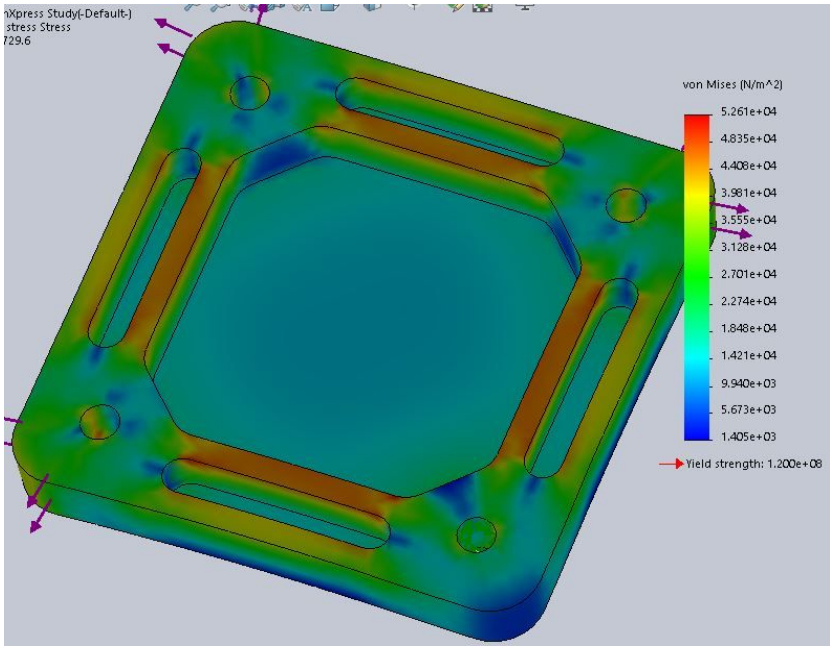


Figure 4.14: Design Alternative 2: Analysis of von Mises Stress showing decreased stresses in the center of the well and slightly lower stresses at the sacrificial well edges.

This analysis showed an even lower and more uniform area of uniform stress in the middle of the well and slightly lower areas of high stress at the pins and along the sacrificial well edges. Once again, this analysis was repeated for the third design alternative with rounded inner corners, as seen in figure 4.15 below.

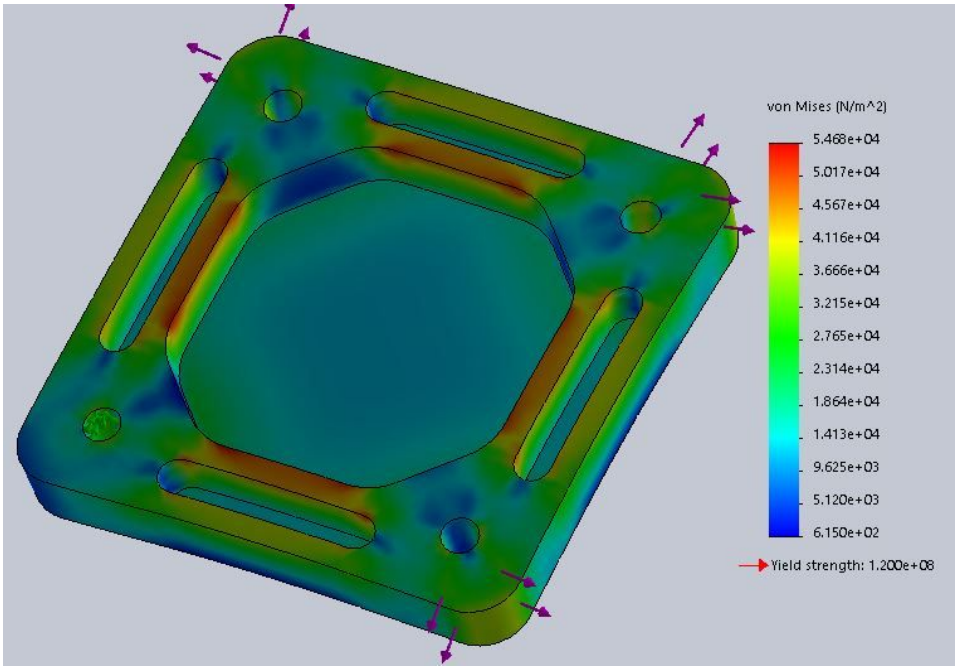


Figure 4.15: Design Alternative 3: Analysis of von Mises Stress showing low stresses in the center of the well and higher stresses at the sacrificial well edges

The more rounded center in the third design alternative showed a large area of low uniform stress in the center, but there was an increase in the area of high stress along the edges of the sacrificial wells indicting a greater chance of well failure.

Based on the FEA analysis of uniform strain area and SolidWorks analysis of von Mises stresses for each of the well design alternatives, our team decided to use the second design idea with 17.5 x 3 mm sacrificial wells on each edge as our final well design. This design was incorporated into the new mold design, described in section 3.

Section 4: Prototyping the Sacrificial PDMS Well Mold

The second alternate design with 17.5 x 3 mm sacrificial wells was chosen to create a new well mold. The purpose of creating a new well mold was to reduce the number of manufacturing defects on the wells to improve well durability.

A resin mold was created using a Form 2 Labs SLA printer for the new well design with sacrificial wells as seen in figure 4.16.

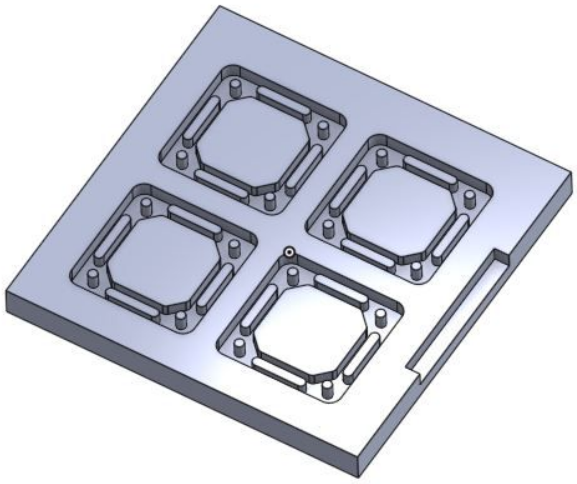


Figure 4.16: Well Mold for 3D Printing

Additive manufacturing was chosen as the method to form the mold prototype for the new wells due to the complexity of the well design(many small channels) that required clean, high resolution, lines. The comparison between machining an acrylic mold or printing a resin mold indicated that we could achieve a smoother surface finish by 3D printing with a resolution of 25 microns. Machining acrylic would result in dimensional changes and potential stress concentrators that could propagate and cause material failure (Plastics Distributor & Fabricator, 2019). While micromachining techniques could be used to achieve high resolution (0.1 micrometers) in some cases, the cost to generate parts is extremely high, often exceeding $2000 (Plastics Distributor & Fabricator, 2019; “CNC Micromachining”, 2019). Even by using a lower cost ($900) subtractive manufacturing machine, such as a CNC machine, the feature size is limited by cutter clearance (Zalewski, 2015). Our goal for developing a new mold was to

decrease stress concentrators in the mold to prevent failure of the wells. Therefore, we initially chose a 3D printed resin to achieve a high level of detail at the lowest possible cost. The 3D print could retain small features and eliminate post-processing altogether. Additional information for comparing the resin and acrylic materials and resin material properties is found in Appendix B.

A four-well mold was generated in SolidWorks then exported to an STL file and oriented in the PreForm print preparation software. Figure 4.17 shows the printed mold product which was used as a proof of concept for manufacturing the new wells.

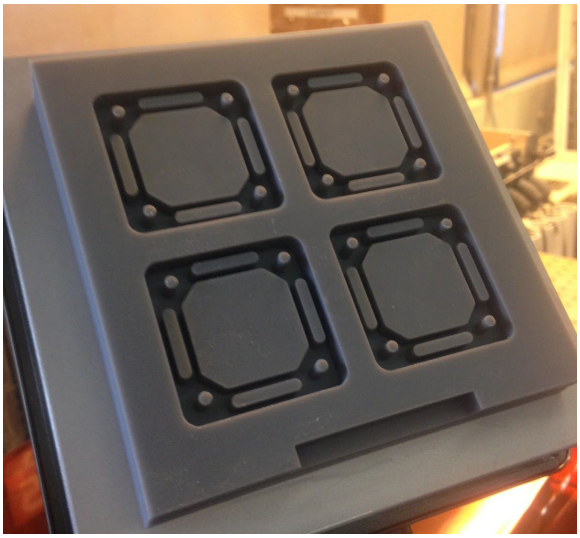


Figure 4.17: The 3D printed resin well mold.

Section 5: Alternative Device Designs

Subsection A: Validation Testing of 2013 Device

In order to determine the best design approach to fix the 2013 MQP device, we needed to find the cause of the issues impairing device performance. The motors did not go as fast or at the frequency set by the user. However, the motors should be able to run at the speed needed of 10mm/second. Appendix C contains the motor velocity specifications for the motors used in the 2013 device. Speed tests were conducted by timing the motors as they operated at various frequencies and strain percentages. The results of these tests are presented in in Tables 4.1 and 4.2 below.

Table 4.1: 2013 Device Set to 1 Hz and Tested for Time Elapsed at Various Strains

	10 Cycles	10 Cycles	10 Cycles
Set Strain	5% Strain	10% Strain	15% Strain
Trial 1	12.76 s	15.85 s	18.72 s
Trial 2	12.53 s	15.46 s	18.21 s
Trial 3	12.51 s	15.5 s	18.15 s
Average	12.6 s	15.6 s	18.36 s
Expected (s)	10	10	10
Actual Hz	0.79	0.64	0.55

As seen in Table 4.1, we set the device to run at 1 Hz frequency and tested various strains. The device was run for three trials of 10 cycles at each strain input. The data in the table above shows that increasing the strain will result in slower movement of the motors. The program prioritized moving the motors to the correct distance over the cycle frequency.

Table 4.2: 2013 Device Set to 10% Strain and Tested for Time Elapsed at Various Frequencies

Cycles Elapsed	5 Cycles	5 Cycles	10 cycles
Set Frequency	0.25 Hz	0.5 Hz	0.75 Hz
Trial 1	25.88 s	11.3 s	18.08 s
Trial 2	25.68 s	11.21 s	18.03 s
Trial 3	25.18 s	11.2 s	18.01 s
Average	25.58	11.23	18.04
Expected (s)	20	10	13.3
Actual Hz	0.2	0.45	0.56

As seen in Table 4.2, we set the device to run at 10% strain and tested various frequencies. The device was run for three trials at each frequency input. The trials at 0.25 and 0.5 Hz ran for 5 cycles, and the trials for 0.75 Hz ran for 10 cycles. As the results show, setting the motors to lower frequencies resulted in more accurate timing of the motors.

Based on the speed tests conducted, we determined that failure to achieve the set frequency was caused from a combination of inefficient code, a slow motor control processor, and inadequate power supply. The section of the code that was used to run the motors at the desired strain is found in Appendix D. The code uses three loops to tell the motor how to move in step degrees. To complete one cycle at 10% strain, the motors need to move 416 steps. Therefore, the code was rather inefficient because to move at 10% strain, the computer had to loop 416 times per cycle.

Additionally, we used an oscilloscope to evaluate the duty cycle of the motors. When the motors changed direction, the duty cycle increased which altered the operating frequency. The cause of this time lag was a lack of code to account for the time to change direction in the motor. To correctly account for the duty cycle, the time to change direction must be included in the

frequency calculation of the code. The current motor controller used in the device is a ChipKit Max 32, and its processor can only run at 80MHz (ChipKIT, 2018). We decided to buy an Arduino as a new motor controller as it has a faster processor that runs at 84MHz (Arduino, 2018). One alternative design that we developed uses two motors. This design would maximally reduce the total amount of code needed and the power required.

Subsection B: Motor Type Analysis

Upon analysis of the original device design from the previous MQP team, our team conducted time and frequency analyses on the device. Our findings showed that the device did not achieve the desired frequency for stretching. Initially, our team attempted to resolve this limitation by modifying the initial device. We first updated the Arduino control to have a higher processing speed, adjusted the power input in the code to increase the speed output of the device, we also tried to modify the CPU settings on the computer to increase the speed of processing and frequency. All of these methods failed to allow the device to reach the necessary 1 Hz frequency, so our team decided to re-build the device from scratch.

To address the concerns that the speed of the motors limited the maximum strain rate of 1 Hz, alternative motor types and actuator types were considered. Direct current, or DC, motors were determined to be necessary because these motors operate at a constant voltage (Hughes & Drury, 2013). Arduino boards, such as the one used to power this device and connect it to a computer, provide an output that runs at a constant voltage. This voltage is used to run any device run by the Arduino, and any “changes” in voltage are provided by rapidly fluctuating between having voltage and no voltage output from the Arduino. Arduinos utilize a constant voltage to power the attached motors. For stepper motors, such as the motors used by the existing

design, rotation is achieved in “discrete angular intervals” each time a command impulse is received; this angle can be anywhere from 1.8°-90° depending on the individual motor (Hughes & Drury, 2013). Stepper motors also hold the advantage of being controlled directly by computers and microcontrollers (Hughes & Drury, 2013). For the purpose of our design, we considered a stepper type motor that rotated at a smaller angle because thousands of steps per second are possible using this type of motor. Stepper motors, however, increase the complexity of coding the device, especially in the calculation for duty cycle (time lag for motor turn-around).

Servo motors, or brushless permanent magnet motors, are generally considered higher performance than stepper motors, but come with some serious limitations. The first of these limitations is that with continuous use, the motors rise can above the allowable temperature, which causes loss of copper and eventually iron in the motor. This material ion lost decreases its subsequent operating efficiency (Hughes & Drury, 2013). This device is intended to be used continuously for long periods of time (up to 24 hours), which makes this constraint a serious drawback. Moreover, through either continuous use or high speeds, field weakening can occur. Once the magnetic field is weakened, the motor can reach dangerous voltages that will both fry the motor and compromise user safety (Hughes & Drury, 2013). Overall these motors have a higher performance over stepper motors, but fail to operate effectively when used continuously. Stepper motors on the other hand can operate continuously and have improved motion precision and produce higher forces.

Linear actuators are used to convert motor rotation into linear motion (Lewotsky, 2007). The two main types of linear actuators are screw-driven actuators and belt-driven actuators.

Screw-driven actuators are best for heavier loads used for short durations (Lewotsky, 2007). They are quiet and less prone to back driving due to the use of screw threads. However, due to a tendency for the screws and nuts to have high friction, they wear quickly and become slow and inefficient (Lewotsky, 2007). Belt-driven actuators are beneficial for use with low forces in the horizontal direction (Lewotsky, 2007). They are faster but slightly less accurate than the screw systems (Lewotsky, 2007). Linear actuators are small in size and overall profile. The size can be reduced further when micro-linear actuators are used. Our device required only a small force output at specific displacements. For coding purposes, calculation and coding of the duty cycle could more easily be implemented with the use of linear actuators. Moreover, the step angle does not have to be calculated or coded, only a displacement input from the user. These factors greatly can increase the efficiency of the code processing. Moreover, because the original device failed to reach the desired strain rates due mainly to processing speed, a more efficient code would most effectively target the past device shortcomings. Due to the small size and necessary force output and accuracy for our device, screw-driven linear actuators were determined to be the most appropriate motor choice.

To further validate our decision to build the device with linear actuators, our team considered the cost and exact specifications of different linear actuators compared to stepper motors. The amount of force needed to deform the stretch wells at the appropriate speed falls at or below 25 pounds force similar to the motors used in the previously created stepper motor design. Thus, we anticipated we could purchase low force linear actuators that would be the most affordable to suit their purpose. One potential model, SOVik 6” small stroke linear actuator has a 12V DC motor speed deforming at 16.002 mm/s and can accomplish a 44 lb maximum lift force

(Amazon, 2019). Even so, the force required to displace the PDMS wells at the maximum 20% strain nears only 1 pound of force. Searching for linear actuators that produce a low force output, we discovered that micro-linear actuators could perform at the desired force output and speed to deform the PDMS wells at the appropriate strains and rates. These micro-linear actuators were more affordable, $69.99, than the stepper motors, $175.43, and the regular linear actuators, ~$100-$150(Lewotsky, 2007; Progressive Automations, 2019).

Section 6: Comparing Device Configurations

One limitation to the previous stepper motor device design was the L-shaped brackets that connected the motors to the stretching wells. These L-shaped brackets protruded beneath the well surface causing interference with the microscope objective. This was not ideal because it limited the amount of imaging area available by obstructing the objective lens. Additionally, it introduced the potential to scratch the objective lenses. To fix this shortcoming, our team decided to flip the L-shape bracket orientation so that wells were loaded from underneath the device and the metal brackets remained above the silicone rubber stretching well. This allowed for the PDMS membrane to be the only contacting surface in the line of the microscope objective. By doing so, we could prevent lens scratching and increase the total area that could be imaged.

Our first proposed design configuration consisted of a two-motor design utilizing stepper motors and a drive belt. This model replicated the market gold standard, Strex Cell, model. The device contains two stepper motors attached to threaded rods that are axially driven with rod rotation to achieve x and y stretching (Duoba et al., 2012). Figure 4.18 illustrates the configuration described. In the figure, the stepper motors are represented by red arrows, the threaded rods are represented by green arrows.

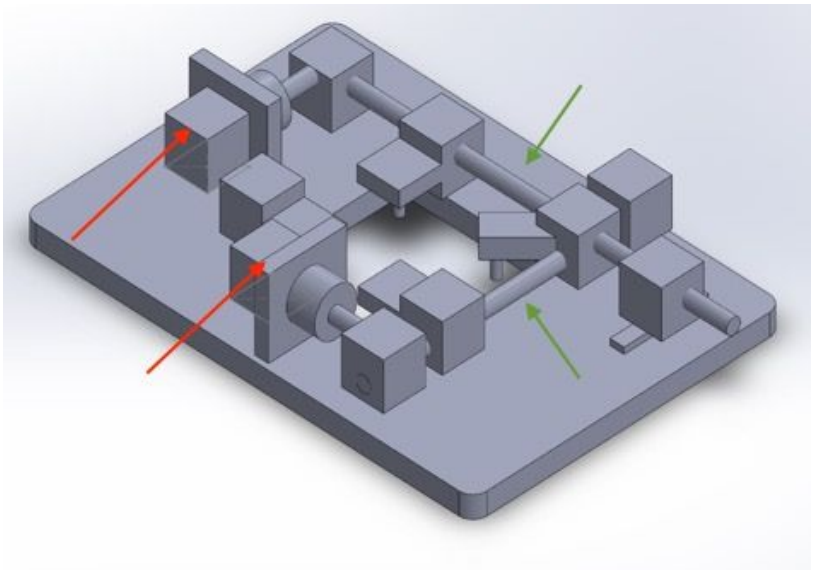


Figure 4.18: Two Stepper Motor Design Alternative. The red arrows point to the stepper motors, the green arrows point to the threaded rods.

This design utilizes two linear motors to move pegs affixed to the PDMS well in order to create an equibiaxial strain. The motors are attached to sliders, which are attached to rods that move the pegs. One motor moves two pegs, the upper right and lower right pegs, in the x-direction while the other motor moves two pegs, the lower left and lower right pegs, in the y-direction. The upper left peg remains stationary. This type of model was theorized previously by the 2012-2013 MQP group. This model draws less power than the original four motor design by cutting the number of motors in half. Additionally, because the use of stepper motors the step angle and duty cycle has to be re-calculated for each additional motor, reducing the total motors would improve the processing efficiency. This is ideal for use of an Arduino as the device controller. Eliminating two motors also creates more space on the microscope stage allowing more room for the condenser. This design idea also makes an additional modification by inverting the pegs that attach the well so that the well is loaded from the bottom instead of the top. As previously described, this clears space for the microscope objectives.

To account for our team's motor selection, we sketched a second design idea consisting of two linear actuator motors in CAD. Figure 4.19 shows a labeled CAD drawing.

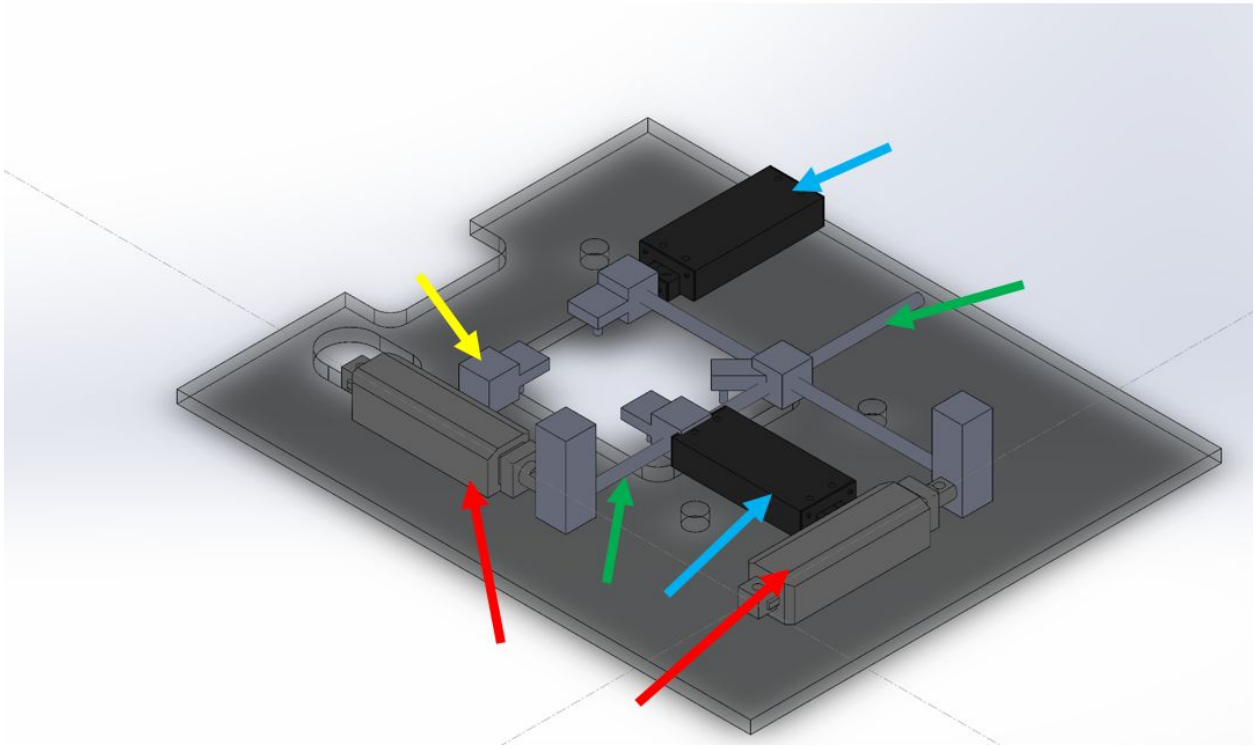


Figure 4.19: Two motor design with linear actuators. The linear actuators are identified by the red arrows. The blue arrows indicate the sliders. The green arrows indicate the rods. The yellow arrow points to the stationary peg.

The two linear actuator design retains the overall configuration of the device, but replaces threaded rods with smooth connecting rods that slide in and out of bushings. Linear slides guide the motion of the corner brackets attached to the stretching well to achieve either uniaxial or biaxial motion. By using linear actuators, this design simplifies the coding parameters by eliminating the step angle calculation, and simplifying the duty cycle calculation. The resulting code can be processed more efficiently by the Arduino controller. The linear actuators also

provide an extremely low profile that would improve the fit of the device on the microscope stage, allowing for more room for the condenser of the microscope.

After modeling the device in SolidWorks, early prototypes the brackets and connector pieces were 3D printed and a model of the base was laser cut out of plywood. Figure 4.20 shows the assembled version of this prototype.

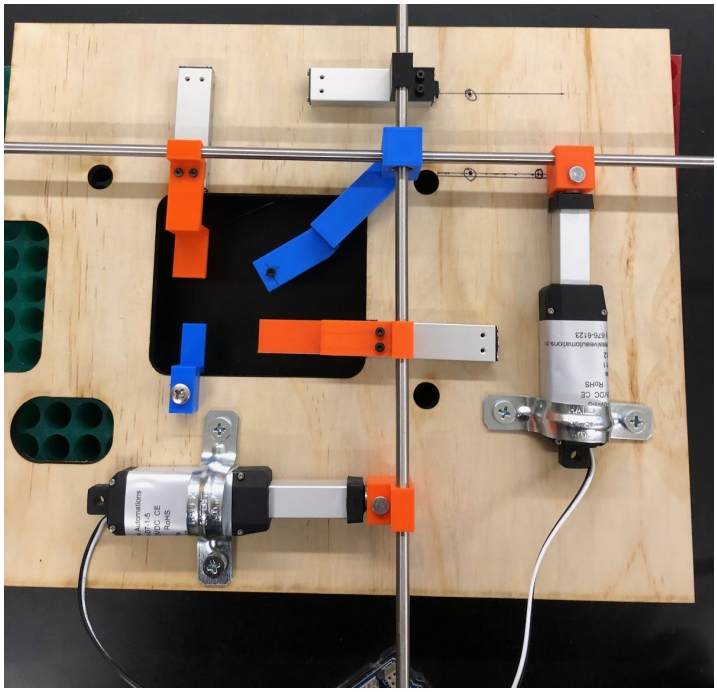


Figure 4.20: Prototype of Two Linear Actuator Design

After initially testing the prototype, it was concluded that the motors created a moment arm about the connector rods. To fix this, the team tested a single orientation of the motors and connector rods with a second linear slider attached to the far end of the rod. This led to increased stability of the rod, so a second set of two linear sliders were ordered.

The limitation to this design is that even with the additional slider support, the two linear actuators push on the end of a long connector rod which creates a lever arm. We can attempt to

correct this moment with linear slides at the opposite end of the rod; however, this is not guaranteed to resolve the uneven forces on either side of the connecting rod.

In order to reduce the moment arm from the two linear actuator design, we created another design alternative with three linear actuators, seen below in figure 4.21.

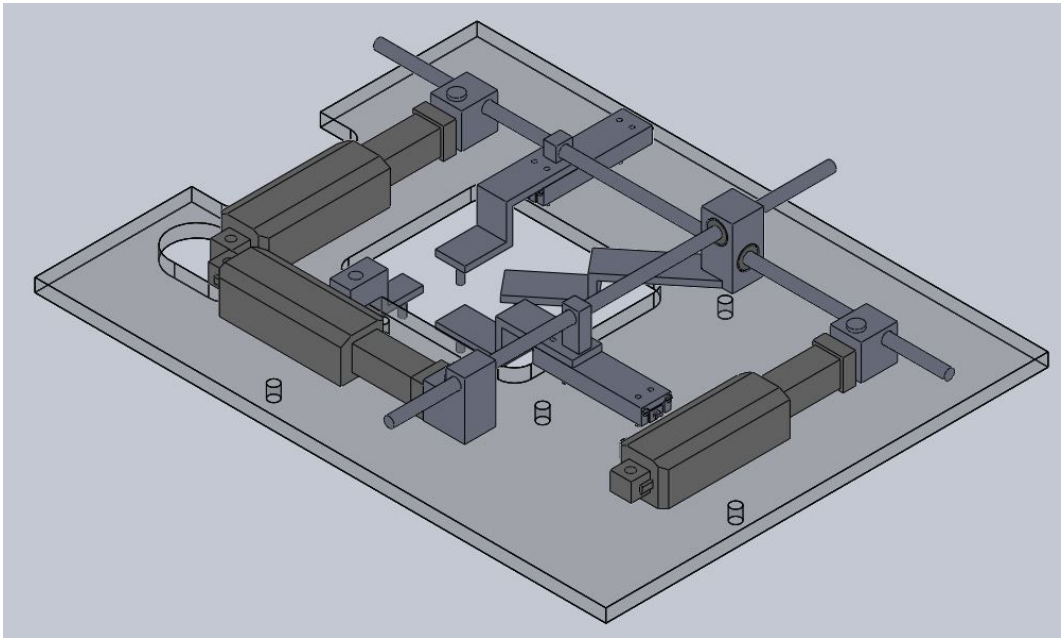


Figure 4.21: Three Linear Actuator CAD Design.

A linear actuator was added to the left side of the device to create a parallel force to the right side linear actuator. The additional linear actuator was intended to balance the force to significantly reduce the moment and improve the tolerance of the device. As a result, the error margin for strain was could also be reduced. Nonetheless, this device design alternative did not reduce the level arm in the opposite direction due to an imbalanced force. Thus, we considered another design alternative for our final device design.

A four linear actuator design alternative was created in CAD as seen in figure 4.22 below.

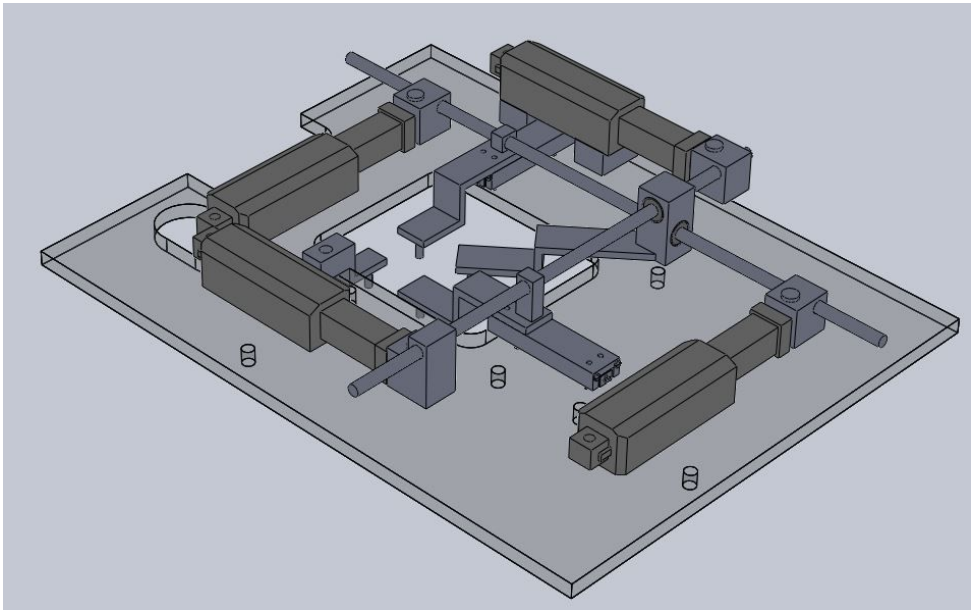


Figure 4.22: Four Linear Actuator CAD Design.

