6-13-2017

Beta: Bioprinting engineering technology for academia

Max Abrams
Santa Clara Univeristy, mabrams@scu.edu

Jeffrey Barone
Santa Clara University, jbarone@scu.edu

Cynthia Le
Santa Clara University, cwle@scu.edu

Jacob Ososke
Santa Clara University, jososke@scu.edu

Franz Plum
Santa Clara University, fplum@scu.edu

See next page for additional authors

Author
Max Abrams, Jeffrey Barone, Cynthia Le, Jacob Ososke, Franz Plum, Emily Takimoto, and Josie Warren

SANTA CLARA UNIVERSITY

Department of Mechanical Engineering
Department of Computer Engineering

I HEREBY RECOMMEND THAT THE THESIS PREPARED

UNDER MY SUPERVISION BY

Max Abrams, Jeffrey Barone, Cynthia Le, Jacob Ososke,

Franz Plum, Emily Takimoto, and Josie Warren

ENTITLED

BETA: BIOPRINTING ENGINEERING TECHNOLOGY FOR ACADEMIA

BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREES OF

BACHELOR OF SCIENCE

IN

MECHANICAL ENGINEERING
COMPUTER SCIENCE AND ENGINEERING

Thesis Advisor — date 6/13/17

Department Chair Mechanical Engineering — date 6/14/2017

Department Chair Computer Engineering — date 6/13/17

BETA: BIOPRINTING ENGINEERING TECHNOLOGY FOR ACADEMIA

By

Max Abrams, Jeffrey Barone, Cynthia Le, Jacob Ososke,

Franz Plum, Emily Takimoto, and Josie Warren

SENIOR DESIGN PROJECT REPORT

Submitted to
the Department of Mechanical Engineering
the Department of Computer Engineering

of

SANTA CLARA UNIVERSITY

in Partial Fulfillment of the Requirements
for the degree of
Bachelor of Science in Mechanical Engineering
Bachelor of Science in Computer Science and Engineering

Santa Clara, California

2017

BETA: Bioprinting Engineering Technology for Academia

Max Abrams, Jeffrey Barone, Cynthia Le, Jacob Ososke,
Franz Plum, Emily Takimoto, and Josie Warren

Department of Mechanical Engineering
Department of Computer Engineering
Santa Clara University
2017

ABSTRACT

Higher STEM education is a field of growing potential, but too many middle school and high school students are not testing proficiently in STEM subjects. The BETA team worked to improve biology classroom engagement through the development of technologies for high school biology experiments. The BETA project team expanded functionality of an existing product line to allow for better student and teacher user experience and the execution of more interesting experiments. The BETA project's first goal was to create a modular incubating Box for the high school classroom. This Box, called the BETA Box was designed with a variety of sensors to allow for custom temperature and lighting environments for each experiment. It was completed with a clear interface to control the settings and an automatic image capture system. The team also conducted a feasibility study on auto calibration and dual-extrusion for SE3D's existing 3D bioprinter. The findings of this study led to the incorporation of a force sensor for auto calibration and the evidence to support the feasibility of dual extrusion, although further work is needed. These additions to the current SE3D educational product line will increase effectiveness in the classroom and allow the target audience, high school students, to better engage in STEM education activities.

Keywords: STEM, Education, Incubation, Thermal Control, User Interface, 3D Bioprinting

Acknowledgements

Special thanks to:

Dr. Chris Kitts
Greg Richmond
Greg Barsky
Chris Paetsch
Sam Varney
Dr. Walter Yuen
Don Maccubbin
Kristen Ronhovde

Table of Contents

List of Figures

List of Tables

Part I. Introduction to the System

1. Introduction

A major challenge that many STEM educators face is getting students interested and engaged in the subject while in the classroom. Currently, middle school and high school students are not testing proficiently in STEM subjects [1]. An educational-technology startup, SE3D, hopes to solve this challenge by bringing 3D bioprinting technology to high school classrooms. Although SE3D has already produced a working prototype printer, it only has basic capabilities and requires further development.

The BETA Senior Design Project sought to expand functionality in SE3D's product line to allow for a better student and teacher experience and more interesting experiments. This involved the creation of an entirely new product, the BETA Box, which incubates, lights and photographs experiments while they culture. The project also included the study of further technologies to be used on SE3D's existing printer, specifically auto-calibration and dual extrusion.

1.1 Motivation: Customer Needs

The BETA team worked to meet the needs of both the direct and indirect customer, SE3D and high school teachers. To achieve this, both customers were interviewed and the responses of both interviews were consolidated to determine the prioritized goals of the project. Below are the interview questions, followed by the mentioned summary.

1.1.1 Customer Demographics And Interview Questions

Since the team worked on the 3D bioprinter for an outside client, it is important to recognize the needs of the company as well as the needs of their customers. The following interview questions were prepared to give the team a better understanding of the customers' needs.

1.1.1.1 Direct Customer: SE3D

The team met directly with the primary client of the project, SE3D. One contact was Kevin Kozel. As a Santa Clara University alumni, Kevin worked directly with the printer during its development and is very aware of its current capabilities. Additionally, the team talked with Mayasari Lim, the CEO and co-founder of the company. Maya led the primary vision for the printer, and was interested in improving the product for her customers: high school teachers and students. The following questions were given to the SE3D employees:

1. Can you provide some background on the printer?
2. What are the current features of your bioprinter?
3. What would you like to add to the printer?
4. Which of these wants are priorities?
5. How do you envision these additions?

Kevin’s and Maya’s full responses to these questions can be found in the meeting minutes in Appendix M.

1.1.1.2 Indirect Customer: High School Teachers and Students

The team also reached out to potential indirect customers. These included Kevin Flack and Judy Schoonmaker. Kevin is a ninth grade biology teacher at a well-funded, academically accelerated public high school. Judy is currently a professor for a university level biology course. Prior to this, she taught Advanced Placement Biology at a public school. Both of these contacts represent prospective customers with much insight into how a 3D bioprinter and other additional technology could fit into a high school curriculum. They were both asked the following questions:

1. SE3D is developing simplified 3D bioprinters for running standard high school biology experiments. Does this seem like an interesting addition to your high school curriculum?
2. Are there any setbacks you have when running an experiment with students related to efficiency, errors, or restrictions on experiments? If so, what are they?
3. The current selling price for SE3D’s printer is $3,300. What is your reaction to this price for the product?
4. What considerations do you make when investing in new equipment for your class?
5. When you think of a 3D bioprinter for your classroom, what would you envision for your students? What abilities would this printer have?

The responses to these questions were combined with SE3D’s responses to determine what both customers would like to see in the development of the product line.

1.1.2 Customer Needs Based on Raw Responses

1.1.2.1 Direct Customer Needs

After the SE3D employees were given the interview question previously stated, their responses were analyzed. Table 1 displays the initial customer needs identified by the company.

Table 1.1: Direct Customer Raw Needs

Need	Current State	Potential Additions
Multiple Material Extruder	• Standard syringes from Amazon • One 5 mL syringe	• Dual/triple extruder • Additional material capability
Auto-Calibration	• Manual calibration • Air bubbles issue	• Create system that automatically calibrates based on material
Experiment Tracking	• Manual image capturing • No thermal or humidity control in external light Box • Use of ImageJ software for color quality tracking	• Automated analysis • Control system for incubating experiments • More accurate color quantifying software
Simplified Software Interface	• No significantly useful user interface • Still some excessive programming required	• LCD screen • Possibly compatible with RepRap controller or application to control all modular equipment

1.1.2.2 Indirect Customer Needs

Using the responses received from Kevin Flack and Judy Schoonmaker, the indirect customer needs assessment was created. The following list indicates criteria of high school teachers in the classroom:

1. Students receive as much direct hands-on experience as possible
2. Product has longevity
3. Human errors are not detrimental, but contribute to the learning process
4. Current price of $3,300 is reasonable for technology, but expensive for budget
5. New technology excites students, will bring new excitement to classroom
6. Consider new experiments that could reach a collegiate classroom

1.1.3 List of Needs

Combining the survey responses from both the direct and indirect clients, the team prioritized the features that could be incorporated into the scope of the Senior Design Project. These features are listed in order of importance in Table 1.2.

Table 1.2: List of Needs from Highest Priority to Lowest

Rank	Need	Solution
1	LCD Interface	LCD screen that is compatible with current RepRap controller
2	Color Quantifying Software	Switch from gray scale to more accurate RGB scale
3	Image Capturing Software	Utilize phone camera through app or institute permanent camera
4	Software	Make compatible with Chromebook and LCD panel
5	Unified Control System	System that allows control of all units together ▪ Incubator ▪ Printer ▪ Thermal control
6	Incubator	▪ Creation of separate incubator ▪ Need more teacher/student feedback ▪ Focus on lighting, thermal, and humidity control for experiments
7	Multi-filament Extruder	▪ Multiple syringes fixed on extruder head ▪ Ability to print multiple materials at once ▪ Potentially ability to print plastic ▪ Auto calibration

Currently, SE3D's main objectives are to improve the educational capabilities and ease of use for the 3D Bioprinter. For this reason, the top priorities of the project focus on enhancing these qualities. Table 1.3 identifies the most important features that the team recognized and included in the scope of the project.

SE3D expressed these primary needs after receiving feedback from current customers. These needs were the highest primary needs for the project, and we prioritized when considered alongside the other criteria listed in Table 1.2.

Table 1.3: Prioritized Project Needs

Project Need	Benefit to User
Low Cost	Ability to be purchased using high school biology budget
LCD screen interface	Eliminates need for external computer screen
Updated image capture software	Ability to provide more accurate and meaningful data
Auto-calibration system for printer syringe	More accurate setup and reduced setup time

1.2 Literature Review

3D printing and 3D bioprinting are fields that have expanded greatly in recent years, and there is abundant research and developments published on all aspects of the technology. Santa Clara University senior design teams have been working with SE3D for several years, iterating upon their product designs to best serve their customers in education. However, incubation for educational purposes is a less densely researched field. The following literature review denotes the key pieces of available information used for understanding low-cost incubation methods and dual-extrusion in 3D printers.

1.2.1 Pertinent Incubation Literature

Yordanova's Low Cost Temperature Controllers

Yordanova used micro-controllers to develop a low-cost temperature control unit [2]. His research illustrates how the temperature in a heat exchanger can be controlled with a micro-controller that controls the amount of radiation given off by the unit. Since our project focuses on the affordability of the Bioprinter for high schools, this method for temperature control demonstrates a way to keep the price low while still achieving accurate temperature control. This experimental procedure was studied to determine how best to control the air surrounding the heat source, such that the entire incubation chamber temperature can be within a few degrees.

Differences

Yordanova's project went into too great of detail for the BETA's incubation requirements, and met more precise specifications than required for this project, at a higher economic cost. The BETA incubator met an even lower cost point, while sacrificing precision in temperature readings.

1.2.2 Pertinent 3D Bioprinter Literature

1.2.2.1 Gao's Coaxial Nozzle Method

The 3D printer developed by Gao [3] experimented with different flow rates and material concentrations to determine the optimum high strength printable biological material. The printer used an interchangeable coaxial nozzle to print the hollow cavity of the filament. This allowed the experimenters to control the inside diameter without significantly altering the outside diameter.

Differences

SE3D's bioprinter uses a single nozzle in the form of a syringe, which has made it difficult to print complex physical structures. Gao's research differs from the mission of the project in that his method for printing multiple materials is not restricted to a coaxial nozzle. However, the experimentation and successful creation of complex 3D structures using multiple materials is vital to the understanding of how to successfully implement true 3D printing.

1.2.2.2 Hergal and Wang on Extrusion Techniques

An article by Hergal discusses techniques that improve print quality when dual extruders are used [4]. The device prints a rampart, a disposable structure around the parts, to catch oozing strings of material when the other extruder is being used to print. The rampart is also used as a wipe station to wipe down the extruder before reaching the part it will be printing on next. In this way, clean printing is ensured.

Differences

This technique was considered for the design of the printer's dual extruders because some of the challenges faced by the researchers are the same as the current challenges of the printer. Printing biomaterial of different viscosities runs the risk of oozing onto the rest of the printed part when not in use.

1.2.2.3 Control Through Software: Rankin and Sinopoli

Previous processes for printing a 3D object involved generating a 3D model in STL format, converting the STL file to GCODE using a slicing software, then using a communication software, such as Pronterface, to load the GCODE and send it to the printer. According to

Rankin, the constraint of this software was that it needed to be run from an external computer [5]. This external computer must be connected to the 3D printer at all times.

Differences

This is the design currently implemented in the bioprinter. However, Rankin recommends software called OctoPrint that processes the 3D printer control software on a web interface, so printing is controlled and monitored over the network and can be run on a Raspberry Pi. Operating over the network allows for the potential to connect more control boards to the main interface, which allows more room to control the additional features planned for the printer.

1.3 Problem Statement

The BETA project is helping to expand functionality in SE3D's product line to allow for a better student user experience and more interesting experiments to be run.

1.3.1 Goals

The 3D bioprinting project had two main goals, each with individual sub goals, as listed below:

1. Modular Incubating Box with sensors to set up custom environments for different experiments based on temperature and light. An automatic image capture to analyze experiment over time. A simple user interface to configure and run experiments
2. Explore solutions—proof of concepts to add functionality to 3D Bioprinter based on multi-material extrusion and auto-homing calibration of syringe

As the functionality was increased, the cost needed to be kept in mind so that the price of the final project remained as low as possible. These additions to the current SE3D 3D Bioprinter will increase its effectiveness in the classroom and allow the target audience, high-school students, to better engage in STEM education activities.

1.3.2 User Scenario

1.3.2.1 Student User Scenario

The client market of the 3D bioprinter is strictly aimed towards middle and high school science classes. Experiments utilize an improved 3D bioprinter design based on SE3D's reBEL printer and a simplified user interface so that students and teachers with no prior experience with 3D printers can easily set up and run an experiment using the following procedure:

1. Insert syringes with the desired biomaterial into the printer filament head.
2. Initiate printing. This is where the printer auto calibrates and begins printing into the petri dish.
3. When print is complete, transfer petri dish to BETA Box.
4. Ensure the Box is set to the desired temperature and lighting, then initiate image capture.
5. When culture is complete, remove experiment and reset BETA Box.
6. Transfer captured images to external drive for documentation.

1.3.2.2 Teacher User Scenario

It is anticipated that high school teachers will be the initial users of both the 3D bioprinter and the BETA Box. Before a high school student enters the classroom to use either of these technologies for an experiment, the teacher will first initiate experimental setup and programming to his or her liking.

Before students can use the Box to run their experiments, the teacher or student must set the temperature of the Box and allow time for the environment to get up to temperature and set the color of lights necessary for the experiment.

1.3.3 Methods

The team took a hands-on approach to design for this project. The project started with the construction of an open source 3D printer, which helped increase understanding of the functionality and the use of different parts in a 3D printer. For actual design work, each member of the group developed his or her own design and then brought that design to the group for review and critique. An open discussion was held where the pros and cons of each design were discussed. After all proposed designs had been vetted, the group voted on what they saw to be the optimal choice for each subsystem. The open communication system within the group facilitated the design of the Box, which was a combination of multiple teammates' ideas for the incubator subsection.

1.3.4 Team Structure

To manage the large team effectively, a communication system was set up between all group members. All group members had one another's cell phone numbers and a cellular application called GroupMe was used for basic group communication, which occurred daily. Regular meetings were scheduled twice per week, and then individual members met outside of those meetings to work on the project. The advisor for the project also met with the group once per week. The minutes from these meetings can be found in Appendix M. This method of constant meetings helped the group stay on track and be successful. The only issue with team management was finding a time for the entire group to meet—the group members committed to early mornings for group meetings.

The team laid out a specific set of goals for each week and a detailed plan for how to achieve those goals. If each individual member of the team had not met these goals, the project faced an increased chance of failure. This type of scenario did not occur in the project and each team member carried his or her own weight, contributing to the successful completion of the project.

2. Design of System

The following image outlines the BETA project, broken down into subsystems and final outcomes.

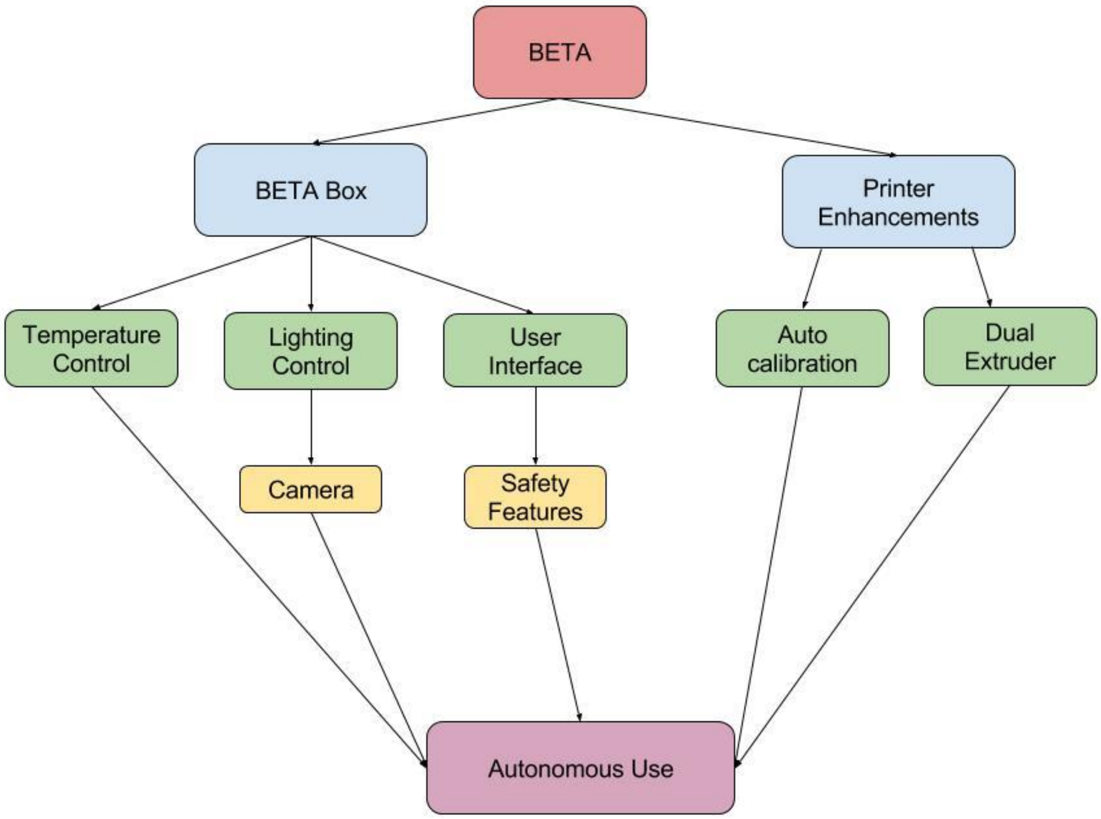


Figure 2.1 System-Level Design

This sketch denotes the three main systems that made the Bioprinter project, in blue. These consist of the incubation chamber denoted "BETA Box," the enhancements to the bioprinter itself, and the user interface for the BETA Box. Each of the main systems is further subdivided into key aspects of each system, denoted by green boxes. Yellow boxes are subcomponents of the individual projects and tied to a specific subproject that directly affects them.

2.1 Deliverable: BETA Box - System Level Overview

The BETA Box is a standalone, low cost, high-tech incubator that can be used with SE3D experiments to culture bacteria and cells over an extended period of time. Figure 2.2 shows an image of the final BETA Box prototype.

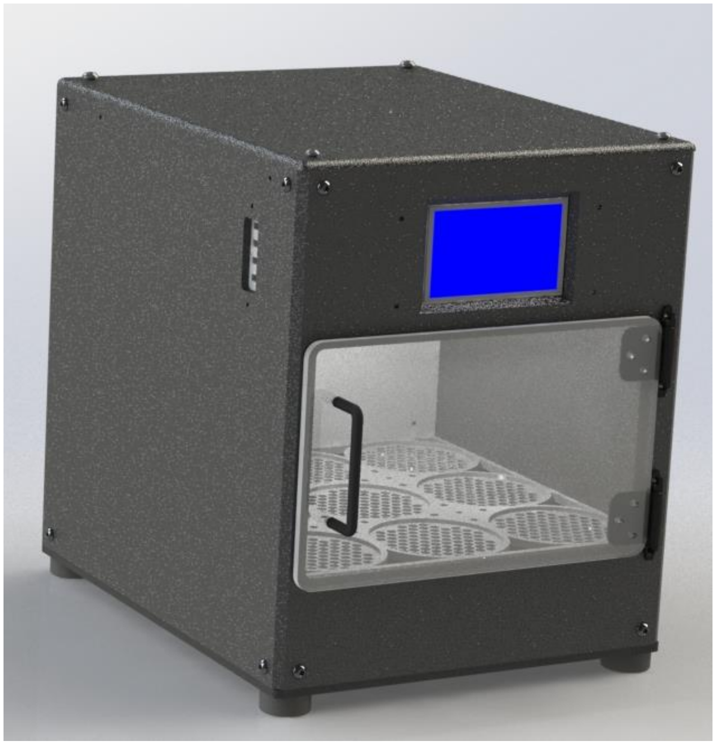


Figure 2.2: BETA Box Final Prototype

There is lighting control inside the Box to provide constant lighting conditions and a camera for documentation of the incubation with an 8-megapixel camera that takes independent photos of each experiment. The user interacts with the BETA Box by using a touch screen graphical user interface that is connected to a main computing system. Figure 2.3 shows a graphical layout of the BETA Box and each of the appropriate subsystems.