This four linear actuator design would correct for the lever arms created by the two and three linear actuator designs by providing a parallel force to offset the moment. The fourth linear actuator was added above one of the linear slides to preserve space and allow for the device to fit on the microscope stage. By balancing all forces in two-directions, the uniformity of the strain could be improved. When the prototype of this design was placed on the microscope, the entirety of the device cleared the microscope condenser. Furthermore, the device did not interfere with any microscope function.

In each of the CAD models we created to represent our design ideas for the device, the baseplate, linear actuators, and slides were drawn to scale. The base plate is 213 mm by 275 mm. Three small holes are in the baseplate for screws so an insert for the microscope stage can be attached to the baseplate. The insert would allow for the baseplate to comfortably fit within different microscope stages. The large hole in the baseplate is so that the baseplate doesn't

interfere with the peg on the microscope stage in Billiar's lab. The motors are 5 inches (127 mm) long and 1 inch (25.4 mm) wide.

Subsection A: Cost Analysis of Device Design Alternatives

To help determine the type of mechanical driver for the final design assembly, we analyzed the cost of each of our device design alternatives.

The four motor stretching device created by the 2012 MQP team can provide uni- or biaxial cell stretching. The device should have been able to provide strain rates between 0.1%-30% with a frequency range of 0.01 Hz-1 Hz, however, upon validation testing by our team, the device could only support a ~10 % strain with 0.5-0.7 Hz frequency. Figure 4.23 displays the four motor device CAD design.

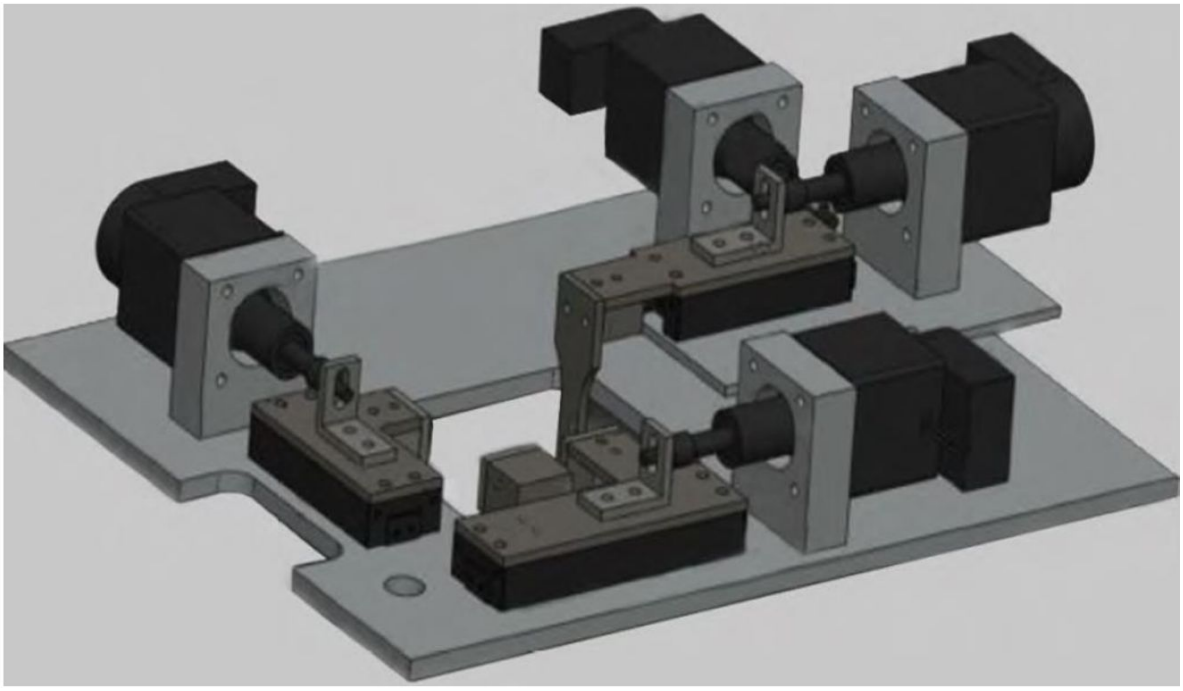


Figure 4.23: 2012 MQP Four Motor Uniaxial and Biaxial Cell Stretching Device CAD drawing.

The cost for this design was obtained from the Worcester Polytechnic Institute Electronic Projects Collection and is displayed in Table 4.3 below (Duoba et al., 2012).

Table 4.3: Four Motor Device Budget

Bill of Materials Table

Materials						
Date	**Company**	**Items**	**Quantity**	**Unit Cost**	**Shipping Cost**	**Notes**
11/28/2011	SSP Inc.	SSP-M823 Silicone Sheeting 0.004" x 12" x 12"	1	$30.00	$12.65	
1/18/2012	Scrap Yard	Aluminum Stock (1/8 in., 1/4 in., 5/8 in. plate)	1	$0	$0	Donated
1/18/2012	Scrap Yard	Stainless Steel (1/8 in., 1/2 in. plate and 3/16 in. angle)	1	$0	$0	Donated
1/20/2012	SparkFun	PRT-09684 Chameleon Face Plate Arduino	1	$4.95	$4.35	
1/20/2012	SparkFun	PRT-10746 Small Heastsink with Thermal Tape	4	$7.95	$4.35	
1/20/2012	SparkFun	ROB-10735 - Big Easy Drivers	4	$22.95	$4.35	
1/23/2012	Hammond Enclosures	Steel Enclosure (1458 Series)	1	$0	$0	Donated
1/25/2012	Haydon Kerk	35000 series size 14 stepper motor linear actuators with quadrature encoders	4	$0	$0	Donated
1/27/2012	Surplus Sales of Nebraska	CNES321AB 21 pin female connectors	4	$6.00	$12.25	
1/27/2012	Surplus Sales of Nebraska	CNEP321CCT-K 21 pin male connectors	4	$17.00	$12.25	
1/27/2012	12V Adapters	12V - 5Amp adapter	1	$32.98	$0	
2/3/2012	RobotShop	RB-Spa-616 Big Easy Driver Bipolar Motor	1	$22.95	$13.11	
2/6/2012	Electronics Salon	5 Amps Voltage Regulator Module	1	$19.80	$0	
3/15/2012	Ace Hardware	#6-32 1/4 in. Cap Screws	6	$0	$0	Donated
3/15/2012	Ace Hardware	#6-32 1/8 in. Cap Screws	26	$0	$0	Donated
3/15/2012	Ace Hardware	#4-40 1/2 in. Flathead Screws	16	$0.27	$0	
3/15/2012	Ace Hardware	#2-56 1/2 in. Flathead Screws	4	$0.27	$0	
3/15/2012	Ace Hardware	#2-56 1/2 in. Female Standoffs	4	$0	$0	Donated
3/15/2012	Ace Hardware	M3x.05 16mm Flathead Screws	16	$0.27	$0	
3/15/2012	Ace Hardware	#4-40 1/2 in. Female Standoffs	20	$0.27	$0	
4/2/2012	Digilent	ChipKit Max32	1	$49.50	$17.80	
Total				**$390.90**	**$81.11**	

Even though the total cost of this device is $472.01 with shipping, the four motors were donated and thus are not included in the final budget. If we chose new motors, four new motors would require purchasing driving up the cost of the design. Each of the four motors in this design cost $175.43(Haydon Kerk, 2019). In total, to purchase four new motors would cost $701.72, making the total device cost $1173.73 with shipping.

A two linear actuator design approach would decrease the overall cost of the device. By purchasing additional motors, the overall budget would increase and more time would be required for complete reassembly. Despite increasing the budget, the total cost of a two linear actuator design would still be less than the $1000 overall budget. It is also less than the four motor design if it were to be built from scratch. An additional advantage to rebuilding a two linear actuator device alongside the existing device is that the existing device can be used in

other research labs, making it more cost effective. Moreover, purchasing new parts to create a separate assembly could help to preserve the function of the four motor device for research during construction and coding. A complete budget to assess the price of the 2 linear actuator design is outlined in Table 4.4.

Table 4.4: Two Linear Actuator Design Budget

Parts	Part ID #	Cost (Total for Multiple Parts)
1 MultiMoto Arduino Shield	LC-82	$48.99
2 Micro Linear Actuators	PA-07	$139.98
4 Linear Slides		$324.00
2 Connector Rods		$10.00
Spool of PLA for 3D prints	BRK-07	$13.99
2 Bushings		$24.00
1 12V DC Power Supply	AC-15	$84.00
0.25" Aluminum Sheet (12" x12")		$25.00
Various Other Hardware		$15.00
Total Cost		$684.96

The total cost to build a complete two-motor device is $684.96 including the cost of shipping.

A three linear actuator device retains the features of the two linear actuator device and reduces the error margins for producing strain by lessening the moment of the actuators pushing on each connecting rod. This configuration requires only a few more lines of code that are replicated from the parallel motor with only a new variable name. Thus, adding code is nearly

insignificant to the feasibility of writing the code. The addition the the linear actuator would increase the overall cost of the device design by $69.99, but would also require to fewer linear slides to create a total of $618.42 including shipping. The total budget for this design is represented in table 4.5 below:

Table 4.5: Three Linear Actuator Design Budget

Parts	Part ID #	Cost (Total for Multiple Parts)
1 MultiMoto Arduino Shield	LC-82	$48.99
3 Micro Linear Actuators	PA-07	$209.97
2 Linear Slides	DA-02	$162.00
2 Connector Rods		$10.00
Spool of PLA for 3D prints	BRK-07	$13.99
2 Bushings	LMU-5	$24.00
1 12V DC Power Supply	AC-15	$84.00
0.25" Aluminum Sheet (12" x12")		$35.47
Various Other Hardware		$30.00
Total Cost		$618.42

A four linear actuator design retains the advantages of the three linear actuator design but improves the error margins for the produced strain by minimizing the moment created by the actuator pushing on the connecting rod. The moment is minimized because equal force is applied by two separate linear actuators on each end of the connecting rod to balance the forces on each end. To create a four linear actuator design, four linear actuators and two linear slides are required. The additional actuator increases the cost of the overall budget compared to the three

linear actuator design by $69.99. The total overall budget for a four linear actuator design is $688.41. The overall budget is presented in table 4.6 below.

Table 4.6: Four Linear Actuator Device Budget

Parts	Part ID #	Cost (Total for Multiple Parts)
1 MultiMoto Arduino Shield	LC-82	$48.99
4 Micro Linear Actuators	PA-07	$279.96
2 Linear Slides	DA-02	$162.00
2 Connector Rods		$10.00
Spool of PLA for 3D prints	BRK-07	$13.99
2 Bushings	LMU-5	$24.00
1 12V DC Power Supply	AC-15	$84.00
0.25" Aluminum Sheet (12" x12")		$35.47
Various Other Hardware		$30.00
Total Cost		$688.41

Section 7. Final Design Selection

Final device selection was proposed to contain four linear actuators. This design uses four PA-07 Micro linear actuators from Progressive Automations, two 5mm bearings from MISUMI, and two DA-2 slides from Deltron, with one fixed stretch-cell corner. The PDMS well mold was redesigned with sacrificial wells on every wall to improve the area of true biaxial strain. The mold that forms the wells was also modified to reduce imperfections which are mirrored in the stretch wells and act as stress concentrators. The addition of the sacrificial wells should also reduce the percentage of failure events. The device was designed within the limitations of the size of the microscope stage mount. The new base plate was manufactured out of 0.125”

aluminum. The new device aimed to achieve 10% strain at 1 Hz frequency for biaxial stretching. The motors from the previous design were replaced by micro linear actuators and new corner pins were manufactured to connect the linear actuators to the PDMS well. New code was written to include biaxial and alternating patterns based on the extending and retracting rates of the linear actuators. Finally, a new graphical user interface was designed and coded to enhance the usability of the device.

The linear actuators selected were the PA-07 Micro Linear Actuators by Progressive Automations. They can be seen in Figure 4.24 below. They have a stroke length of 1”, a max speed of 0.59”/second, and a max force of 5lbs.

Figure 4.24: PA-07 Linear Actuator Motors
(Copyright 2019 Progressive Automations Inc.).

Appendix C shows the velocity vs. force plot for the chosen linear actuators, along with profiles for similar models of actuators.

The linear slides were DA-2 ball-bearing slides from DelTron. They are metric and have a travel distance of 25mm, a max load of 4kgs, and were selected for their low profile and smooth movement. An image of them can be seen in Figure 4.25 below.

Figure 4.25: DA-2 Ball Bearing Slides
(Copyright 2019 Del-tron Precision, Inc.).

Subsection A: Development of a GUI

The new graphical user interface (GUI) contains a home button, run button, stop button, alternating strain button, y-axis strain slider, x-axis strain slider, and frequency slider. Figure 4.26 displays the exact configuration. The code for this display is shown in Appendix F.

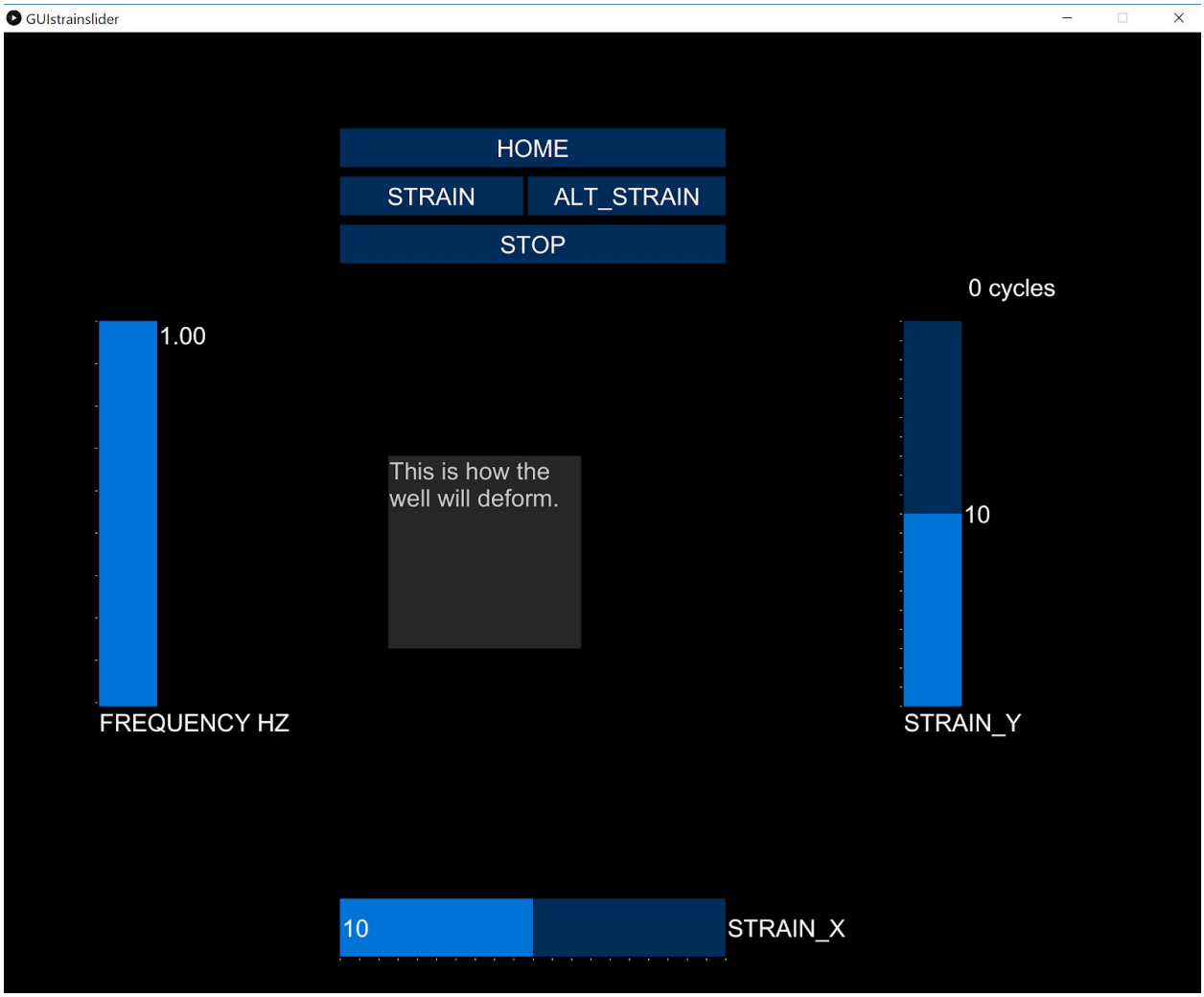


Figure 4.26. Display of the layout and features in the GUI.

The home button fully retracts the actuators, so the PDMS well can be placed into the pegs without stretching the well. The strain button and alt strain buttons trigger the arduino to run a specific strain pattern. The strain button runs the normal non-alternaternating pattern, moving in both directions at the same time for biaxial deformation; while the alt strain button runs the alternating pattern straining in one direction then the other direction (i.e. x direction then y direction). The *pwm_1* and *pwm_2* set the power scale input to the motors. These values regulate the strain magnitude and frequency by controlling the velocity of the motor. Specific values are

tabulated using the x and y strain provided by the GUI. The delay function takes the frequency from the GUI and sets the time the motor is extending and retracting. For 10% strain at 1 Hz, *pwm_1* and *pwm_2* both equal 187 after the power to strain equation we developed converts the strain to power. The delay equals 500 because in the code, delay(1000) equals a 1 second interval, so the time of extension is set to half a second and the delay is set to half a second to equal a 1 Hz frequency. The stop button causes the linear actuators to finish the current cycle then stop running. The GUI contains two strain sliders one for the x-direction and one for the y-direction. Each slider is set to a default 10% strain, but can be modified by clicking clicking higher or lower. The sliders are set to have an increment of 1% strain; however, this could be altered by changing the number of tick marks within each slider in the GUI code:

cp5.addSlider("Strain_Y")

.setNumberOfTickMarks(20)

The current number of tick marks is set to 20 so that the strain can be set from 1-20%. The code also saves the amount of cycles the device has performed, retrieving the data from the arduino. The central grey box gives a visual display of the PDMS well displacement. It is coded as a text box wherein the dimensions of the text box change with the X and Y strain values. The entire code was written in the program 'Processing 3.5.3' which operates the language Java, but can be easily exported to *.pde* which can be can controlled by Arduino software.

Based on the strain values input by the user through the GUI, the code will perform a set of conversion equations to translate that strain into the appropriate amount of extension and retraction for each linear actuator independently. Based on the strain-to-power equation developed for the code to determine how quickly the linear actuator motors move, a set of

pre-calculated strains and power values were determined. The complete list of these values is available in appendix I. For a 1 Hz frequency setting, the maximum attainable strain is 13.6%.

Micro-pulses control the actual movement of a linear actuator. These pulses define the time of movement of the linear actuator stroke. The range 0-255 is preset scalar for 8-bit translation of a 0-100% scale for pulse duration. The pulse duration is minimized when this value is set to 0 in code and maximized when set to 255. This scale cannot exceed 255; therefore, the only way to increase the strain is by purchasing a linear actuator with a higher maximum velocity. The equation we used to calculate the duration of time that power is "on" for the micro-linear actuator is defined in Equation 1 below.

Equation 1

Duration of Power to Motor = (((((Strain Input/2)*0.1)+).5*Strain Input)/25.4)*2)/0.0023137*2 *(Frequency Input)

Our code interprets strain input by the user with this equation. For the specific micro-linear actuators purchased and used in our device, the maximum pulse duration is attained at a 13% strain input parameter. Overall, based on this equation, as strain rate (Hz) increases, the total attainable strain(%) by the motors decreases because the pulse duration is limited. A reference chart for determining the allowable strain and frequency inputs for this device is listed in Appendix I. Furthermore, a diagram of the wiring for the Arduino/MultiMoto controller setup can be found in Appendix K.

In the code there is a variable named "power to speed" that uses equation 1 to produce the appropriate strain output. The exact code is described in sub-section B.

Subsection B: Code Modifications

Along with values for the x strain, y strain, and frequency, a selector value is sent from the GUI to the arduino for telling the arduino what program to run. These are split the three cases in the arduino code. The first, is for the regular strain command but it is also for the stop command. The stop command just tells the arduino to run at 0 strain in this loop. The second is for the home command, which just tells the arduino to fully retract the motors at full speed until they are fully retracted, Finally, the third case is for the alternating strain command. It is set up similar to the run command loop, but has delays between the x and y strain corresponding to the frequency value.

The following code runs the home program automatically when the GUI is launched. The *dir* variables are used for what the *digitalWrite* function sets as the direction of the motors. If dir=0, the motor retracts, and if dir=1, the motor extends. The *pwm* variables are used for what the *analogWrite* function sets as the speed of the motors. The speed is broken up from 0-255, with 255 equaling about .59 inches per second. The delay function sets how long the motors will obey their previously given instructions. A delay of 1000 equals about 1 second. To set the motors to the home position, this program sets both dir variables to 0 to retract and sets both pwm variables to the max of 255 and the deley function at the bottom makes the motors retract at full speed for about 10 seconds.

Initial Set-up code (automatically runs in setup portion of arduino code):

```
dir1=0;
  pwm1=255;
  digitalWrite(DIR_M1, dir1);
  analogWrite(PWM_M1, pwm1); // write to pins
  dir2 = 0;
  pwm2 = 255;
```

```
digitalWrite(DIR_M2, dir2);
analogWrite(PWM_M2, pwm2);
dir3 = 0;
pwm3 = 255;
digitalWrite(DIR_M3, dir3);
analogWrite(PWM_M3, pwm3);
dir4 = 0;
pwm4 = 255;
digitalWrite(DIR_M4, dir4);
analogWrite(PWM_M4, pwm4);

delay(5000);
```

The following code runs the strain and stop button actions in the GUI. Similar to the home code, this code utilizes digitalWrite, analogWrite and delay functions. The first section of the code extends the motor for the x-strain. The second section of the code extends the motor for the y-strain. Both sections of code run simultaneously for the first half of the cycle. The third section of the code retracts the motor for the x-strain. The fourth section of the code retracts the motor for the y-strain. Both sections of code run simultaneously for the second half of the cycle.

Code for running non-alternating strain-pattern and stop command (currently executed with the "strain" and button in GUI respectively):

```
case 1:   //IMPORTANT: code for running strain and stop actions (stop program has power input values for x and y strain as 0, so actuators don't move during either section of this code)
while(Serial.available() ==0){
while(i==1){
        //all actuators extend in this step
        dir1 = 1; //sets direction of motor. 0=retract, 1=extend
        pwm1 = xStrainValue; //sets speed of motor. 0 is minimum value, 255 is the maximum value.
        digitalWrite(DIR_M1, dir1);
        analogWrite(PWM_M1, pwm1); // write to pins, tells motors
        dir2 = 1;
        pwm2 = yStrainValue;
        digitalWrite(DIR_M2, dir2);
        analogWrite(PWM_M2, pwm2);
        dir3 = 1;
        pwm3 = yStrainValue;
        digitalWrite(DIR_M3, dir3);
```

```
        analogWrite(PWM_M3, pwm3);
        dir4 = 1;
        pwm4 = xStrainValue;
        digitalWrite(DIR_M4, dir4);
        analogWrite(PWM_M4, pwm4);
        delay(timePerCycle*500); // length of time motors run this step for, 1000=1 second
        i=i+1;
}
while(i==2){
        //all actuators retract in this step
        dir1 = 0;
        pwm1 = xStrainValue;
        digitalWrite(DIR_M1, dir1);
        analogWrite(PWM_M1, pwm1);
        dir2 = 0;
        pwm2 = yStrainValue;
        digitalWrite(DIR_M2, dir2);
        analogWrite(PWM_M2, pwm2);
        dir3 = 0;
        pwm3 = yStrainValue;
        digitalWrite(DIR_M3, dir3);
        analogWrite(PWM_M3, pwm3);
        dir4 = 0;
        pwm4 = xStrainValue;
        digitalWrite(DIR_M4, dir4);
        analogWrite(PWM_M4, pwm4);
        delay(timePerCycle*500);
        i=i-1;
  }
}
break;
```

The following code runs the home and stop button function. In the home command, it completely retracts the actuators at the maximum velocity. In the stop command, it completely stops the motors. Both actions occur after the current cycle is completed.

Code for running home command (currently executed with the "home" and "stop" button in GUI):

```
case 2:  //IMPORTANT: code for running home action
        // all actuators retract at max speed until they are fully retracted
        dir1 = 0;
        pwm1 = xStrainValue; //set direction and speed
```

```
        digitalWrite(DIR_M1, dir1);
        analogWrite(PWM_M1, pwm1); // write to pins
        dir2 = 0;
        pwm2 = yStrainValue;
        digitalWrite(DIR_M2, dir2);
        analogWrite(PWM_M2, pwm2);
        dir3 = 0;
        pwm3 = yStrainValue;
        digitalWrite(DIR_M3, dir3);
        analogWrite(PWM_M3, pwm3);
        dir4 = 0;
        pwm4 = xStrainValue;
        digitalWrite(DIR_M4, dir4);
        analogWrite(PWM_M4, pwm4);
        delay(timePerCycle*500);  // length of time motors run this step for
break;
```

The following code runs the alternating strain button action, *alt strain*. The alternating strain portion of the code, takes two strain values for the x and y direction and deforms the well in one direction then the other. This creates an alternating pattern (x then y or vice versa) that can be used to test the adaptation behavior of cells. Based on these inputs the code signals to move the linear actuators at different strains in each direction. It is divided into four sections: the first extends the actuators oriented in the x-direction, the second retracts these same actuators, the third extends the actuators oriented in the y-direction, and the fourth retracts these actuators.

Code for running alternating strain-pattern (currently executed with the "alt strain" button in GUI):

```
case 3:  //IMPORTANT: code for running alt strain action
while(Serial.available() ==0){
while(i==1){
  // actuators in x direction extend
        dir1 = 1;
        pwm1 = xStrainValue; //set direction and speed
        digitalWrite(DIR_M1, dir1);
        analogWrite(PWM_M1, pwm1); // write to pins
        dir4 = 1;
        pwm4 = xStrainValue;
        digitalWrite(DIR_M4, dir4);
```

```
        analogWrite(PWM_M4, pwm4);
        dir2 = 1;
        pwm2 = 0;
        digitalWrite(DIR_M2, dir2);
        analogWrite(PWM_M2, pwm2);
        dir3 = 1;
        pwm3 = 0;
        digitalWrite(DIR_M3, dir3);
        analogWrite(PWM_M3, pwm3);
        delay(timePerCycle*500);  // length of time until motors run this step for
        i=i+1;
}
while (i==2){
  // actuators in x direction retract
        dir1 = 0;
        pwm1 = xStrainValue; //set direction and speed
        digitalWrite(DIR_M1, dir1);
        analogWrite(PWM_M1, pwm1); // write to pins
        dir4 = 0;
        pwm4 = xStrainValue;
        digitalWrite(DIR_M4, dir4);
        analogWrite(PWM_M4, pwm4);

        dir2 = 1;
        pwm2 = 0;
        digitalWrite(DIR_M2, dir2);
        analogWrite(PWM_M2, pwm2);

        dir3 = 1;
        pwm3 = 0;
        digitalWrite(DIR_M3, dir3);
        analogWrite(PWM_M3, pwm3);
        delay(timePerCycle*500);  // length of time until motors run this step for
        i=i+1;
}

while(i==3){
// actuators in y direction extend
        dir1 = 0;
        pwm1 = 0;
        digitalWrite(DIR_M1, dir1);
        analogWrite(PWM_M1, pwm1);
        dir4 = 0;
        pwm4 = 0;
        digitalWrite(DIR_M4, dir4);
```

```
        analogWrite(PWM_M4, pwm4);
        dir2 = 1;
        pwm2 = yStrainValue;
        digitalWrite(DIR_M2, dir2);
        analogWrite(PWM_M2, pwm2);
        dir3 = 1;
        pwm3 = yStrainValue;
        digitalWrite(DIR_M3, dir3);
        analogWrite(PWM_M3, pwm3);
        delay(timePerCycle*500);  // length of time until motors run this step for
        i=i+1;
}

while(i==4){
// actuators in y direction retract
        dir1 = 0;
        pwm1 = 0;
        digitalWrite(DIR_M1, dir1);
        analogWrite(PWM_M1, pwm1);
        dir4 = 0;
        pwm4 = 0;
        digitalWrite(DIR_M4, dir4);
        analogWrite(PWM_M4, pwm4);

        dir2 = 0;
        pwm2 = yStrainValue;
        digitalWrite(DIR_M2, dir2);
        analogWrite(PWM_M2, pwm2);
        dir3 = 0;
        pwm3 = yStrainValue;
        digitalWrite(DIR_M3, dir3);
        analogWrite(PWM_M3, pwm3);
        delay(timePerCycle*500);
        i=i-3;
  }
}
break;
```

The code and GUI will be transferable to a new computer or laptop for ease of use. In order to do this, the user must download Arduino's IDE program and Processing with its library "ControlP5". The Arduino code used to run the motors is displayed in total in Appendix G.

Subsection C: Baseplate & Corner Pin Designs

The base plate dimensions, seen in figure 4.27 below, are made to fit the maximum space provided by the Zeiss microscopes. The large middle hole is where the PDMS well is placed. The medium sized hole to the lower left of it is meant to be clearance for the pin in the base of Professor Billiar's Zeiss microscope. The three small holes are where an insert can be screwed underneath the baseplate to help the device fit better into the microscope bases.