The Box senses the temperature inside the incubator by simulating the conditions experienced by biology in a petri dish with distributed analog sensors throughout the system that can each measure temperature within 0.1 °C and digital humidity sensors that can measure the relative humidity inside the incubation chamber to within 5%.

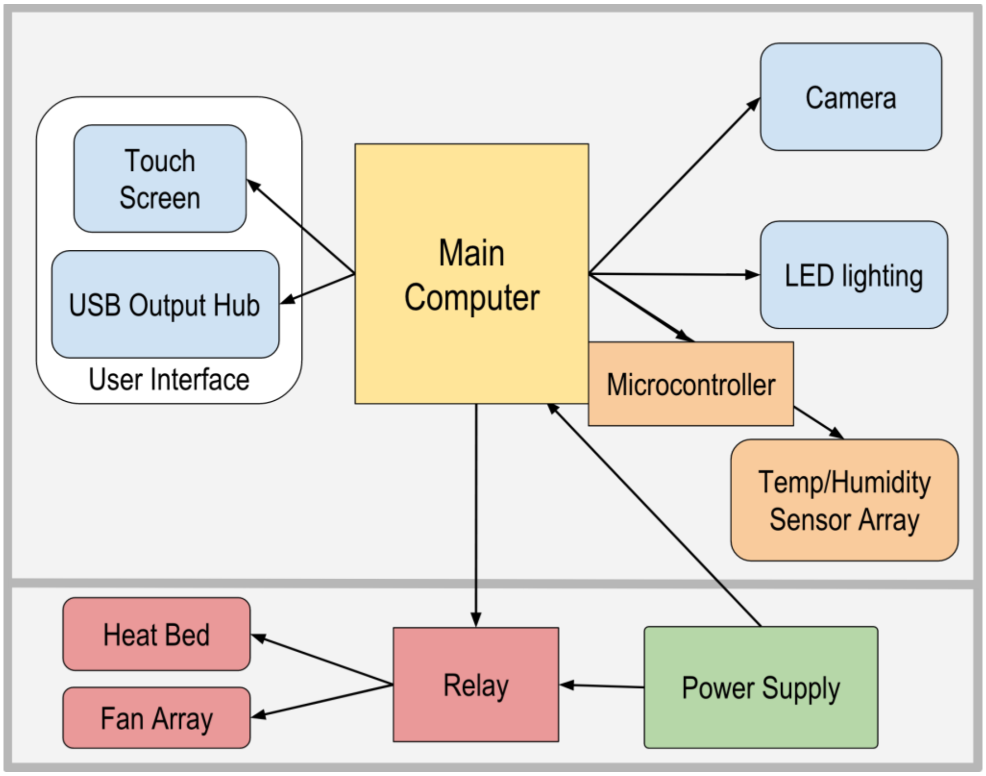


Figure 2.3: System-Level Overview

The Box is able to change lights so that the biology has minimal reflectance and can be exposed to the entire RGB spectrum at different levels of brightness. The heater used in the Box consists of a 150-Watt resistive heat bed with 4 high-speed 66 CFM fans that create a convective heat flow throughout the BETA Box. The BETA Box also has an independent power supply for reliable power with any standard outlet and isolation of power for all of the different systems in the Box.

The Product Design Specification document can be found in Appendix A, which outlines the main requirements for BETA Box. The BETA Box requirements are summarized below:

1. Support timed image capture at a pre-specified image per time interval
2. Have an interactive user interface
3. Allow user to monitor and control light over full spectrum
4. Allow user to monitor and control temperature between 20-50 °C with +/- 2°C variation
5. Allow image capture in +/- 1 second intervals and download those images
6. Be user friendly and intuitive for high school students

7. Meet safety standards according to UL Standards
8. Act as a secure Box with teacher override capabilities
9. Perform reliably for over 10,000 hours
10. Function without being connected to a desktop or laptop computer
11. Cost the customer $500

2.1.1 Deliverable System Operation

The BETA Box system was designed to be operated by high school biology students in the classroom. The BETA Box is meant to be user-friendly and intuitive for high school students to be able to use with no prior training and for teachers to be able to operate and control with very few instructions.

2.1.2 Product Design

The BETA Box was designed to be a fully functional, stand alone product for SE3D and their product line. The design process took into account cost, functionality, and aesthetics to create a product that would appeal to the most number of high school classrooms. There were many iterations on the CAD design, system functionalities, and overall system integration to create the most robust and aesthetic product.

2.2 Feasibility Study: Bioprinter Enhancement

2.2.1 Auto-Calibration

The auto-calibration feature of the Bioprinter eliminates the user's need to manually lower the plunger of the device to meet the top of the syringe. This automation reduced experiment setup time and user error. Also, less material will be wasted since inexperienced users, such as young students, may put too much force on the plunger and extrude biomaterial before the print begins. This enhancement to the 3D Bioprinter makes for a more user friendly experience and increase in customer satisfaction.

To accomplish the parameters of the auto-calibration, a force sensor is placed at the end of the actuating arm connected to the stepper motor, as seen in Figure 2.4 below.

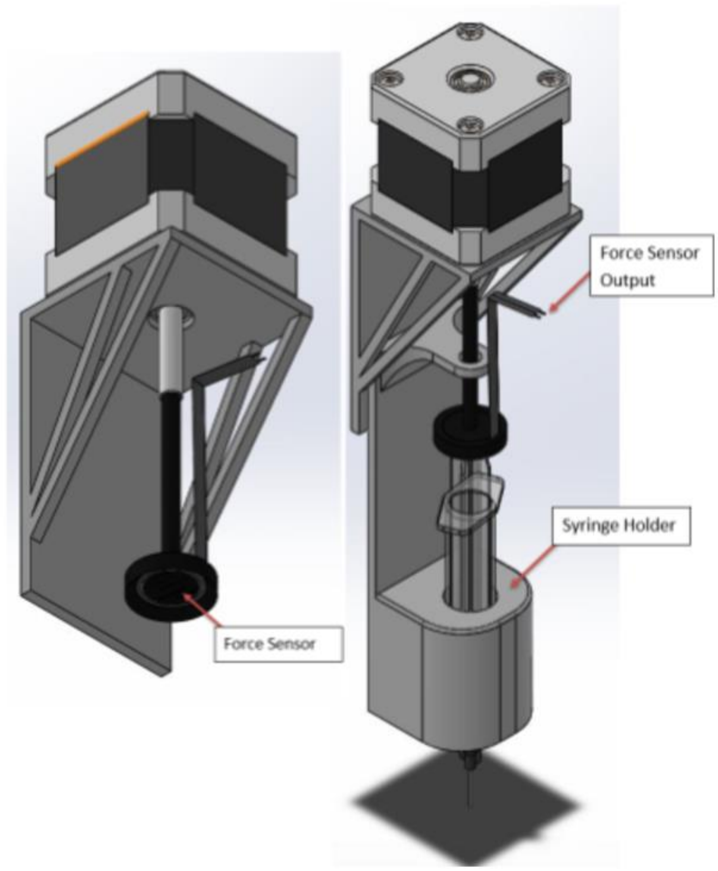


Figure 2.4: Auto Calibration Final Design

The arm is lowered until it presses upon the top of the syringe plunger, which then begins printing of the biomaterial. Through multiple tests, our group gathered data from the force sensor that allowed us to calibrate when the biomaterial was beginning to be extruded and when the syringe was empty, which allowed the auto-calibration feature to prime the bio extruder for printing and then also register when the syringe was empty and stop the bioprinter.

2.2.2 Dual Extruder

The dual extrusion feature of the Bioprinter was a feasibility study designed to explore the ability to improve upon the existing bio extruder to allow for multi-material printing and true 3-dimensional bioprinting capabilities. The current SE3D bioprinter model is limited to a single bio extruder, which currently prints biomaterial in 2D arrays. The bioprinter is unable to create 3D structures since the biomaterial cannot keep its shape while hardening. Currently, the bio extruder has to print one set of biomaterial, then be reloaded and recalibrated for the second print, lengthening the process and potentially invalidating the experiment. The final design can be seen in Figure 2.5 below.

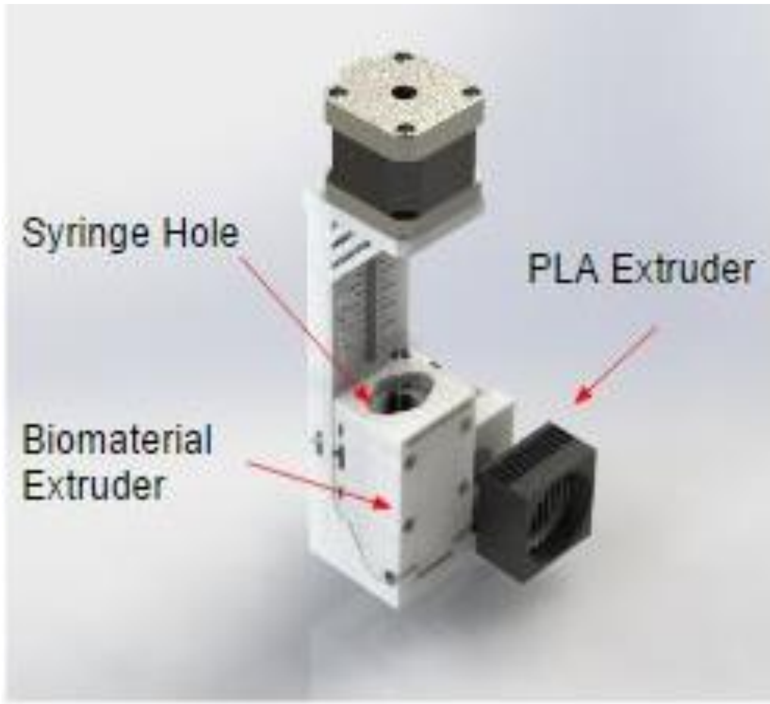


Figure 2.5: Final Dual Extruder Design

This design is important for the 3D bioprinting since the plastic can create a mold to support the biomaterial as it hardens into its 3D structure. Furthermore, this design proves the feasibility of a dual extrusion system, which establishes that two materials can be printed at the same time, satisfying the second goal.

2.3 System Requirements

Working with SE3D, the team was given guidelines and customer needs that would ideally be incorporated into the final products. These needs were prioritized and quantified to establish a list of specifications that needed to be met. This provided the team with a metric in order to measure the success of the project. Although some criteria have been mentioned in section 2.1, the full list of product design specifications for the incubation unit and printer enhancements can be found in Appendix A.

Part II. Incubation Box Deliverables

3. Structure

The Box is durable plastic that can operate for many years in the classroom at a range of warm temperatures. The primary elements of the structure are the large plastic panels that make up the cubic shape of the Box. Other elements of the BETA Box structure include joint and other supplementary hardware, and all non-electric components. The design process, required specifications, iterations of the BETA Box, and material selection for the primary panels are detailed below.

3.1 Design Process

The initial conception of the BETA Box was derived through the concept sketches previously mentioned. These primary concepts were compared in the decision matrices found in Appendix B. As a result, the "Box" design was selected. Through our iterative design of the BETA Box's structural components, the best final prototype was created using SolidWorks. The designs evolved from the foam core board Box to the HDPE final Box. Modifications were made when necessary to incorporate all the additional subsystem space updates and to address structural problems with the successive prototypes. Drawings for the final prototype can be found in Appendices C and D.

3.2 Requirements

When designing the different prototypes of the BETA Box, the team came up with different requirements for functionality of the Box structure, listed below:

1. Aesthetic appearance to appeal to high school students
2. Durable through shipping and extended high school use in laboratory setting
3. Designed using as much COT parts as possible for easy replacement if broken
4. Easy disassembly if anything needs to be modified or replaced on the inside of the Box
5. Room in back to house all the electronic components out of the way
6. Room under the base to house the heating unit
7. Robust integrity to endure large impact stresses

3.3 Alternative Prototypes

3.3.1 Prototype 1: Foam Core Board

The first prototype of the Box was a 12in. cube made of foam core board and was designed in Fall 2016 as a proof of concept. A SolidWorks model of this design can be seen below.

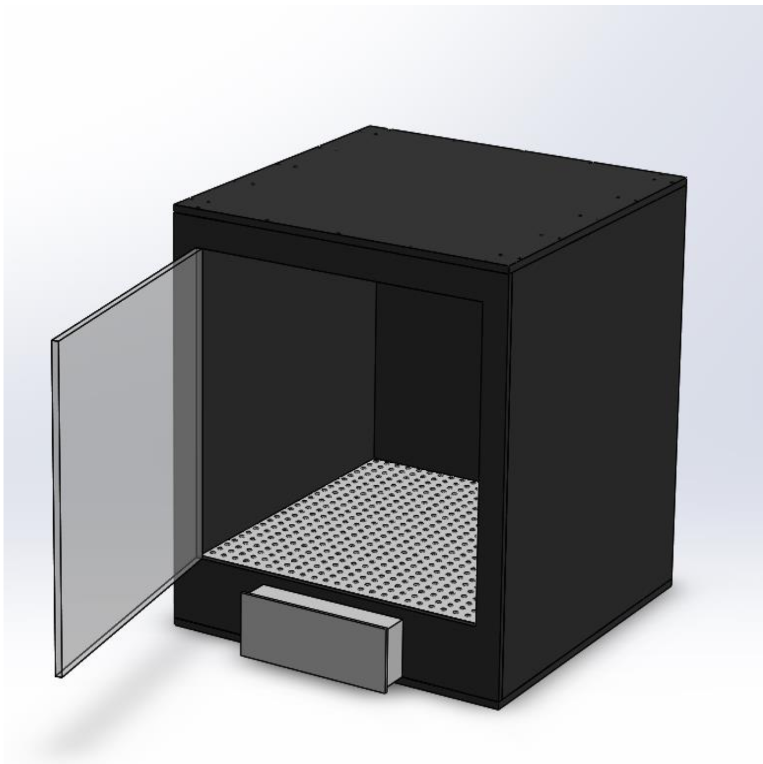


Figure 3.1: BETA Box Prototype 1

This initial model included double-paneled walls for optimal insulation and hidden wiring. The perforated heat bed and hinged door in the final model are seen here. Features include a modular thermal drawer (seen as a pull out underneath the heatbed) and a removable camera (not pictured). The removable camera was designed to plug directly into the computer using a USB. The goal of these removable, modular features was to sell them individually, to lower initial costs for teachers.

3.3.2 Prototype 2: Acrylic

The second prototype successfully fit nine petri dishes, and met the dimensions of 15in. in width, 14in. in depth, and 12in. in height.

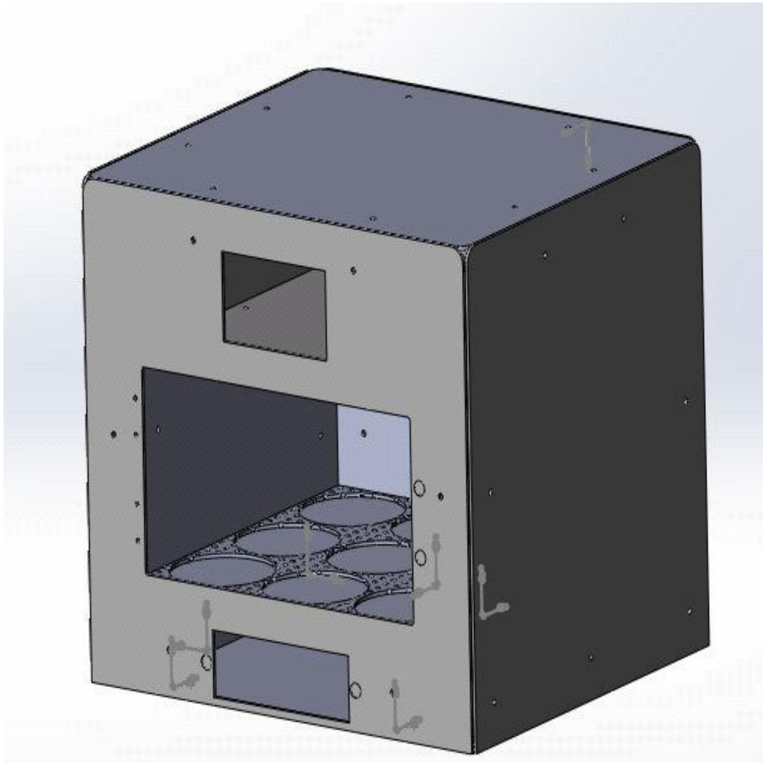


Figure 3.2: BETA Box Prototype 2

The second prototype of the BETA Box was made out of acrylic, intended to match the material and aesthetic design of the existing r3bEL printer. The material, however, lacked the desired durability of the final product. This model still shows the modular thermal compartment below. Additionally, the left, right, and back are all double-paneled to hold wiring and electronics. However, this double-panel design made the Box larger than desired. The computer system and additional wiring also had ample space in the top compartment, more so than was needed.

3.3.3 Prototype 3: High-Density Polyethylene

Prototype 3 was the final design of the BETA Box, and can be seen in Figure 4.3.

The final design of the BETA Box features ¼ in. HDPE, instead of ⅛ in. thick material. All sides are single-paneled, with the exception of the back and lower compartments. As a result, the dimensions of the final Box are reduced from prototype 2, cutting down on structural material costs.

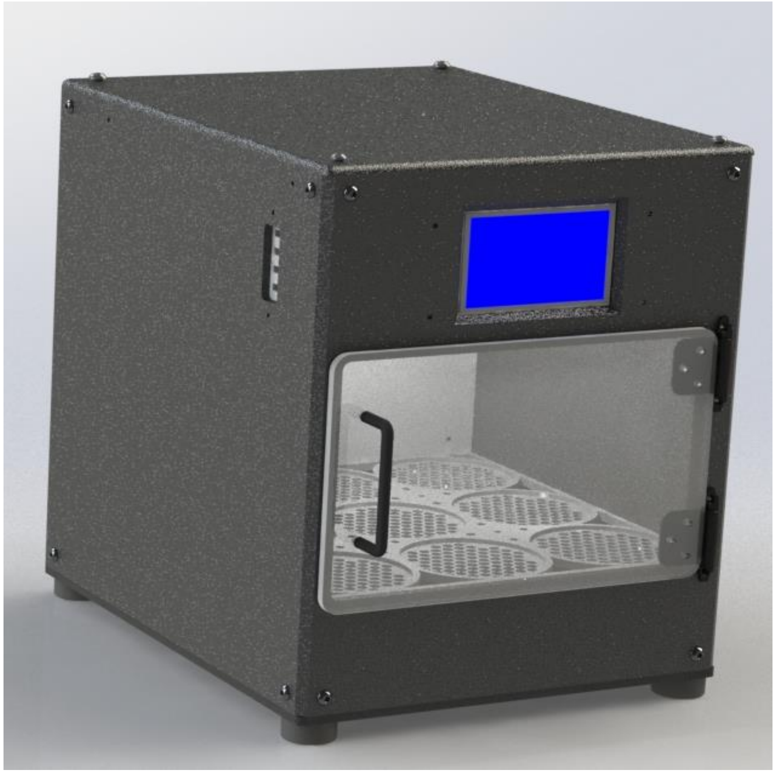


Figure 3.3: BETA Box Prototype 3

The modular thermal compartment was removed, after determining that the cost saved was outweighed by the added complexity of creating a modular design. All wiring is fed directly to the back, with a slim channel in the ceiling to connect the touch screen and computer to the rest of the equipment. This design is optimal for minimizing costs and optimizing durability.

3.4 Material Analysis

3.4.1 Material Selection

The structural material of the BETA Box was carefully selected to meet the following criteria, listed in order of importance:

1. Color Availability—black, white, clear
2. Manufacturable via laser-cutting or routing
3. Low-Cost
4. No deformation at or below 90ºC
5. High impact resistance

Although the original material of choice was acrylic, because it was easily manufactured and because SE3D previously used acrylic for its printers, this material was too expensive for meeting the target price of the Box. Additionally, it was very brittle and did not meet the temperature criteria very well. As a result, a vast range of materials was explored, as can be seen in Appendix E. A summary of materials considered and used follows.

3.4.2 Material Evaluation

The top five materials that met the listed criteria above were acrylic, low-density polyethylene, high-density polyethylene, polypropylene, and polystyrene. All five materials were tested to evaluate which material optimally met all five criteria.

3.4.2.1 Testing Procedure

All materials were tested as ¼ in. thick samples, matching the design thickness of the prototype 3 model. Each material was tested using the following procedure:

1. If applicable, cut the material into a 12 in. x 12 in. square.
2. Laser cut the established pattern into the material. If laser cutting fails, end experiment. If laser cutting is successful, review aesthetics of laser cut edges and overall material.
3. Place material directly on heatbed. Secure thermocouple end between material and heatbed.
    a. Plug in heatbed and allow to reach full temperature, 100ºC.
    b. Measure the temperature of the opposite side of material from that touching heat bed. If below 40ºC, touch with palm, and evaluate if material is safe to touch in classroom.
    c. Remove material from heatbed and visually review that no melting occurred.
    d. Attempt to bend material in half back and forth and determine if needed structural integrity will remain at 100ºC.
4. Attach M3 bolt and ¼ in. bolt in laser cut holes. Pound head of each bolt one time so that bolt is in full shear stress with the material. Remove bolts and review damage accrued on material.
5. For all tests, rate on scale 1-5.

3.4.2.2 Testing Results

A summary of the results of the final material testing can be seen below. Detailed results and scaled ratings can be seen in Appendix E.

Table 3.1: Material Testing Evaluation

Material	Cost/ in.^{2}	Laser Cut	Heat Deflection @264 psi	Brittleness
Acrylic	$0.025	Very good	82.2ºC	High
LDPE	$0.020	No	36.7ºC	Low
HDPE	$0.021	Good	80.0ºC	Low
Polypropylene	$0.017	No	43.3ºC	Low
Polystyrene	$0.012	No	93.3ºC	Very Low

As can be seen above, high-density polypropylene optimally met the design criteria for the BETA Box structure. The final choice was Starboard, a specific brand of HDPE. Matte black was chosen for the outer structure, and glossy white for the dish bed and back inner panel.

3.4.3 Additional Considerations

3.4.3.1 White Glossy HDPE

Although the tested HDPE (matte) could be laser cut, the glossy HDPE did not manufacture well in the laser cutter—it melted rather than cut. As a result, the dish bed and inner back panel were both manufactured using a CNC router table.

3.4.3.2 Clear Acrylic

It was determined that the door of the BETA Box must be transparent, so that students may observe their experiments while they are inside the Box. Because acrylic was the only material on the list of top five materials that was available as ¼ in. thick and clear, it was selected as the choice material for the door.

3.5 Conclusion

All the design decisions for the structure of the BETA Box were iterated and developed to optimize cost, durability, and aesthetic appeal. The material selection was HDPE for the panels of the Box. Also, the use of extrusion to mount the panels led to simplified construction as well as easily accessible electronics in case maintenance is needed. All these final considerations led to a sturdy, durable, yet cheap and simple structure for the BETA Box.

4. Thermal Subsystem

The thermal package of the BETA Box was developed to keep the biological experiments at a specified temperature over an extended period of time. Consequentially, accurate heating was determined to be one of the most critical functions of the Box. It was understood that most low-cost desktop incubators sell for a minimum of $1000 and since this Box had to be sold for $500 with expensive computing equipment, the heating element had to be easy to manufacture, low cost, and reliable.

4.1 Requirements

Table 4.1 lists the requirements for the BETA Box. These requirements were validated from research of similar commercial incubators used in the industry.

Table 4.1: BETA Box Requirements

Function	Requirement
Temperature Average	± 2ºC setpoint
Even Dish Bed Temperature	± 2ºC entire heatbed area from setpoint
Warm-Up Time	<30 minutes
Safety	U.L. Standards

4.2 Design Process

The design process for the thermal subsystem consisted of constant iterations with low cost heaters to find the results that met the specifications. The main parameter most difficult to achieve was evenly heating the dish bed at ± 2ºC entire heatbed area from setpoint. The parameters tuned during this design process consisted of:

1. Heater
2. Fan position and alignment
3. Fan volume flow rate
4. Insulation

Figure 4.1 shows the methodical closed loop design process used to achieve even heating of the dish bed within ± 2ºC.

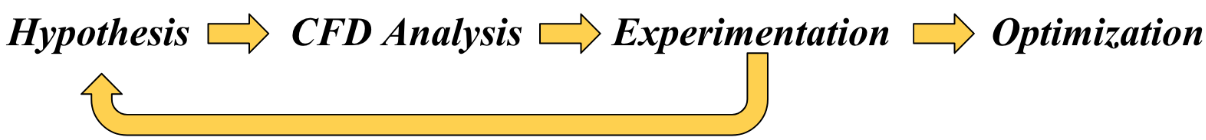


Figure 4.1 Thermal Subsystem Design Process

This process led to a successful design after 18 different physical tests and multitudes of computational simulation tweaks.

4.3 Heat Transfer Analysis

4.3.1 Finite Element Analysis

The analysis of the initial design for the heater and the interior of the Box were simulated using the computer program, Abaqus, to ensure the material would not fail while in use. This primary design for the heating system used a heated plate to warm the air beneath the interior of the incubation Box. Although the target temperature was 36-38°C, the heated plate could reach 100ºC when powered fully. The warmed air was blown upwards by fans in order to evenly distribute the heat throughout the interior of the Box. The setup of the system can be seen in Figure 4.2.

This method of convective heat transfer would have taken longer than heating through conduction. For this reason, it was optimal to supply a large amount of power to the heater to reduce the time it took for the system to reach to the desired temperature.

Heating the plate below the Box to a temperature much higher than the target range allowed the system to warm up quickly; however, it also meant that the structure of the Box was exposed to higher temperatures. The first prototype of the incubation Box was made out of acrylic, which softens around 90°C.

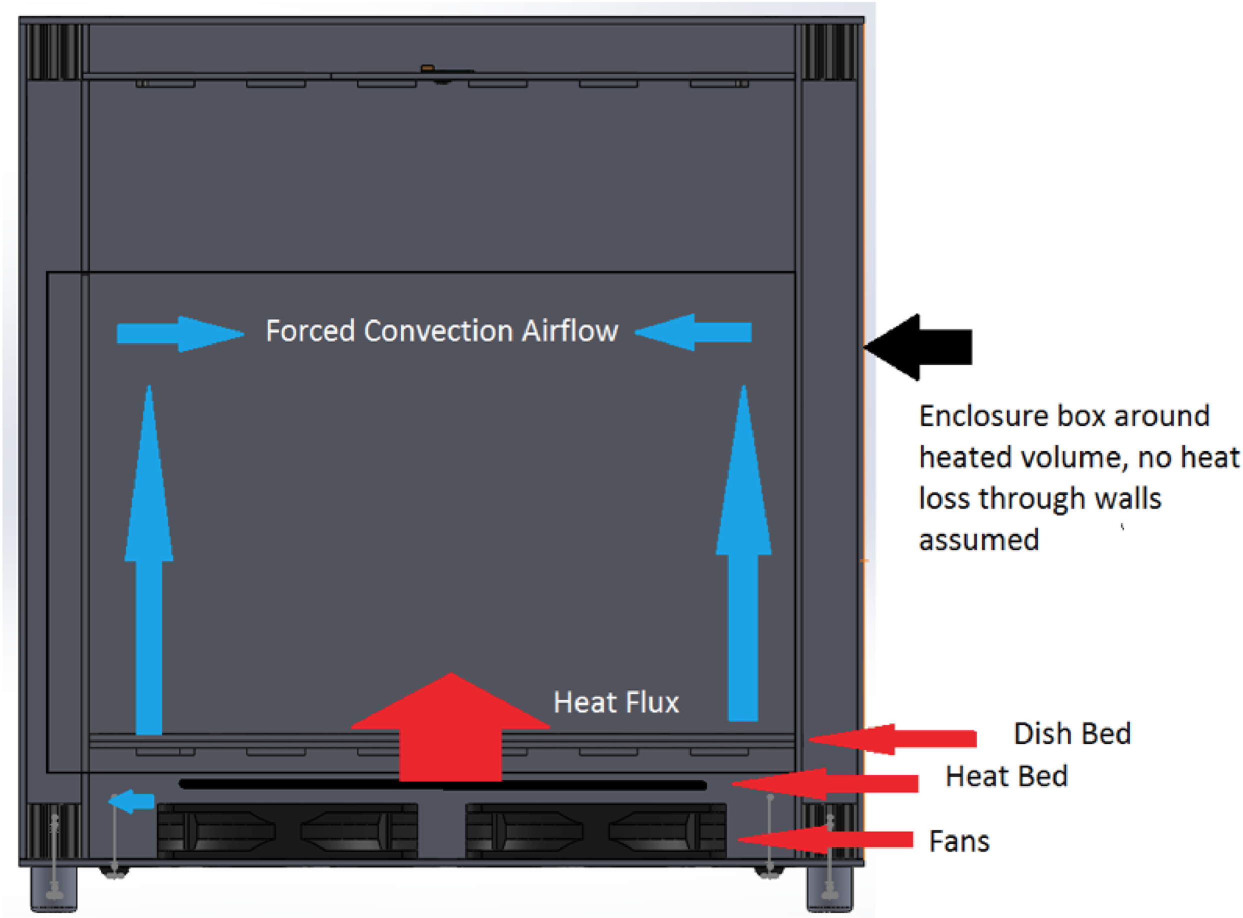


Figure 4.2: System Diagram for FEA Modeling

For this reason, it was important to make sure that the base of the incubation Box was not subjected to temperatures above this value. Since the bottom of the Box was most exposed and closest to the heater, it was the area of main concern. The finite element analysis not only evaluated temperature distribution, but also established that structural integrity would not be compromised due to the heat.

4.3.1.1 Simplifying Elements

The simplifications made for the analysis can be seen in Figure 4.3. The left image shows the second iteration of prototypes of the BETA Box. The right image shows the simplification of the Box for FEA testing.

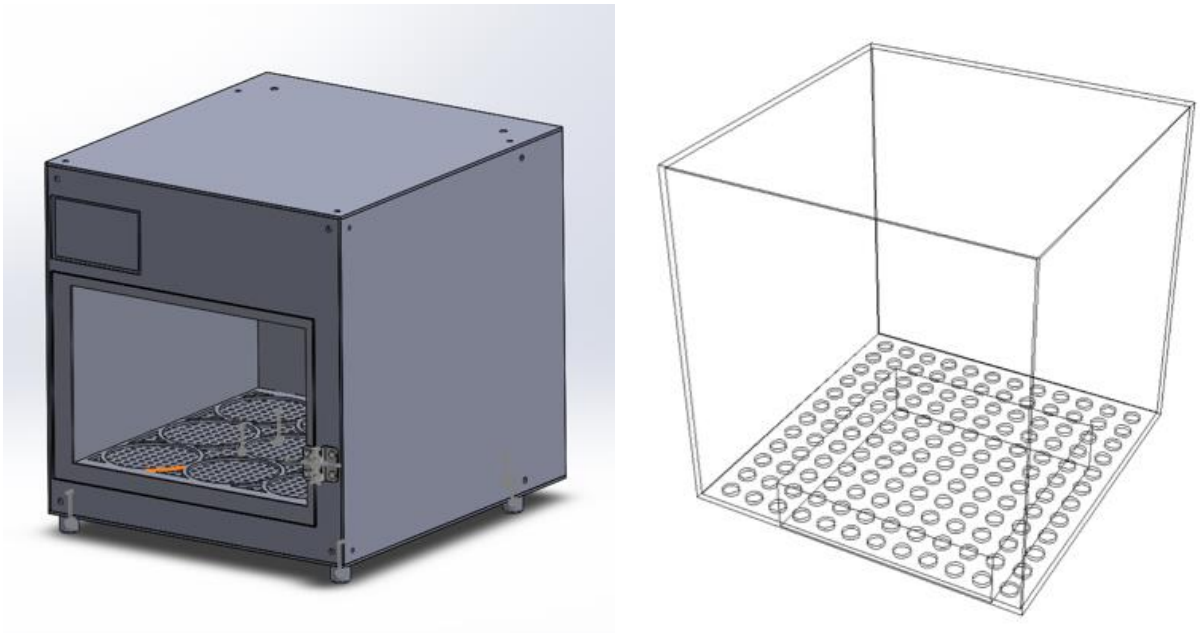


Figure 4.3: BETA Box Prototype 2 and Simulation Simplifications

When doing the thermal analysis of the incubation chamber, it was assumed that the heat was traveling in a single direction, beginning at the heating plate in the base of the chamber, and traveling upwards towards the ceiling of the chamber. In this simulation, radiation was ignored and the incubation Box was treated as a simplified Box structure, with the door being treated as a solid plate since the entire structure was insulated. The Box experienced conduction at the plate and convection in the space above as the hot air rose. The convective coefficient was assumed to be constant throughout the heating process and the heating bed was represented to heat the chamber uniformly. The material properties of the air, including density, thermal conductivity, and viscosity were also assumed to remain constant. A further simplification was taken, in an effort to reduce render time and have a smoother simulation, by changing the sizing of the holes in the dishbed while decreasing the number of total holes, keeping the same ratio of plate area to holes on the incubation plate.

The only materials necessary to consider for thermal transfer in this analysis were the acrylic and air. This was because the setup of the system dictated that as the fans moved heated air away from the heat plate, only the air and acrylic affected the heating of the target product, with the exception of the petri dish. The following properties in Table 4.2 were relevant for the execution of this thermal analysis.

Table 4.2: Relevant FEA Material Properties

Solid Material	Conductive Coefficient (W/m-K)	Thickness (m)
Acrylic	0.19	6.35×10^{-3}
Fluid Material	**Convective Coefficient (W/m^2-K)**	**Density (kg/m^3)**
Air	38.25	1.225

4.3.1.2 Model Results

In order to ensure that the results of the Abaqus analysis were as precise as possible, it was necessary to adjust the seed size of the simulation mesh until the values began to converge. The FEA elements needed to be small enough to create an accurate image of the temperature distribution in the system. The final seed size was set to 0.5 and can be seen in Appendix F along with other mesh sizes tested.

The simulation was run for power inputs of 120W, 100W, and 80W to show the maximum temperatures of both the heater and the structure of the Box. Table 4.3 shows the results for each input.

Table 4.3: Abaqus Results

Power Supplied to Heater	Heat Flux Generated	Max Temperature of Heater (°C)	Max Temperature of Structure (°C)
120W	0.257W/m2	92°C	86°C
100W	0.215W/m2	80°C	75°C
80W	0.172W/m2	69°C	65°C

As shown in Table 4.3, when the highest power was supplied to the heater, the temperature of the incubation plate came close to the deformation temperature of acrylic but did not reach it. In order to ensure that the plastic would not soften, the team decided to keep the temperature at least 10 degrees below the deformation temperature. In that case, 120W would be too much power to apply to the system. For 100W and 80W, the structure’s temperature seemed more reasonable and the risk of deformation was lower. It was also noted that in the development of the controller, the 120W power supplied to the heater would saturate the controller and therefore, it was planned that the heater would never run at maximum power for optimal control.

The analysis obtained by the finite element model seemed consistent with the hand calculations. The values were slightly elevated, as expected due to the simplifications made. This error was mainly due to the fact that Abaqus could not simulate convection between two separate parts, and therefore required this heat transfer to be modeled by conduction.

The heat distribution for the base plate of the incubation Box can be seen in Figure 4.4. The figure shows the results when the full 120W was supplied to the heater.

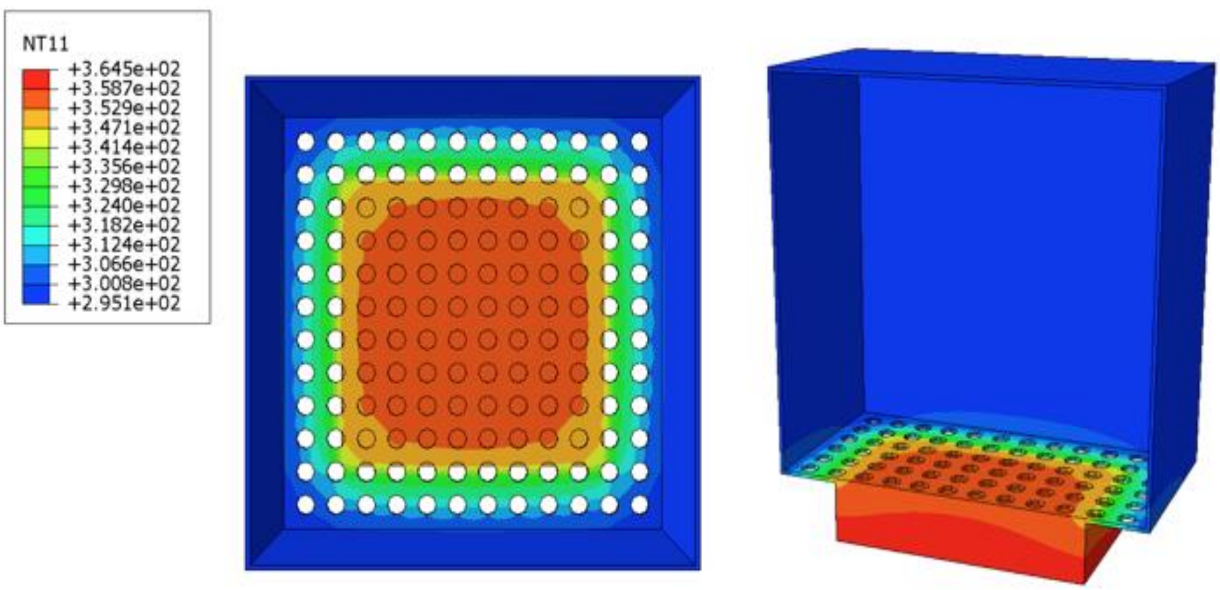


Figure 4.4: Heat Distribution Powered by 120W

Figure 4.4 also showed that the temperature was not evenly distributed across the base of the Box. There was a hot spot in the center and the edges did not receive much heat. Based on the simulation, the sides of the Box were not heated at all; however, this was again due to the inadequacies of the simulation since Abaqus only allowed for convective interaction from a solid to the air. The lack of information in regard to the walls of the Box was not ideal, but the main concern of the analysis was the base of the Box and its proximity to the heater.

It was found that in the simulation that if the full 120W were supplied to the heater, this would have induced a maximum heater temperature of 92°C but a base plate temperature of 86°C. This was determined to be too close to the limit at which acrylic begins to soften, indicating that 120W was too much power to apply to the system. The simulation showed that for 100W and 80W, the structure's temperature appeared more reasonable and the risk of deformation was lower. Since the heating element was sized so that a temperature controller could be effectively utilized without saturation, then the FEA results state the design would be acceptable to meet the target specifications without overheating of the material. However, the finite element analysis was a preliminary simulation, and further physical testing was conducted.

4.3.2 Computational Fluid Dynamics Analysis

CFD, or computational fluid dynamics, was used to simulate the overall heat distribution of the heater on the heat plate. Unlike the FEA, which was used to understand the conductive heat maximums for different heaters, the CFD software was used to understand the airflow distribution, and how that affected the heat flow. To do this, a simplified model with only the heat bed, Box frame, and fans was created and imported into the software. A simplified model was created to streamline the modeling time and to analyze only the important features. The variables modified to find the best overall distributions were heater, fan position, and fan volume flow rate. The first simulation is shown in Figure 4.5.

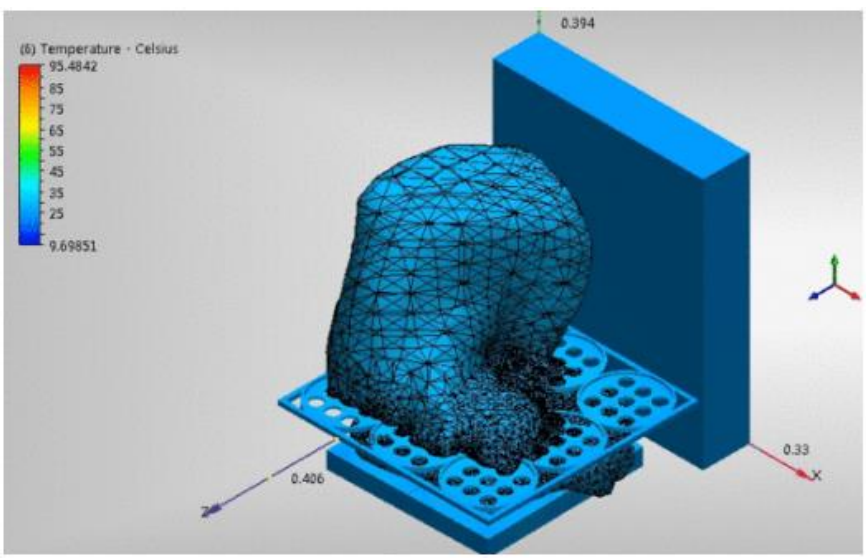


Figure 4.5: Initial Simulated Temperature Distribution

This simulation shows the primary challenge for even heating: attaining the same temperature around the entire edge of the dish bed. After multiple simulations, a more even heating distribution was achieved, shown in Figure 4.6.