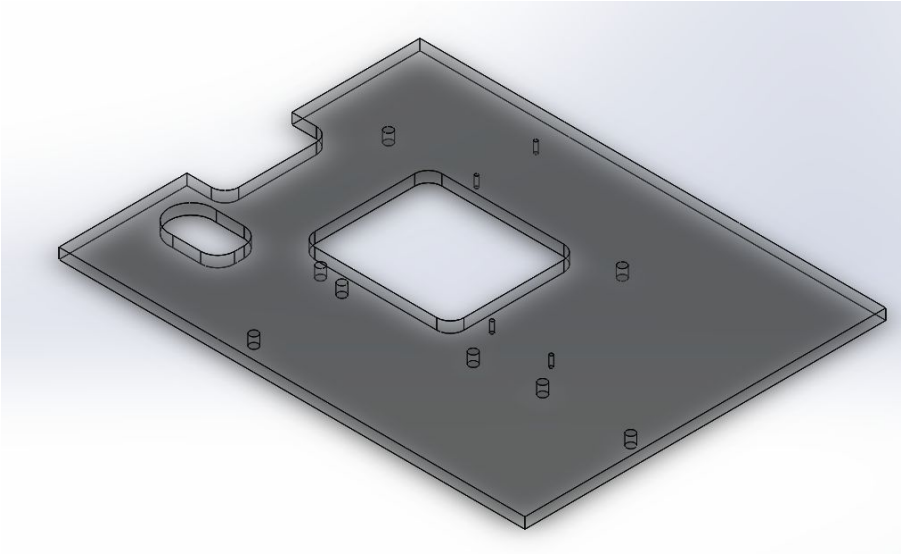


Figure 4.27: Base Plate CAD Drawing, Isometric View

The fixed corner piece shown in Figure 4.28 isn't attached to a motor so it fixes one of the corners of the PDMS well in place while the other corners are stretched.

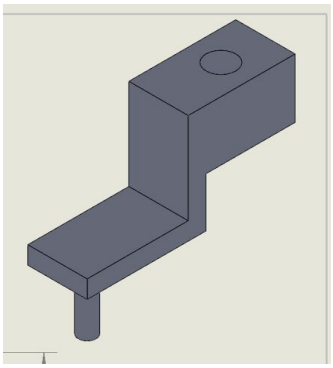


Figure 4.28: Fixed Corner Pin CAD Drawing

The Right Side Slider Bracket shown in Figure 4.29 is attached to one of the linear sliders to move the device in the x-direction.

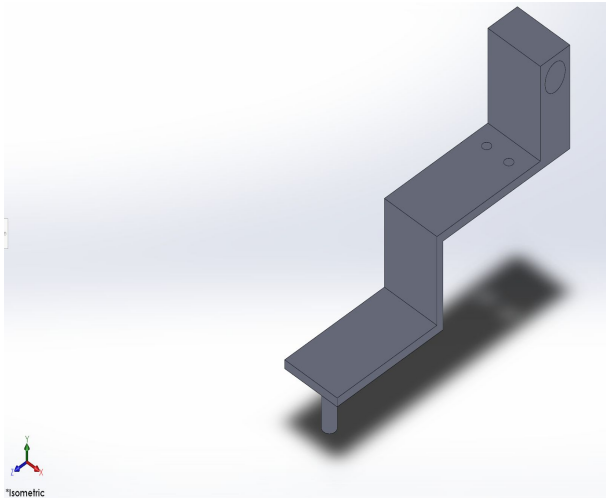


Figure 4.29: Right Side Slider Bracket CAD Drawing

The Top Side Slider Bracket shown in Figure 4.30 is attached to one of the linear sliders to move the device in the y-direction.

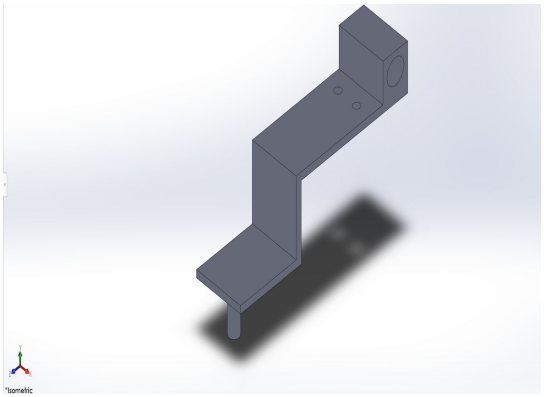


Figure 4.30: Top Slider Bracket CAD Drawing

The Corner Bracket shown in Figure 4.31 is free floating and move the corner the farthest away from the fixed point in both the x and y-directions.

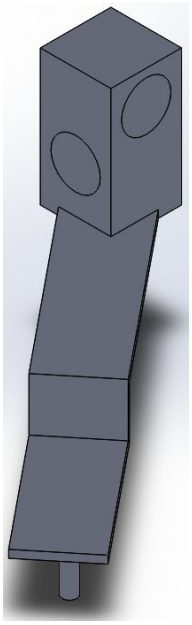


Figure 4.31: Top Right Corner Bracket CAD Drawing

Subsection D: Final Assembly

Figure 4.32 below shows a model of the final device design. Motors are fixed to the base plate and push stainless steel rods that connect to the brackets and sliders.

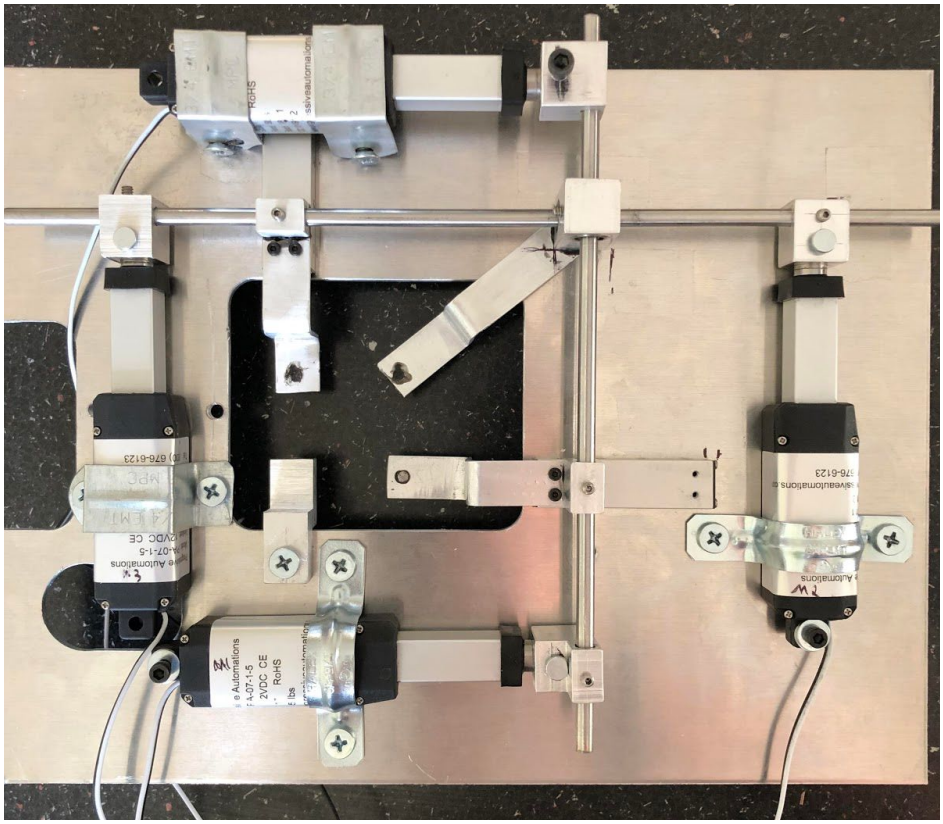


Figure 4.32: Finished Construct of Final Device Design

This final assembly has several improvements over the old design. The device properly fits on the Zeiss microscope, while the other is unstable. It also has a much lower profile, with the linear actuators at 0.75” and the old motors 1.4”. The height difference helps keep the device from hitting the condenser overhead. The new brackets are top loading, while the old ones dropped far below the plate. These would interfere with the objectives of the microscope. The new code is much easier to work with and is easily transferable to different computers. Combining the new code, hardware, and linear actuators, we are able to obtain the desired 10% strain at 1Hz.

Chapter 5: Final Device Verification Results

Section 1: Statistical Analysis Methodology for Validation

To analyze our strain validation data, a one-sample t test was performed. A one sample t test compares the sample mean of a data set to an accepted standard "test value" (Kent State University, 2019). For a sample to work in this test, a set of independently-gathered data that follows an approximate normal distribution is needed. The mean value and standard deviation of the data set are taken and a two-tailed p value is calculated comparing this average to the accepted average value. A p-value below 0.05 corresponds to a rejection of the null hypothesis that there is no statistical significance.

Section 2: Instron Uniaxial Well Testing

Our fourth major project goal was to improve the overall well durability. Our team addressed this goal by redesigning the wells. To test if the new well design was durable enough to withstand 10% cyclical strain at 1 Hz for 24 hours, a program was designed using an Instron E3000. Two brackets as shown in Figure 5.1 were 3D printed to allow the wells to interface properly with the Instron grips. One of the brackets was placed into each grip and the well was placed across the pegs on the brackets.

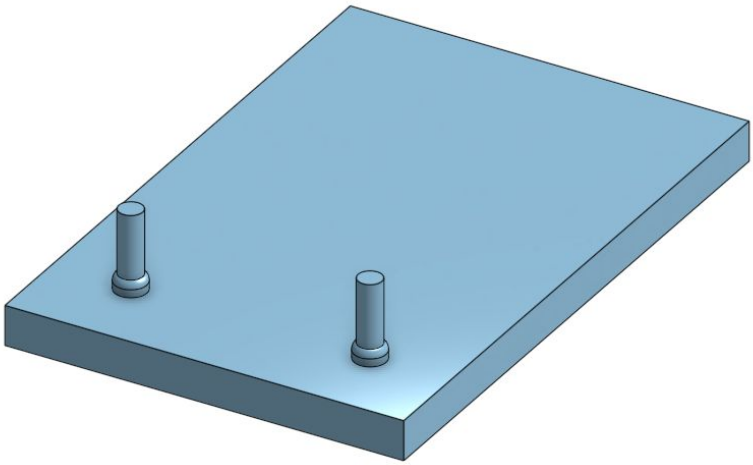


Figure 5.1: CAD drawing of Instron bracket.

A WaveMatrix method was programmed to perform the cyclical stretching tests. Figure 5.2 shows the inputs to this method, which programmed the wells to have a 5 mm displacement over 86400 cycles at 5 Hz.

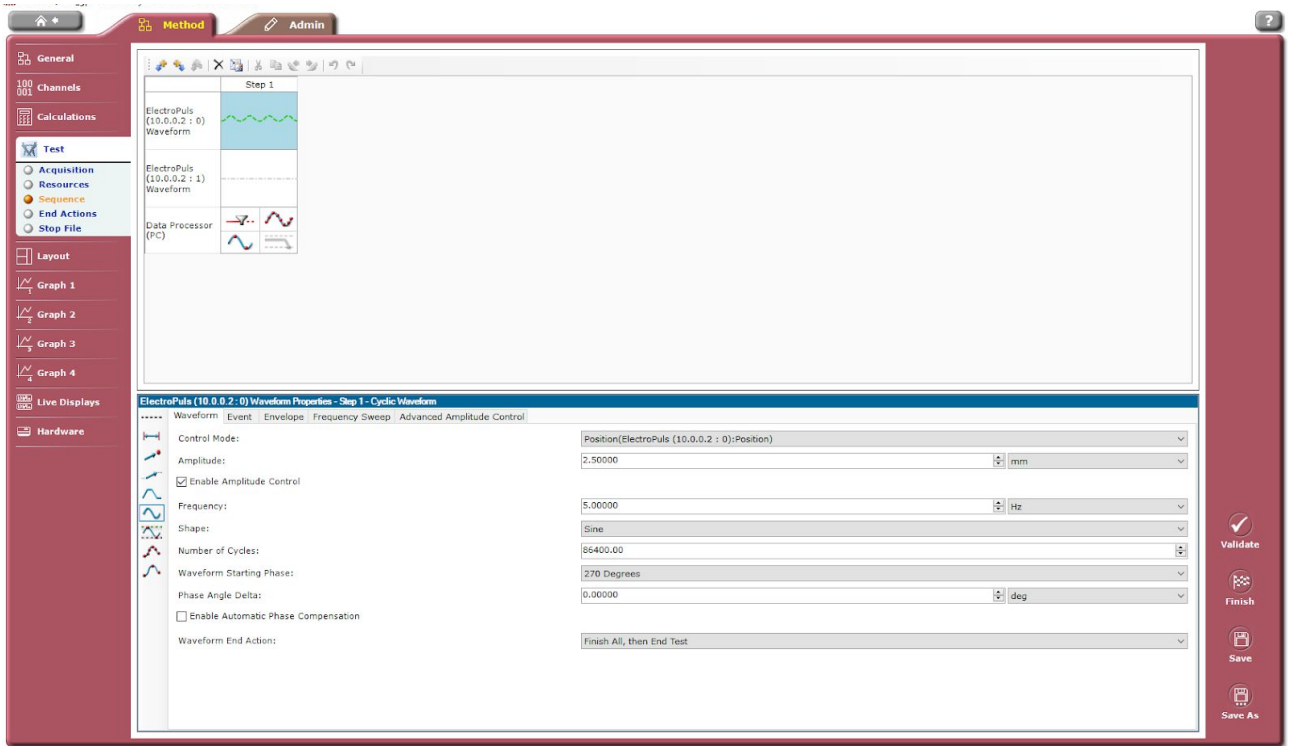


Figure 5.2: WaveMatrix Method Inputs.

This number of cycles was chosen as the number of cycles which would be completed in 24 hours at a rate of 1 Hz, and the 5 Hz rate was used to decrease the amount of time that each test would take.

As a control, one of the original wells was first tested. Figure 5.3 shows the maximum force sustained by the wells in each cycle.

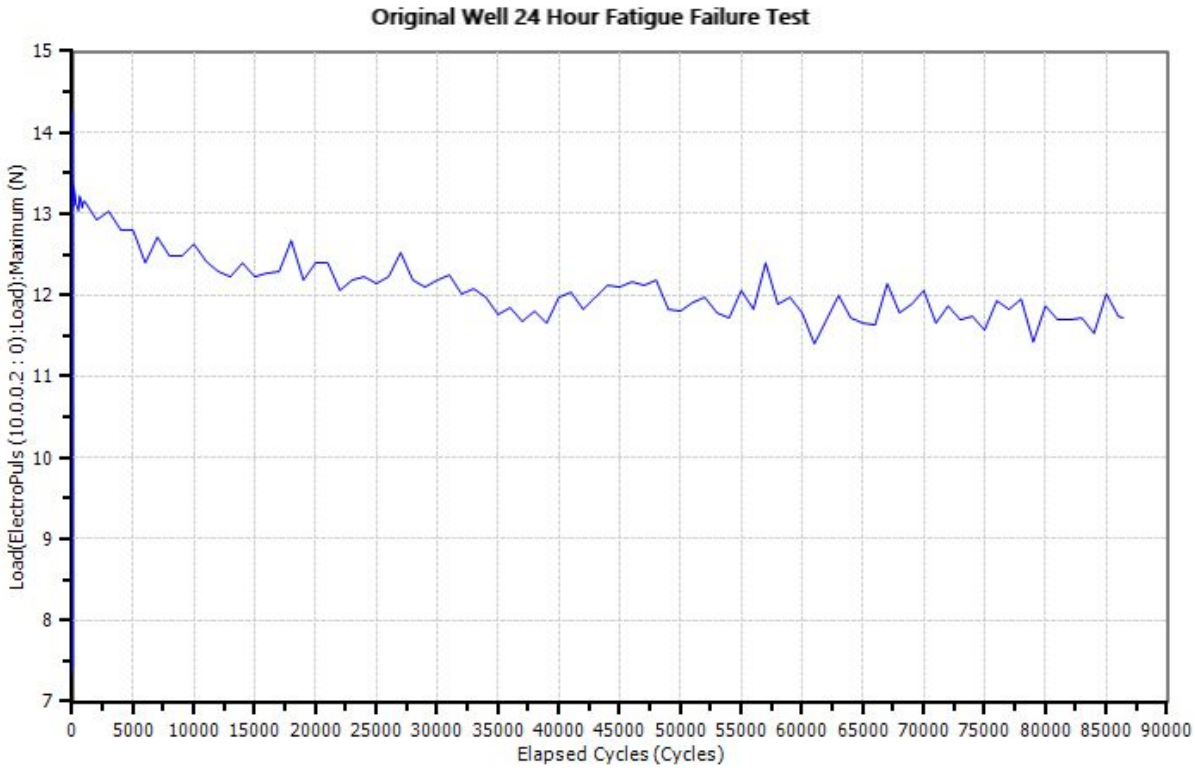


Figure 5.3: Original Well 24 Hour Fatigue Test

This well showed a slight decreasing trend of maximum load sustained over time. As time increased, the silicone well underwent some plastic deformation due to stress relaxation. The initial max load was 13.50 N at 0 hour and 11.71 N at 24 hours.

The same fatigue failure test was performed using one of our newly-designed wells, with the output shown in Figure 5.4.

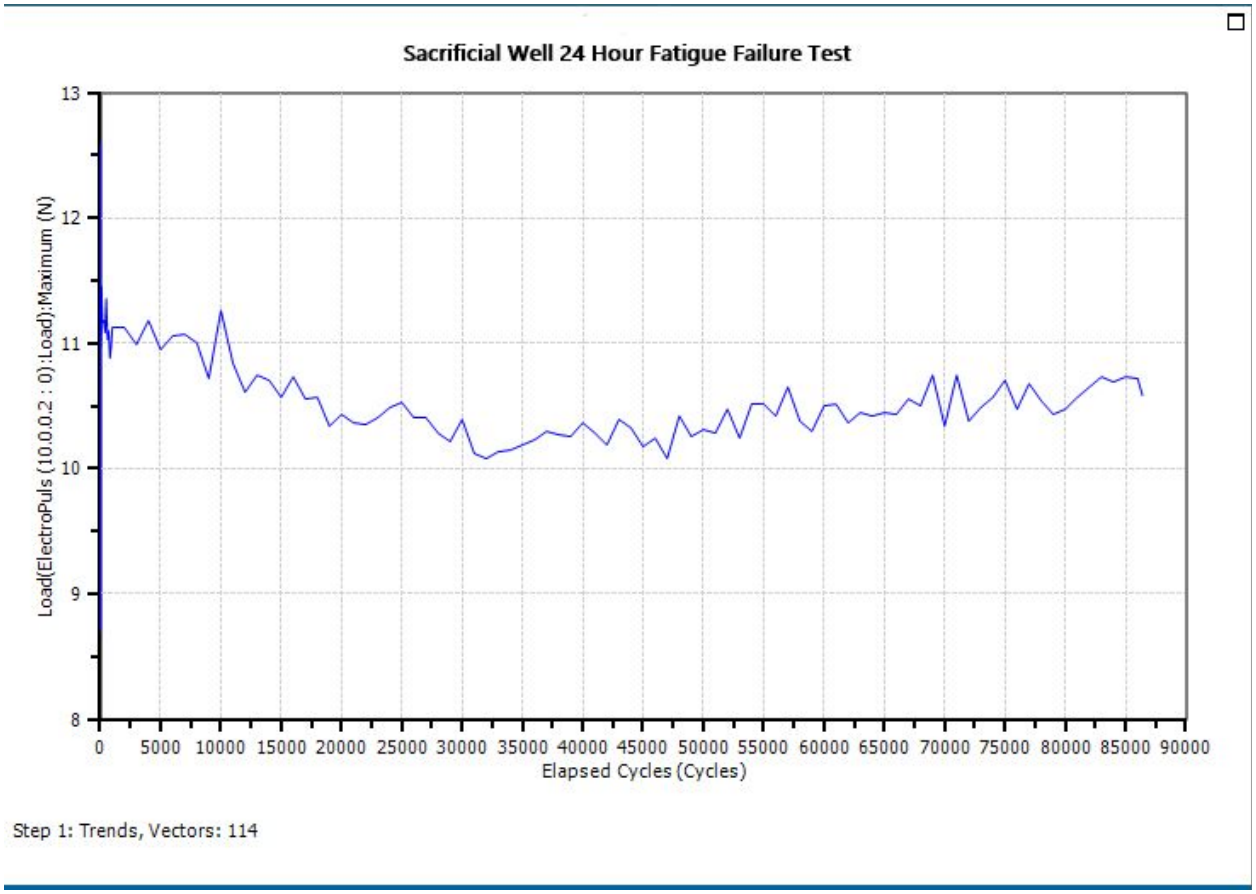


Figure 5.4: Sacrificial Well 24 Hour Fatigue Test.

These wells were as durable to cyclic stretching as the original well design because they did not fail after 86400 cycles. The wells also experience some plastic deformation with time due to stress relaxation of the silicone. This was expected because the same material was used to produce the new wells. Notably, the maximum load required to deform these wells was reduced overall compared to the original wells. A maximum load of 12.58 N was recorded at 0 hours and 10.60 N was recorded at 24 hours. The Instron records the force at the time that 10% strain is attained. This force value is recorded on the chart. The new well design test required a lower

force to attain 10% strain. With the same material properties, this force difference can be attributed to the variability in height between the original well batches. Our wells were manufactured with the same height between batches, but the original well height depends on the amount of silicone poured into the mold.

Section 3: Validating Strain on the Well Produced by the Final Device

To ensure the final device is able to achieve 10% strain, analysis of the wells stretching was conducted through video and Image J analysis. The acceptable margin of error was 1%, so the goal was to achieve 9-11% strain. A PDMS well was marked four dots in the center region and recorded through video, and images of full extension and retraction were taken from various time points. The set distance between the dots was 5 mm, so the analysis measured the change in distance. As seen in figure 5.5 below, it was found that the well at each time had between 9-11% strain. The average strain for the well over the time period was 10.49%. Standard deviation error bars were added for each average. The small variations could be accounted for by user error when measuring the distances between the dots.

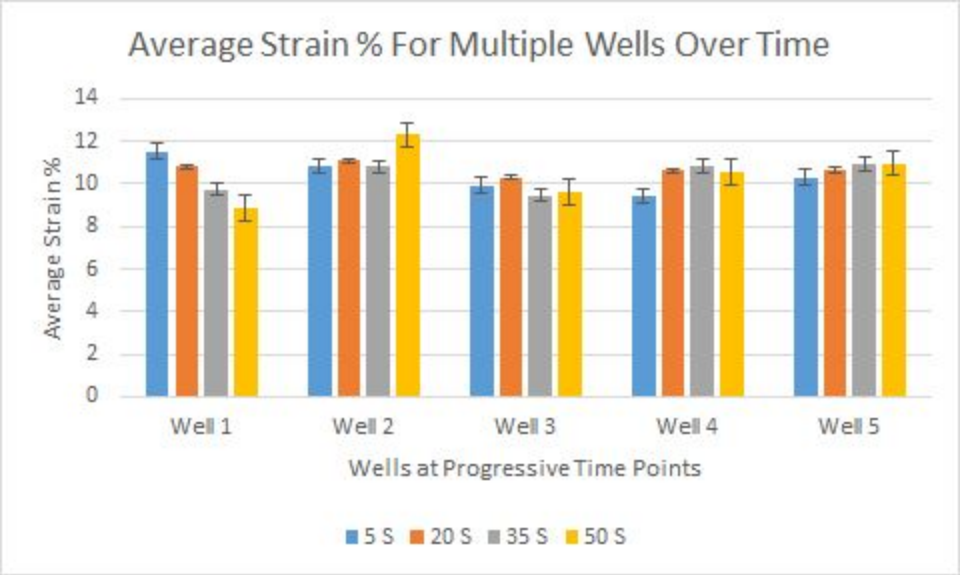


Figure 5.5: Average Strain % in Wells on Final Device Over Time.

Statistical analysis was conducted to determine the significance of our data. A single sample T test determined the P value to be 0.0193, indicating the data is statistically significant. The standard deviation was 1.8, with a standard error of the mean of 0.2. The average strain of 10.49 is within our acceptable error range of 9-11%.

Section 4: Validating the Final Design Durability and Operating Time

Operation time was tested by running the device continuously at 3, 6, 12, and 24 hour intervals. Due to time constraints, the device was only tested once in each time interval. For each of the time intervals, however, the device ran continuously without complications. Complications were defined as one or both of the linear actuator attenuating, the linear actuators overheating, or any of the device the parts breaking. Figure 5.6 below shows the linear actuator motor temperature over a 25 hour period of continually running the device. A trendline was calculated to show a nearly horizontal trendline. The percent error was calculated for change in motor temperature over time and was found to be 1%.

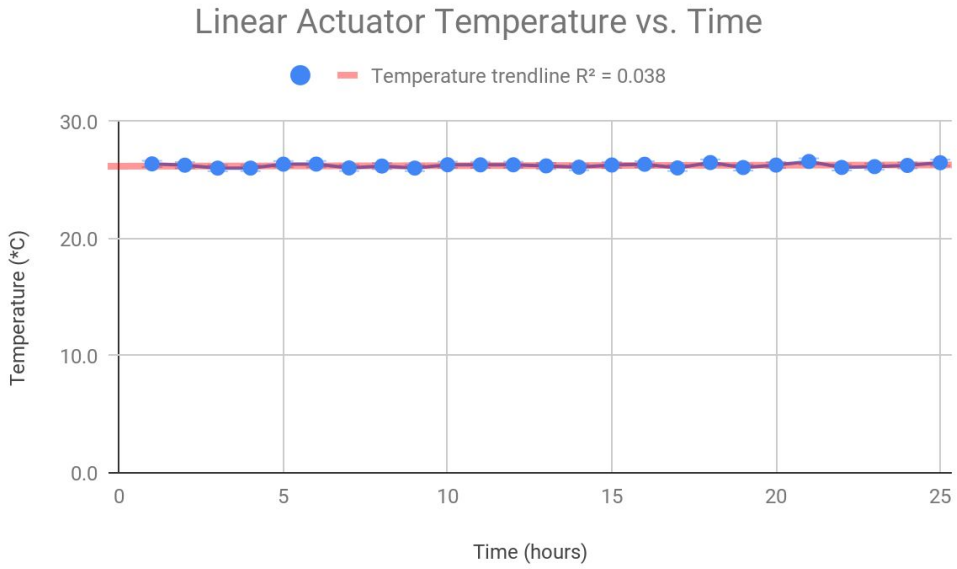


Figure 5.6: Motor temperature versus time for the device run continuously over 25 hours. The linear actuator motors remained at approximately the same temperature with +/- 0.3 *C difference over this time.

Chapter 6: Discussion

The existing four linear actuator device has some limitations. This device cannot operate at 1 Hz at strains above 13% due to our choice of micro-linear actuator. Although the micro-linear actuators conserve space to allow the device to fit properly on the microscope stage, their maximum velocity inhibits higher strain production. Theoretically, the device can operate at lower frequencies to attain strains up to 20%; however, this functionality has not been fully validated.

The new sacrificial wells, after formation in the prototyped well mold, are optically translucent, not transparent. The wells appear cloudy after removal from the mold. It was determined that by adding clear liquid to the well, they become transparent. This was tested with cell media to confirm that the wells are optically transparent. This cloudy surface is likely due to a rough microstructure imposed on the surface of the well from the well mold. We predict that the rough micro-surface of the mold is due to the 3D print resolution and micro-surface scratches of the resin material. Thus, the wells appear cloudy from deflection of light in many divergent directions from the well surface. In our future work section, we recommend that the final well mold design be manufactured from new acrylic sheets. Acrylic was proven to produce smooth surface structure and optical transparency in the old well design. A rough surface structure is not ideal for cell-culture; however, cells will not be seeded directly on the PDMS surface, but on a PA-gel substrate that sits on the well surface. Thus, the cells should not be affected by the rough surface even if the mold prototype is used for well manufacturing and new well testing.

The code can be improved to allow for different strain patterns other than an alternating strain in each direction. Although the alternating strain feature makes this device beneficial for

our client over market models - by allowing for the investigation of cell adaptation response - the ability to test more patterns would enhance the robustness of the device capability.

Additionally, the sturdiness of the device could be increased by using steel or another stiff metal, greater than aluminum. A stiffer material would guarantee less motion of individual parts within the device.

To protect the device, our code contains extension limits for the linear actuators. Our device lacks physical limits which would improve the safety of the device operating at larger deformations by acting as failsafes to the code limits.

Subsection A: Contemporary Issues Associated with Mechanotransduction

Calcification aortic valve disease (CAVD) is the most common acquired valvular disease and the third leading cause of cardiovascular disease in developed countries (Roger et al., 2010). Calcification of aortic heart valves results in the impaired ability of the valves to properly open and close (Chen et al., 2012). This malfunction results in abnormal blood flow and an increased load rendered on the muscle tissue of the heart (Merryman, 2010).Tissue Engineered Heart Valves (TEHVs) have been developed to replace failed aortic heart valves. Currently, TEHV's contract excessively which causes leaflet shortening and eventually, *in vivo* failure (Schmidt et al. 2010).

Heart valves are constantly under significant stresses due to the cardiac cycle. The development of a benchtop model to accurately mimic in vivo mechanical stimulation of aortic heart valves can provide researchers with a method to develop a fundamental understanding how mechanobiology affects pathophysiology. Specifically, understanding the cellular response of heart valves in an accurate *in vitro* model can help improve current TEHV designs. A benchtop

simulator of *in vivo* conditions can also provide a mechanism for mechanically conditioning cells outside the body. This concept can catalyze the creation of complex tissue engineered models by allowing live cells to grow, proliferate, and acquire traits outside the body before *in vivo* implantation. Several additional factors would be required for growing cells outside the body on a benchtop with the intention for future implantation. The environment of the cells and sterilization of the device would need to be strictly controlled and monitored. Additionally, the device environment would require strict regulation. Although, these considerations are outside the scope of this project, long term implications were considered in this device design.