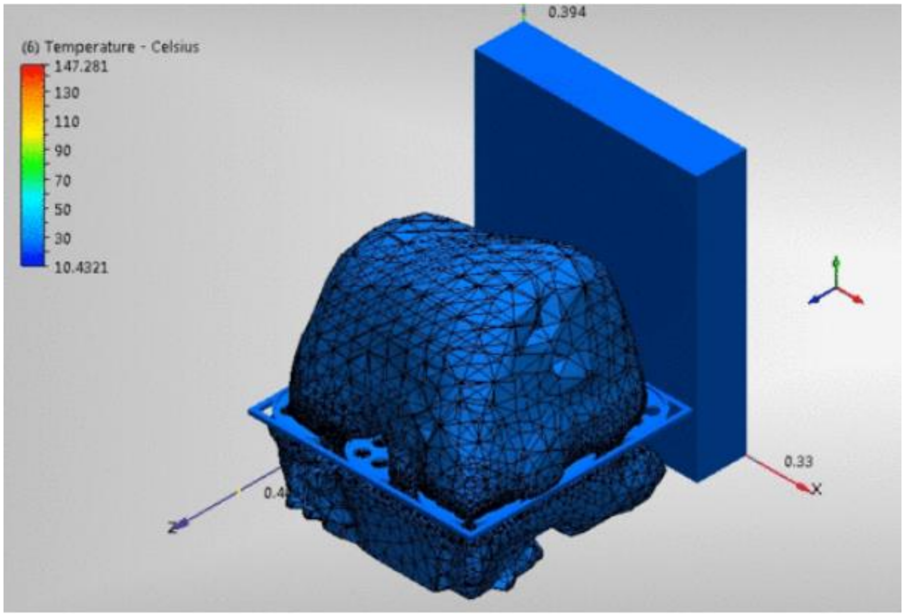


Figure 4.6: Final Simulated Temperature Distribution

This image shows that the even temperature extends to the corners of the dish bed. The parameters for this simulation are summarized in Table 4.4.

Table 4.4: CFD Variables

Variable	Final Selection
Heater	150 W, 8x8 inch Heat plate
Fan position and alignment	Below heat source; Airflow directly up
Fan Volume Flow Rate	Four 66 CFM Fans

4.4 Experimentation

Using the initial findings from both analysis methods described above, physical experimentation of the thermal subsystem was performed to determine the exact setup and positioning of each element to reach the established specifications. The purpose of the experimentation was to evaluate the optimal equipment setup in the Box undercarriage to keep all 9 petri dishes placed on the bed at a steady state temperature of 37ºC.

4.4.1 Experimental Procedure

The following variables were tested physically to determine the best thermal setup:

1. Heat source
2. Fan volume flow rate
3. Fan volume flow rate
4. Insulation

During this experiment, the equipment in Table 4.5 was used.

Table 4.5: Thermal Subsystem Experimentation Equipment

Equipment	Model
DC Brushless Fan 12V 0.10A	AV-F4010MB
SFS 120mm Cooling Fan 38.2CFM	ROCF-13001
Spire ORION 120mm Cooling Fan 66CFM	LYSB01HP4GDXQ-ELECTRNCS
Lightobject 120mm Cooling Fan 200 CFM	EFAN-FFB1212EHE
Uxcell 10W 10Ohm Cement Resistor	a11101400ux0166
SainSmart RepRap MK2B 3D printers Dual Power PCB HeatBed	20-029-101
Adasong Digital Temperature-Humidity Sensor	AM2302
Thermal Camera	FLIR E4

4.4.1.1 Setup

For all tests run, the following setup and calibration was required:

1. Ensure entire inner Box is closed—thermal plug is sealed, door is closed, no major holes are uncovered.
2. Adhere thermostat to back inside wall centered immediately above heatbed.
3. Establish distance and emissivity for thermal camera settings.
   (Emissivity of white acrylic = 0.84)

4.4.1.2 Procedure

For each test the following procedure was used:

1. Capture an image every 2 minutes for all following steps unless otherwise indicated.
2. Ensure entire Box and all equipment is cooled to below 32ºC. Record approximate initial temperature.
3. Turn on heat plate and fans (as applicable), close door.
4. Allow system to heat for approx. 10 minutes, until air just above bed is 37ºC according to thermostat.
5. Open door and add a covered petri dish to front middle location, and uncovered petri dish to front right location.
6. Wait approximately 5 minutes or until both petri dishes reach 37ºC.
7. Open door for 10 seconds and close. Capture image every 1 second.
8. Wait 2 minutes or until dishes return to 37ºC. Or observe temperature change in dishes.
9. Turn off system.

4.4.2 Results

The picture on the left of Figure 4.7 below shows the area of Prototype 2 captured during thermal experimentation using the FLIR camera, and on the right is the final thermal distribution.

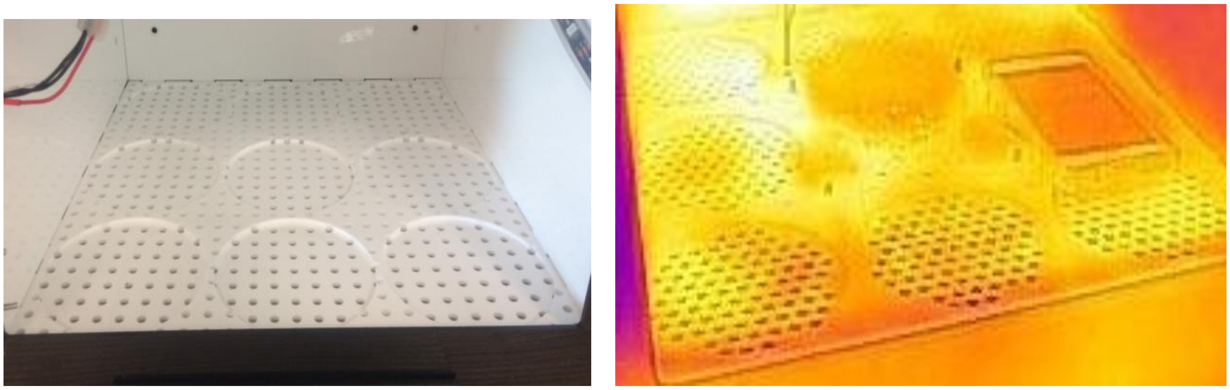


Figure 4.7: BETA Box Prototype 2 Dish Bed and Final Temperature Distribution

Table 4.6 outlines the results of each trial. The variables of the subsystem were modified based on the previous trial results.

Table 4.6: Thermal Subsystem Experimental Results

Test	Image Range	Initial Box Internal Temp. (ºC)	Ramp up time (room temp to 32ºC, 30ºC) (min)	Initial Test Box Internal Temp. (ºC)	Time to reach 37ºC (min)	Time for covered dish lotion to reach 37ºC (min)	Notes
1- heat plate only, test center dish	84-88	24	5:15	32.2	6:11	0:56	Heat plate gets super hot with large surface area, bad bed sag
2 - heat plate, 4 38cfm fans, test center dish	89-100	22.8	6:12	30	11:15	5:03	Good sound level, lowered heat bed to rest on fans
3 - heat plate, 4 38cfm fans, test front left dish	101-120	26.6	3:17	30	14:31	11:14	Changed to images every 2 min after 10:17
4 - heat plate, 4 66cfm fans, test center dish	121-124	23.1	3:21	33	4:16	0:55	
5 - heat plate, 4 66cfm fans, test front left dish	125-138	26.8	2:18	30	7:34	5:16	
6 - 2 resistors, test center	139 - 147	25.1	7:05				
7 - heat plate, 4 66cfm fans angled toward corners	148-173	20.8	11:45	30.3	22:45	11:00	Moved to images every 2 min after 16min
8 - heat plate, 4 66cfm fans angled toward corners, test center	174-183	30.8	0	30.8	7:24	7:24	
9 - heat plate, 4 66cfm fans angled across plate 2 on each side, test center	184-203	21.5	4:20	21.5	13:20	13:20	Still too hot in center
10-heat plate on top with a thick layer of fiberglass insulation over the heat plate, 4 66cfm fans (Corner)	0-6 (SEEK)	21.2	11:00	30.5	19:00	-	
11-heat plate on top with a thick layer of fiberglass insulation over the heat plate, 4 66cfm fans (Middle)	0-6 (SEEK)	20.7	19	29.5	-	-	
12- pinwheel 66 cfm fans angled over heat plate	Null	-	-	-	-	-	7º temp difference between center and corner
13- pinwheel 66 cfm fans angled over heat plate, after median temp was 37, turned off heat bed and ran fans until even	Null	-	-	33	8.53	-	Unplugged at 512 sec, center cooled but front stayed constant
14-Heat plate in corner with 4 66cfm fans (Corner)	7-10 (SEEK)	27.6	5.5	29.8	11.5	35.1	
15-Heat plate in corner with 4 66cfm fans (Middle)	7-10 (SEEK)	25.7	5.5	30	11.5	38	
16- 4 Power resistors in corner (Corner)	11-13 (SEEK)	21.1	6	26.1	10	37	
17- 4 power resistors in corner (Middle)	11-13 (SEEK)	22.4	6	29.7	10	33.6	
18 - Heat plate in center with 4 66cfm fans and insulation	204	21.5	3:00	21.5	4:50	4:50	

As shown in Table 4.6, the optimal setup for the thermal subsystem was established in trial 18, using the following choice variables:

1. 150W, 8x8in. Heat plate
2. Fan placement directly below heat plate, blowing directly upwards
3. Four 66 CFM fans
4. 6x6in. fiberglass insulation centered above heat plate

An image from each trial can be seen in Appendix G. (For some trials, the FLIR camera was unavailable for use, so a SEEK thermal camera was used to capture thermal distribution images.)

4.4.3 Sources of Error

There were two primary potential sources of error in this experimentation analysis, the improvement of which could optimize thermal distribution further in the future. These potential sources of error were that the reflection of the walls affected the emissivity readings on the dish bed, and that the camera was positioned outside of the Box for image capture, requiring the door to be opened for thermal image capture, causing heat loss from the front of the dish bed when images were captured.

4.5 Control System Analysis

A proportional integral control system was implemented for this heating system to stabilize the response. Integral control was used in this system to reduce the steady state error that exists from the heat escaping the system. It was found that open loop control would have too large of an overshoot and would not give the required response from the system. The thermal controller functioned by accepting a temperature setpoint given by the user within the range of 20-50 °C and adapting by sensing actual temperature and evaluating the error. Figure 4.8 shows the control loop diagram for the thermal controller.

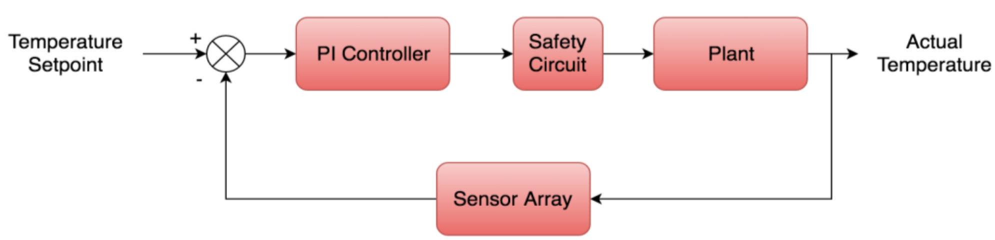


Figure 4.8: Temperature Control Loop Diagram

There is a safety circuit in the control loop diagram, so that if something fails within the BETA Box, such as the heatbed malfunctions or one of the circuits malfunction or breaks, the Box will detect the error with independent circuitry, alert the user, and shutdown the system. The plant consists of the 150-watt resistive heat bed, which is coupled to four 66 CFM fans that cause airflow to go through the holes in the bed of the Box, which was shown in the CFD analysis and physical experimentation to heat the biology. The sensors in the feedback loop consist of an array of high precision analog temperature sensors, which are placed in simulated petri dishes and different points on the heatbed. The temperature sensors all have a ± 0.1 °C accuracy.

The data for the control, created in Simulink, can be found in Appendix L. This control system allowed the BETA Box to beat the design specification of ± 2 °C error and a 30 minute ramp up time, and after appropriate tuning of the proportional and integral gains through the Ziegler-Nichols method, it was found that the BETA Box had a ±1 °C error and a 5 minute heat up time. This smaller error was found without heat addition and voltage differences from the other subsystems in the BETA Box.

Figure 4.9 shows data from the temperature controller with a setpoint of 37°C and all four temperature sensors overlaid on the graph.

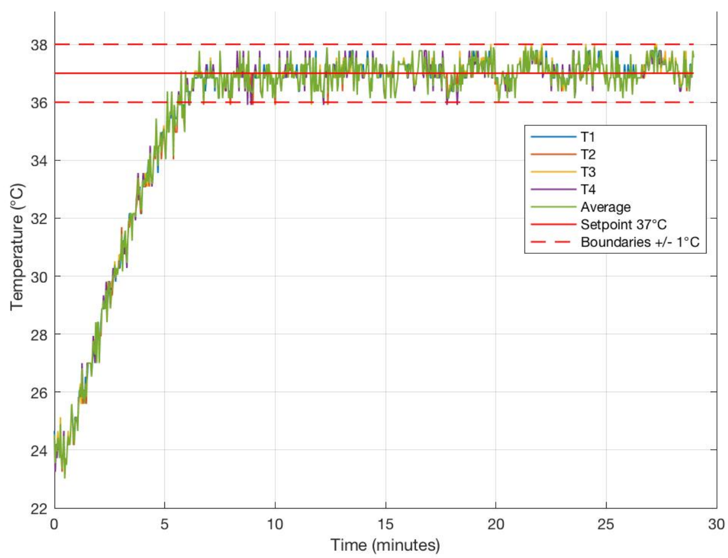


Figure 4.9: Temperature Controller Response Curve

The analog values show that they closely follow the average line, which means that the temperature is consistent along all four temperature sensors. The spikes in the steady state response of this curve come from the noise associated with reading analog temperature measurements. It can be seen that the tuned control system exceeded the specification of ± 2 °C error and a 30 minute ramp up time with a ±1 °C error and a 5 minute heat up time.

4.6 Conclusion

The Thermal Package will maintain the biology inside the BETA Box at a specified temperature over an extended period of time. The average temperature had to remain within ± 2ºC setpoint, and the temperature distribution along the dish bed had to be ± 2ºC between all temperature sensors. The warm up time was well under 30 minutes and the Box had to be safe for students. Numerous computational analyses and 18 experiments were iterated with different heaters, fans, fan placement, and insulation. Specific analyses included a finite element analysis, which indicated the softening points of each of the structure. CFD analysis assisted in the determination of the appropriate fan orientation and size. Experimentation was then run, which verified that the best head distribution would be achieved with a 150 Watt resistive heat bed and four 66 CFM fans placed directly upward and a 6in.^{2} piece of insulation on top of the heatbed for isolation.

Thermal imaging, thermocouples placed inside a covered petri dish, and high precision analog temperature sensors in petri dishes for a coupling effect were all used during the experimentation stage. A proportional integral controller was implemented to reduce the steady state error and the overshoot of the uncontrolled dynamic response of the system. It was found that the BETA Box had a ±1 °C error and a 5-minute heat up time. It should be noted that the smaller error was found without heat addition and voltage differences from the other subsystems in the BETA Box and with the entire system operational and running the BETA Box had a ±2 °C error and a 5 minute heat up time.

5. Electrical

The electrical system required to support the Box included the following:

1. Full RGB Lighting
2. > 5MP images ± 1s frequency
3. Independent power supply

These subsystems are described below.

5.1 Lighting

5.1.1 Design Process

Lighting inside of the Box was necessary for optimized picture quality. The team decided to implement an RGB spectrum that could be controlled through the interface screen and would give uniform lighting throughout the Box without creating extreme glare on the camera. The first iteration of the lighting system was implemented in the initial prototype and featured a simple LED strip zip tied around the inside of the Box and about halfway up the wall, with no lights on the wall with the door. The power cable was run through a hole in the back wall and the strip came with a remote that enabled a full control of the RGB spectrum.

However, this positioning did not function satisfactorily, casting a glare directly into the camera, and this specific LED strip was incompatible with the interface software. Consequentially, a similar LED strip was obtained that could properly interface with the BETA Box and was acquired as a more cost effective alternative. This LED strip was implemented in the second prototype and was mounted to the top inner edges of the Box between the walls and the ceiling via the adhesive on the back of the strips. This design still left the front panel, where the door is located, free of light and required the power cable to be run through the front. It was also found that the adhesive wasn’t strong enough to support the strip at the desired angle and that the wire warped as it went around the perimeter of the Box, preventing uniform lighting throughout the Box. Furthermore, the first LED strip was waterproof, an expensive commodity that was unnecessary for an incubator.

The final light selection was a less expensive, non-waterproof version of the first LED strip that was purchased, and this time it was fastened to the top using 3D printed couplings to ensure that proper orientation would always be achieved. This final design provided no direct glare to the camera and had very little glare affecting the petri dish lids. The LED strip also ran along the entire inside perimeter of the Box creating uniform lighting throughout the Box and was wired through a small hole on the back, which made it very easy to connect with the rest of the BETA Box's power systems. This LED strip featured the entire RGB spectrum controlled through the interface screen on the front of the Box and met all the requirements the team had set out to accomplish.

5.1.2 Requirements

It was decided that the BETA Box's lights needed to reach the full RGB spectrum to allow the widest range of possible experiments to be implemented. It was also a requirement that the Box had uniform lighting and created no glare on the camera, which was central to usability of the BETA Box. The lights also needed to be relatively low cost to effectively reduce the BETA Box's overall cost. The lighting also needed to have full control over the intensity of the lights and the uniformity of the color.

5.1.3 Test

Based on the tests done with image capture of the camera, the final design met the specifications established by the team. The camera received no direct glare, and the entire interior had uniform lighting, which was proven by comparing images of petri dishes in multiple spots on the bed of the BETA Box. Furthermore, the LED strip shone the full spectrum of the RGB scale and was fully customizable through the user interface screen.

5.1.4 Conclusion

The BETA Box's LED strip met all of the requirements set by the team. The chosen LED light strip created no direct glare on the camera and had minimal glare on the reflective surfaces of the petri dishes, which did not impact image capture by the camera. The orientation of the LED strip provided uniform lighting throughout the BETA Box and the strip itself was the most cost effective product that ran full RGB spectrum and was fully customizable through the user interface.

5.2 Imaging

5.2.1 Design Process

A trade-off analysis was made between a camera that could take images of all nine petri dishes or one that could take images of each petri dish and then be moved. Multiple cameras were evaluated for their pixilation, lens, and cost. A 5MP wide-angle lens camera was tested to reduce the overall size of the Box, but it was found that the wide-angle lens had large distortion effects that created blurry images. As a result, a single 8MP camera supported by the Raspberry Pi was implemented.

5.2.2 Requirements

For the imaging subsystem the requirements did not come directly from the customer. As a result, specifications were created and quantified by establishing a minimum pixel count. Table 5.1 summarizes the requirements of the camera system.

Table 5.1 Camera system requirements

Function	Requirement
Take images of biology	Capture images of all 9 petri dishes with ± 1s delay between images
High resolution images	> 5MP Camera

5.2.3 Analysis

To have an autonomous documentation system, a camera was sized to be 8MP with a 3280 x 2464 pixel size and an adjustable focal length. A normal lens was implemented so there was no distortion between images. The camera calculations were computed using a working distance of 12-inches. The sensor resolution was provided by the manufacturer, and the smallest feature resolvable was found by Equation 1 below.

$$\text{Sensor Resolution} = \text{Image Resolution} = 2(\text{Field of View (FOV)} / \text{Smallest Feature}) \quad (1)$$

The field of view was found using Equation 2. The focal length of the camera and the sensor size were provided by the camera specifications.

$$\text{Focal Length} \times \text{FOV} = \text{Sensor Size} \times \text{Working Distance} \quad (2)$$

Figure 5.1 shows how the camera views the sample.