Subsection B: Economics, Environmental and Societal Impact and Political Ramifications

The device created was intended solely for research purposes to study mechanobiology and will not be marketed. Therefore, this device will not directly have an impact on the local or global economy. The use of this device for research purposes could further the understanding of mechanobiology to contribute valuable biological and biomedical information to society. The results of these studies could lead to significant biomedical advancements or applications. If marketed, these potential advancements could contribute to a local or global economy.

The long term implications of this device included its intended purpose to be used to understand cardiomyocyte mechanotransduction for aortic tissue engineered heart valve (TEHV) research. Therefore, the target demographic for the research application of this device is individuals with or in need of a tissue engineered aortic heart valve replacement. The use of TEHV might range from infants and children through adults and older adults. For patients requiring an aortic heart valve replacement, it is essential to acquire a replacement that has a long lifetime and durability. To replace an aortic heart valve would require a high risk, expensive

procedure. For infants and older adults, longer hospital stays might be required for a high risk surgery. Thus, an ideal solution would be to have one surgery to replace the damaged valve. If our device works effectively on a benchtop to simulate the heart valve microenvironment, cardiomyocytes could be conditioned outside the body. Conditioning the cells outside the body would allow for a prolific cardiomyocyte valve replacement for immediate functionality and long term sustainability in the body. This could provide a major societal impact by increasing the quality of care for cardiac diseases.

Cardiac disease can be expensive, requiring long term medications to treat over a person's lifetime. Medications can also be required for artificial heart valves to prevent an immunological reaction and rejection. Although this device does not eliminate the need for a surgical procedure, it does have the potential to prevent the need for lifelong medications. Taken as a whole, this long term treatment concept could reduce the overall cost of care by minimizing medications required and total surgical procedures and hospital stays. If a TEHV was successfully conditioned *in vitro* and implanted in a human, the FDA would classify the device as class III for its high risk, life-sustaining features. As a result, the design, manufacture, and distribution of this device would be highly regulated and monitored.

The final device created during our project poses no significant human health risk when operated according to the instructions provided. To safely use live cells with this device, silicone wells should be sterilized before contact with cells then discarded in biohazardous waste after contact with cells according to ASTM . F813-07 and ASTM STP810 (ASTM, 2012; Brown, 1983). Safe handling of cells during device use and cell culture, including personal safety attire, is required.

This device does not pose any adverse risk to the environment. No toxic chemical are given off or leached into the local environment of the device. The device does not create a laboratory hazard because it is low profile and can be stored outside an incubator on a benchtop. The most risk associated with this device is the use of cells in the silicone stretching wells. These risks are well known an mitigated by adhering to cell-culture techniques, wearing proper lab attire, and using BioSafety hoods for cell handling.

Subsection C: Device Manufacturability, Device Sustainability, and Ethical Concerns

The device was created with the intention that it could be easily replicated at a low cost and the code could be easily modified for customization. If the device is altered, we recommend that the device is re-validated before use. The parts used to fabricate the final device were designed in SolidWorks. Parts can be manufactured by CNC machining or hand machining with jigs. All electronic products used in the creation of the device are commercially available on the market. Appendix J contains the original web links, part names, and part numbers for obtaining parts in the future. The custom open source program that our device is built off of provide for long term sustainability of the technology. As research objectives or findings change over time, our device can be customized to accommodate the changes. The coding language is well-known and easy to learn or find references for making it an accessible program for a wide range of people.

Ethical concerns associated with the operation of this device consider the use of animal cells and animal cell lines in cell culture for research purposes. Using animal cells in research can improve societal understanding of disease pathogenesis and potentially improve human lives. By conservatively using immortalized cell lines, by culturing cells at a responsible rate, ethical

concerns can be minimized. If this device were to be used in a TEHV application wherein the valve was implanted into a animal model or human, the safety of both the animal and human subjects would be a large concern. Adequate safety and efficacy data should be acquired for in vitro experiments prior to in vivo animal experiments. The safety should again be evaluated with a high degree of efficacy before moving into human subjects in clinical trial phases. Clinical trials that use this device for cell condition should properly educate and inform the patient before obtaining consent. The patient should be made aware of all the possible treatment options and risks before engaging in a clinical trial.

Chapter 7: Conclusions and Recommendations

Although this device can attain 0.1Hz-1Hz frequencies and 0%-13% strains, the device did not meet the requirements for every value in the initial specifications defined by the client statement. The device was validated for operation up to 1 Hz and 10% strain. The first future assignment should be to validate the device at variable strains up to 13% with frequencies from 0.1Hz-1Hz. This will fully validate the operational capabilities of the device within its theoretical range.

Moreover, the current linear actuators have a 0.59”/s maximum speed. To increase the strain produced at the same frequency range, a different linear actuator motor would need to be purchased with a higher maximum speed. Each pulse to move the linear actuator forward is dependent on the maximum velocity of the motor. To increase the strain, micro-linear actuators could not be used; instead, linear actuators would need to be purchased. Linear actuators are made to operate at higher maximum speeds, but will be larger and take up more space on the device.

To improve the physical stability of the device, the L-brackets that hold the pins attaching to the silicone stretching well could be machined out of steel. The pins could also be metal-welded onto the L-bracket. Both of these changes would prevent unwanted movement of the corner locations or stretch well by increasing material stiffness. This would better prevent an accidental bump into a bracket from offsetting the corner position.

Additionally, a mold should be created for the new sacrificial well design. This well design has been found to greatly increase the uniform strain area, as mentioned previously. This improvement of the wells will help to ensure cells are uniformly strained during testing. It is

recommended that the new well mold is made out of acrylic with a CNC. The factory edges of acrylic are very smooth and allow the well to be transparent after polymerization. The 2013 well mold is a three part acrylic mold with a thin metal spacer. The metal spacer creates the well surface atop which the cells are seeded on PA. Metal pins are dropped into the corners of each well in the mold through holes in the acrylic. Similarly, we recommend that the new well mold have three detached parts, wherein the middle part defines the thickness of the well. The three parts should be fixed together with removable screws during assembly. The new mold should also contain a 3* draft angle to facilitate the removal of the wells after polymerization. This 3* draft angle is already incorporated into the accompanying CAD model for the new well mold. The pins and sacrificial wells of the new mold design should be included as pins that insert into the silicone after the mold is assembled. This means holes should be drilled in the top piece of acrylic to allow insertion. By including removable pins, upon disassembly, the wells can be more easily extracted from the mold. This will help prevent well failure. Moreover, we redesigned the new well so that the corner pegs are the same distance apart as the original well design. This means that no changes in the device or code are necessary to accommodate the new wells after manufacturing.

To further improve the device, physical limits should be added to the device setup. The code written for the four micro-linear actuator device already contains limits to prevent system malfunction; however, the inclusion of physical limits on the device would act as an additional failsafe to malfunction. The old device used encoders to track the position of the motors, and other marketed-devices contain limit switches. Both of these prevent the motors from going too far and crashing into the baseplate, or binding. The physical limits would cut the power to protect

the motors, device, and test sample. The addition of a physical limit switch to this device would provide a safe backup to code or system failure, as well as power outage or computer system crash.

To create different strain patterns wherein the device stretches and pauses for different lengths of time, a new section of code would be required. The new code should add a delay parameter between the extending and retracting cycles based on two value inputs from the user. One value should represent the dealy after retraction and the other value should represent the delay after extension. This delay parameter should be added to both the non-alternating and alternating stretch pattern code sections. The actual code to accomplish this delay could mimic the duty cycle calculation wherein the device holds a position for the time specified by the user.

References

Abiko, H., Fujiwara, S., Ohashi, K., Hiatari, R., Mashiko, T., Sakamoto, N., Sato, M., and Mizuno, K. “Rho guanine nucleotide exchange factors involved in cyclic-stretch-induced reorientation of vascular endothelial cells,” *Journal of Cell Science,* vol. 128, no. 9, pp. 1683–1695, May 2015 [Online]. Available: https://www.ncbi.nlm.nih.gov/pubmed/25795300

Anonymous Biological evaluation of medical devices - part 5: Tests for in vitro cytotoxicity. 2009. Available: https://global.ihs.com/doc_detail.cfm?gid=YOSTPCAAAAAAAAAA&input_doc_number=SN V SN EN ISO 10993-5.

Anonymous F813 standard practice for direct contact cell culture evaluation of materials for medical devices. 2012. Available: http://www.astm.org/Standards/F813. DOI: 10.1520/F0813- 07R12.

“Arduino Due,” 2018. [Online]. Available: https://store.arduino.cc/usa/arduino-due

ASTM. “ASTM International,” 2019. [Online]. Available: https://www.astm.org

Balestrini, J., Skorinko, J., Hera, A., Gaudette, G., Billiar, K. “Applying controlled non-uniform deformation for in vitro studies of cell mechanobiology,” *Biomechanics and Modeling in Mechanobiology,*vol. 9, no. 3, pp. 329–344, Jun. 2010 [Online]. Available: https://www.ncbi.nlm.nih.gov/pubmed/20169395

Boudou, T., Ohayon, J., Picart, C., Pettigrew, R., and Tracqui, P. “Nonlinear elastic properties of polyacrylamide gels: Implications for quantification of cellular forces,”

Biorheology, vol. 46, no. 3, pp. 191–205, Jul. 2009 [Online]. Available: https://hal.archives-ouvertes.fr/hal-00670214

Brown, S. Cell-Culture Test Methods. United States: 1983. ASTM. Available: https://www.astm.org/DIGITAL_LIBRARY/STP/SOURCE_PAGES/STP810.htm

Butcher, J. and Nerem, R. "Porcine aortic valve interstitial cells in three-dimensional culture: comparison of phenotype with aortic smooth muscle cells," *The Journal of Heart Valve Disease,* vol. 13, no. 3, p. 478, May 2004 [Online]. Available: https://www.ncbi.nlm.nih.gov/pubmed/15222296

"ChipKIT Max32," 2018. [Online]. Available: https://chipkit.net/wpcproduct/chipkit-max32/

"CNC Micromachining," 2019. [Online]. Available: http://www.owensind.com/CNCServices/Micromachining. [Accessed: 02-Mar-2019]

"Computer Controlled Cell Deforming - Cell Stretcher," 2019. [Online]. Available: https://www.emsdiasum.com/microscopy/products/histology/cell_stretcher.aspx. [Accessed: 02-Mar-2019]

Denisin, A. and Pruitt, B. "Tuning the Range of Polyacrylamide Gel Stiffness for Mechanobiology Applications." *ACS Applied Materials & Interfaces*, vol. 8, no. 34, pp. 21893–21902, Aug. 2016 [Online]. Available: https://www.ncbi.nlm.nih.gov/pubmed/26816386

Duoba, B., Lombardo, J., Thu Minn, K., and J. Rodriguez, J. "Device to dynamically stretch cells during microscopic visualization," 2012 [Online]. (Undergraduate Major Qualifying Project No. KLB1101). Available:

https://web.wpi.edu/Pubs/E-project/Available/E-project-042512-150124/unrestricted/KLB1101_Final_Report.pdf

Fincan, M. "Assessing Viscoelastic Properties of Polydimethylsiloxane (PDMS) Using Loading and Unloading of the Macroscopic Compression Test," University of South Florida, 2015 [Online]. Dissertation. Available: https://scholarcommons.usf.edu/etd/5480/

FlexCell International Corporation, "StageFlexer," 2019. [Online]. Available: http://www.flexcellint.com/StageFlexer.htm. [Accessed: 01-Mar-2019]

FlexCell International Corporation. "Applications of FlexCell Products," 2011. [Online]. Available: http://www.flexcellint.com/applications4.htm. [Accessed: 23-Sep-2018]

Garcia, D., Kelley, J., Pruden, J., and Mann, J. "Advanced Cell Culture Well for Mechanobiology," 2013 [Online]. (Undergraduate Major Qualifying Project No. KLB 1201). Available: https://web.wpi.edu/Pubs/E-project/Available/E-project-042413-102313/unrestricted/Cell_Well_Final_Report.pdf

Gould, S., Srigunapalan, S., Simmons, C., and Anseth, K. "Hemodynamic and Cellular Response Feedback in Calcific Aortic Valve Disease," *Circulation Research,* vol. 113, no. 2, pp. 186–197, Jul. 2013 [Online]. Available: http://ovidsp.ovid.com/ovidweb.cgi?T=JS&NEWS=n&CSC=Y&PAGE=fulltext&D=ovft&AN=00003012-201307050-00016

Haydon Kerk."35H4N-2.33-907", 2019. [Online]. Available: https://prototypes.haydonkerk.com/ecatalog/hybrid-linear-actuators/en. [Accessed: 24-Apr- 2019].

"How to Choose the Right 3D Printing Material," Jan-2017. [Online]. Available: https://archive-media.formlabs.com/upload/how-to-choose-the-right-3D-printing-material.pdf

Hsu, H., Lee, C., Locke, A., Vanderzyl, S., and Kaunas, R. "Stretch-Induced Stress Fiber Remodeling and the Activations of JNK and ERK Depend on Mechanical Strain Rate, but Not FAK," *PLOS One,* vol. 5, no. 8, p. e12470, Aug. 2010 [Online]. Available: https://www.ncbi.nlm.nih.gov/pubmed/20814573

Huang, W., Sakamoto, N., Miyazawa, R., and Sato, M. "Role of paxillin in the early phase of orientation of the vascular endothelial cells exposed to cyclic stretching," *Biochemical and Biophysical Research Communications,* vol. 418, no. 4, pp. 708–713, Feb. 2012 [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0006291X12001179

Hughes, A. and Drury, B. Electric Motors and Drives, 4th ed. Newnes, 2013 [Online]. Available: http://lib.myilibrary.com?ID=451449

International Electrotechnical Commission. *Dimensions and output series for rotating electrical machines part 1: Frame numbers 56 to 400 and flange numbers 55 to 1080* (1991). (Sixth; Incorporating Amendments 1 and 2 ed.)

Johnston, I., McCluskey, D., Tan, C., and Tracey, M. “Mechanical characterization of bulk Sylgard 184 for microfluidics and microengineering.” *Journal of Micromechanics and Microengineering*, vol. 24, no. 3, p. 35017, Mar. 2014.

Kambic, H. & Yokobori, A. T. (1994). Biomaterials' mechanical properties. Philadelphia, PA: ASTM

Kent State University. (2019). SPSS Tutorials: One Sample t Test. Retrieved from https://libguides.library.kent.edu/SPSS/OneSampletTest

Kloxin, A., Benton, J., and Anseth, K. “In situ elasticity modulation with dynamic substrates to direct cell phenotype,” *Biomaterials,* vol. 31, no. 1, pp. 1–8, 2009 [Online]. Available: https://www.clinicalkey.es/playcontent/1-s2.0-S0142961209009557

Lewotsky, K. “Choosing the Right Linear Actuator,” 2007. [Online]. Available: https://www.motioncontrolonline.org/content-detail.cfm/Motion-Control-Technical-Features/Choosing-the-Right-Linear-Actuator/content_id/1051. [Accessed: 30-Sep-2018]

Low voltage motors motor guide (2014). Asea Brown Boveri Ltd. Available: https://new.abb.com/docs/librariesprovider53/about-downloads/low-voltage-motor-guide.pdf

Merryman, W. D. “Mechanobiology of the aortic valve interstitial cell,” University of Pittsburgh, 2007. Dissertation Abstracts International. Vol. 68, no. 09, suppl. B, 186 [Online]. Available: https://search.proquest.com/docview/304838460

Merryman, W. D. “Mechano-potential etiologies of aortic valve disease,” *Journal of Biomechanics,* vol. 43, no. 1, pp. 87–92, 2009 [Online]. Available: https://www.clinicalkey.es/playcontent/1-s2.0-S0021929009005041

Merryman, W. D., Engelmayr, G., Liao, J., and Sacks, M. “Defining biomechanical endpoints for tissue engineered heart valve leaflets from native leaflet properties,” *Progress in Pediatric Cardiology,* vol. 21, no. 2, pp. 153–160, 2006 [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1058981305000809

O’Brien, K. “Pathogenesis of Calcific Aortic Valve Disease: A Disease Process Comes of Age (and a Good Deal More),” *Arteriosclerosis, Thrombosis, and Vascular Biology* vol. 26, no. 8, pp. 1721–1728, Aug. 2006 [Online]. Available: http://atvb.ahajournals.org/cgi/content/abstract/26/8/1721

Park, J., Yoo, S., Lee, E., Lee, D., Kim, J., and Lee, S. “Increased poly(dimethylsiloxane) stiffness improves viability and morphology of mouse fibroblast cells,” *BioChip Journal,* vol. 4, no. 3, pp. 230–236, Sep. 2010.

Pritchard, R., Lava, P., Debruyne, D., and Terentjev, E. (2013). Precise determination of the Poisson ratio in soft materials with 2D digital image correlation. *Soft Matter*. 9. 6037-6045. 10.1039/C3SM50901J.

Progressive Automations. “PA-07 Micro Linear Actuator,” 2019. [Online]. Available: https://www.progressiveautomations.com/products/micro-linear-actuator

Roh, C., Lee, J. and Kang, C. "The Deformation of Polydimethylsiloxane (PDMS) Microfluidic Channels Filled with Embedded Circular Obstacles under Certain

Circumstances", *Molecules*, vol. 21, no. 6, p. 798, 2016. Available: 10.3390/molecules21060798.

Schmidt, D., Dijkman, P. E., Driessen-Mol, A., Stenger, R., Mariani, C., Puolakka, A., . . . Hoerstrup, S. P. (2010). Minimally-invasive implantation of living tissue engineered heart valves. *Journal of the American College of Cardiology*, 56(6), 510. doi:10.1016/j.jacc.2010.04.024

Simmons, C., Grant, G., Manduchi, E., and Davies, P. "Spatial Heterogeneity of Endothelial Phenotypes Correlates With Side-Specific Vulnerability to Calcification in Normal Porcine Aortic Valves," *Circulation Research,* vol. 96, no. 7, pp. 792–799, Apr. 2005 [Online]. Available: http://circres.ahajournals.org/cgi/content/abstract/96/7/792

"SOViK 6" 6 inch Stroke Linear Actuator, 12V DC 22lbs 44lbs 55lbs Maximum Lift, Built in Limit Switch, Includes Mounting Brackets," 2019. [Online]. Available: https://www.amazon.com/SOViK-Actuator-Maximum-Mounting-Brackets/dp/B071S74TFX

SpecialChem. "Hardness of Plastics- When to Use RockWells/Shore Scales," 2019. [Online]. Available: https://omnexus.specialchem.com/polymer-properties/properties/hardness-introduction

StrexCell Inc. "Automated Cell Stretching System STB-140," 2018. [Online]. Available: https://strexcell.com/cell-stretching-system/models/automated-cell-stretching-system/. [Accessed: 23-Sep-2018]

Tissue Mechanics & Mechanobiology Lab. "Research," 2016. [Online]. Available: https://wp.wpi.edu/billiarlab/research/

"USE CORRECT TOOLS FOR PROBLEM-FREE ACRYLIC SHEET MACHINING," Plastics Distributor & Fabricator, vol. 20, no. 6. 2019 [Online]. Available: http://www.plasticsmag.com/ta.asp?aid=2332. [Accessed: 02-Mar-2019]

Zalewski, M. "Guerrilla guide to CNC machining, mold making, and resin casting," 2015. [Online]. Available: http://lcamtuf.coredump.cx/gcnc/full/. [Accessed: 02-Mar-2019]

Appendices

Appendix A: Additional Background Information

Section A1: Physiological and Mechanical Context In Vivo

Compared to vascular aortic endothelial cells, valve endothelial cells (VECs) have demonstrated unique morphological properties. VECs have cells that align perpendicularly to the direction of blood flow whereas endothelial cells align parallel (Merryman et al., 2006). Butcher and Nerem demonstrated this effect in vitro for porcine fibroblastic VECs grown on collagen (Butcher & Nerem, 2004). Moreover, VECs are shear sensitive wherein VECs elongate and align parallel to blood flow; however, in bifurcated regions with disturbed flow, ECs become polygonal and rounded (Gould et al., 2013). Furthermore, Simmons et al., have determined based on differential gene expression of inflow and outflow surface of VECs that the aortic side of the valve is more prone to disease and calcification with implications of an aortic valve disease initiating mechanism available from VECs on the aortic side of the valves (Simmons et al., 2005). The ventricular surface of the AV is exposed to pulsatile shear stresses has been shown to be prone to endocarditis and bacterial growth whereas the aortic surface experiences disturbed oscillatory shear stress has demonstrated a tendency toward sclerotic or fibrotic tissue formation (Merryman, 2009). It has been postulated then that distinct pathologies aries from different valve surfaces under unique shear stress profiles.

Similarly, Gould et al. described how the function of the aortic valve is compromised with changes in the stiffness of the ECM (Gould et al., 2013). They report that ECM remodeling occurs when myofibroblasts degrade and disrupt the normal tissue components and replace them

with disorganized rigid collagens and bone (Gould et al., 2013). Therefore, the stiffer substrate decreases the flexibility of the valves which ultimately augments aortic stenosis. Calcification is a result of calcium and phosphate and hydroxyapatite valvular deposits. The result is stiffened valves which obstruct ventricular flow (O'Brien, 2006).

VECs form in a monolayer on heart valve leaflets beneath which is a basal membrane. This membrane sits on an interstitial extracellular matrix with varying protein composition throughout the thickness (Merryman, 2009). In an adult atrioventricular valve, collagen type I makes up most of the ECM. These collagen cells align circumferentially, aligned with the VECs, making up about a third of the ECM thickness. This collagen layer is primarily responsible for the planar, orthogonal mechanical response during leaflet closing. The central portion of the leaflet is composed of sulfated glycosaminoglycans (GAGs) making up the next third of the ECM.This layer serves biomechanically as a lubricant, shock absorber. The final layer on the ventricular side of the atrioventricular valves is composed of both collagen and elastin which is thought to facilitate early valve closing (Merryman, 2007).

Aortic valvular tissue is created and maintained by valvular interstitial cells (VICs). In an adult, aortic valve, VICs display a dormant fibroblast phenotype with a less than 5% population of myofibroblast population (Gould et al., 2013). The VIC phenotype and function are regulated by the surrounding environment including the biomechanical properties of the supporting ECM, the mechanical stimuli induced by hemodynamic forces, inflammatory cytokines, and paracrine signalling molecules (Gould et al., 2013). The mechanical environment plays a critical regulatory role for cell morphology. For instance, if the transvalvular pressure increases, such as in hypertension, the tissue will undergo greater stretching resulting in a higher myofibroblastic

phenotype. This can lead to fibrosis of the tissue (Gould et al., 2013). Kloxin et al. demonstrated that as the culture substrate elastic modulus increased, the number of myofibroblasts also increased (Kloxin et al., 2009). They determined a threshold for VIC activation, described as greater than 50 percent myofibroblasts, at a Young's Modulus of 15 kPa. Nonetheless, to understand the mechanism of aortic heart valve disease, the development of culture platforms and models with multiple precisely controllable mechanical and biochemical cue regulations are required.

Section A2: Mechanical Properties of PDMS

Polydimethylsiloxane (PDMS) is the stretching well material used in this device and polyacrylamide gel (PA) is the substrate onto which cells are seeded. The mechanical properties of each material was assessed and compared to the values obtained for aortic heart valve cells in vivo. PDMS is a highly crosslinked semi-crystalline thermoplastic material (Fincan, 2015). The siloxane component has a glass transition temperature well below room temperature which causes PDMS to behave as an elastomer (Fincan, 2015). Solid state PDMS is hydrophobic therefore it must be surface treated in order to adhere cells. Plasma treatment ads a layer of silanol groups to the surface converting the surface to be hydrophilic (Fincan, 2015). The molecular structure for PDMS can be seen in figure A1 below.

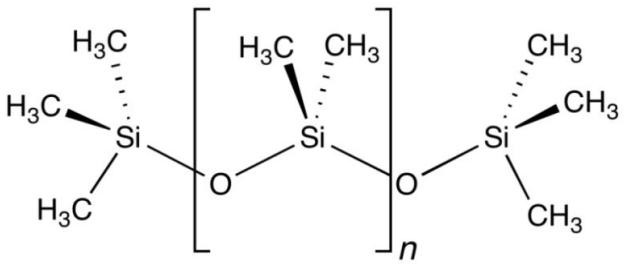


Figure A1: Molecular Structure of PDMS

For PDMS in a 10:1 ratio of base to crosslinker, the reported Young's Modulus is *E*=580 kPa (Park et al., 2010). Decreasing the ratio to 5:1 produces higher Young's Modulus values recorded at *E*=1,000 kPa; however, increasing the ratio to 20:1, the Young's Modulus produced is 280 kPa (Park et al., 2010). The following tensile stress to tensile strain curve was recorded by Park et al. in figure A2 below concluding that the mixing ratio of PDMS strongly controls the material stiffness (Park et al., 2010).

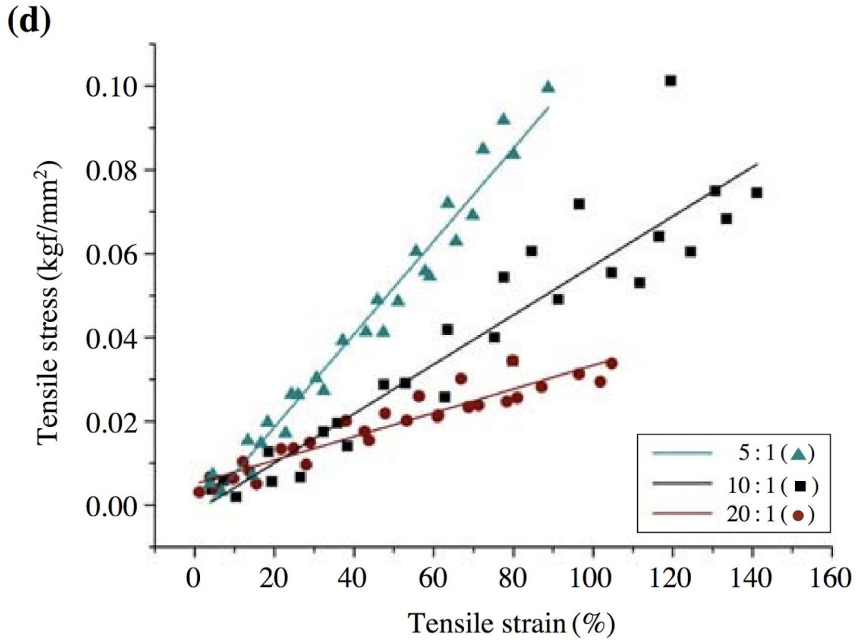


Figure A2: Tensile stress versus tensile strain for increasing ratios of PDMS. Linear fit lines determined the modulus for each set of data(© The Korean BioChip Society and Springer 2010).

The elastic modulus for PDMS can also be affected by the curing temperature. I.D. Johnston et al. determined the elastic modulus for increasing curing temperatures for Sylgard 184 PDMS (Johnston et al., 2014). The curve in figure A3 was obtained. All of the information about the

mechanical properties of PDMS is required to effectively match the required loading conditions with those of the device motors to accurately achieve 10% strain at 1 Hz.

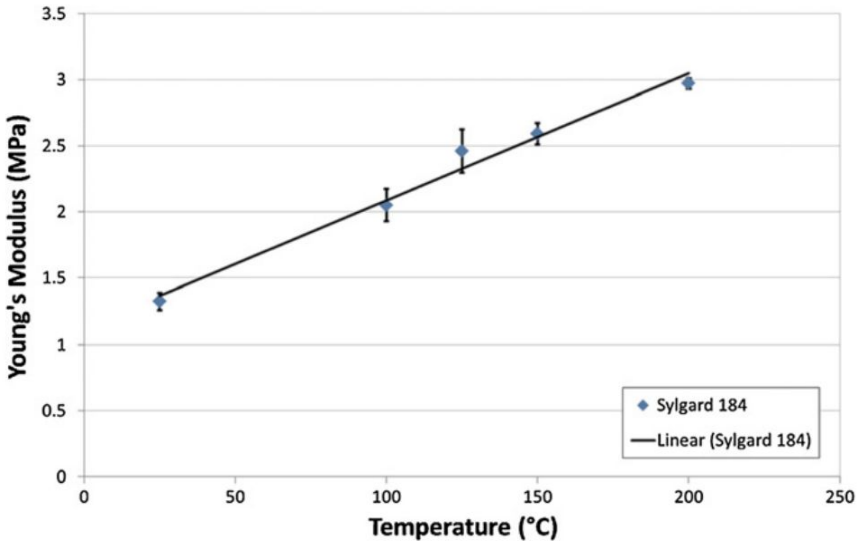


Figure A3. Relationship between curing temperature of Sylgard 184 and the resultant Young's modulus.

Section A3: Mechanical Properties of PA Gel

To accurately assess cellular response to stretch conditioning related to those found in vivo for aortic heart valves, the mechanical properties of Polyacrylamide (PA) also had to be determined. Polyacrylamides are commonly used as cell substrates to study how cells sense and respond to physical characteristics of their environments (Denisin & Pruitt, 2016). The stiffness of polyacrylamide can be altered to observe differences in cell shape, spreading, functional maturity and differentiation (Denisin & Pruitt, 2016). Polyacrylamide hydrogels (PAG) are highly swollen networks of cross-linked acrylamide units. The elastic modulus of PA can be adjusted by altering the ratio of acrylamide to bis-acrylamide. PA is considered a linear elastic, time-independent, material exhibiting a storage modulus throughout a wide range of strains.

However, due to the water content of PA, a non-linear elastic behavior is observed at large strains. Denisin and Pruitt determined that the elastic modulus of PA varies significantly with composition and strain magnitude. They determined hydrogel formulations and polymerization conditions for 'ideal' PA, tabulated in table A1. Denisin and Pruitt describe as ideal PA gel as one which, "exhibits the maximum elasticity because each tetrafunctional bis-acrylamide molecule is connected to four of the nearest neighboring acrylamide groups and spaced following a Poisson distribution" (Denisin & Pruitt, 2016).

Additionally, Boudou et al. determined that the non-linear behavior of PA gels is necessary to consider when quantifying cellular forces(2009). The authors found that assuming linear elastic behavior of PA gels results in underestimated traction forces for displacements later than 2μm (Boudou et al., 2009). The authors also noted that a 6μm deformation is typical on PA for fibroblasts; however, traction forces are underestimated by about 30% for a gel thickness about 50 μm and underestimation rises exponentially to about 50% for thinner gels. Boudou et al. determined that finite size effects were non-significant for PA thicknesses greater than 60 μm (Boudou et al., 2009). The effect of displacement magnitude and gel thickness on traction force calculation is displayed in figure A4 below.

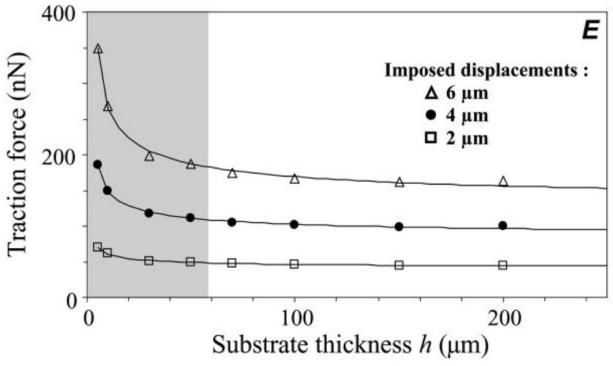


Figure A4. Traction force versus substrate thickness for imposed displacements on PAG(©Biorheology).

The information about the thickness of PAG gel compared to the accuracy of cell traction force calculations for gels under 60 μm is essential to develop a stretch well model that can maximize the accuracy of the device for research purposes. Furthermore, Boudou et al. also determined the elastic modulus for polyacrylamide with varying amounts of acrylamide and bis-acrylamide (Boudou et al., 2009). The authors compared their experimental values to those obtained from literature to produce a linear fit of the data to describe predicted values for Young's modulus of PAG. The results can be observed below in Figure A5.

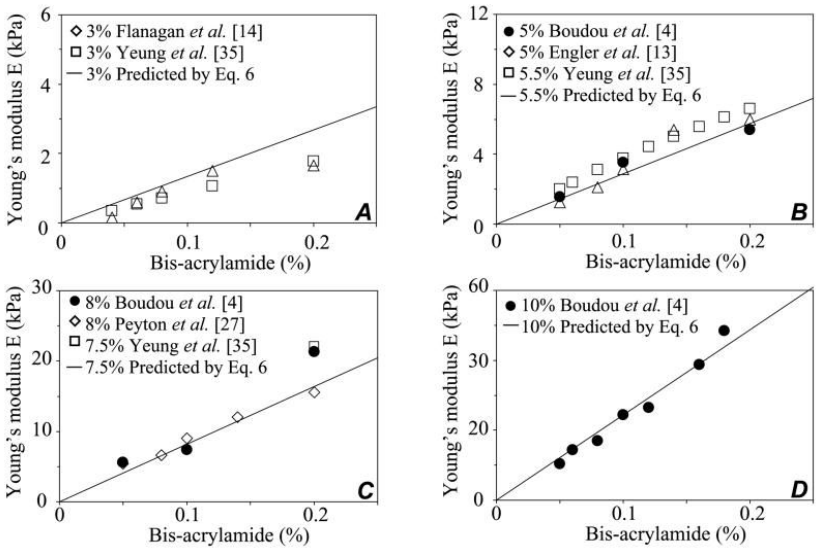


Figure A5. Measured and predicted (linear fit) Young's Modulus for reported values of varying acrylamide concentrations in the amounts of: 3% (A), ~5% (B), ~8% (C), 10% (D), and bis-acrylamide in the range 0.02–0.2

Table A1 below represents the ideal formulation for PA polymerization for obtaining various elastic moduli.

Table A1. Ideal hydrogel formulations and polymerization conditions

Expected Elastic Modulus (kPa)	%T	%C	milliQ water (µl)	Acrylamide (µl of 0. 5 mg/ml stock)	Bis-acrylamide (µl of 0.025 mg/ml stock)	APS (µl) of 10% solution)	TEMED (µl)
6.29	10	1	756.50	198	40	5	0.50
14.04	10	2	718.50	196	80	5	0.50
13.79	10	3	680.50	194	120	5	0.50
19.61	10	4	644.50	190	160	5	0.50
32.29	15	1	636.50	297	61	5	0.50
38.33	15	2	579.50	294	121	5	0.50
69.75	15	4	464.50	289	241	5	0.50
93.46	15	6	352.50	282	360	5	0.50
10.14	20	0.2	578.50	400	16	5	0.50
27.25	20	1	517.50	397	80	5	0.50
51.41	20	2	441.50	392	161	5	0.50
99.25	20	3	441.50	392	161	5	0.50
142.91	20	6	136.50	377	481	5	0.50
163.77	20	7.8	0.50	369	625	5	0.50

Section A4: Detailed Descriptions of StageFlexer Microscope Mountable Stretching Systems

The Flexcell StageFlexer is a microscope mountable cell stretching device for observing cell signaling responses to strains in real time (FlexCell, 2019). This device houses the BioFlex culture plate, also manufactured by Flexcell International Corporation. The culture plate hold a 35 mm well consisting of a matrix bound to a silicone rubber silicone rubber membrane (FlexCell, 2019). The membrane is then deformed using either of the corresponding Flexcell® FX-6000™ or Flex Jr.™ Tension Systems. Figure A6 below shows the complete device.

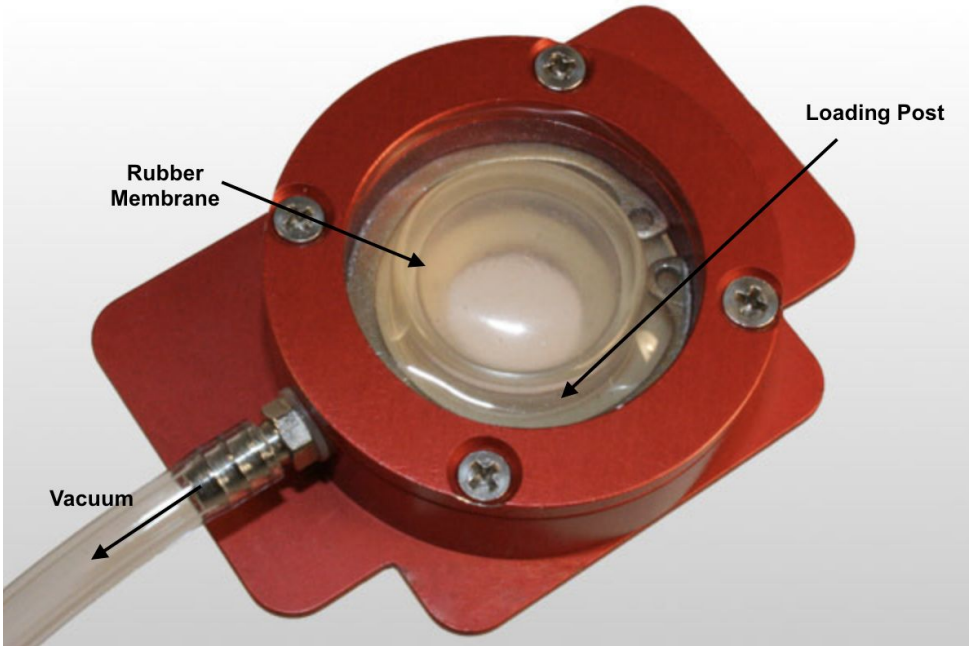


Figure A6: Flexcell StageFlexer microscope mountable cell stretching well composed of a Bioflex cell culture plate, loading post, and silicone rubber-matrix (silicone rubber membrane) well. To stretch the cells, this assembly is connected to either the Flexcell

FX-6000 or the Flex Jr. Tension Systems(Copyright © 2011 Flexcell International Corporation).

The cells are seeded on the silicone rubber membrane in a monolayer and cell activity is monitored under a microscope. The Flexcell FX-6000 and Flex Jr. Tension Systems control the strain, frequency, amplitude, waveform, and cycles of cell stretch. Figure A7 shows and example of the complete cell stretching configuration with the Flex Jr. Tension System.

Figure A7: The complete cell stretching system using the Flex Jr. Tension System and StageFlexer microscopy device(Copyright © 2011 Flexcell International Corporation).

According to the manufacturer, the Flexcell can achieve a 14% elongation of the substrate (FlexCell, 2019). Table A2 below shows the substrate elongation compared to the loading station diameter in mm.

Table A2: StageFlexer Substrate Elongation with Induced Strain (Copyright © 2011 Flexcell International Corporation)

Device	Strain Range
StageFlexer®	1.6% - 14.8% (25 mm) 1.9% - 13.4% (28 mm) 2.1% - 8.6 % (31 mm)
StageFlexer® Jr.	1.8% - 13. 8% (BioFlex®) 2.5% - 17.7% (Tissue Train®) 1.5% - 7.9% (UniFlex®)
FlexFlow™	0.8% - 4.3%

A vacuum is used in the StageFlexer device to produce a displacement of substrate. Figure A8 gives a visual representation from the side view of how the StageFlexer generates a strain in an equibiaxial application.

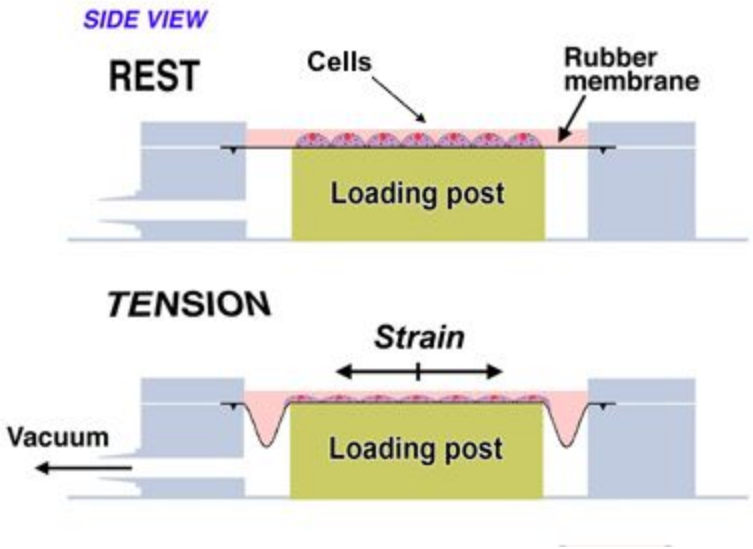


Figure A8: A side view of the StageFlexer producing an equibiaxial strain on a silicone rubber membrane pulled over a loading post by a pressure force generated by a vacuum(Copyright © 2011 Flexcell International Corporation).

This company also manufactures several variations of the StageFlexer including the StageFlexer Jr., FlexFlow, and the StagePresser. The StageFlexer Jr. deforms membranes free of a culture plate allowing for real time cell observation during stretching, shown in figure A9.

Figure A9: StageFlexer Jr. assembly and available loading posts(Copyright © 2011 Flexcell International Corporation).

The FlexFlow applies a laminar shear stress or cyclic strain to cells in culture and allows observation during testing on standard upright microscope, shown in figure A10.

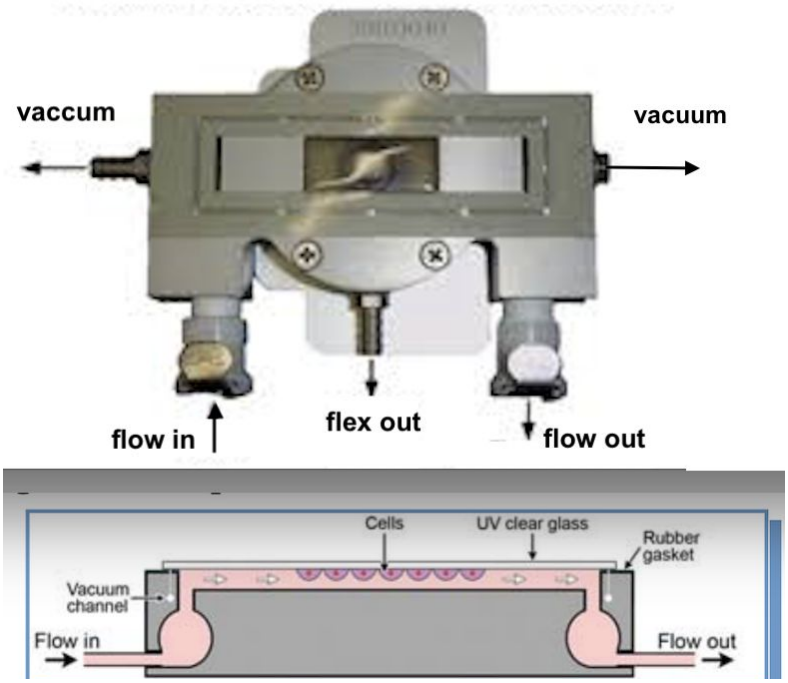


Figure A10: FlexFlow stretch device showing a top and side view of the assembly and fluid flow to induce strain(Copyright © 2011 Flexcell International Corporation).

The StagePresser is controlled by the FX-5000 System and is built to compress single tissue samples or 3D cultured cells and allows real time cell or tissue activity viewing on a microscope. Figure A11 shows the loaded and unloaded size view schematic of the StagePressure and its components.

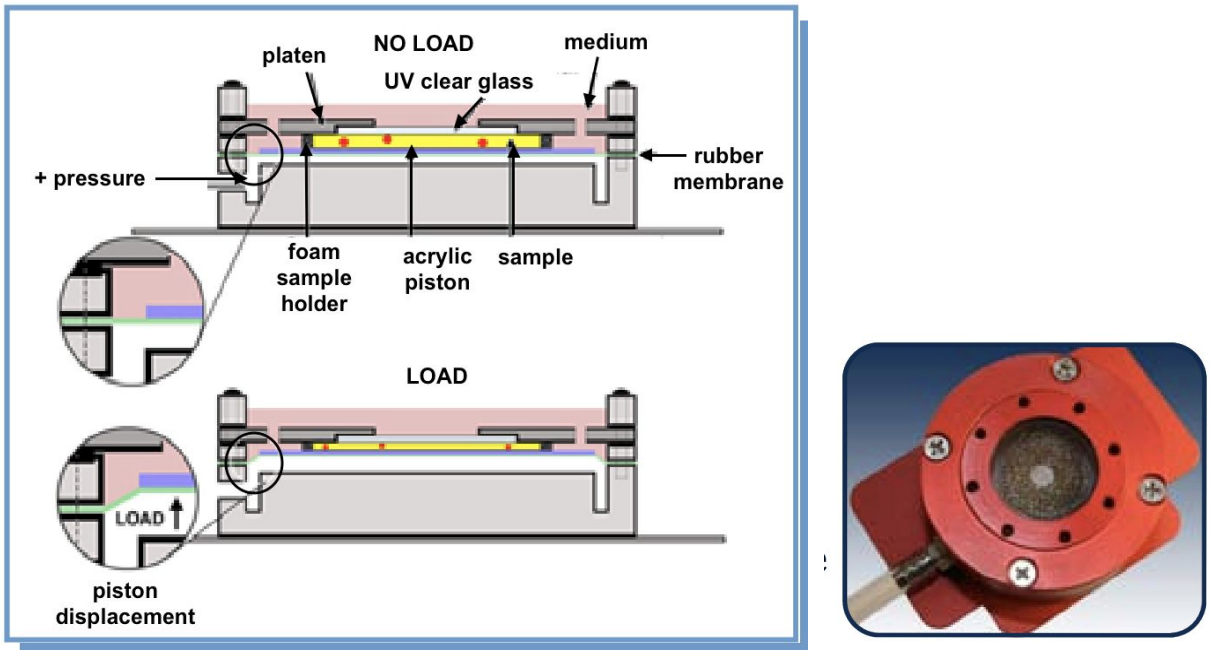


Figure A11: (Left) Side view of the StagePresser and its components in an unloaded and loaded state; (Right) Complete microscope mountable cell compressing assembly (Copyright © 2011 Flexcell International Corporation).

Section A5: Detailed Descriptions of Cell Stretcher CS-10 Series

The Cell Stretcher CS-10 Series is a microscope mountable uni-axial straining and compression system that provides motion compensation for accurate cell tracking ("Computer Controlled Cell Deforming - Cell Stretcher", 2019). A ball screw driven compensator motor allows for cells of the same region on a membrane well to be continuously monitored over a microscope. Thus, a specific region remains centered over the microscope objective. This device is run by two DC servo motors to create uniaxial strains or compressions on a PDMS membrane. The DC motors and the membrane well set up are mounted to the microscope stage. The components of this system are diagrammed in figure A12. The red arrows show the displacement

and lateral compensation motion during movement. The SC motor generates deformation on the membrane by moving the sliding blocks.

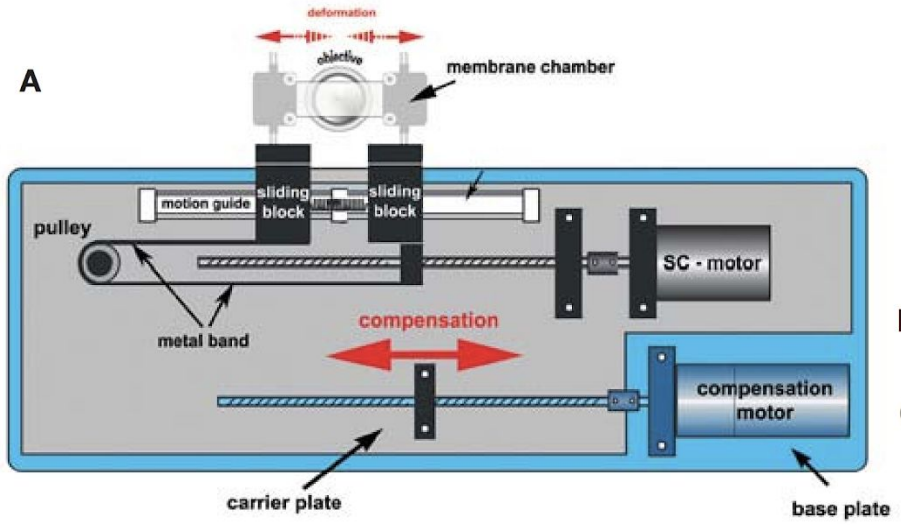


Figure A12: Cell Stretcher CS-10 uniaxial cell straining and compressing device schematic of components and direction of deformation and lateral compensation(Copyright © Electron Microscopy Sciences, 2019).

The computer system required to run the motors in this system is based on JAVA and the user interface allows the user to control the distance of deformation, activate the compensation motor, and define the stretching pattern protocol (i.e. sinusoidal). Figure A13 illustrates the design and features of the GUI. Letter A shows the stretch profile of the membrane during a deformation cycle. Letter B is an enlargement of that deformation to isolate details. Letter C represents the calibration features and manual controls. Letter D shows the parameter input table to control stretch pattern, strain, and duration ("Computer Controlled Cell Deforming - Cell Stretcher", 2019).

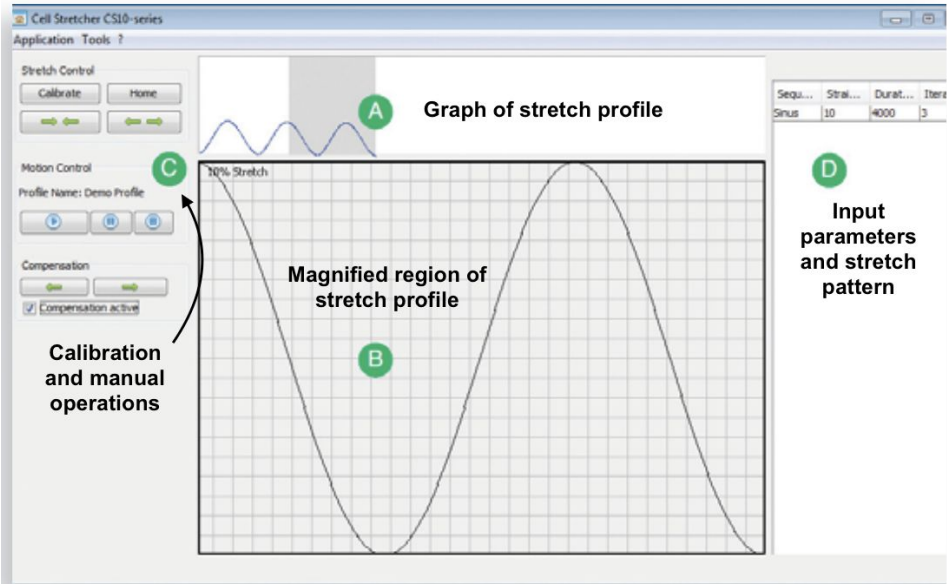


Figure A13: Cell Stretcher CS-10 Uniaxial control GUI containing a calibration button to alter deformation distance, a graphical representation of the stretch profile, and access to other features like calibration(Copyright © Electron Microscopy Sciences, 2019).

The total cost of this system is $39,000 for the compensation model and $25,000 for the linear model without compensation ("Computer Controlled Cell Deforming - Cell Stretcher", 2019).

A biaxial attachment for this system is also available through Electron Microscopy Systems. The dc-CS10 Biaxial system is different from the uniaxial/compression model only in the configuration of the membrane assembly. The platform for the user interface remains unchanged, however, there is no available information about parameter options. Unfortunately, the attachment is not specifically marked. The only specifications given were the dimensions and power supply - 24(w)x 23(h)x11 cm; 110/220V 50Hz switchable ("Computer Controlled Cell

Deforming - Cell Stretcher", 2019). The total cost for this system is $30,000. Figure A14 shows the dcCS-10 Biaxial membrane assembly.

Figure A14: dcCS-10 Biaxial Cell Stretcher membrane assembly. This assembly replaces the membrane chamber shown in figure A12 in the microscope mounted system(Copyright © Electron Microscopy Sciences, 2019).

Appendix B: Material Properties & Safety Data Sheets for Resin Stretch Well Mold

Section B1: Assessing Mold Material Hardness Properties

The Shore hardness scale gives a relative ranking of the hardness of two materials(resistance to penetration from a harder material). Methacrylic acid ester resin provides a Shore A hardness of 75-85 whereas acrylic (PMMA) has a Shore D value of 90-99 (Boudou et al., 2009; FlexCell, 2019). Figure B1 shows the hardness relationship between MMA Resin and PMMA.

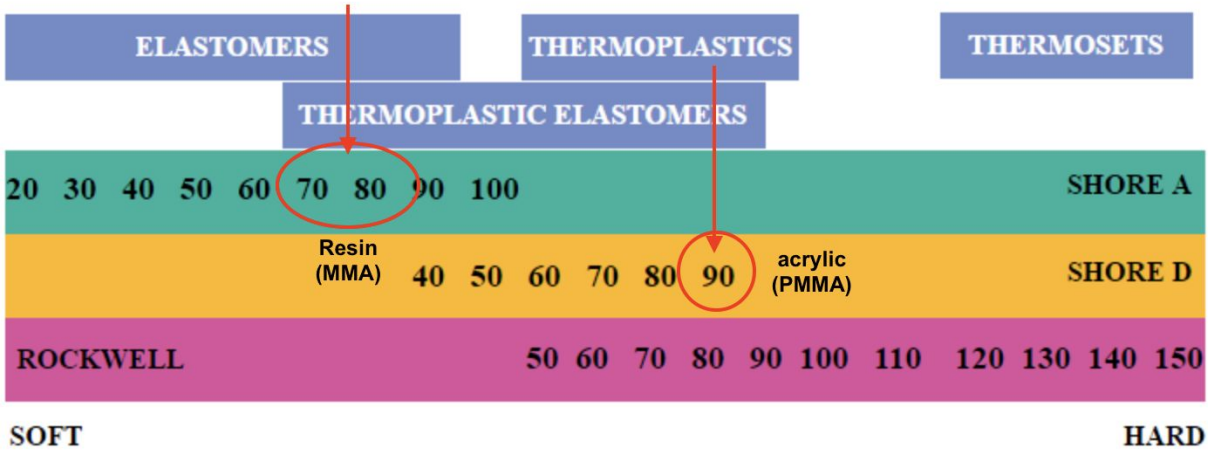


Figure B1: Hardness Scales for Different Polymer Materials showing MMA resin having a lower hardness value compared to acrylic (PMMA) (Copyright © Omnexus 2019).

Even though the resin (MMA) material had a lower hardness, we chose to manufacture the new well molds by 3D SLA printing for the high possible resolution. Figure B2 shows the notched IZOD test results for the resin material used in the well mold prototype. The material properties, printing resolution, and curing time and temperature were the primary considerations

for choosing the 3D printed resin mold material.

STANDARD RESINS

CLEAR FLGPCL04 | WHITE FLGPWH04 | GREY FLGPGR04 | BLACK FLGPBK04 | COLOR BASE FLGPCB01

	METRIC[1]		IMPERIAL[1]		METHOD
	Green[2]	Post-Cured[3]	Green[2]	Post-Cured[3]	
Tensile Properties					
Ultimate Tensile Strength	38 MPa	65 MPa	5510 psi	9380 psi	ASTM D 638-10
Tensile Modulus	1.6 GPa	2.8 GPa	234 ksi	402 ksi	ASTM D 638-10
Elongation at Failure	12 %	6.2 %	12 %	6.2 %	ASTM D 638-10
Flexural Properties					
Flexural Modulus	1.25 GPa	2.2 GPa	181 ksi	320 ksi	ASTM C 790-10
Impact Properties					
Notched IZOD	16 J/m	25 J/m	0.3 ft-lbf/in	0.46 ft-lbf/in	ASTM D 256-10
Temperature Properties					
Heat Deflection Temp. @ 264 psi	42.7 °C	58.4 °C	108.9 °F	137.1 °F	ASTM D 648-07
Heat Deflection Temp. @ 66 psi	49.7 °C	73.1 °C	121.5 °F	163.6 °F	ASTM D 648-07

Figure B2: Notched IZOD test results to determine resin hardness based on the ASTM D 256-10 standard.

Figure B3 compares the printing resolution for different resins. The grey resin used prints with a resolution of 25 microns.

	200 Microns	100 Microns	50 Microns	25 Microns
Clear Resin	Form 1+	✔	✔	✔
Black Resin		✔	✔	✔
White Resin		✔	✔	Form 1/1+
Grey Resin		✔	✔	✔
Color Resin (Form 2 only)		✔	✔	✔
Flexible Resin		✔	✔	

Figure B3:Printing resolution for different resin types. The grey resin used to create the well mold prototype has a 25 micron printing resolution.

Appendix C: Motor Specifications

Section C1: Stepper Motor from 2013 MQP Device Specifications

Figure C1 shows the linear velocity of different linear motors from Haydon. The motor that the previous MQP used is identified as "Q" with a blue color. The X-axis shows the linear velocity in mm/sec in parentheses.

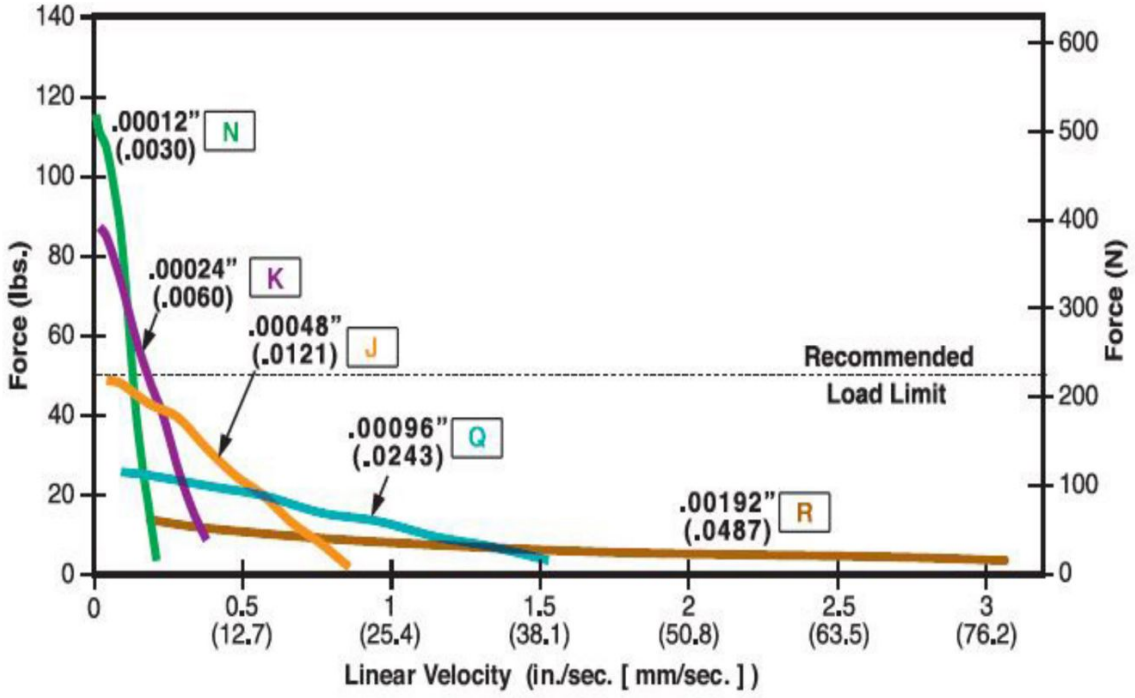


Figure C1. Force Velocity Graph for Hybrid Linear Actuators (Labeled Q)

The desired speed for 10% strain at 1Hz is 10 mm/sec. As seen above, the motor is able to handle that speed, and much faster. However, there is an issue with the device and the motor cannot reach the speeds given by the manufacturer.