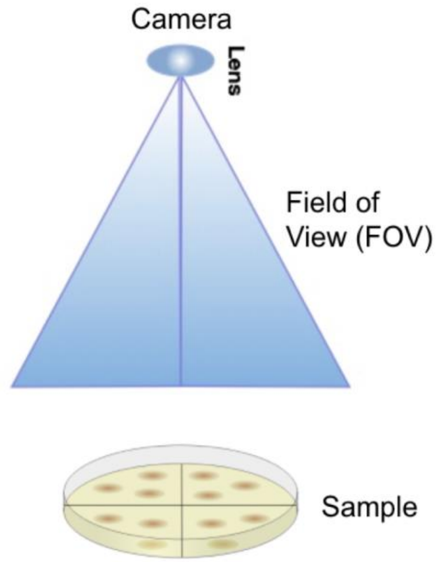


Figure 5.1: Cameral Analysis

It was determined from Equations 1 and 2 with a 12 inch working distance and an 8MP camera that the smallest resolvable feature would be 66 μm. This allows for imaging down to the cellular level, since a single cell can range from 15 to 40 μm in length and each array droplet has approximately 100,000 cells. The current camera meets the specifications needed for proper imaging analysis and determining color change in the droplets for the use in experiments with the ImageJ analysis tool.

5.2.4 Testing

An image analysis tool was used to count the number of pixels in each of the images that were taken to determine if the image of the sample had the proper number of pixels. Other tests that were done included sizing tests to determine the appropriate working distance and verify that the calculations done in the analysis section matched physical testing for the camera.

5.2.5 Conclusion

It was found that an 8MP camera would produce images of all 9 petri dishes in the incubation Box with a working distance of 12 inches. The images had an acceptable resolution to be fully analyzed using the ImageJ analysis tool. Figure 5.2 shows a sample of images taken using the final camera design.

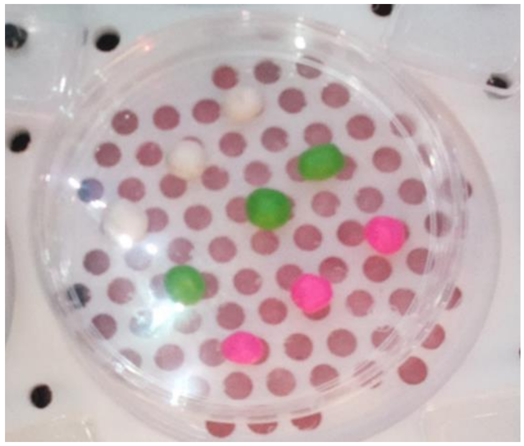


Figure 5.2: Cameral Imaging Results

The frequency of the imaging was handled in the software section and produced images of all 9 petri dishes independently under the required ± 1s delay between images.

5.3 Power

5.3.1 Design Process

Once the electronics were mapped out and organized, power requirements were decided based on the specified electric components. Each component has a required voltage and amperage to run properly, which affected the power requirements. Once the requirements were determined, a low cost power supply was purchased that met the total power specification. The power supply was mounted in the back of the Box, behind the inner wall of the incubator. In addition, our electronics setup required both 5 volts and 12 volts, so a buck boost circuit was researched and implemented, so that the entire Box could run off a single power supply unit. A buck boost circuit, seen in Figures 5.3, and 5.4, is a small board that regulates the voltage down from 12 volts to 5 volts by turning the power on and off, in a similar manner to pulse width modulation (PWM) works for motors.

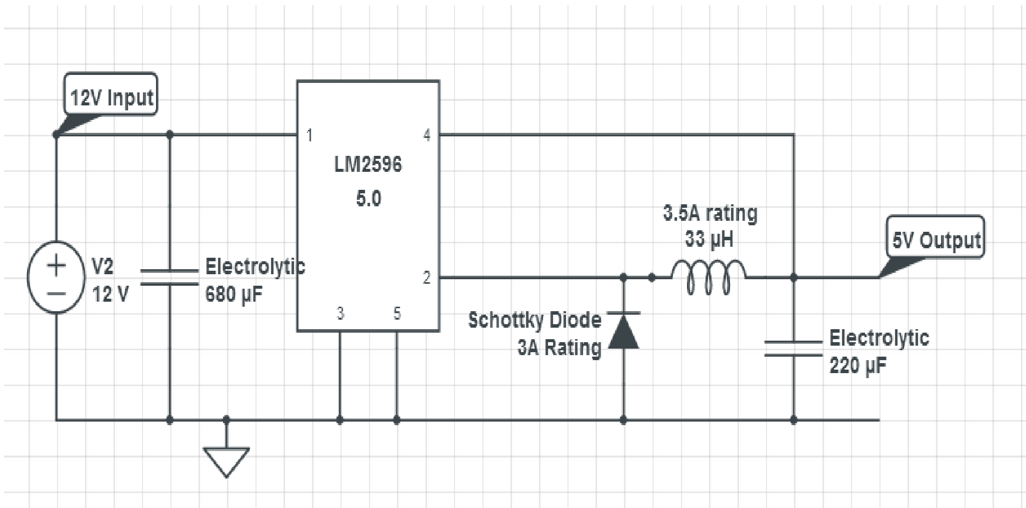


Figure 5.3: Buck Boost Schematic

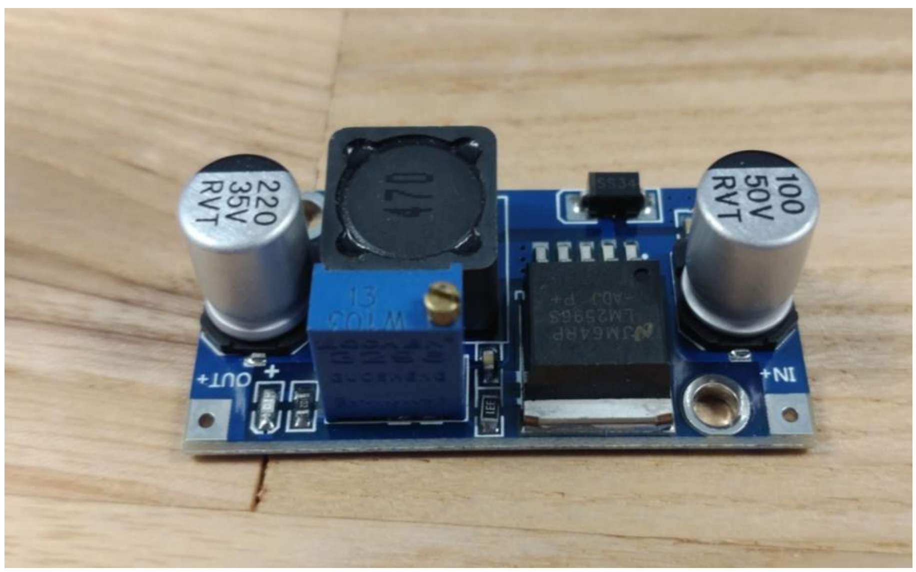


Figure 5.4: Buck Boost Controller

This component however, is all hardware and doesn't require a microcontroller to regulate the power. In addition, all high power circuits were controlled via a transistor, to protect against electricity backflowing into the controllers.

5.3.2 Requirements

The following table indicates the power requirements for the BETA Box.

Table 5.2 Power Subsystem Requirements

Electronics Component	Max Amperage Requirement	Voltage Requirement
Raspberry Pi	1A	5V
Arduino	200mA	5V
Heat Bed	12.5A	12V
Relays	500mA	5V
LED light strip	12A	12V

5.3.3 Testing

The entire electronics subsystem was run under maximum stress on the system to “burn-in” the electronics. The setup was run for 1 hour to test that all the electronics components were receiving the correct power requirements.

5.3.4 Conclusion

The power subsystem, though seemingly simple, required modification of the standard power supply. In order to incorporate all the electronics needed to allow the Box full functionality, a power distribution system ensured that each of the electronics components were receiving the correct voltage and amperage for safe and extended operation.

6. Computing System

6.1 Design Process

When considering the computing requirements of this project, it was important to consider the design constraints as discussed with our customer. These requirements, outlined in Table 6.1 below, list the most important requirements in order of highest to lowest priority.

Table 6.1. Computing Requirements

Requirements	Design Constraints
● Support timed image capture ● Control lighting and temperature ● Download captured image ● Safe ● User friendly and intuitive	● Low-cost ● Cannot be connected to a desktop or laptop computer

The system must support timed image capture so that students can perform image analysis and see the changes in their experiment. The incubator must also have full lighting and temperature control to support various kinds of experiments. Finally, there must be a way for users to download the images captured during their experiment run.

In regards to non-functional requirements, it is important to remember that this device will be used in a high school setting, so the incubator must be safe. This means there are redundant safety systems that can safely shutdown the box incase of failure. The incubator must also be controlled in a user friendly and intuitive way as to appeal to the high school students. Additional design constraints include keeping the unit low cost and ensuring the incubator can operate independently of a laptop or desktop computer thus eliminating the use of special software and additional cost for the school.

6.2 System Sequence Diagrams

Figure 6.1 outlines the typical use cases a user will experience while using the incubator.

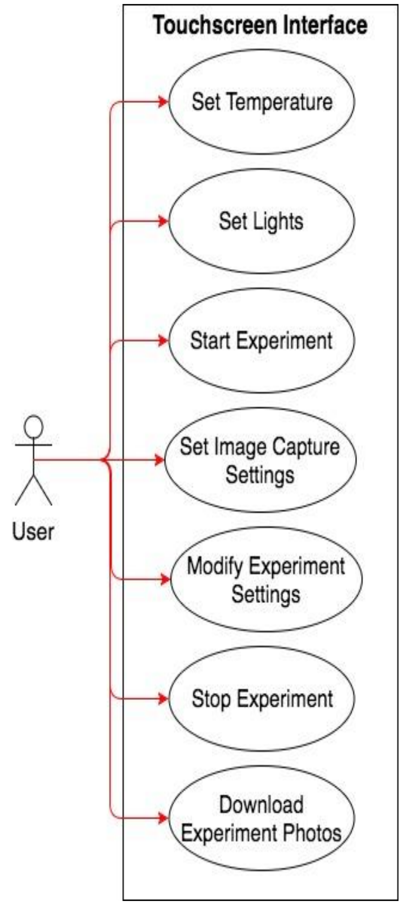


Figure 6.1: Use Case Diagram

The figure above shows the typical order in which a user would participate in each experience. Through the touch screen interface, a user can set the temperature and lighting levels of the entire Box. A user can then start an experiment and specify the image capture rate and duration of that experiment. Each petri dish can have its own unique image settings. At any point during an experiment run, the user should be able to modify the experiment settings, such as image capture rate. The user should be able to stop and cancel an experiment at any point. Finally, the user can download the experiment photos after an experiment is completed.

Figure 6.2 outlines the activity flow a user will follow while operating the incubator Box.

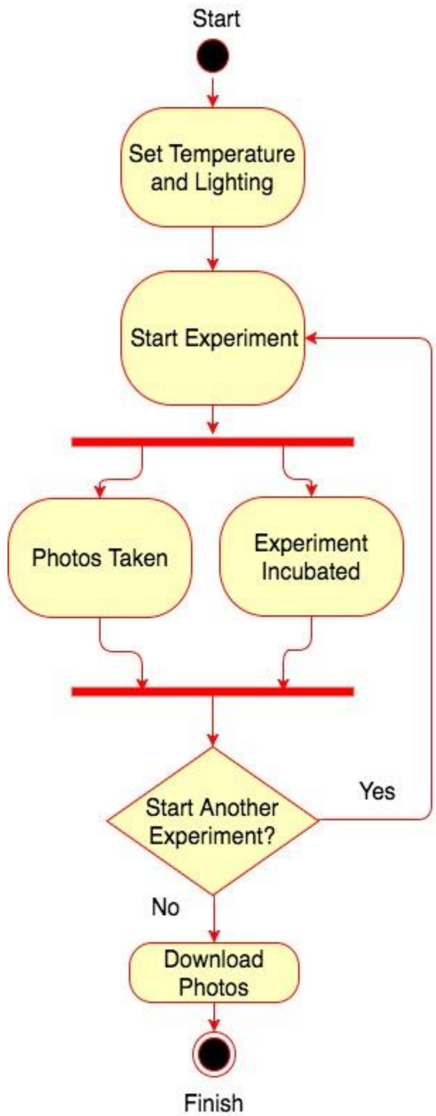


Figure 6.2: Activity Diagram

First, the user will set the overall temperature and lighting levels in the Box. Then, the user will start an experiment by specifying the experiment name, the interval at which a photo should be captured, and the duration of the experiment. Once the experiment is started, the Box will automatically begin taking photos at the specified interval and maintain the desired temperature and lighting incubation levels. At that point, another experiment can be added in a similar fashion for a total of nine simultaneous experiments. Once the desired experiment is completed, the user can then download the photos captured during the incubation of that experiment.

6.3 Architectural Design and Conceptual Model

Figure 6.3 outlines the technologies used in the BETA Box.

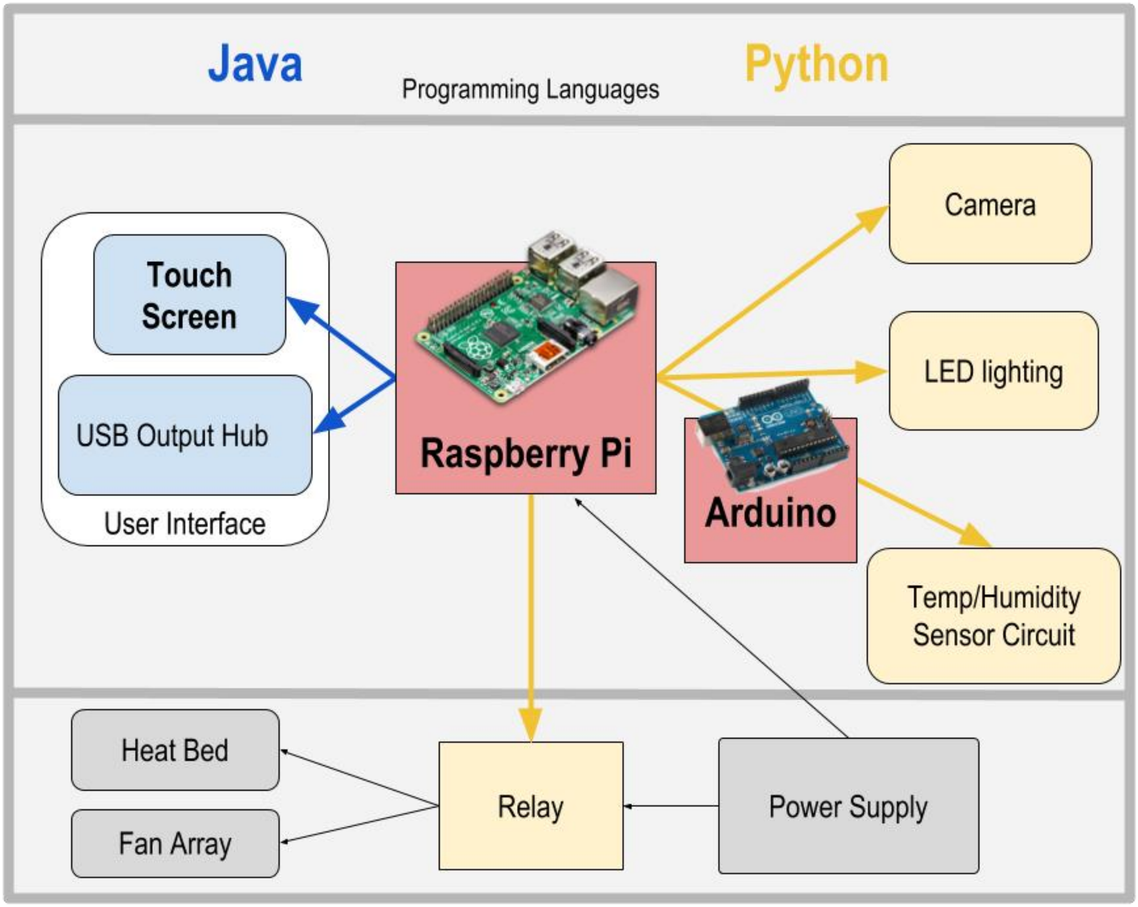


Figure 6.3: System Overview

Components in blue are part of the front-end graphical user interface, or GUI, and are coded in Java. Yellow components are controlled in the back end through Python. The main computer is the Raspberry Pi. Java worked well as the front end due to its extensive graphic libraries for the GUI and its ease of multithreading. However, the Raspberry Pi has the most hardware support with Python. So we created a front-end / backend architecture to allow the GUI to control light Python scripts that would manipulate the hardware components. Because of the high number of general-purpose inputs and outputs our low cost touch screen required, we were forced to add a second microprocessor. As described in detail in section 6.5 below, an Arduino was the optimal choice for adding more inputs and native analogue sensing (something the Raspberry Pi does not have). This allowed the Box to have higher precision analog temperature and humidity sensors. To accomplish this, we created a TCP/IP-like serial communication protocol to communicate between the devices. This allowed the Arduino to have real time sensing while the Raspberry Pi could control the temperature PID controller and lighting using its powerful multi threading capabilities. All code for both the Raspberry Pi and Arduino in the BETA Box can be found in Appendix P.

6.4 Technologies Used and Rationale

Programming languages:

- Java: A class based, object oriented computer programming language that is used for general purpose computing. Java is commonly found in projects around the world.
- Python: A high level general-purpose programming language that has strong hardware support on the Raspberry Pi.

Version Control System:

- Git: A commonly used version control system. Git allows distributed workflow with high data integrity.

Hardware:

- Touch screen: A low cost 5in. touch screen to drive an intuitive GUI.
- Arduino: An inexpensive real time system microprocessor platform with integrated analog support and serial communication abilities.
- Raspberry Pi: A small, commonly used open-source computer with large community support. The Raspberry Pi has General Purpose Input/Output (GPIO) pins for interfacing with other devices as well as built in WIFI, Bluetooth, GPU with HDMI output, and USB support.

Tables 6.2 and 6.3 below highlights the benefits and drawbacks of both the Raspberry Pi and Arduino and systems and show the rationale for including both a Raspberry Pi and Arduino in Prototype 3 of the BETA Box.

Table 6.2. Software Comparison

	Raspberry Pi + Touch screen	Arduino Mega + Ramps
Benefits	● GPU for much higher resolution UIs ● Supports USB drives ● General Purpose Input/Output (GPIO) pins can control simple devices- light and temperature controls ● Directly control camera module	● Modular design is easy to replace burnt chips ● Natively supports LCD screen + buttons for navigation ● Supports SD card ● Much cheaper than all other options ● Well supported development community
Drawbacks	● Requires expensive touch LCD or external computer to operate ● Cost: Pi + touch screen / external computer + camera module added to printer ● Possible print failure caused by Pi hanging	● Limited to LCD display (not a modern touch LCD screen) ● Does not natively support network access

Table 6.3. Design Rationale For Using Raspberry Pi and Arduino

Raspberry Pi	Arduino
● Integrated GPU for Quality Graphics	● Allows More Inputs
● Full Operating System	● Real-time System
● Multi-Thread support	● Native Analog Hardware
● Does Not Require Use of PC	● Low-cost

As highlighted in Table 6.3, the team used a Raspberry Pi for the main computer because of its onboard hardware and low price. The Pi has an integrated GPU, which allows for high quality graphics and animations on the touch screen. Because the Pi runs a full Linux-like operating system, it has built in support for the camera and USB devices like a flash drive which is crucial for offloading experiment images. The Pi has four compute cores, which allows the multithreading performance we need to manage the experiment settings and take photos simultaneously. Finally, the Pi can run independently of any other PC, which was one of the design constraints. It also gives the platform room to grow with wireless capabilities. However, with so many devices to interface with, the team quickly ran out of usable inputs on the Pi. To add more inputs, an Arduino Uno was added. The Arduino is real time system that has built in analog hardware, which is something the Raspberry Pi does not have. This allowed for the creation of a distributed system where the Arduino could more precisely measure temperatures on an interval and send it back to the Pi for control. Arduinos are also low cost which kept the overall cost of the system to a minimum. In fact, the Arduino and five sensors were about the same price as one digital sensor for the Pi.

6.5 Conclusion

Because of the architecture implemented, the team successfully created a system that achieved all of the requirements. The BETA Box can support full RGB color while simultaneously maintaining temperature internally. Up to nine simultaneous experiments can be incubated. The Box can take photos of all experiments at a specified time interval for the duration of the experiment, and can offload the images onto a USB device. For safety, the system is utilizes redundant circuitry that shuts down the electrical components if there is a temperature or wiring fault.

7. User Interface

The Box has a touch screen interface that allows users to monitor, control, and interact with the other subsystems of the Box. The user interface is a screen through which the user can configure the Box's settings for the current experiment, begin and end experiments, and download images that are captured during the experiment. The user interface performs the following:

1. Runs off of a microcomputer.
2. Allows users to choose custom light, temperature, and image capture settings.
3. Organizes all images from a single experiment into a folder per petri dish so that they each can be exported to USB.