PA-07 Specifications

Input Voltage:	12 VDC
Stroke:	0.5-12 inches
Force:	5 lbs
Speed:	0.59"/sec
Protection Class:	IP66
Feedback:	No
Operational Temperature:	0°C~50°C
Noise:	db<45(A)
Duty Cycle:	10%

Limit Switch:	Built In, Non-Adjustable
Current (full load):	0.24A
Mounting Holes:	0.16"
Motor Type:	Brushed DC Motor
Screw Type:	ACME
Housing Type:	Plastic and Aluminum Alloy
Fully Retracted:	3.81" + stroke, 3.18" for stroke 2" and under
Fully Extended:	3.81" + stroke + stroke, 3.18" for stroke 2" and under
Certifications:	CE and RoHS
Unit Weight:	0.15 lbs
Warranty:	18 Months

Figure C2: Specifications and Velocity Profiles for the Current Device's Linear Actuator Motors

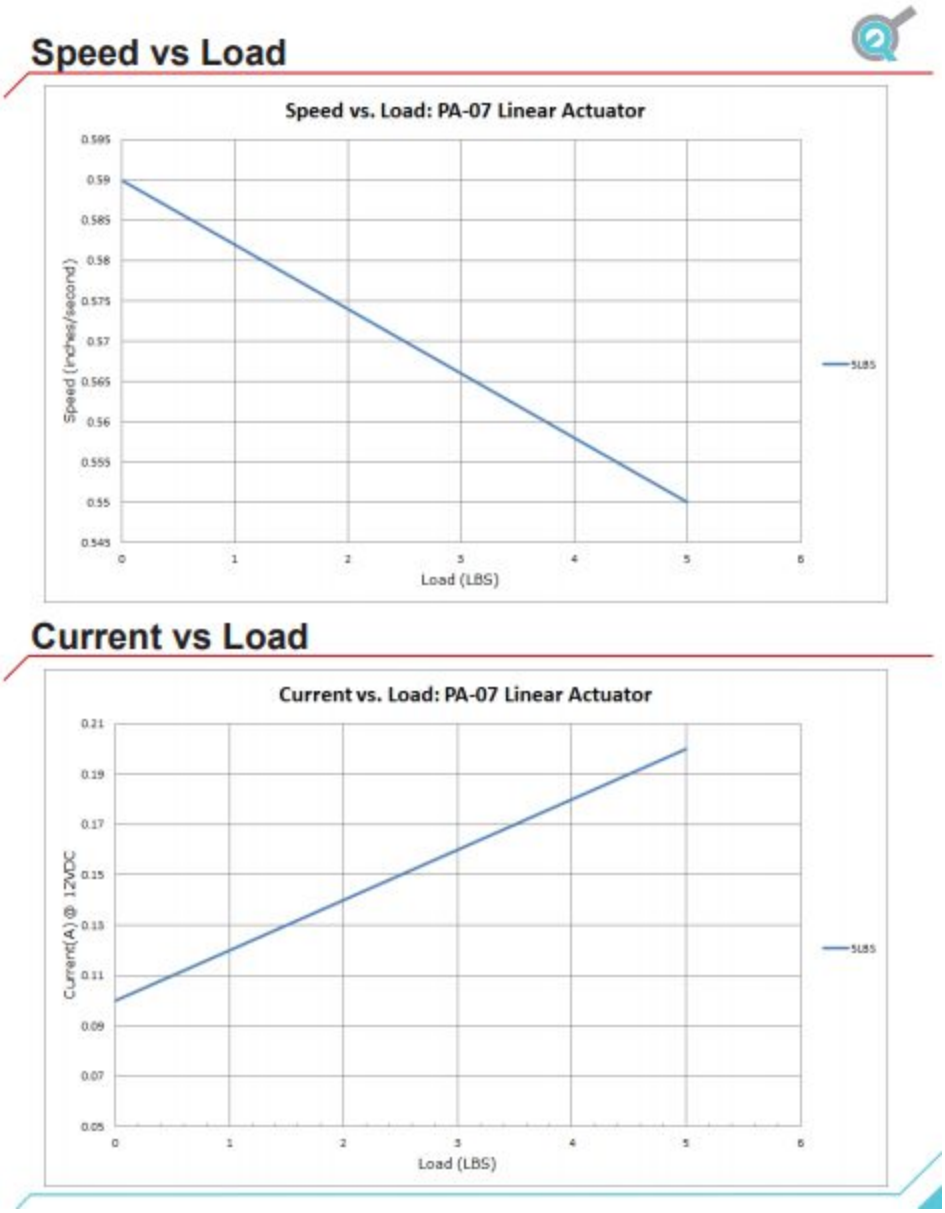


Figure C3: Velocity and Current Profiles for the Current Device's Linear Actuator Motors: (A) Shows the max velocity attained by PA-07 micro-linear actuators; (B)Shows maximum current of ~0.20A

Appendix D: Example Segment of the Former MQP Device

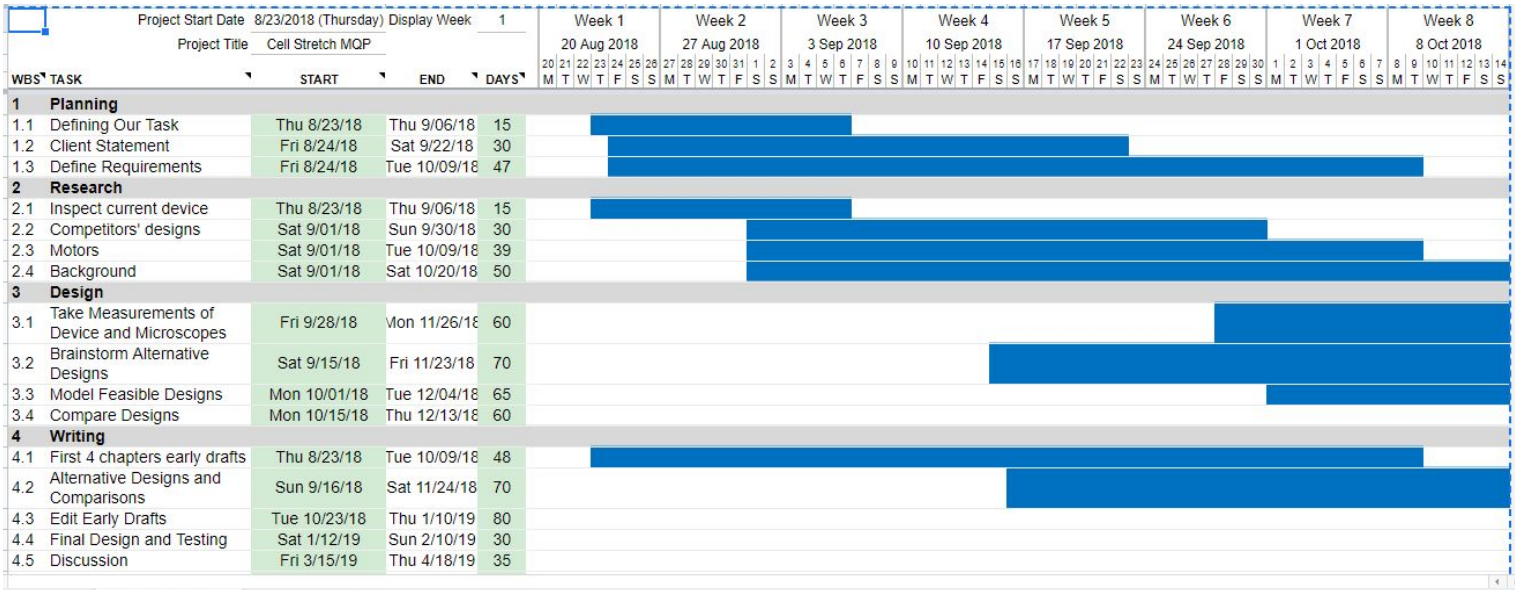
Code to Run Motors:

```
case 2: // Runs standard non-alternating strains

while (Serial.available() == 0){

 while (i == 1){
 stepper1.setMaxSpeed(xMaxSpeedRetract);
 stepper2.setMaxSpeed(xMaxSpeedRetract);
 stepper3.setMaxSpeed(yMaxSpeedRetract);
 stepper4.setMaxSpeed(yMaxSpeedRetract);

 stepper1.moveTo(0);
 stepper2.moveTo(0);
 stepper3.moveTo(0);
 stepper4.moveTo(0);

 stepper1.run();
 stepper2.run();
 stepper3.run();
 stepper4.run();

 if (stepper1.distanceToGo() == 0 &
 stepper2.distanceToGo() == 0 &
 stepper3.distanceToGo() == 0 &
 stepper4.distanceToGo() == 0){
 i=i+1;
 break;
 }
 }

 while (i == 2){
 stepper1.setMaxSpeed(xMaxSpeedExtend);
 stepper2.setMaxSpeed(xMaxSpeedExtend);
 stepper3.setMaxSpeed(yMaxSpeedExtend);
 stepper4.setMaxSpeed(yMaxSpeedExtend);

 stepper1.moveTo(stepsX);
 stepper2.moveTo(stepsX);
```

```
stepper3.moveTo(stepsY);
stepper4.moveTo(stepsY);

stepper1.run();
stepper2.run();
stepper3.run();
stepper4.run();

if (stepper1.distanceToGo() == 0 &
stepper2.distanceToGo() == 0 &
stepper3.distanceToGo() == 0 &
stepper4.distanceToGo() == 0){
i = i+1;
cycleCount = cycleCount+0.5;
Serial.println(cycleCount);
delay(timeHeldExtended*1000);
break;
}
}

while (i == 3){
stepper1.setMaxSpeed(xMaxSpeedRetract);
stepper2.setMaxSpeed(xMaxSpeedRetract);
stepper3.setMaxSpeed(yMaxSpeedRetract);
stepper4.setMaxSpeed(yMaxSpeedRetract);

stepper1.moveTo(0);
stepper2.moveTo(0);
stepper3.moveTo(0);
stepper4.moveTo(0);

stepper1.run();
stepper2.run();
stepper3.run();
stepper4.run();

if (stepper1.distanceToGo() == 0 &
stepper2.distanceToGo() == 0 &
stepper3.distanceToGo() == 0 &
stepper4.distanceToGo() == 0){
i=i-1;
cycleCount = cycleCount+0.5;
Serial.println(cycleCount);
delay(timeHeldRetracted*1000);
break;
```

```
 }
 }
}
break;
```

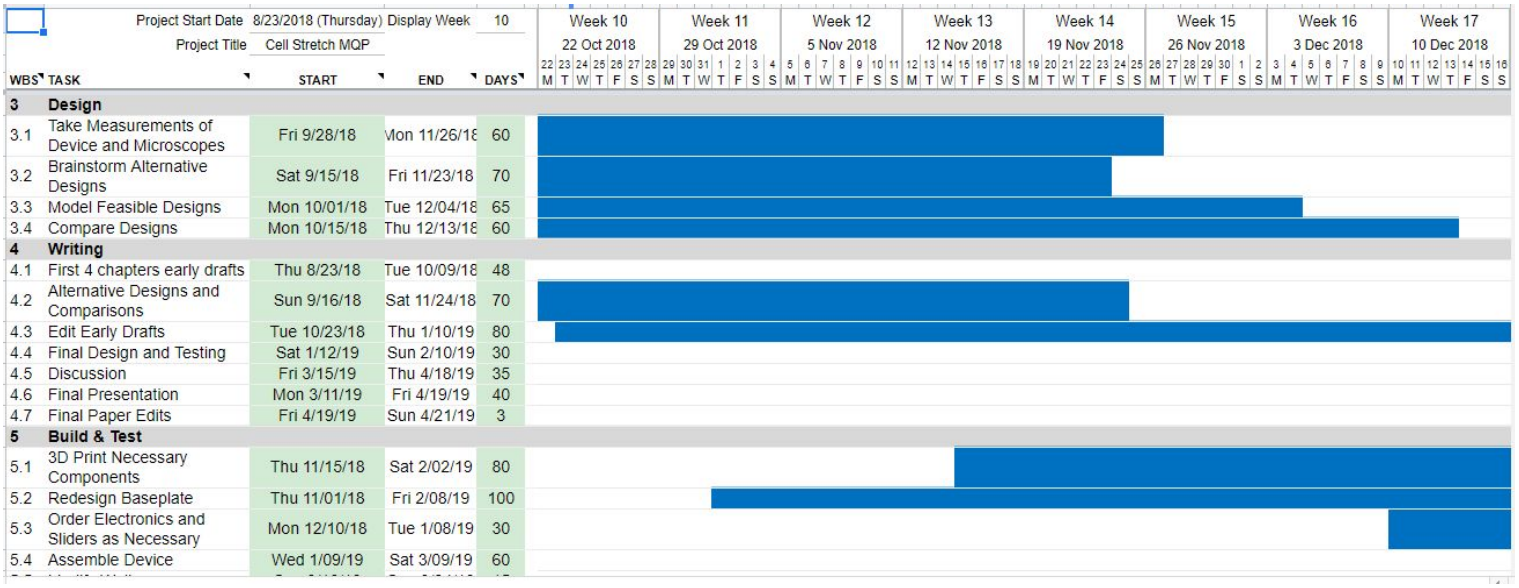
Appendix E: Gantt Charts Used in Management Approach



Project Start Date 8/23/2018 (Thursday) Display Week 1
Project Title Cell Stretch MQP

WBS	TASK	START	END	DAYS
1	**Planning**			
1.1	Defining Our Task	Thu 8/23/18	Thu 9/06/18	15
1.2	Client Statement	Fri 8/24/18	Sat 9/22/18	30
1.3	Define Requirements	Fri 8/24/18	Tue 10/09/18	47
2	**Research**			
2.1	Inspect current device	Thu 8/23/18	Thu 9/06/18	15
2.2	Competitors' designs	Sat 9/01/18	Sun 9/30/18	30
2.3	Motors	Sat 9/01/18	Tue 10/09/18	39
2.4	Background	Sat 9/01/18	Sat 10/20/18	50
3	**Design**			
3.1	Take Measurements of Device and Microscopes	Fri 9/28/18	Mon 11/26/18	60
3.2	Brainstorm Alternative Designs	Sat 9/15/18	Fri 11/23/18	70
3.3	Model Feasible Designs	Mon 10/01/18	Tue 12/04/18	65
3.4	Compare Designs	Mon 10/15/18	Thu 12/13/18	60
4	**Writing**			
4.1	First 4 chapters early drafts	Thu 8/23/18	Tue 10/09/18	48
4.2	Alternative Designs and Comparisons	Sun 9/16/18	Sat 11/24/18	70
4.3	Edit Early Drafts	Tue 10/23/18	Thu 1/10/19	80
4.4	Final Design and Testing	Sat 1/12/19	Sun 2/10/19	30
4.5	Discussion	Fri 3/15/19	Thu 4/18/19	35

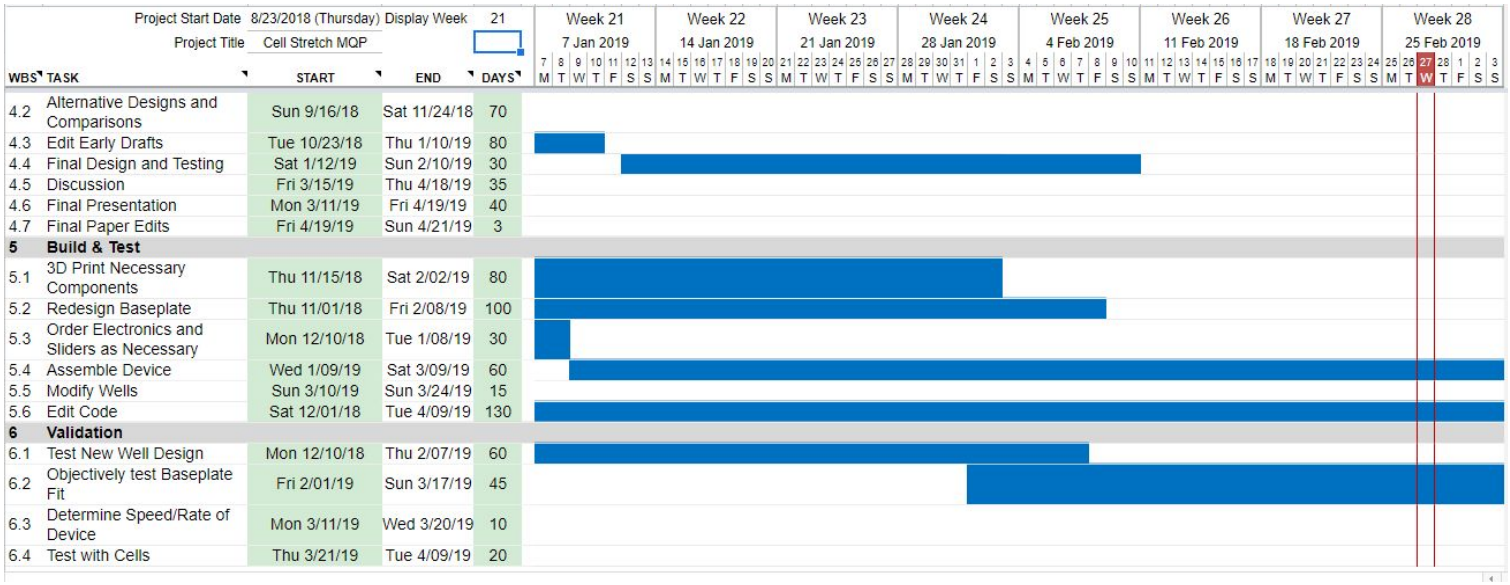
Figure E1: Gantt chart to plan project management approach for A-term focused on planning and research followed by initial designing and writing.



Project Start Date 8/23/2018 (Thursday) Display Week 10
Project Title Cell Stretch MQP

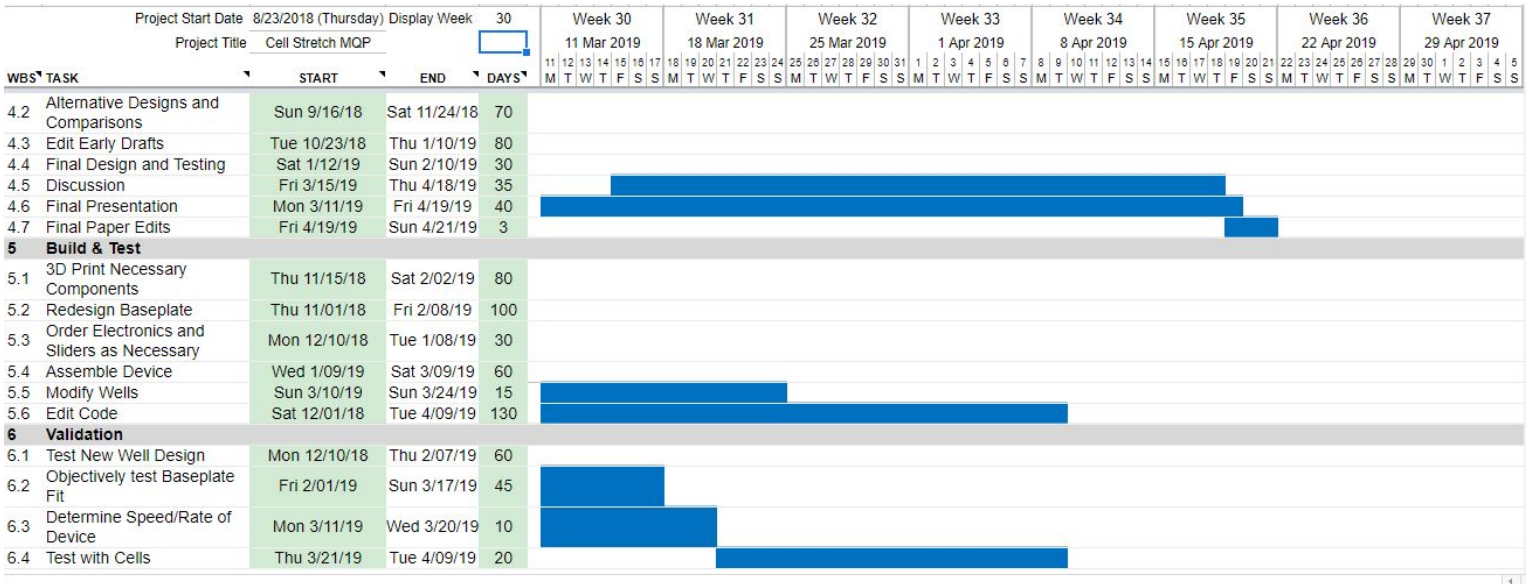
WBS	TASK	START	END	DAYS
3	**Design**			
3.1	Take Measurements of Device and Microscopes	Fri 9/28/18	Mon 11/26/18	60
3.2	Brainstorm Alternative Designs	Sat 9/15/18	Fri 11/23/18	70
3.3	Model Feasible Designs	Mon 10/01/18	Tue 12/04/18	65
3.4	Compare Designs	Mon 10/15/18	Thu 12/13/18	60
4	**Writing**			
4.1	First 4 chapters early drafts	Thu 8/23/18	Tue 10/09/18	48
4.2	Alternative Designs and Comparisons	Sun 9/16/18	Sat 11/24/18	70
4.3	Edit Early Drafts	Tue 10/23/18	Thu 1/10/19	80
4.4	Final Design and Testing	Sat 1/12/19	Sun 2/10/19	30
4.5	Discussion	Fri 3/15/19	Thu 4/18/19	35
4.6	Final Presentation	Mon 3/11/19	Fri 4/19/19	40
4.7	Final Paper Edits	Fri 4/19/19	Sun 4/21/19	3
5	**Build & Test**			
5.1	3D Print Necessary Components	Thu 11/15/18	Sat 2/02/19	80
5.2	Redesign Baseplate	Thu 11/01/18	Fri 2/08/19	100
5.3	Order Electronics and Sliders as Necessary	Mon 12/10/18	Tue 1/08/19	30
5.4	Assemble Device	Wed 1/09/19	Sat 3/09/19	60

Figure E2: Gantt chart to plan project management approach for B-term focused on developing alternative designs, writing, and building prototypes.



WBS	TASK	START	END	DAYS
4.2	Alternative Designs and Comparisons	Sun 9/16/18	Sat 11/24/18	70
4.3	Edit Early Drafts	Tue 10/23/18	Thu 1/10/19	80
4.4	Final Design and Testing	Sat 1/12/19	Sun 2/10/19	30
4.5	Discussion	Fri 3/15/19	Thu 4/18/19	35
4.6	Final Presentation	Mon 3/11/19	Fri 4/19/19	40
4.7	Final Paper Edits	Fri 4/19/19	Sun 4/21/19	3
5	**Build & Test**			
5.1	3D Print Necessary Components	Thu 11/15/18	Sat 2/02/19	80
5.2	Redesign Baseplate	Thu 11/01/18	Fri 2/08/19	100
5.3	Order Electronics and Sliders as Necessary	Mon 12/10/18	Tue 1/08/19	30
5.4	Assemble Device	Wed 1/09/19	Sat 3/09/19	60
5.5	Modify Wells	Sun 3/10/19	Sun 3/24/19	15
5.6	Edit Code	Sat 12/01/18	Tue 4/09/19	130
6	**Validation**			
6.1	Test New Well Design	Mon 12/10/18	Thu 2/07/19	60
6.2	Objectively test Baseplate Fit	Fri 2/01/19	Sun 3/17/19	45
6.3	Determine Speed/Rate of Device	Mon 3/11/19	Wed 3/20/19	10
6.4	Test with Cells	Thu 3/21/19	Tue 4/09/19	20

Figure E3: Gantt chart to plan project management approach for C-term focused on building prototype designs and testing design performance standards.



WBS	TASK	START	END	DAYS
4.2	Alternative Designs and Comparisons	Sun 9/16/18	Sat 11/24/18	70
4.3	Edit Early Drafts	Tue 10/23/18	Thu 1/10/19	80
4.4	Final Design and Testing	Sat 1/12/19	Sun 2/10/19	30
4.5	Discussion	Fri 3/15/19	Thu 4/18/19	35
4.6	Final Presentation	Mon 3/11/19	Fri 4/19/19	40
4.7	Final Paper Edits	Fri 4/19/19	Sun 4/21/19	3
5	**Build & Test**			
5.1	3D Print Necessary Components	Thu 11/15/18	Sat 2/02/19	80
5.2	Redesign Baseplate	Thu 11/01/18	Fri 2/08/19	100
5.3	Order Electronics and Sliders as Necessary	Mon 12/10/18	Tue 1/08/19	30
5.4	Assemble Device	Wed 1/09/19	Sat 3/09/19	60
5.5	Modify Wells	Sun 3/10/19	Sun 3/24/19	15
5.6	Edit Code	Sat 12/01/18	Tue 4/09/19	130
6	**Validation**			
6.1	Test New Well Design	Mon 12/10/18	Thu 2/07/19	60
6.2	Objectively test Baseplate Fit	Fri 2/01/19	Sun 3/17/19	45
6.3	Determine Speed/Rate of Device	Mon 3/11/19	Wed 3/20/19	10
6.4	Test with Cells	Thu 3/21/19	Tue 4/09/19	20

Figure E4: Gantt chart to plan project management approach for D-term focused on building the final design concept and performing validation testing.

Appendix F. GUI Code

```
// by Megan Hendrie and Joshua Boynton

//imported librarires

import controlP5.*;

import processing.serial.*;

//defining variables

ControlP5 cp5;

Textarea myTextarea;

Serial port;

Slider frequencySlider;

Slider xStrainSlider;

Slider yStrainSlider;

float timePerCycleTempTemp;

float powertoSpeedX;

float powertoSpeedY;

float j;

Textlabel cycleTextLabel;

int myColor = color(51,153,255);

float valueMessage;

String cycleNumber = "0";

int sliderTicks1 = 100;

Slider abc;
```

```
//sets the size of the window
void setup() {
  size(2500,2500);
  cp5 = new ControlP5(this);
  // creates a text boxwith the following parameters
  myTextarea = cp5.addTextarea("txt")
                  .setPosition(800,880)
                  .setSize(800,800)
                  .setFont(createFont("arial",50))
                  .setLineHeight(56)
                  .setColor(color(204))
                  .setColorBackground(color(255,100))
                  .setColorForeground(color(255,100));
                  ;
  myTextarea.setText("This is how the well will deform.");
        // this creates a vertical and horizontal slider each with a pre-set strain of 10%
        xStrainSlider=cp5.addSlider("Strain_X")
        .setRange(0,20)
        .setValue(10)
        .setPosition(700,1800)
        .setSize(800,120)
        .setNumberOfTickMarks(21)
```

```
        .setFont(createFont("arial",50))
        ;
        yStrainSlider=cp5.addSlider("Strain_Y")
        .setPosition(1870,600)
        .setSize(120,800)
        .setRange(0,20)
        .setValue(10)
        .setNumberOfTickMarks(21)
        .setFont(createFont("arial",50))
        ;
        noStroke();
        cp5 = new ControlP5(this);

  //String portName = Serial.list()[0]; //change the 0 to a 1 or 2 etc. to match your port
//IMPORTANT, this code sets up the connection from the GUI to the Arduino
  //port = new Serial(this, portName, 9600);
 // create the frequency slider with tick marks, starts at 1 Hz.
        frequencySlider=cp5.addSlider("Frequency Hz")
        .setPosition(200,600)
        .setSize(120,800)
        .setRange(0.1,1)
        .setValue(1)
```

```
.setNumberOfTickMarks(10)
.setFont(createFont("arial",50))
;
//sets the home position of the linear actuators to be at "zero"
cp5.addButton("Home")
.setValue(0)
.setPosition(700,200)
.setSize(800,80)
.setFont(createFont("arial",50))
;
//tells program to initiate non-alternating strain pattern
cp5.addButton("Strain")
.setValue(0)
.setPosition(700,300)
.setSize(380,80)
.setFont(createFont("arial",50))
;
//tells program to initiate alternating strain pattern
cp5.addButton("Alt_Strain")
.setValue(0)
.setPosition(1090,300)
.setSize(410,80)
```

```
        .setFont(createFont("arial",50))
        ;
        cycleTextLabel = cp5.addTextlabel("cyclelabel") //creates label for number of cycles
        .setPosition(2000,500)
        .setFont(createFont("arial",50));
        //tells program to cease running
        cp5.addButton("STOP")
        .setValue(0)
        .setPosition(700,400)
        .setSize(800,80)
        .setFont(createFont("arial",50))
        ;
}
// maintaines text when adjusting box area
void keyPressed() {
  if(key=='r') {
        myTextarea.setText("This is how the well will deform.");
  } else if(key=='c') {
        myTextarea.setColor(0xffffffff);
  }
}
//changes the size of the text as box area changes andd updates the number of cycles
```

```
void draw() {
  background(0);
  if(keyPressed && key==' ') {
        myTextarea.scroll((float)mouseX/(float)width);
  }
  if(keyPressed && key=='l') {
        myTextarea.setLineHeight(mouseY);
  }
  cycleTextLabel.setText(cycleNumber + " cycles");
}
//changes the size of the box area when the sliders change
void Strain_X(int theValue) {
  myTextarea.setWidth(theValue*40);
}
void Strain_Y(int theValue) {
  myTextarea.setHeight(theValue*40);
}
void slider(float theColor) {
  myColor = color(theColor);
  println("a slider event. setting background to "+theColor);
}
```

```
public void Strain(){ //program that runs when pressing strain button, runs non-alternating strain
pattern
println("I'm running");
float frequencyValue = frequencySlider.getValue(); // value taken from frequency slider
int timePerCycleTemp =round((100/frequencyValue)); // converts frequency to time value,
multiplies it by 100 for easier conversion to arduino. Divides it by 100 later in arduino program.
println(timePerCycleTemp);
float displacementXinput = xStrainSlider.getValue(); // value taken from x strain slider
float displacementYinput = yStrainSlider.getValue(); // value taken from y strain slider
int
powertoSpeedX=round(((((((displacementXinput/2)*0.1)+0.5*displacementXinput)/25.4)*2)/0.0
023137)*frequencyValue); // IMPORTANT: converts x_strain value to power input for x
direction. 0=minimum, 255= maximum. This equation needs to be changed to change the length
the well is streteched in this direction.
int
powertoSpeedY=round(((((((displacementYinput/2)*0.1)+0.5*displacementYinput)/25.4)*2)/0.0
023137)*frequencyValue); // IMPORTANT: converts y_strain value to power input for y
direction. 0=minimum, 255= maximum. This equation needs to be changed to change the length
the well is streteched in this direction.
println(powertoSpeedX);
if(powertoSpeedX>255 || powertoSpeedY>255){
```

```
println("ERROR parameters too high for motors, lower frequency and/or strain");} //error for
when power input values are too high for motor to handle
else{ //values given to arduino to run code
port.write(str(timePerCycleTemp)); // time value
port.write(",");
port.write(str(powertoSpeedX)); // x power input value
port.write(",");
port.write(str(powertoSpeedY)); // y power input value
port.write(",");
port.write(str(1)); // selector value for running different programs
port.write("q"); // random variable to end string reading in arduino
}
}
public void STOP(){ //program that runs when pressing stop button, stops actuators
println("I'm stopping");
//values given to arduino to run code
port.write(str(1)); // time value
port.write(",");
port.write(str(0)); // x power input value
port.write(",");
port.write(str(0)); // y power input value
port.write(",");
```

```
port.write(str(2));// selector value for running different programs
port.write("q");   // random variable to end string reading in arduino
}
public void Home(){ //program that runs when pressing home button, sets actuator to home
postition
println("Going home");
//values given to arduino to run code
port.write(str(1)); // time value
port.write(",");
port.write(str(255)); // x power input value
port.write(",");
port.write(str(255)); // y power input value
port.write(",");
port.write(str(2)); // selector value for running different programs
port.write("q"); // random variable to end string reading in arduino
}
public void Alt_Strain(){ //program that runs when pressing alt strain button, runs alternating
strain pattern
println("I'm running");
float frequencyValue = frequencySlider.getValue(); // value taken from frequency slider
int timePerCycleTemp =round((100/frequencyValue)); // converts frequency to time value,
multipites it by 100 for easier conversion to arduino. Divides it by 100 later in arduino program.
```

```
println(timePerCycleTemp);
float displacementXinput = xStrainSlider.getValue(); // value taken from x strain slider
float displacementYinput = yStrainSlider.getValue(); // value taken from y strain slider
int
powertoSpeedX=round(((((((displacementXinput/2)*0.1)+0.5*displacementXinput)/25.4)*2)/0.0
023137)*frequencyValue); // IMPORTANT: converts x_strain value to power input for x
direction. 0=minimum, 255= maximum. This equation needs to be changed to change the length
the well is streteched in this direction.
int
powertoSpeedY=round(((((((displacementYinput/2)*0.1)+0.5*displacementYinput)/25.4)*2)/0.0
023137)*frequencyValue); // IMPORTANT: converts y_strain value to power input for y
direction. 0=minimum, 255= maximum. This equation needs to be changed to change the length
the well is streteched in this direction.
println(powertoSpeedX);
if(powertoSpeedX>255 || powertoSpeedY>255){
println("ERROR parameters too high for motors, lower frequency and/or strain");} //error for
when power input values are too high for motor to handle
else{ //values given to arduino to run code
port.write(str(timePerCycleTemp)); // time value
port.write(",");
port.write(str(powertoSpeedX)); // x power input value
port.write(",");
```

```
port.write(str(powertoSpeedY)); // y power input value
port.write(",");
port.write(str(3)); // selector value for running different programs
port.write("q"); // random variable to end string reading in arduino
}
}
void serialEvent(Serial p) { //returns the current amount of cycles from arduino to GUI
  // get message till line break (ASCII > 13)
  String message = port.readStringUntil(13);
  if (message != null) {
          valueMessage = float(message);
          cycleNumber = str(valueMessage);
          //  println("This many cycles: "+ value);
  }
}


}
```

Appendix G. Code to Operate Motors

```
/*      Code to control up to 4 actuators, using the Robot Power MultiMoto driver.
  Hardware:
        - Robot Power MultiMoto
        - Arduino Uno
        Wiring:
 - Connect actuators to the M1, M2, M3, M4 connections on the MultiMoto board.
 - Connect the negative (black) to the right connection, positive (red) to the left.
 - Connect a 12 volt source (minimum 1A per motor if unloaded, 8A per motor if fully loaded)to
the BAT terminals. Ensure that positive and negative are placed in the correct spots.
  Code modified by Progressive Automations from the example code provided by Robot Power
        <a href="http://www.robotpower.com/downloads/" rel="nofollow">
http://www.robotpower.com/downloads/</a>
        Robot Power MultiMoto v1.0 demo
        This software is released into the Public Domain
*/
// include the SPI library:
#include <SPI.h>       //defines variable names
// L9958 slave select pins for SPI
#define SS_M3 13
#define SS_M2 12
#define SS_M1 11
```

```
#define SS_M4 14
// L9958 DIRection pins
#define DIR_M1 2
#define DIR_M2 3
#define DIR_M3 4
#define DIR_M4 7
// L9958 PWM pins
#define PWM_M1 9
#define PWM_M2 10 // Timer1
#define PWM_M3 5
#define PWM_M4 6
// L9958 Enable for all 4 motors //define variables to be used in code
#define ENABLE_MOTORS 8
int pwm1, pwm2, pwm3, pwm4;
boolean dir1, dir2, dir3, dir4;
const int numberOfFields = 4;          // How many values are being sent from GUI to arduino
int fieldIndex = 0;                    // The current field index being received
int values[numberOfFields];   // An array which holds values of all the fields
int i = 1;                     // A variable for running the motor loop
float cycleCount = 0.0;          // A variable to store the number of cycles
void setup() {
  // setup code here, to run once:
```

```
Serial.begin(9600);              // Open serial connection, 9600 baud to interface with GUI
unsigned int
configWord;
 // setting up definitions for power and direction
pinMode(SS_M1, OUTPUT); digitalWrite(SS_M1, LOW);  // HIGH = not selected
pinMode(SS_M2, OUTPUT); digitalWrite(SS_M2, LOW);
pinMode(SS_M3, OUTPUT); digitalWrite(SS_M3, LOW);
pinMode(SS_M4, OUTPUT); digitalWrite(SS_M4, LOW);
// L9958 DIRection pins
pinMode(DIR_M1, OUTPUT);
pinMode(DIR_M2, OUTPUT);
pinMode(DIR_M3, OUTPUT);
pinMode(DIR_M4, OUTPUT);
// L9958 PWM pins
pinMode(PWM_M1, OUTPUT);  digitalWrite(PWM_M1, LOW);
pinMode(PWM_M2, OUTPUT);  digitalWrite(PWM_M2, LOW);          // Timer1
pinMode(PWM_M3, OUTPUT);  digitalWrite(PWM_M3, LOW);
pinMode(PWM_M4, OUTPUT);  digitalWrite(PWM_M4, LOW);
// L9958 Enable for all 4 motors
pinMode(ENABLE_MOTORS, OUTPUT);
digitalWrite(ENABLE_MOTORS, HIGH);  // HIGH = disabled
```

```
// set to max current limit and disable ISR slew limiting
configWord = 0b0000010000001100;
SPI.begin();
SPI.setBitOrder(LSBFIRST);
SPI.setDataMode(SPI_MODE1);  // clock pol = low, phase = high
// defines 4 motors and what variables go to them
// Motor 1
digitalWrite(SS_M1, LOW);
SPI.transfer(lowByte(configWord));
SPI.transfer(highByte(configWord));
digitalWrite(SS_M1, HIGH);
// Motor 2
digitalWrite(SS_M2, LOW);
SPI.transfer(lowByte(configWord));
SPI.transfer(highByte(configWord));
digitalWrite(SS_M2, HIGH);
//Motor 3
digitalWrite(SS_M3, LOW);
SPI.transfer(lowByte(configWord));
SPI.transfer(highByte(configWord));
digitalWrite(SS_M3, HIGH);
        //Motor 4
```

```
  digitalWrite(SS_M3, LOW);
  SPI.transfer(lowByte(configWord));
  SPI.transfer(highByte(configWord));
  digitalWrite(SS_M4, HIGH);

  //Set initial actuator settings to pull at 0 speed for safety
  dir1 = 0; dir2 = 0; dir3 = 0; dir4=0; // Set direction
  pwm1 = 0; pwm2 = 0; pwm3 = 0; pwm4=0;// Set speed (0-255)
  digitalWrite(ENABLE_MOTORS, LOW);// LOW = enabled
// IMPORTANT: this is the code that automatically places the motor in the home position when
powered on/booting up the GUI
  dir1=0;
  pwm1=255;
  digitalWrite(DIR_M1, dir1);
  analogWrite(PWM_M1, pwm1); // write to pins
  dir2 = 0;
  pwm2 = 255;
  digitalWrite(DIR_M2, dir2);
  analogWrite(PWM_M2, pwm2);
  dir3 = 0;
  pwm3 = 255;
  digitalWrite(DIR_M3, dir3);
```

```
  analogWrite(PWM_M3, pwm3);
  dir4 = 0;
  pwm4 = 255;
  digitalWrite(DIR_M4, dir4);
  analogWrite(PWM_M4, pwm4);


  delay(5000);
} // End setup
void loop()
  {
  if( Serial.available()>0) {    // following code is for reading what information is was sent from
GUI to arduino
          char ch = Serial.read();          // Store the serial input
          if(ch >= '0' && ch <= '9')       // Check if this an ascii digit between 0 and 9
          {
          // If it is, accumulate the value in the array
          values[fieldIndex] = (values[fieldIndex] * 10) + (ch - '0');
          }
          else if (ch == ',')  // If it's a comma (the chosen separator), move on to the next field
          {
          if(fieldIndex < numberOfFields - 1)
          fieldIndex++;  // Move onto the next field index
```

```
        }
        else
        {
        // Each of the values collected in the array are stored as variables
        int timePerCycleTemp = (int)values[0]; //variable that will be used to generate
timePerCycle value
        int xStrainValue = (int)values[1]; //variable for the velocity of the actuators to strain in
the x direction
        int yStrainValue = (int)values[2]; //variable for the velocity of the actuators to strain in
the y direction
        int selectorvalue = (int)values[3]; //variable that will be used to select case to perform a
specific action, all these value were taken from GUI

        float timePerCycle=(timePerCycleTemp/100); //variable for setting the time per cycle,
        for(int j=0; j <= fieldIndex; j++)
        {
        values[j] = 0; // Clear the array values now that they are stored elsewhere
        }
        fieldIndex = 0;  // Reset the field index - ready to start over
switch (selectorvalue){ //selects which case to run based on selector value
case 1:   //IMPORTANT: code for running strain and stop actions (stop program has power input
values for x and y strain as 0, so actuators don't move during either section of this code)
```

```
while(Serial.available() ==0){
while(i==1){
        //all actuators extend in this step
        dir1 = 1; //sets direction of motor. 0=retract, 1=extend
        pwm1 = xStrainValue; //sets speed of motor. 0 is minimum value, 255 is the maximum value.
        digitalWrite(DIR_M1, dir1);
        analogWrite(PWM_M1, pwm1); // write to pins, tells motors to run at the set speed and direction
        dir2 = 1;
        pwm2 = yStrainValue;
        digitalWrite(DIR_M2, dir2);
        analogWrite(PWM_M2, pwm2);
        dir3 = 1;
        pwm3 = yStrainValue;
        digitalWrite(DIR_M3, dir3);
        analogWrite(PWM_M3, pwm3);
        dir4 = 1;
        pwm4 = xStrainValue;
        digitalWrite(DIR_M4, dir4);
        analogWrite(PWM_M4, pwm4);
        delay(timePerCycle*500); // length of time motors run this step for, 1000=1 second
```

```
        i=i+1;
}
while(i==2){
        //all actuators retract in this step
        dir1 = 0;
        pwm1 = xStrainValue;
        digitalWrite(DIR_M1, dir1);
        analogWrite(PWM_M1, pwm1);
        dir2 = 0;
        pwm2 = yStrainValue;
        digitalWrite(DIR_M2, dir2);
        analogWrite(PWM_M2, pwm2);
        dir3 = 0;
        pwm3 = yStrainValue;
        digitalWrite(DIR_M3, dir3);
        analogWrite(PWM_M3, pwm3);
        dir4 = 0;
        pwm4 = xStrainValue;
        digitalWrite(DIR_M4, dir4);
        analogWrite(PWM_M4, pwm4);
        delay(timePerCycle*500);
        i=i-1;
```

```
        cycleCount = cycleCount+1;
        Serial.println(cycleCount); //send number of cycles to GUI to be displayed
    }
}
break;
case 2:  //IMPORTANT: code for running home and stop action
        // all actuators retract at max speed until they are fully retracted
        dir1 = 0;
        pwm1 = xStrainValue; //set direction and speed
        digitalWrite(DIR_M1, dir1);
        analogWrite(PWM_M1, pwm1); // write to pins
        dir2 = 0;
        pwm2 = yStrainValue;
        digitalWrite(DIR_M2, dir2);
        analogWrite(PWM_M2, pwm2);
        dir3 = 0;
        pwm3 = yStrainValue;
        digitalWrite(DIR_M3, dir3);
        analogWrite(PWM_M3, pwm3);
        dir4 = 0;
        pwm4 = xStrainValue;
        digitalWrite(DIR_M4, dir4);
```

```
        analogWrite(PWM_M4, pwm4);
        delay(timePerCycle*500);  // length of time motors run this step for
break;
case 3:  //IMPORTANT: code for running alt strain action
while(Serial.available() ==0){
while(i==1){
  // actuators in x direction extend
        dir1 = 1;
        pwm1 = xStrainValue; //set direction and speed
        digitalWrite(DIR_M1, dir1);
        analogWrite(PWM_M1, pwm1); // write to pins
        dir4 = 1;
        pwm4 = xStrainValue;
        digitalWrite(DIR_M4, dir4);
        analogWrite(PWM_M4, pwm4);
        dir2 = 1;
        pwm2 = 0;
        digitalWrite(DIR_M2, dir2);
        analogWrite(PWM_M2, pwm2);
        dir3 = 1;
        pwm3 = 0;
        digitalWrite(DIR_M3, dir3);
```

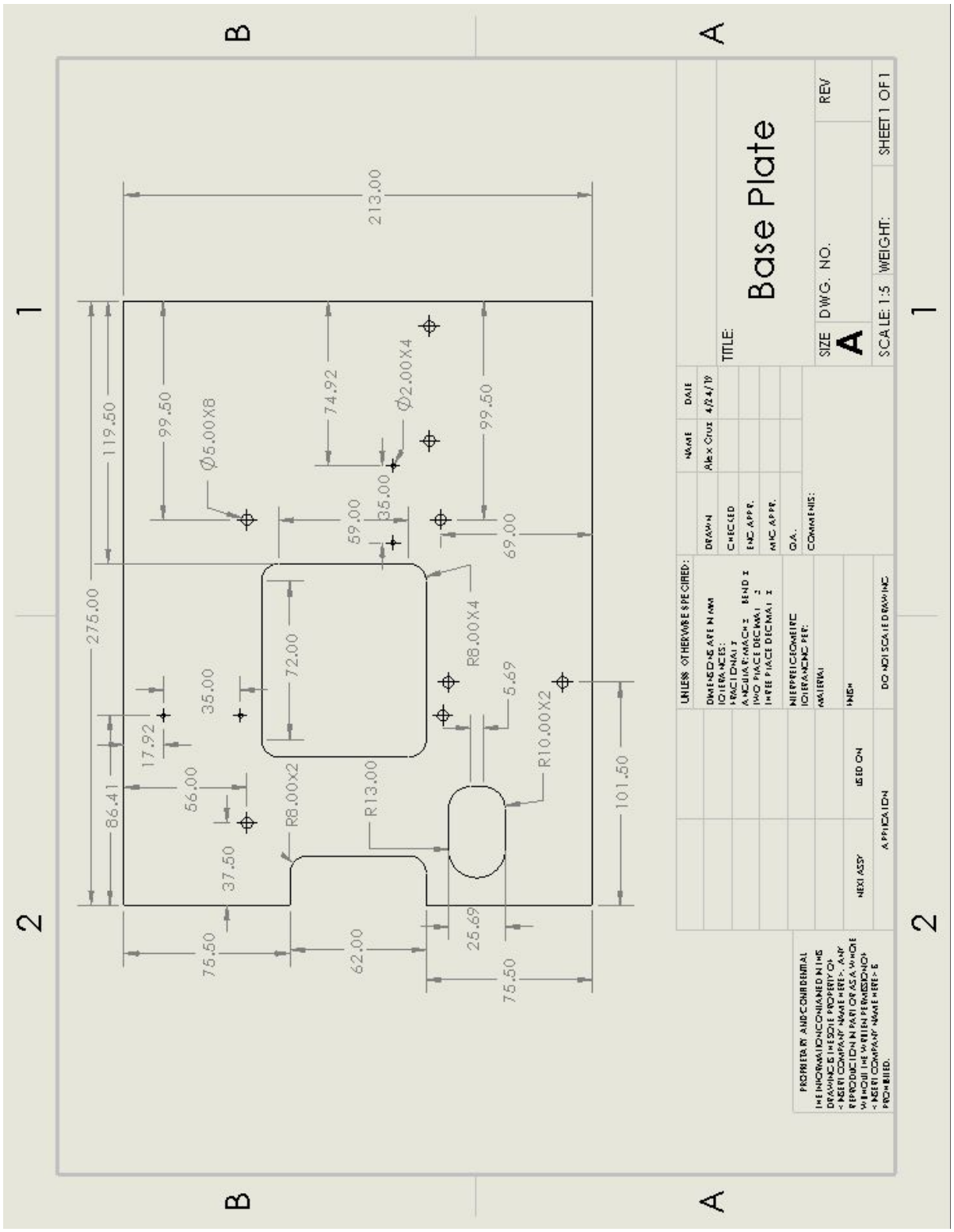
```
        analogWrite(PWM_M3, pwm3);
        delay(timePerCycle*500);  // length of time until motors run this step for
        i=i+1;
}
while (i==2){
  // actuators in x direction retract
        dir1 = 0;
        pwm1 = xStrainValue; //set direction and speed
        digitalWrite(DIR_M1, dir1);
        analogWrite(PWM_M1, pwm1); // write to pins
        dir4 = 0;
        pwm4 = xStrainValue;
        digitalWrite(DIR_M4, dir4);
        analogWrite(PWM_M4, pwm4);

        dir2 = 1;
        pwm2 = 0;
        digitalWrite(DIR_M2, dir2);
        analogWrite(PWM_M2, pwm2);

        dir3 = 1;
        pwm3 = 0;
```

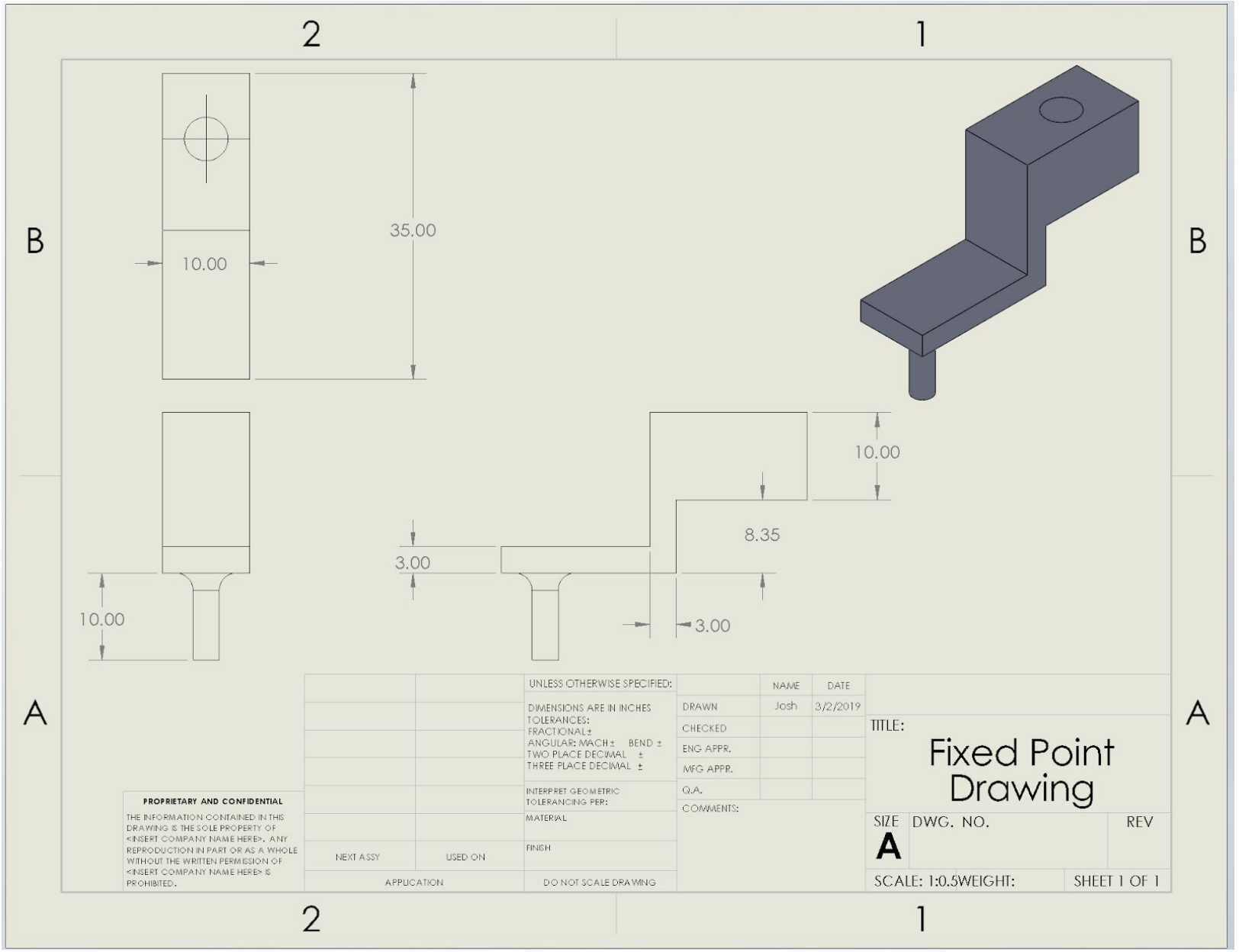
```
        digitalWrite(DIR_M3, dir3);
        analogWrite(PWM_M3, pwm3);
        delay(timePerCycle*500);  // length of time until motors run this step for
        i=i+1;
}

while(i==3){
// actuators in y direction extend
        dir1 = 0;
        pwm1 = 0;
        digitalWrite(DIR_M1, dir1);
        analogWrite(PWM_M1, pwm1);
        dir4 = 0;
        pwm4 = 0;
        digitalWrite(DIR_M4, dir4);
        analogWrite(PWM_M4, pwm4);
        dir2 = 1;
        pwm2 = yStrainValue;
        digitalWrite(DIR_M2, dir2);
        analogWrite(PWM_M2, pwm2);
        dir3 = 1;
        pwm3 = yStrainValue;
```

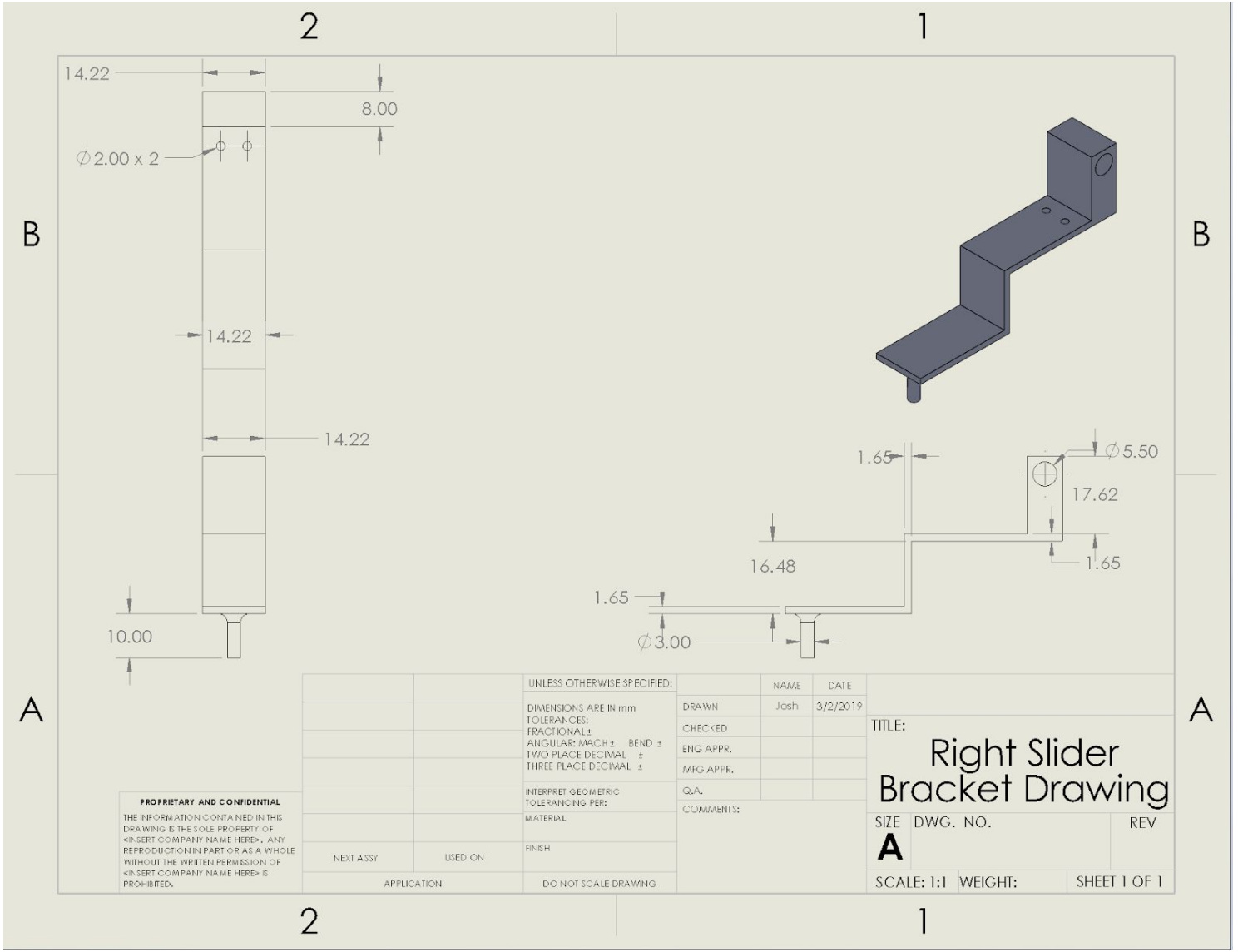
```
        digitalWrite(DIR_M3, dir3);
        analogWrite(PWM_M3, pwm3);
        delay(timePerCycle*500);  // length of time until motors run this step for
        i=i+1;
}

while(i==4){
// actuators in y direction retract
        dir1 = 0;
        pwm1 = 0;
        digitalWrite(DIR_M1, dir1);
        analogWrite(PWM_M1, pwm1);
        dir4 = 0;
        pwm4 = 0;
        digitalWrite(DIR_M4, dir4);
        analogWrite(PWM_M4, pwm4);

        dir2 = 0;
        pwm2 = yStrainValue;
        digitalWrite(DIR_M2, dir2);
        analogWrite(PWM_M2, pwm2);
        dir3 = 0;
```

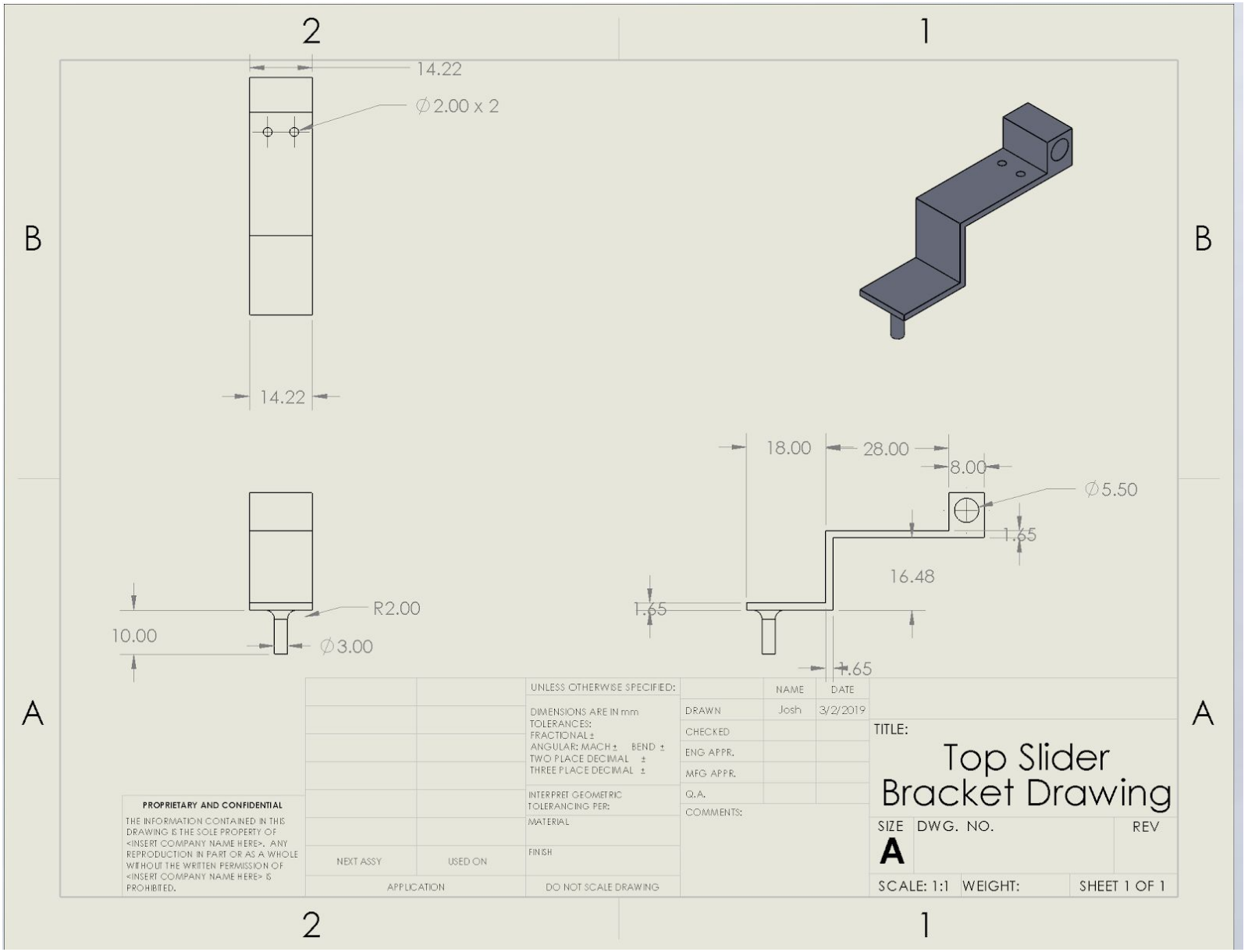
```
        pwm3 = yStrainValue;
        digitalWrite(DIR_M3, dir3);
        analogWrite(PWM_M3, pwm3);
        delay(timePerCycle*500);
        i=i-3;
        cycleCount = cycleCount+1;
        Serial.println(cycleCount); //send number of cycles to GUI to be displayed
   }
}
break;
}
  }
  }
  }
```