7.1 Design Process

7.1.1 Touchscreen Design Rational

A touchscreen interface was the most effective solution for the simplest user experience for both high school students and instructors. A touchscreen interface provides better image quality and intuitive interaction for users. Users are familiar with touchscreens because of the popularity of touchscreen smartphones and tablets.

The user interface is optimized for a touch screen experience. Since users will be using fingertips to navigate the interface, instead of a mouse cursor or stylus, all functions are tied to simple and prominent buttons. The interface was kept as simple as possible, only showing options and information that are necessary. A simpler interface allows users to more easily navigate the system. In addition, since users will be high school students, limiting the options they have on each screen reduces confusion on what to do next and lowers the chance of error due to unintentional clicks of other buttons.

The navigation bar on the left side of the display is intentionally ordered from top to bottom- Dashboard, Export Images, then Settings tabs. Users should only need to visit the settings tab once, before the experiment begins to set up the Box. Once the Box is set up, users should only need to visit the top two tabs, Dashboard and Export Images, for running the experiments.

There is minimal direct user input of values to reduce the need for on-screen keyboards that can be hard to navigate on a small screen. Instead, numbers are increased or decreased with up and down arrows, images, text, and color highlighting are used for signaling if certain dishes are selected or in use, and sliders are used in configuration. Reducing direct user input reduces the likelihood of human error.

The Box allows users to input their own settings to provide more flexibility for the user to customize experiments and create their own experiments. Since there are only a few criteria to set— light, temperature, image capture, and duration of experiment— it does not add too much overhead in setup time.

7.1.2 Graphical User Interface Iterations

The user interface went through three main iterations. First, during fall quarter, paper mockups were created and shared with the client to walk through the general flow and layout of the interface, as shown in Figure 7.1.

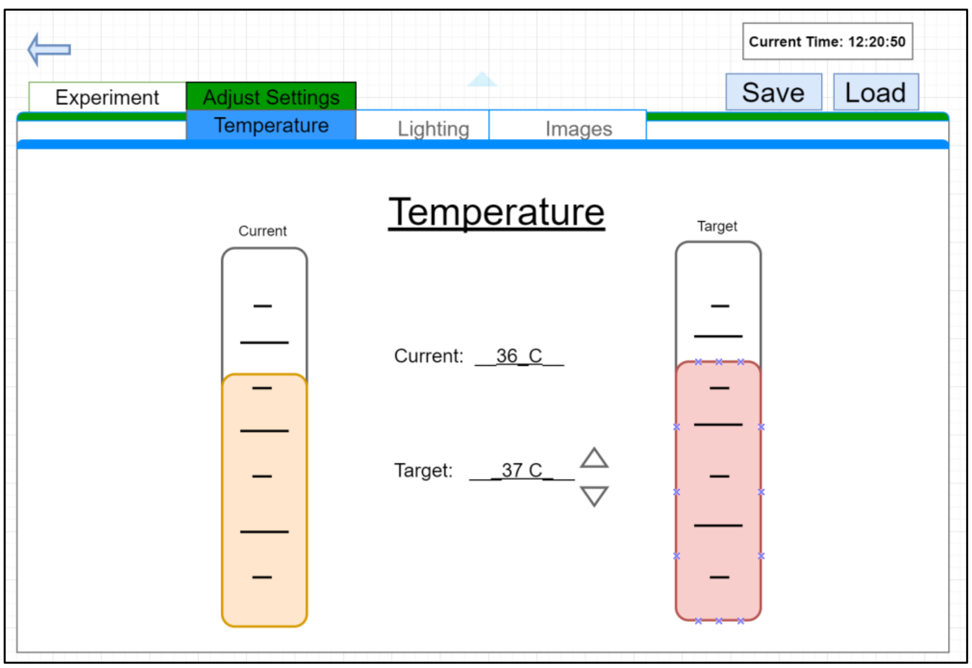


Figure 7.1: Paper Mockup

Next, Java was used to implement the interface to be displayed on the touch screen, shown in Figure 7.2.

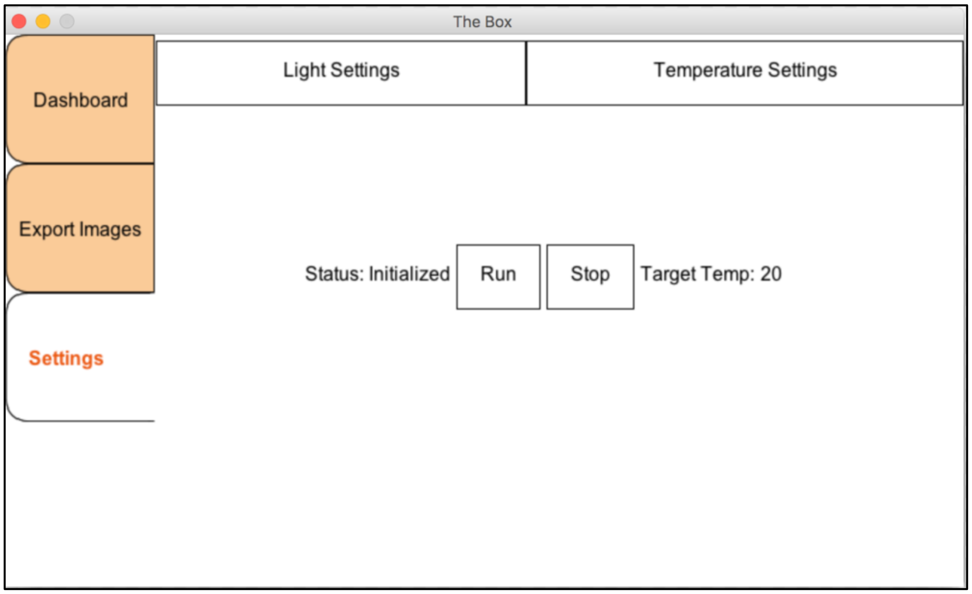


Figure 7.2: Version 1 of GUI

Feedback was gathered from this version through running a usability test with high school students. Using that feedback, we came up with our final prototype implementation of the interface, shown in Figure 7.3.

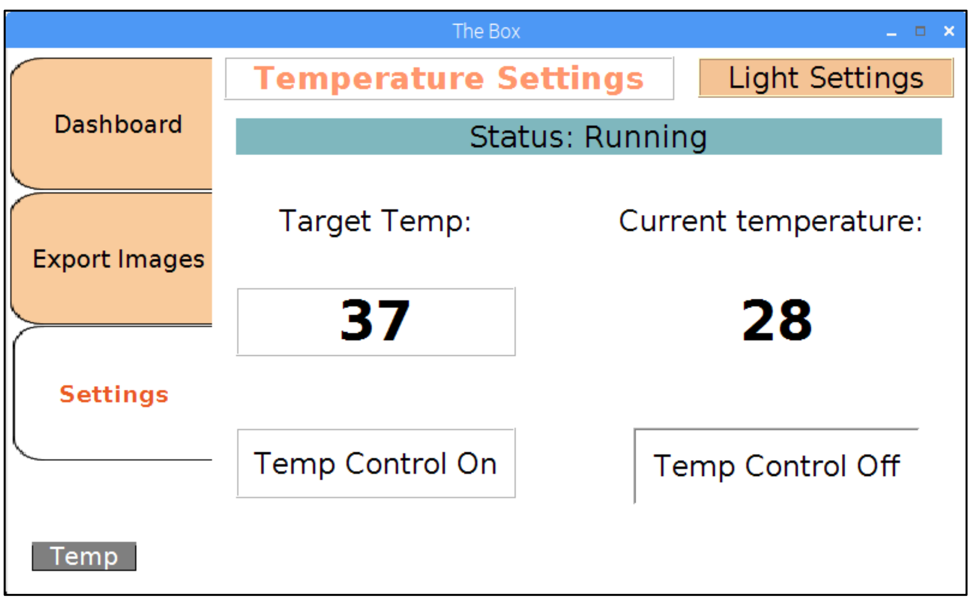


Figure 7.3 Final GUI

7.2 UML Class Diagram

The team took advantage of the object-oriented nature of Java when writing the code for the user interface. There is one main GUI class that utilizes different panels customized for each feature of the Box. Using the inheritance properties of Java, each panel could reuse the code of a base panel class and add additional features that were specific to the panel. For example, Figure 7.4 shows how the status panel also uses a custom DishConfig object to configure each dish.

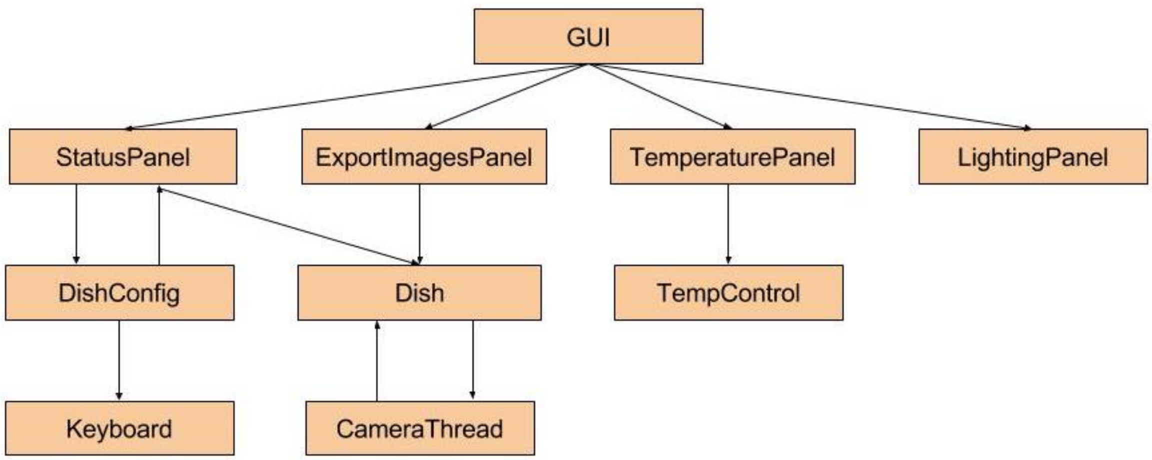


Figure 7.4: UML Class Diagram for Touch Screen UI Java Program

Only the StatusPanel needs that object, so the code is encapsulated to that class. In this way, the team could keep code reusability high, and also keep each component in its own object for security and organization. Because each object inherits from a base class, the team could also make use of Java's polymorphism so that panels can be used interchangeably in the GUI. In this way, the team created custom panels for each subsystem and bring them together in the GUI. The code for the GUI can be found in Appendix P.

7.3 Custom Java Look and Feel

The graphical user interface is optimized for the touch screen with a custom designed Java Look and Feel using the Synth Look and Feel package. The Synth L&F package provides the behavior of the dynamic components, such as menus, dropdowns, and buttons. An external XML file was created to define the look of all of the elements on the screens. This included color, shape, layout, and typefaces of the elements. Thus, a custom skin was created to make the graphical user interface more elegant and usable.

7.4 User Interface Overview

The touch screen interface is split into three main tabs shown in the navigation bar on the left side of the screen: Dashboard, Export Images, and Settings, shown in the following figure.

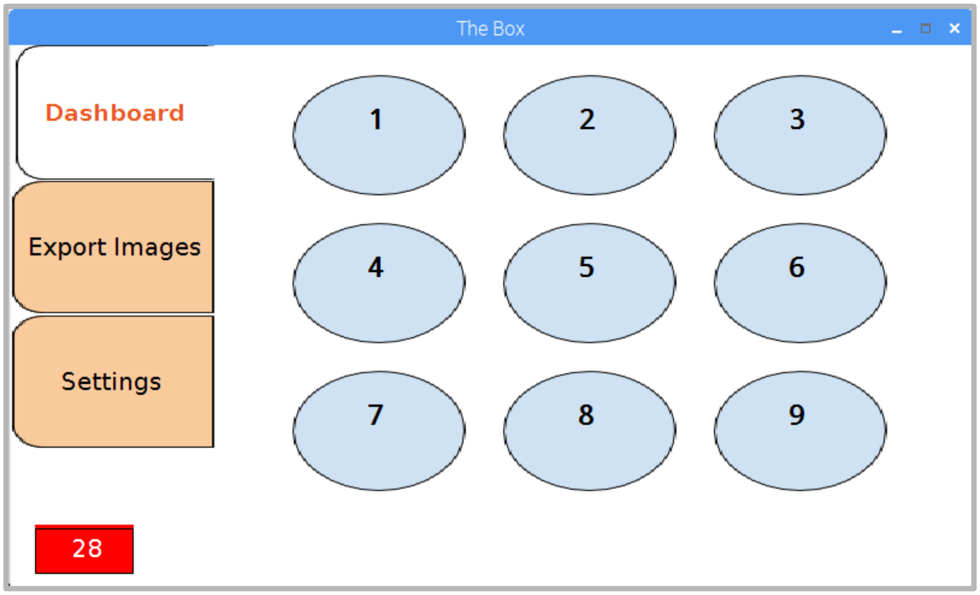


Figure 7.5: Touch screen Dashboard

The Dashboard tab is where current status of the Box is shown. From here, users can see the current temperature and lighting of the Box from the colored information Box in the lower left hand corner of the screen. This screen is also where students will initiate, monitor, and end their experiments.

The Export Images tab is where users go after the experiment is finished to export the images captured during their experiment onto their USB drive.

The Settings tab is where users set up the environment of the Box before the experiments are ran. In this tab, the user can configure the temperature and lighting color inside of the Box.

User Interface Walkthrough

The following screenshots walk through a typical sequence of steps a user would follow to run a complete experiment using the interface.

1. Configure Settings

   The first thing a user will do after powering on the Box is to configure the temperature and lighting settings of the Box to prepare for the experiments to be run. The user can do so from the Settings tab on the left navigation bar. In the Settings panel, there are two subpanels, Temperature Settings and Light Settings.

   a. Preheat Box

      The user will set the target temperature by clicking on the Box under the "Target Temp" label as seen in Figure 7.6.

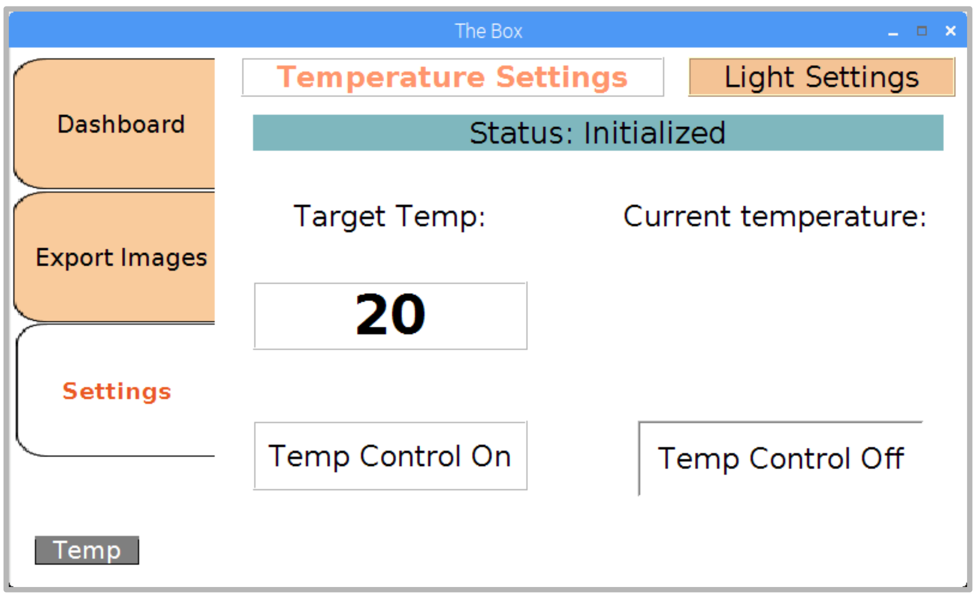


Figure 7.6: Temperature Settings Screen

A numerical keyboard will pop up, Figure 7.7, and the user can input their desired target temperature for the given experiment.

Once the temperature is set, the user must click the button labeled "Temp Control On" to turn the heater on and begin heating the Box. Once the button is pressed, the current temperature field will show the Box's current temperature, Figure 7.8.

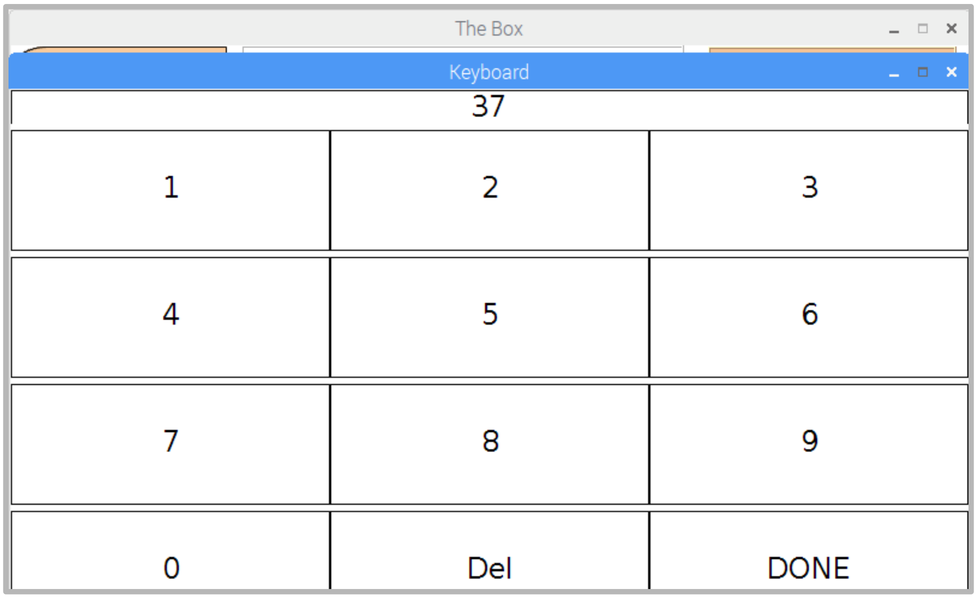


Figure 7.7: Numerical Keyboard for Setting Target Temperature

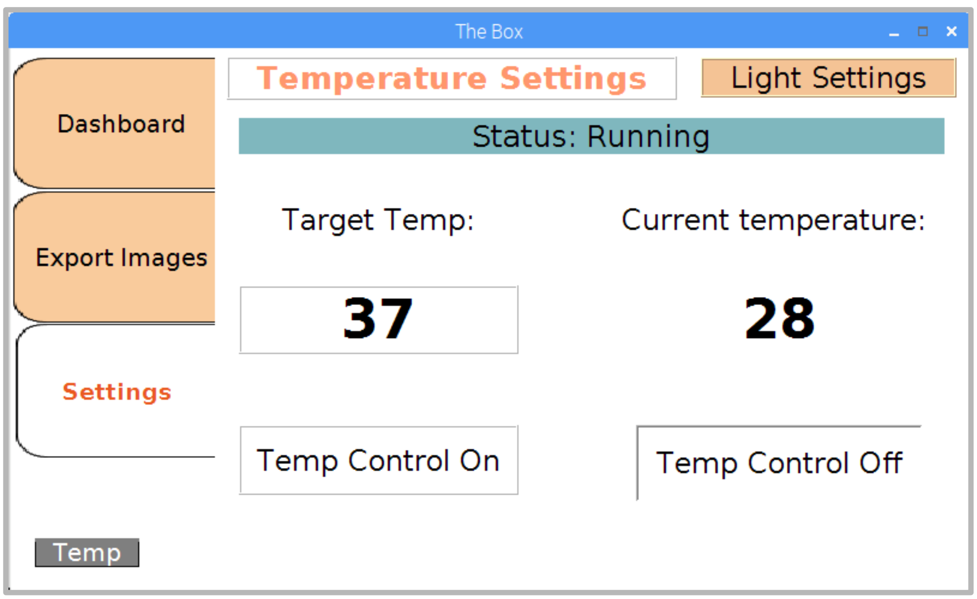


Figure 7.8: Temperature Panel When Heating Unit is Turned On

b) Set lights

From the Light Settings tab, the user can set the desired lighting color of the Box that the particular experiment calls for. The user is able to choose from six preset color values, as seen in Figure 7.9.