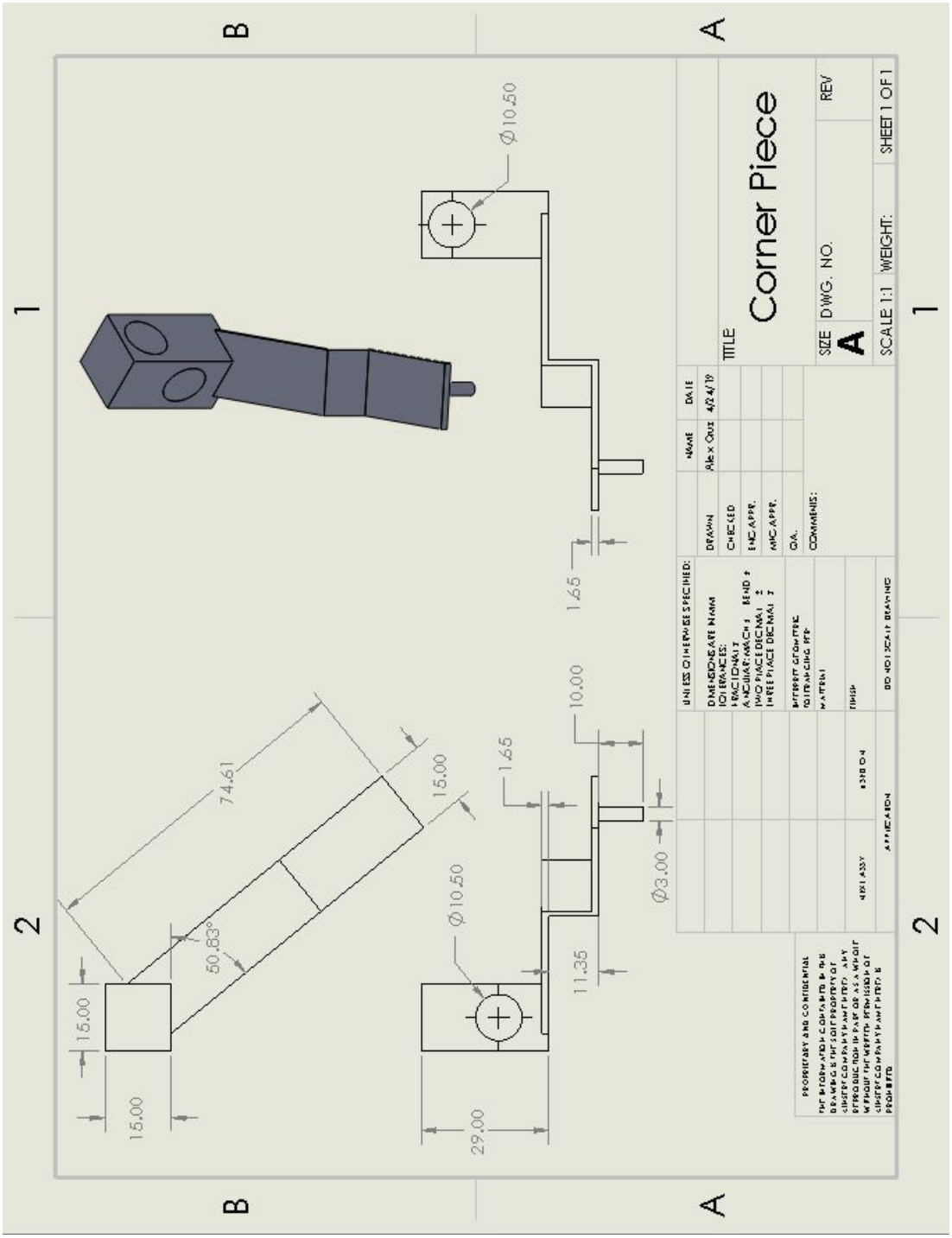
Appendix H: CAD Drawings of Parts Used in Final Device

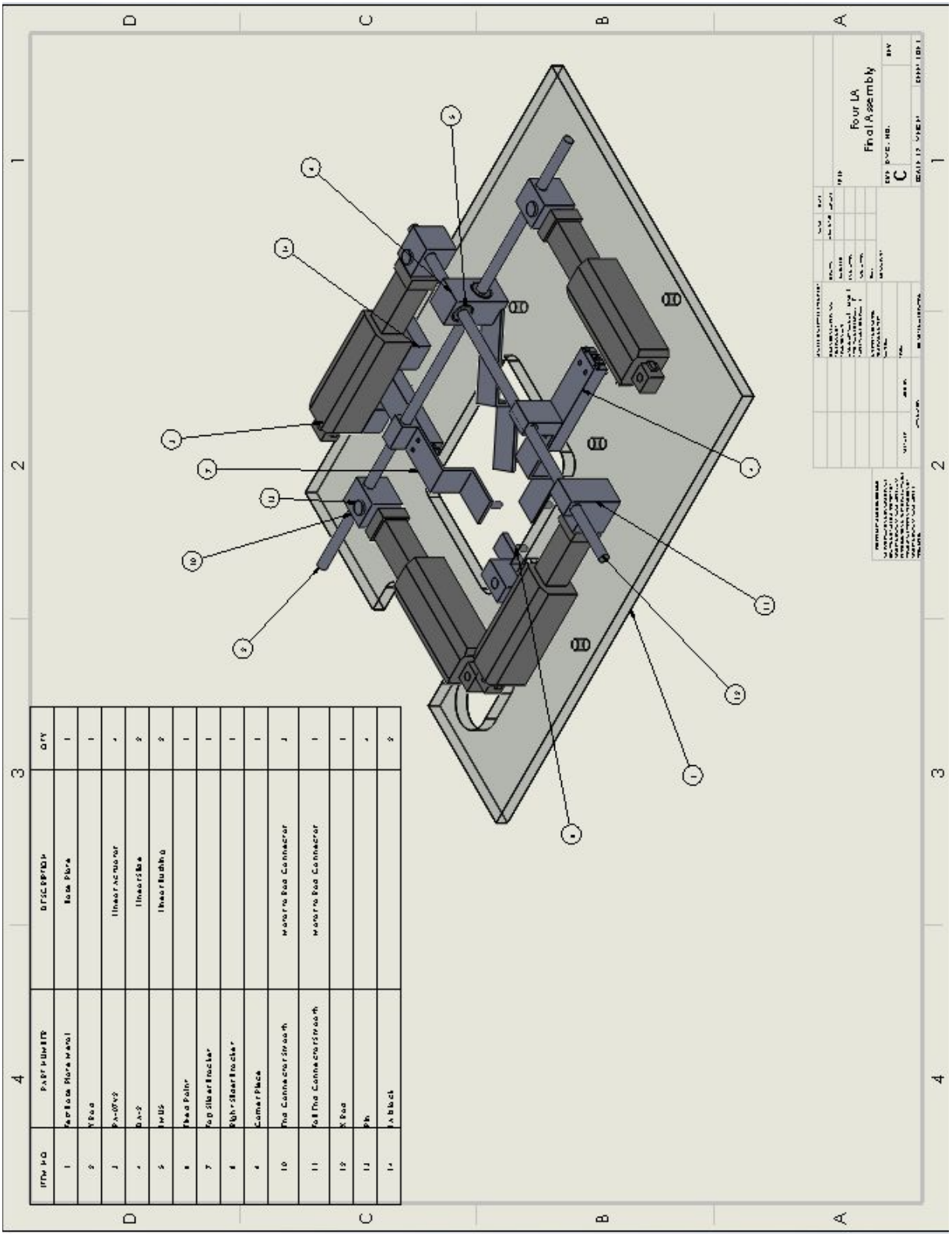



2
1
B
A
35.00
10.00
10.00
8.35
3.00
10.00
3.00
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN INCHES
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±
INTERPRET GEOMETRIC TOLERANCING PER:
MATERIAL
FINISH
NEXT ASSY
USED ON
APPLICATION
DO NOT SCALE DRAWING
NAME
DATE
DRAWN
Josh
3/2/2019
CHECKED
ENG APPR.
MFG APPR.
Q.A.
COMMENTS:
PROPRIETARY AND CONFIDENTIAL
THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF <INSERT COMPANY NAME HERE>. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF <INSERT COMPANY NAME HERE> IS PROHIBITED.
TITLE:
Fixed Point Drawing
SIZE
A
DWG. NO.
REV
SCALE: 1:0.5
WEIGHT:
SHEET 1 OF 1


2
1
B
A
14.22
8.00
∅2.00 x 2
14.22
14.22
10.00
1.65
∅5.50
17.62
16.48
1.65
1.65
∅3.00
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN mm
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±
INTERPRET GEOMETRIC TOLERANCING PER:
MATERIAL
FINISH
NEXT ASSY
USED ON
APPLICATION
DO NOT SCALE DRAWING
NAME
DATE
DRAWN
Josh
3/2/2019
CHECKED
ENG APPR.
MFG APPR.
Q.A.
COMMENTS:
TITLE:
Right Slider Bracket Drawing
SIZE
A
DWG. NO.
REV
SCALE: 1:1
WEIGHT:
SHEET 1 OF 1
PROPRIETARY AND CONFIDENTIAL
THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF <INSERT COMPANY NAME HERE>. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF <INSERT COMPANY NAME HERE> IS PROHIBITED.


2
1
B
A
14.22
∅2.00 x 2
14.22
18.00
28.00
8.00
∅5.50
1.65
16.48
1.65
1.65
R2.00
10.00
∅3.00
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN mm
TOLERANCES:
FRACTIONAL ±
ANGULAR: MACH ± BEND ±
TWO PLACE DECIMAL ±
THREE PLACE DECIMAL ±
INTERPRET GEOMETRIC TOLERANCING PER:
MATERIAL
FINISH
NEXT ASSY
USED ON
APPLICATION
DO NOT SCALE DRAWING
NAME
DATE
DRAWN
Josh
3/2/2019
CHECKED
ENG APPR.
MFG APPR.
Q.A.
COMMENTS:
PROPRIETARY AND CONFIDENTIAL
THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF <INSERT COMPANY NAME HERE>. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF <INSERT COMPANY NAME HERE> IS PROHIBITED.
TITLE:
Top Slider Bracket Drawing
SIZE
A
DWG. NO.
REV
SCALE: 1:1
WEIGHT:
SHEET 1 OF 1


Corner Piece
TITLE
SIZE
A
DWG. NO.
REV
SCALE 1:1
WEIGHT:
SHEET 1 OF 1
UNLESS OTHERWISE SPECIFIED:
DRAWN
CHECKED
ENG APPR.
MFG APPR.
Q.A.
COMMENTS:
NAME
DATE
Alex Cruz
4/24/19
74.61
50.83°
15.00
15.00
15.00
10.00
1.65
1.65
11.35
29.00
Ø10.50
Ø10.50
Ø3.00



ITEM NO.	PART NUMBER	DESCRIPTION	QTY.
1	Surface Plate Material	Base Plate	1
2	Y Rod		1
3	PA-07V2	Linear Actuator	4
4	PA-2	Linear Rail	8
5	Inline	Linear Bushing	8
6	Rod Collar		1
7	Top Bar Bracket		1
8	Right Bar Bracket		1
9	Corner Plate		1
10	Pin Connector Smooth	Motor to Rod Connector	4
11	Roll Pin Connector Smooth	Motor to Rod Connector	1
12	X Rod		1
13	Pin		4
14	Pin Block		8

Appendix I: Pre-Calculated Strain to Power Values for Specific Frequencies

Table I1: Reference Chart for Different Frequencies for Pre-Calculated Power Values Based on Strain Input from User

1 Hz		0.5 Hz		0.1Hz	
Strain [%]	Power to Speed [0-255]	Strain [%]	Power to Speed [0-255]	Strain [%]	Power to Speed [0-255]
1.0	18.7	1.0	9.36	1.0	1.87
2.0	37.4	2.0	18.7	2.0	3.74
3.0	56.2	3.0	28.1	3.0	5.62
4.0	74.9	4.0	37.4	4.0	7.49
5.0	93.6	5.0	46.8	5.0	9.36
6.0	112	6.0	56.2	6.0	11.2
6.5	122	6.5	60.8	6.5	12.2
6.7	125	6.7	62.7	6.7	12.5
6.8	127	6.8	63.6	6.8	12.7
6.9	129	6.9	64.6	6.9	12.9
7.0	131	7.0	65.5	7.0	13.1
8.0	150	8.0	74.9	8.0	15.0
9.0	168	9.0	84.2	9.0	16.8
10.0	187	10.0	93.6	10.0	18.7
11.0	206	11.0	103	11.0	20.6
12.0	225	12.0	112	12.0	22.5
13.0	243	13.0	122	13.0	24.3
13.5	253	13.5	126	13.5	25.3
13.6	**255**	13.6	127	13.6	25.5
14.0	262	14.0	131	14.0	26.2
15.0	281	15.0	140	15.0	28.1
16.0	299	16.0	150	16.0	29.9
17.0	318	17.0	159	17.0	31.8

18.0	337	18.0	168	18.0	33.7
19.0	356	19.0	178	19.0	35.6
20.0	374	**20.0**	**187**	**20.0**	**37.4**

Table I2: Reference Chart Continued for Different Frequencies for Pre-Calculated Power Values Based on Strain Input from User

2Hz		**1.5 Hz**		**1.3 Hz**	
Strain [%]	**Power to Speed [0-255]**	**Strain [%]**	**Power to Speed [0-255]**	**Strain [%]**	**Power to Speed [0-255]**
1.0	37.4	1.0	28.1	1.0	24.3
2.0	74.9	2.0	56.2	2.0	48.7
3.0	112	3.0	84.2	3.0	73.0
4.0	150	4.0	112	4.0	97.3
5.0	187	5.0	140	5.0	122
6.0	225	6.0	168	6.0	146
6.5	243	6.5	182	6.5	158
6.7	251	6.7	188	6.7	163
6.8	**255**	6.8	191	6.8	165
6.9	258	6.9	194	6.9	168
7.0	262	7.0	197	7.0	170
8.0	299	8.0	225	8.0	195
9.0	337	9.0	253	9.0	219
10.0	374	**9.1**	**255**	9.1	221
11.0	412	9.2	258	9.2	224
12.0	449	9.4	264	9.4	229
13.0	487	9.5	267	9.5	231
13.5	505	9.9	278	9.9	241
13.6	509	10.0	281	10.0	243
14.0	524	11.0	309	10.3	251
15.0	562	12.0	337	10.4	253

16.0	599	13.0	365	**10.5**	**255**
17.0	636	13.5	379	11.0	268
18.0	674	13.6	382	12.0	292
19.0	711	14.0	393	13.0	316
20.0	749	15.0	421	13.5	328
		16.0	449	13.6	331
		17.0	477	14.0	341
		18.0	505	15.0	365
		19.0	533	16.0	389
		20.0	562	17.0	414
				18.0	438
				19.0	462
				20.0	487

Appendix J: Final Device Budget and Part Specification Sheets

Table J1: Final Device Budget

Number of Parts and Part Description	Part ID #	Website	Cost (Total for Multiple Parts)
1 Arduino Due	A000062	https://www.amazon.com/gp/product/B00A6C3JN2/ref=ppx_yo_dt_b_asin_title_o09_s00?ie=UTF8&psc=1	$39.44
1 MultiMoto Arduino Shield	LC-82	https://www.progressiveautomations.com/products/lc-82	$48.99
4 Micro Linear Actuators	PA-07	https://www.progressiveautomations.com/products/micro-linear-actuator	$279.96
2 Linear Slides	DA-02	https://www.deltron.com/search/Slides_Ball_Crossed_Roller_Model.aspx?pkid=3358	$162.00
2 Connector Rods	B07KJ8DGF3	https://www.amazon.com/gp/product/B07KJ8DGF3/ref=ppx_yo_dt_b_asin_title_o02_s00?ie=UTF8&psc=1	$10.00

Spool of PLA for 3D prints	BRK-07	https://www.amazon.com/AIO-Robotics-AIOORANGE-Filament-Dimensional/dp/B01HYYPIVA/ref=pd_yo_rr_rp_3/131-6632370-5670961?_encoding=UTF8&pd_rd_i=B01HYYPIVA&pd_rd_r=2f02efe9-a883-444d-90ab-f7b55b0b98ed&pd_rd_w=PUMWm&pd_rd_wg=C9352&pf_rd_p=8d661bcd-e088-478d-a296-0f0c547bdf7e&pf_rd_r=VXBCH6G2DKWPMKZXPV04&psc=1&refRID=VXBCH6G2DKWPMKZXPV04	$13.99
2 Bushings	LMU-5	https://us.misumi-ec.com/	$24.00
1 12V DC Power Supply	AC-15	https://www.progressiveautomations.com/products/ac-15	$84.00
0.25" Aluminum Sheet (12" x12")	7902	https://www.onlinemetals.com/en/buy/aluminum/aluminum-sheet-6061-t4/pid/7902	$35.47
Various Other Hardware (details below)			$30.00
i Excell M2 Screws	M2-041253-12.9-D	https://www.amazon.com/gp/product/B07197MQM4/ref=ppx_yo_dt_b_asin_title_o01_s00?ie=UTF8&psc=1&fbclid=IwAR34VqiE6OWNG8Nxk2pnuUcxAJ_iCOGWqL6779UVt41pqsUzkMNY_hlxYVY	
Halex 3/4" Bracket/	96162	Home Depot	
M5-0.8 Screws - Everbilt	80328	Home Depot	
Total Cost			$727.85

PA-07 Specifications

Input Voltage:	12 VDC
Stroke:	0.5-12 inches
Force:	5 lbs
Speed:	0.59"/sec
Protection Class:	IP66
Feedback:	No
Operational Temperature:	0°C~50°C
Noise:	db<45(A)
Duty Cycle:	10%

Limit Switch:	Built In, Non-Adjustable
Current (full load):	0.24A
Mounting Holes:	0.16"
Motor Type:	Brushed DC Motor
Screw Type:	ACME
Housing Type:	Plastic and Aluminum Alloy
Fully Retracted:	3.81" + stroke, 3.18" for stroke 2" and under
Fully Extended:	3.81" + stroke + stroke, 3.18" for stroke 2" and under
Certifications:	CE and RoHS
Unit Weight:	0.15 lbs
Warranty:	18 Months

Figure J1: PA-07 Micro-Linear Actuator: Specifications

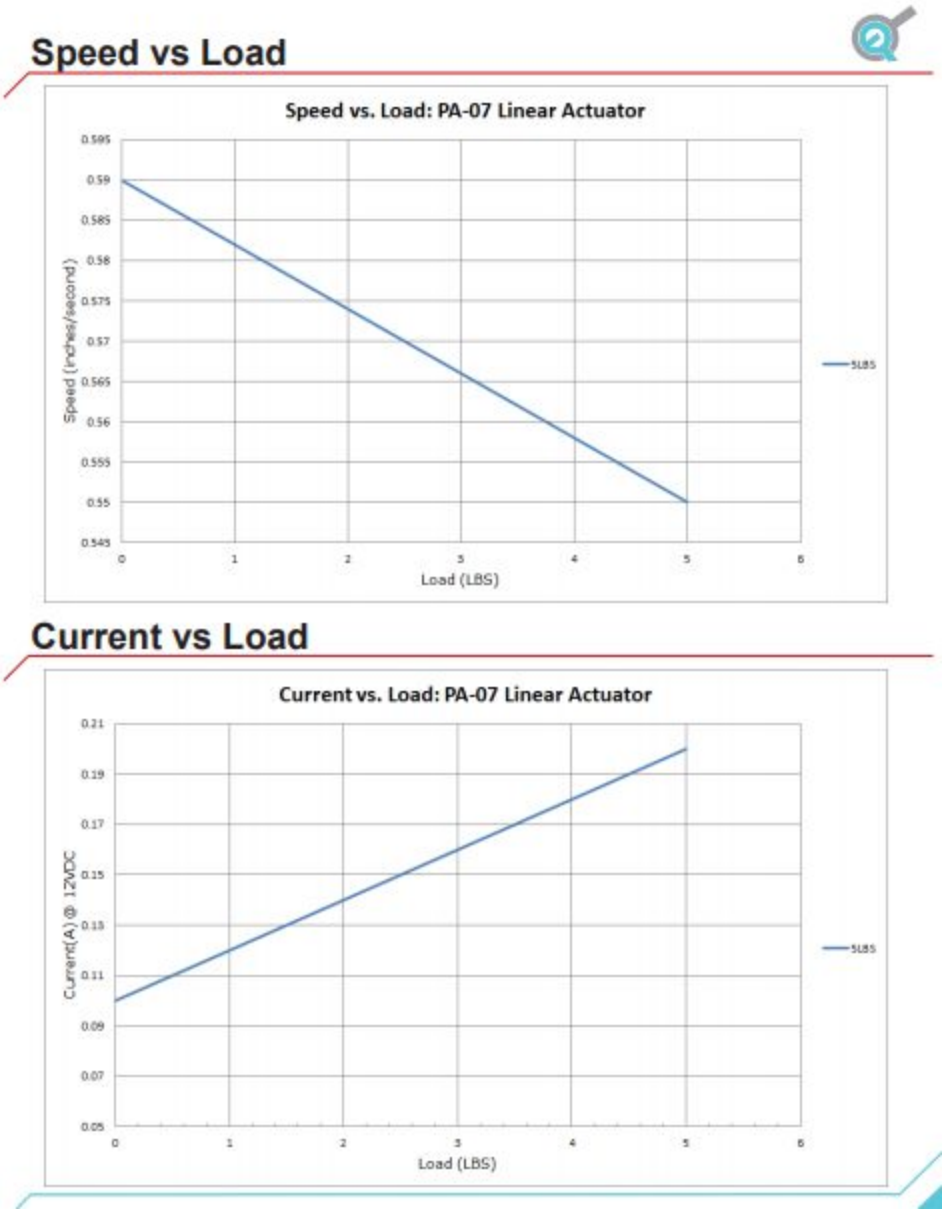


Figure J2: PA-07 Micro-Linear Actuator: Velocity and Current Profiles (top) Shows the max velocity of 0.59”/s at 0 lbs and min velocity of 0.55”/s at 5 lbs; (bottom) Shows maximum current of 0.20A at 5 lbs

LC-82 Specifications

Input Voltage:	6-36V	Size:	2.7" x 1.0" x 2.1"
Operating Current:	6.5A continuous (8A peak) on each channel	Unit Weight:	0.05 lbs

Figure J3: MultiMoto Arduino Shield Specifications

Material Meets These Standard(s):
AMS4026,
ASTMB209,
Weight/Lineal Foot: 1.77

Dimension Name	Value
Thickness	0.125
Max Width	48
Max Length	144
Alloy	6061
Temper	T4
Covering	Pvc 1 Side
MTR Availability	false
Shape	Sheet
Material	Aluminum
Custom Cut Warehouse	1

Figure J4: Aluminum Sheet Specifications

AC-15 Specifications

Input Voltage:	110-240 VAC	Unit Weight:	1.6 lbs
Output Voltage:	12 VDC	Certifications:	UL
Current Rating:	10A	Warranty:	18 Months

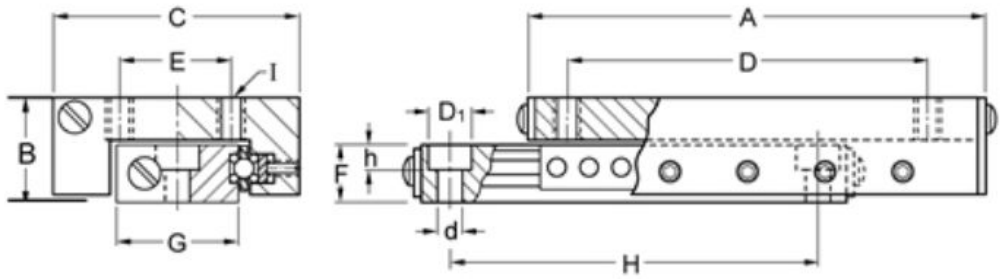
Figure J5: 12V DC Power Supply Specifications

Microcontroller	AT91SAM3X8E
Operating Voltage	3.3V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-16V
Digital I/O Pins	54 (of which 12 provide PWM output)
Analog Input Pins	12
Analog Output Pins	2 (DAC)
Total DC Output Current on all I/O lines	130 mA
DC Current for 3.3V Pin	800 mA
DC Current for 5V Pin	800 mA
Flash Memory	512 KB all available for the user applications
SRAM	96 KB (two banks: 64KB and 32KB)
Clock Speed	84 MHz
Length	101.52 mm
Width	53.3 mm
Weight	36 g

Figure J6: Arduino Due Specifications

A

Carriage 4 Holes (I)	M2 Thread
Base Hole d	2.2
Base Hole D_1	4.0
Base Hole h	2.2
Counter Bore Screw Size	M2



B.

All dimensions in millimeters unless otherwise specified													
								Carriage Hole Spacing		Base Dimensions			
Model	Price $	Travel	Load (kgs)	Weight (gr)	Length A	Height B	Width C	D	E	F	G	H	Accuracy (mm/25mm of travel)
DA-2	81.00	25	4	14	52.0	8.0	14.2	41.0	6.0	4.7	6.4	35.0	.013

Figure J7: (A) DA-2 Linear Slide Dimensional Diagrams (B) Dimension Specifications

Appendix K: Arduino/MultiMoto Wiring Diagram

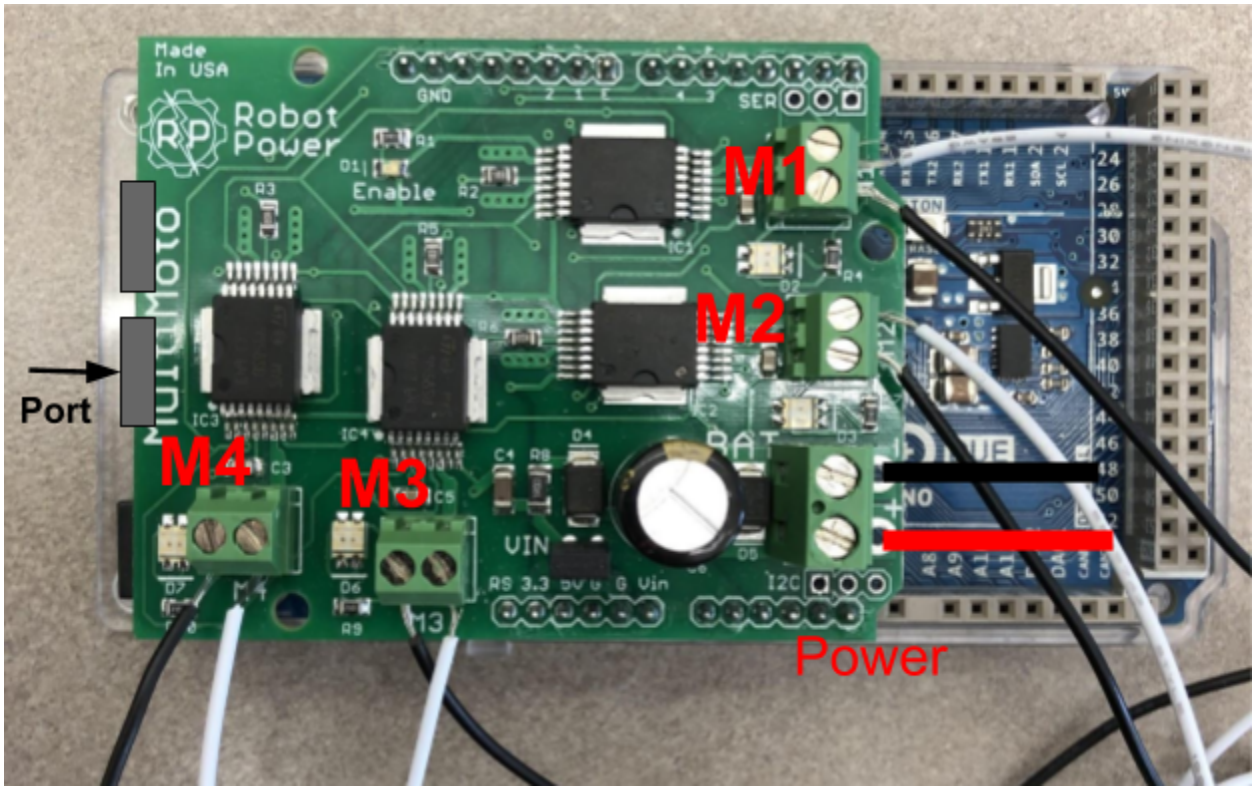


Figure K1: Wiring diagram of Arduino and MultiMoto

Figure K1 above shows the Arduino/MultiMoto wired to control the device. The wire ports for the device, M1 through M4, are labeled in red. The white wire from the motor goes into the right side of the chip input. The black wire from the motor goes into the left side of the input. The red cable from the AC/DC adaptor goes into the left, positive (+) port labeled on the chip. The black cable goes into the right, negative (-) port labeled on the chip. There are two ports for the USB cable to connect to from the computer. To run the device, the USB must go into the programming port, indicated by the labeled arrow.