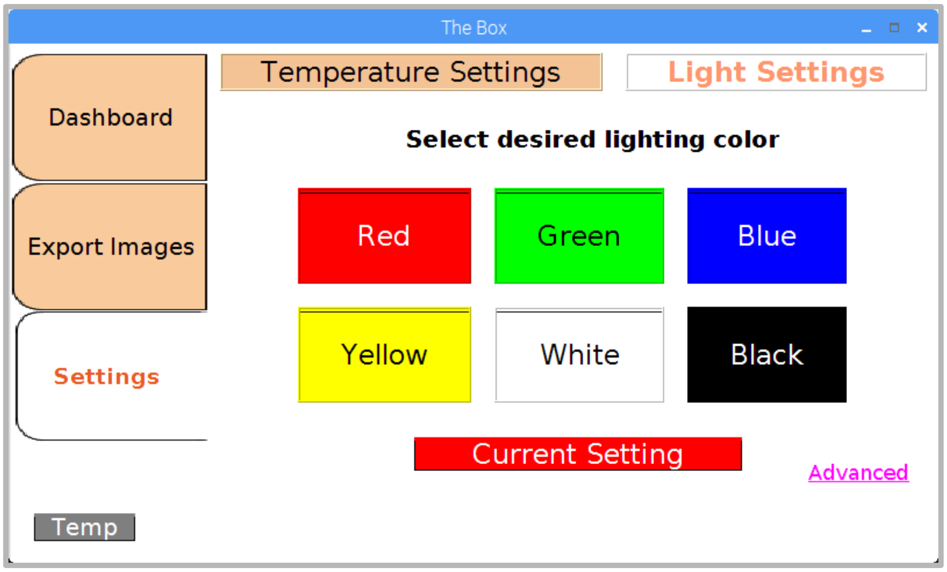


Figure 7.9: Preset Color Values for LED Lights

If the experiment calls for a color that is not a preset value, the user can set a custom RGB color for the lights by clicking the Advanced button on the lower right corner of the screen, Figure 7.10.

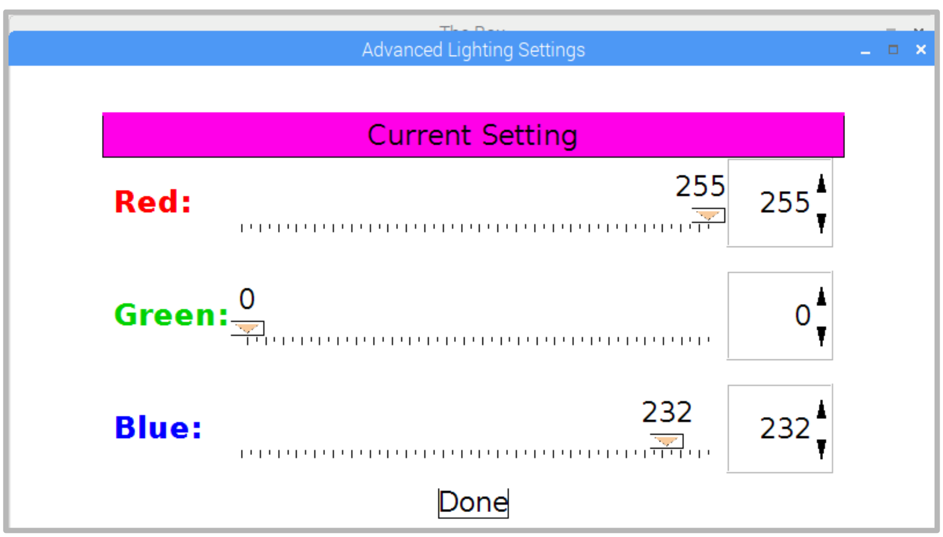


Figure 7.10: RGB Advanced Light Settings

2. Begin Experiment

Once the Box is up to temperature and the lights are all set, the students can begin placing their dishes with the biomaterial samples into the Box to begin incubating and/or capturing images.

a. Place dish in the Box

The student will first place their petri dish into an empty slot on the dish bed. We recommend that the students fill the Box from the back of the Box to the front, to minimize disturbance of the experiment as other students add in or remove their dishes from the Box.

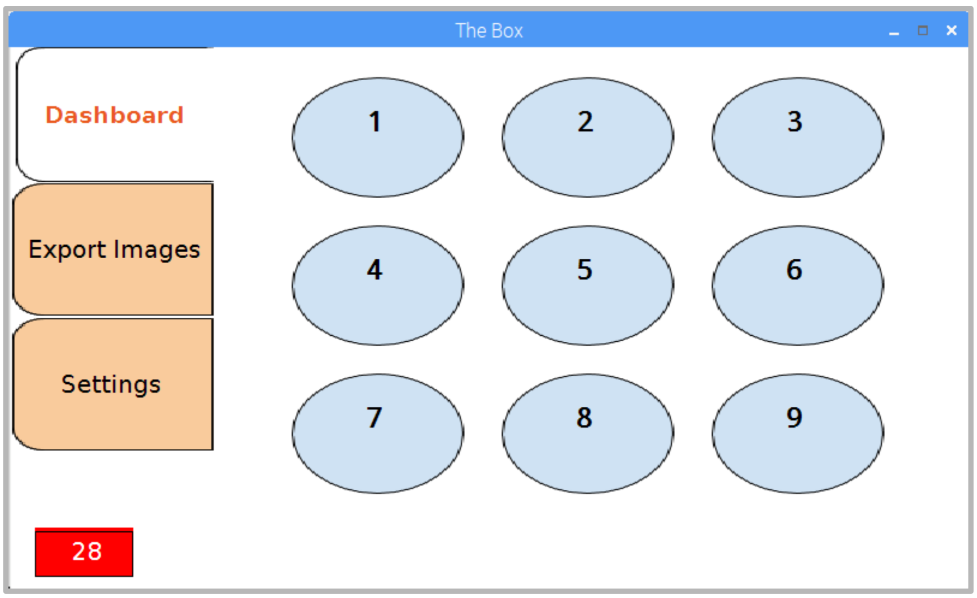


Figure 7.11: Dishes on Screen Corresponding to Box

b. Configure Image Capture

Once the physical dish is inside of the Box and the door is closed, the student will click on the corresponding dish on the screen. The dishes on the screen map directly to the slots on the dish bed inside of the Box.

Clicking the dish will bring up a configuration screen for the student to enter information for image capture:

- File name: the name of the user, which the images will be saved under

- Image frequency: the time interval between photos
- Total time of the experiment: the total amount of time in which images will be captured

In Figure 7.12, the experiment shows that it calls for images to be taken every 10 seconds for a minute. Thus, the experimenter should expect six images at the end of their experiment.

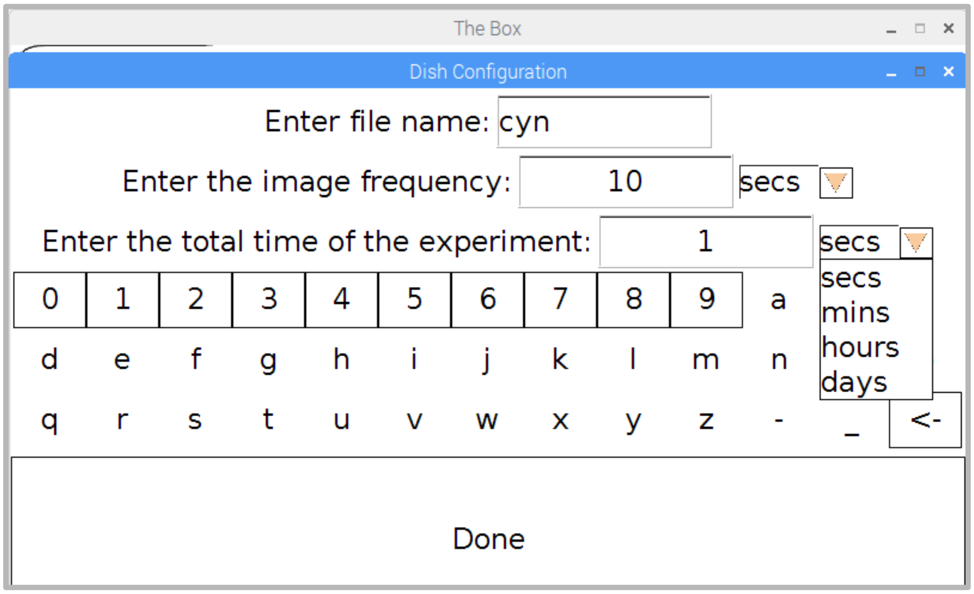


Figure 7.12: Image Capture Configuration for the Dish

After the settings are complete, the status of the dish is shown on the dashboard.

3. Wait for Experiment to Finish
   The dashboard will update as images are taken for each experiment to let users know how many images have been captured for their experiment. During this time, other students can add their dish into the Box aznd their images will be captured in parallel, Figures 7.14 and 7.15.

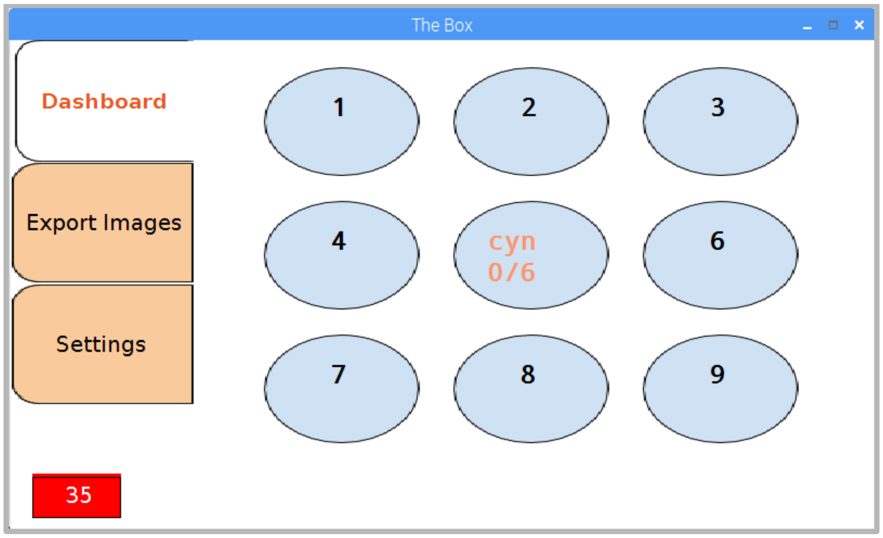


Figure 7.13: Dashboard Showing Running Experiment

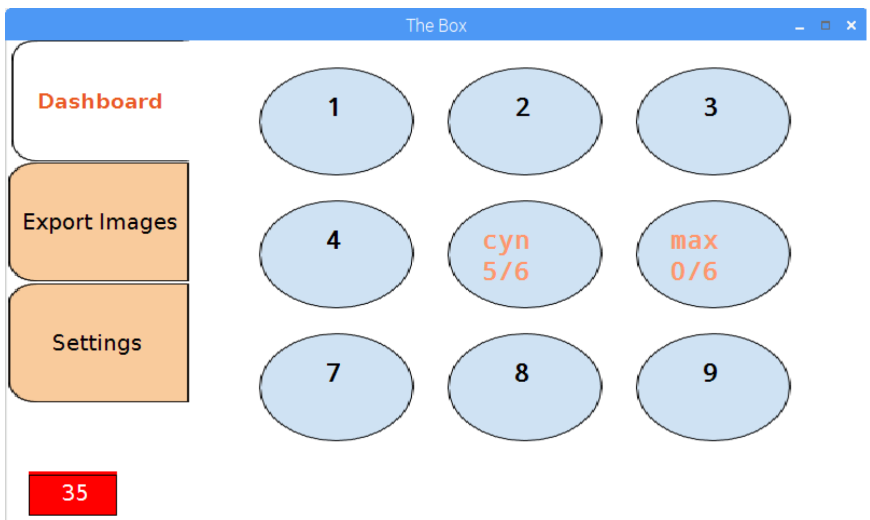


Figure 7.14: Simultaneous Experiments

4. Export Images

   When all the images for a dish have been captured, and its display turns green, the students can export their images. They plug a USB drive into one of the ports on the left side of the Box. They can then go to the Export Images tab, Figure 7.16, select their experiment and USB drive from the corresponding drop down menus, and export the images. The images will be organized in a folder with the name that was specified.

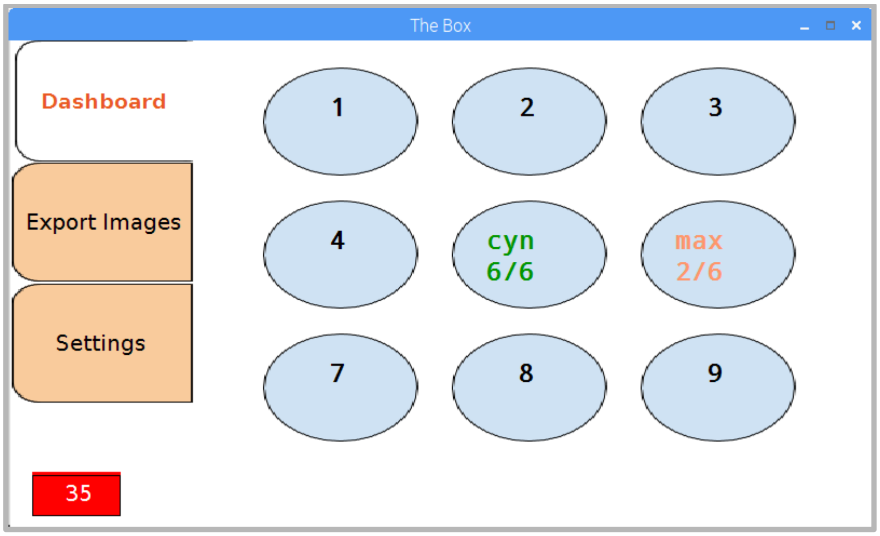


Figure 7.15: End of First Experiment

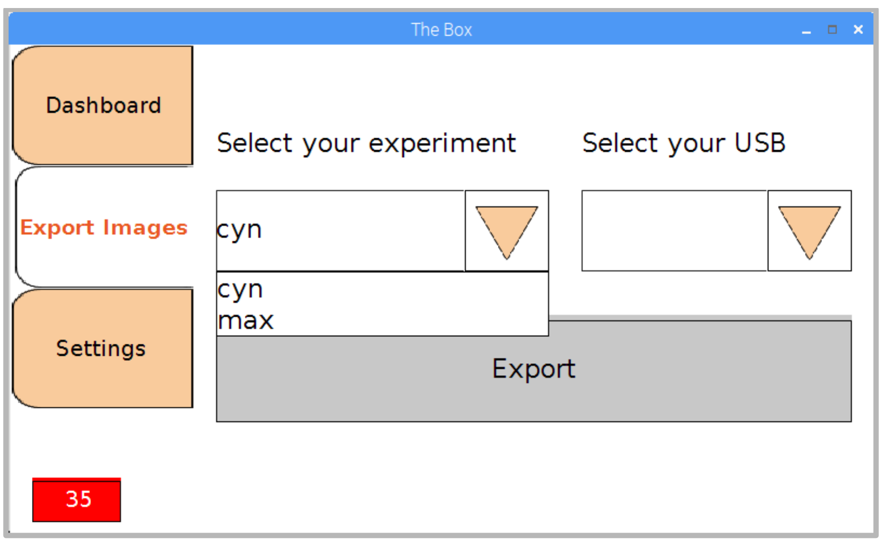


Figure 7.16: Export of Images to USB

5. End Experiment

   After the images are exported, the student must return to the dashboard, Figure 7.17, and click on their dish again to clear it.

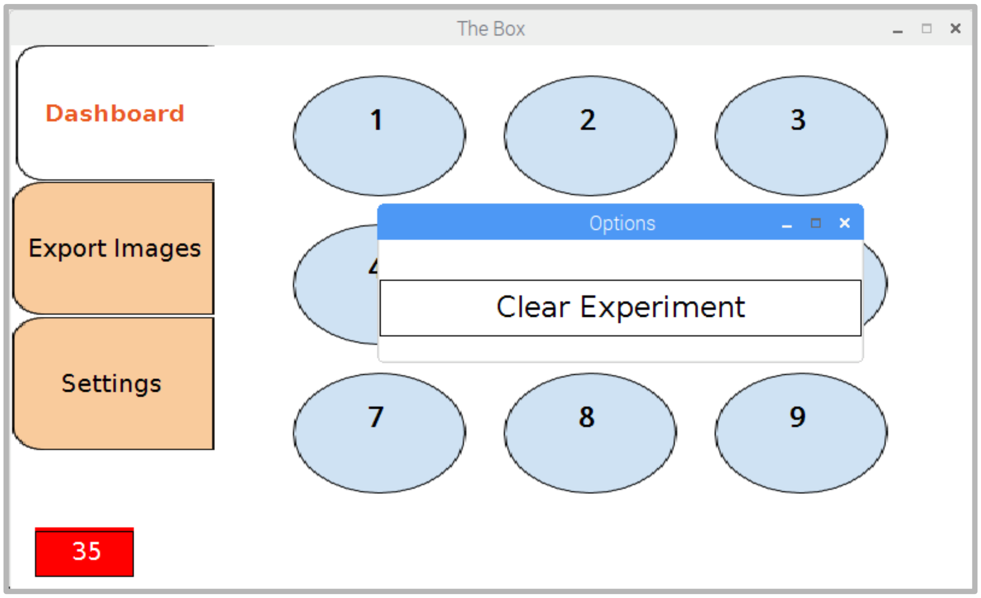


Figure 7.17: Clearing Experiment

Once the dish is cleared on the screen, the student can remove their experiment dish from the inside of the Box. Now, in Figure 7.18, the space can be used for the next student if necessary.

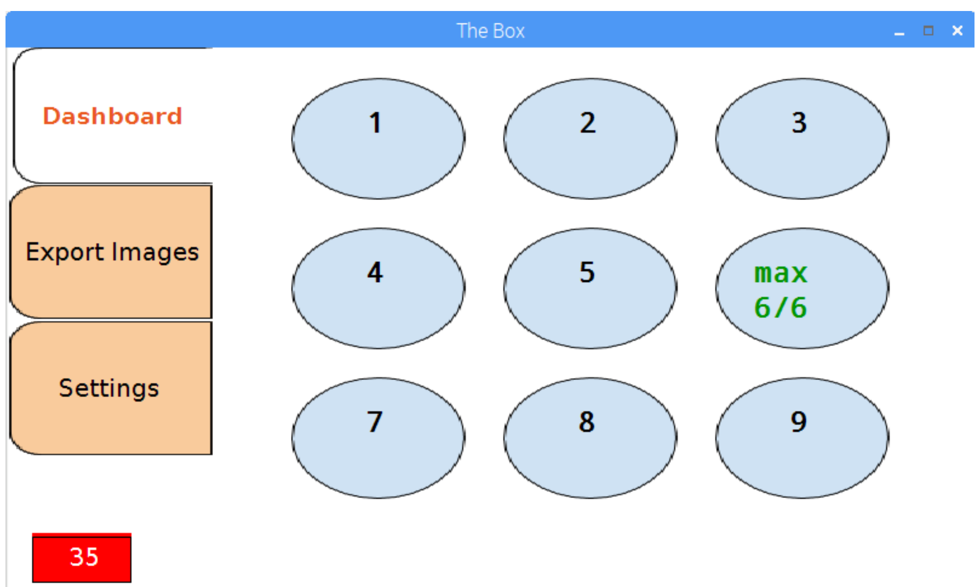


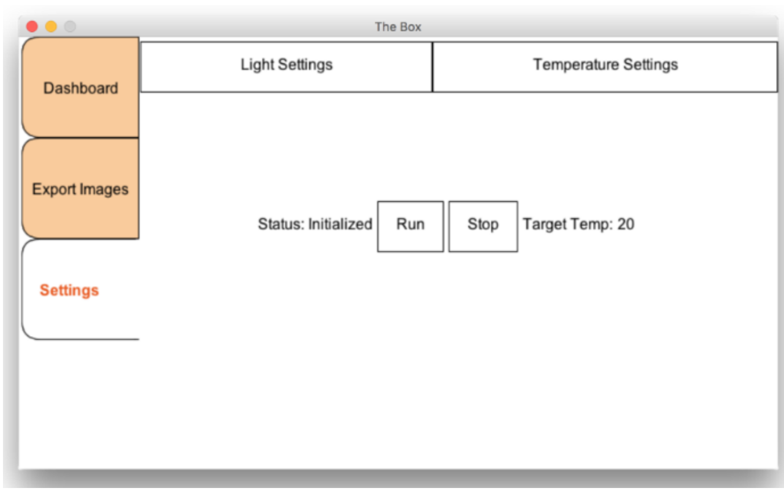
Figure 7.18: Dashboard After Experiment is Cleared

7.5 Usability Test Procedure and Results

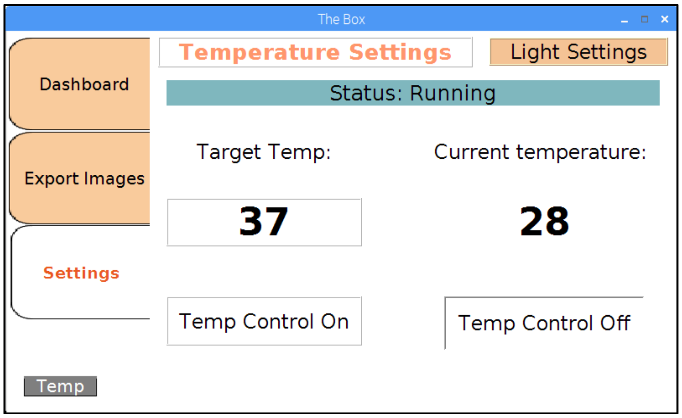
A usability test was run to gather feedback about the first version of the touch screen user interface, as shown in Figure 7.2. The usability test was hosted at an on campus engineering outreach event, early spring quarter. The goal of the test was to see how students interacted with our Box - what was intuitive for them, and what was not. The students were given directions to perform a series of tasks with the Box as if they were running an experiment and then were asked to rate their confidence levels in performing the tasks on a scale from 1 (clueless) to 5 (very confident). The twenty students worked in pairs, yielding ten sets of responses. Details from this test can be found in Appendix O.

Overall, the feedback from the test was positive. From all the responses, we averaged a ⅘ in terms of confidence level that the students felt when completing the tasks. Following is a summary of the main points of feedback that we received from the students:

1. The Box's physical dish bed did not need to be numbered because it was intuitive to students how the location on the screen corresponded to locations in the Box.
2. Students wanted a feature that automatically returned the screen to the home dashboard screen after a certain amount of time of inactivity on the other screens
3. Students needed more visual confirmation on the screen when tasks were completed. For example, when the temperature was set, there was no visual feedback on the screen that indicated that the heater turned on as can be seen in Figure 7.19.



(a)



(b)

Figure 7.19: Rev 1 Temperature Setting Screen Design(a) and Rev 2 After User Input(b)

4. Students wanted the touch screen interface to be able to be maneuvered solely with their fingers. Since the touch screen is so small, it came with a stylus to help manipulate the small screen, which the students were given to use. Students were very turned off by the idea of needing to use the stylus, so we needed to make some buttons bigger to eliminate the need for the stylus.
5. Students had constructive suggestions about the visual styling of components on the screen, such as how the buttons and dropdown menus looked. For some students, the function of the buttons was not intuitive based on their styling.

After evaluating the feedback and our time constraints, we incorporated points 3-5 into our final prototype of the touch screen. Figure 7.20 shows the final user interface in use.

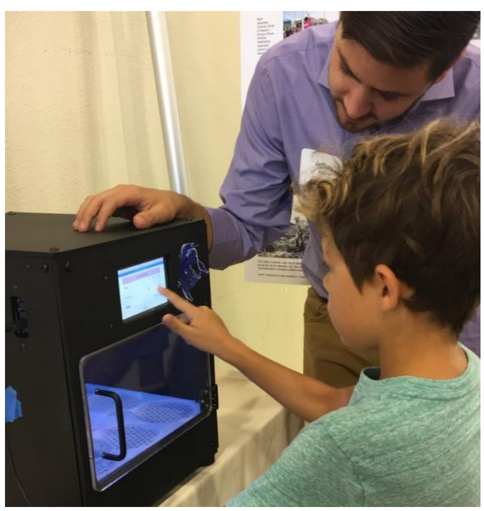


Figure 7.20: Final BETA Box Demo at Research with a Mission Presentation

7.6 Conclusion

The touch screen user interface facilitates a simple and intuitive process for teachers and students to interact with the Box and run their experiments. It was shown that the touch screen was intuitive to the user and the design of the user interface helped facilitate the observation and incubation of biology experiments. The design averaged a 4/5 on a customer usability survey and refinements have since been made to address the key critiques of that survey.

8. Integration

All subsystems were integrated into prototype 3. The design, assembly, and testing of the integrated system, detailed below, were addressed with the entire system in mind, rather than through a step-by-step process.

8.1 Design Process

During the design of the Box, the manufacturing and assembly of the physical Box were taken into account. This allowed for the ease of assembly on the first attempt, with very minimal modifications. Once the Box structure was built, the subsystem components were placed in their respective spots and the Box was run and tested for functionality.

8.2 Assembly

Using the techniques developed during the design process, the structure of the Box was assembled with relative ease. The aluminum extrusion system was used to mount the outer panels, as seen in Figures 8.1 and 8.2.

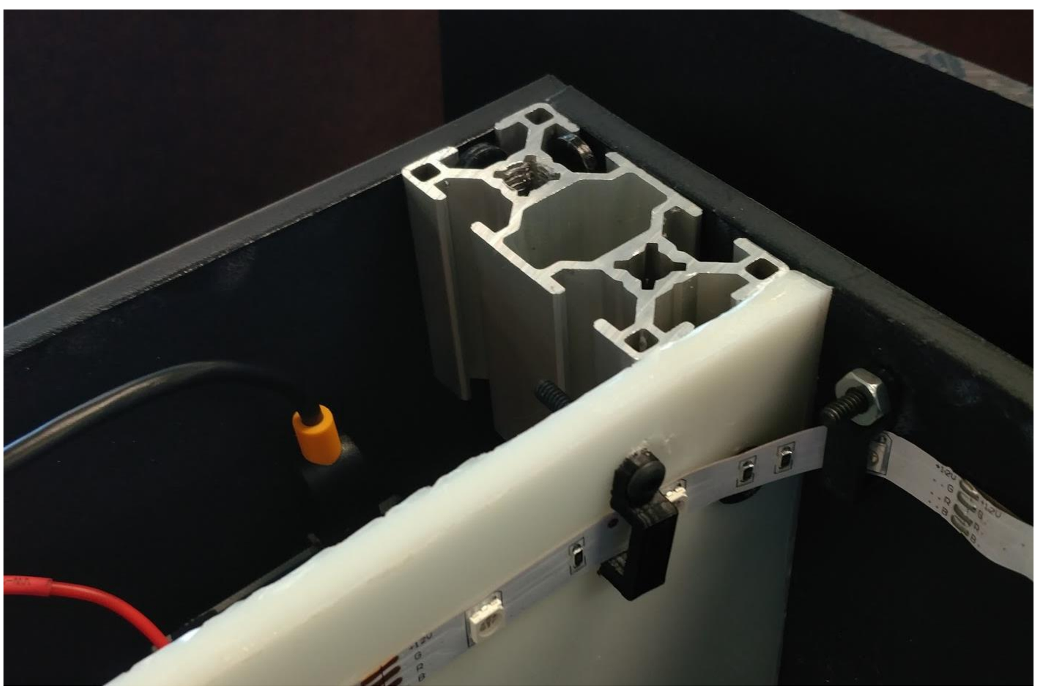


Figure 8.1: Back Mounting Extrusion and Electronics Space

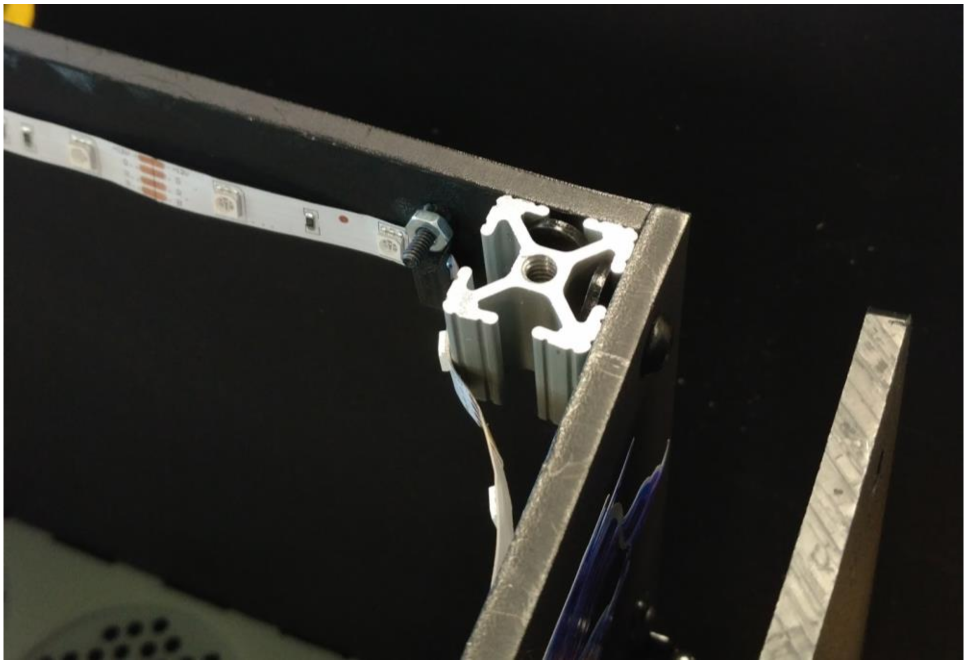


Figure 8.2: Front Mounting Extrusion

The panels required some minimal modification to fit with each other because they were manufactured on the router table, which had relatively loose tolerances. Once all the panels fit, the electronics and power supply were secured into the back compartment. 3D printer brackets were created for the touch screen, Raspberry Pi, and USB hub, and then mounted to the holes initially cut for these components. Lastly, the temperature and humidity sensors were placed in their correct positions, and the camera was mounted to the top panel and screwed in. The Box then moved on to system testing.

8.3 Testing Results

8.3.1 System Integration Test

To ensure that the BETA Box was fully functional after all systems were integrated in prototype 3, a systems integration test was performed. This test involved simulating the execution of an experiment, in the same manner that a student would do so.

A colored array, mimicking the typical prints from the r3bEL printer, was created out of white, pink, and green clay within a petri dish. The Box was set up using only the power button the touch screen. The Box was warmed to a set temperature of 39ºC, and after allowing 15 minutes to ensure complete warm-up, the petri dish was placed in dish position 5. White lighting was set, and the 39ºC temperature was sustained. Image capture was set for a duration of 20 minutes with an image frequency of 1 image every 2 minutes.

8.3.2 Incubation Evaluation

The thermal subsystem was re-evaluated for even heating distribution and for meeting the controlled temperature specification. The FLIR thermal imaging camera was again used to capture the temperature distribution on the dish bed over the duration of the test. Figure 8.3 below shows that the even temperature distribution was achieved in the final integration test.

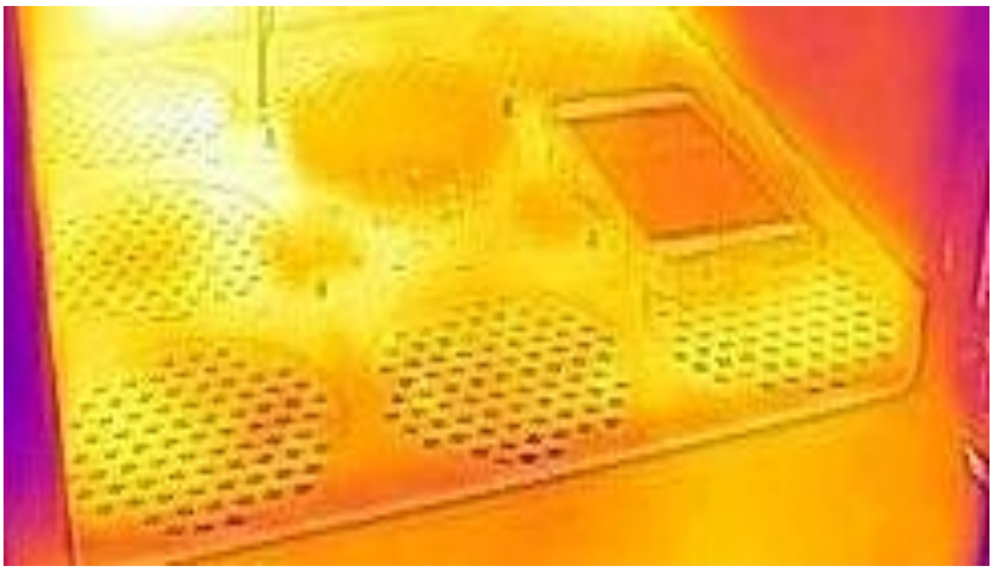


Figure 8.3: Even Temperature Distribution in Integrated System

The slightly brighter spot in the upper left corner of the image is due to reflection from the back wall of the incubator, not from the heating underneath. The phone shown in the image was placed in the incubator for other purposes, which will be discussed in section 8.3.4.

The figure below shows the temperature readings from the sensors. The black plot shows the final, singular temperature reading, taken as the average of the four sensors.

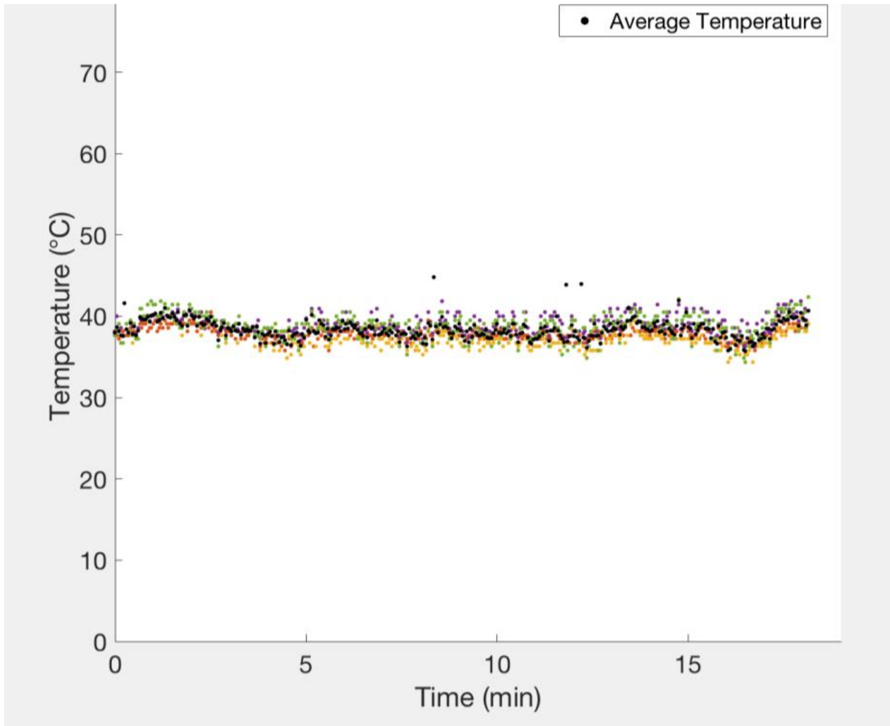


Figure 8.4: Controlled Temperature Readings in Integrated System

The average reading remained within 39ºC-41ºC, meeting the +/-2ºC specification. This change in precision was likely due to the door of the incubator being opened and closed throughout the test, to simulate other students adding or removing petri dishes. Although this precision was lower than that met by the thermal subsystem alone, it still met the established requirement, and proved that opening the door periodically can occur while still meeting the desired temperature control specification.

8.3.3 Lighting Evaluation

An image of the white lighting in the BETA Box during the system integration test is shown in Figure 8.5.

The lighting system was observed over the entire duration of the simulation to ensure that no color change or brightness changed occurred at any point in time during the test. The lighting system did not fail per these qualifications, thus meeting the requirements for lighting in the Box.

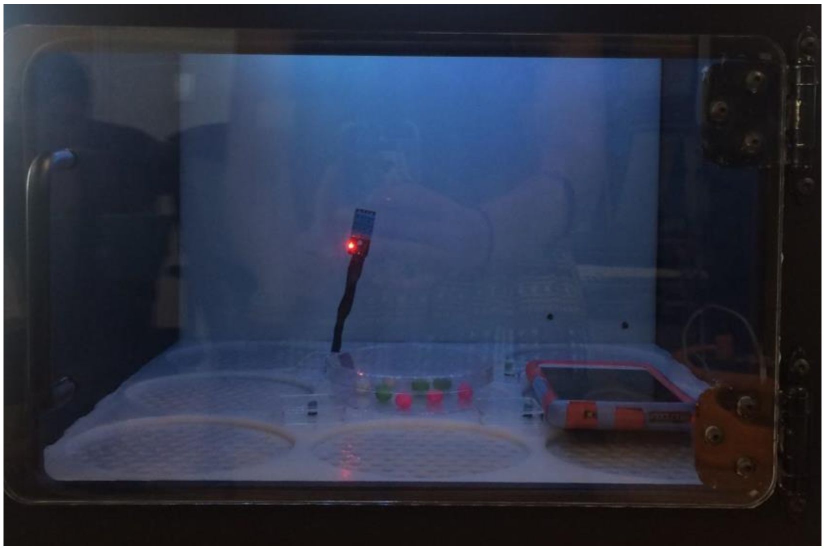


Figure 8.5: White Lighting in Integrated System

8.3.4 Imaging Evaluation

The camera did capture ten images, identifying with one image every 2 minutes for 20 minutes. Images 4 and 5 from the set can be seen below.

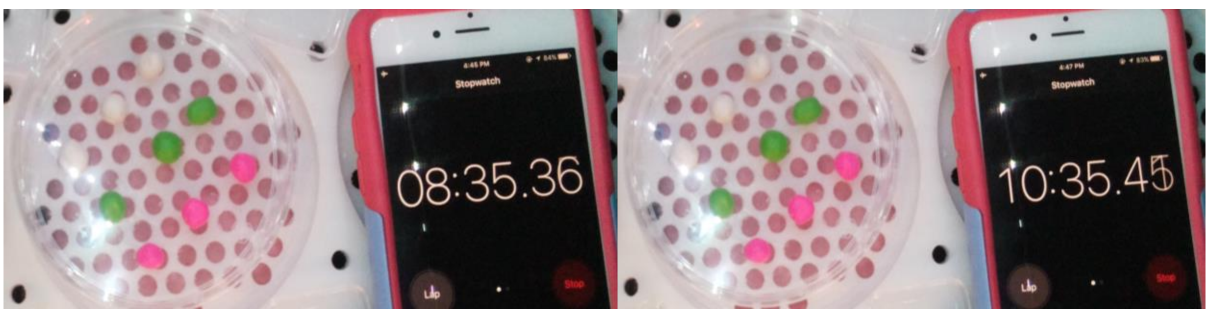


Figure 8.6: Image Capture in Integrated System

An iPhone was placed next to the petri dish, and the stopwatch was started prior to setting up image capture on the GUI. The first image was taken at 2 min 35 sec. As can be seen above, Images 4 and 5 were taken within 0.09 seconds of each other. All images taken were well within the time specification of being captured within +/-1s of the frequency setpoint.

The image of the colored assay in the petri dish itself shows that colored assays are easily visible, but white or clear assays might be difficult to distinguish for a student evaluating his or her experiment. This concern can be addressed either in the biology experiment design, or in modification of the dish bed. This image analysis, nevertheless, does meet the resolution specification establishing that the imaging subsystem met its requirements.

8.4 Conclusion

The BETA Box is fully functional for a student to execute a complete experiment using the Box today. The entire test was set up and run through the computing system and user interface, indicating that all communication between the computing system and physical subsystems was successful. After integrating all subsystems, the Box met all of the specifications that were established for the project, and are listed in Appendix A.

Part III. Bioprinter Feasibility Study

9. Auto-Calibration

In addition to the BETA Incubation Box, the team also developed enhancements for the existing SE3D Bioprinter. One upgrade to the printer was an auto-calibration sequence so that the user would not need to manually lower the motor shaft to the top of the syringe before each print. The purpose of an automated process was to reduce the setup time for experiments and decrease user error. Furthermore, inexperienced users, such as young students, waste less material. This enhancement to the 3D Bioprinter was developed to increase user-friendliness and customer satisfaction.

9.1 Design Process

Initially, the team considered many methods of calibrating the motor shaft with the syringe. Hall sensors, strain gauges, and forces sensors were all considered as possible tools that could be used; however, it was ultimately decided to use force sensors as they were readily available, low cost, and easy to integrate into the existing system.

After initial sketches were drawn up, a CAD model was created to show how the force sensor would be integrated into the existing printer. This is shown in Figure 9.1.

A small part was 3D printed with plastic to ensure that the sensor had a solid backing and that the force from the motor shaft was evenly distributed. The force sensor was easily controlled using an Arduino, which was compatible with the firmware currently used by SE3D.

9.2 Requirements

Since the feasibility study was not commissioned by SE3D, the company did not provide any strict guidelines for the requirements of the printer enhancements. However, the team created its own set of objectives to accomplish. The primary goal was to use the force sensor to empirically determine if the contact force between the motor shaft and syringe plunger would be definite enough to identify a point of contact.

The process involved the following steps:

1. Identify the initial point of contact between the motor shaft and syringe
2. Recognize when material has run out
3. Detect location of air bubbles

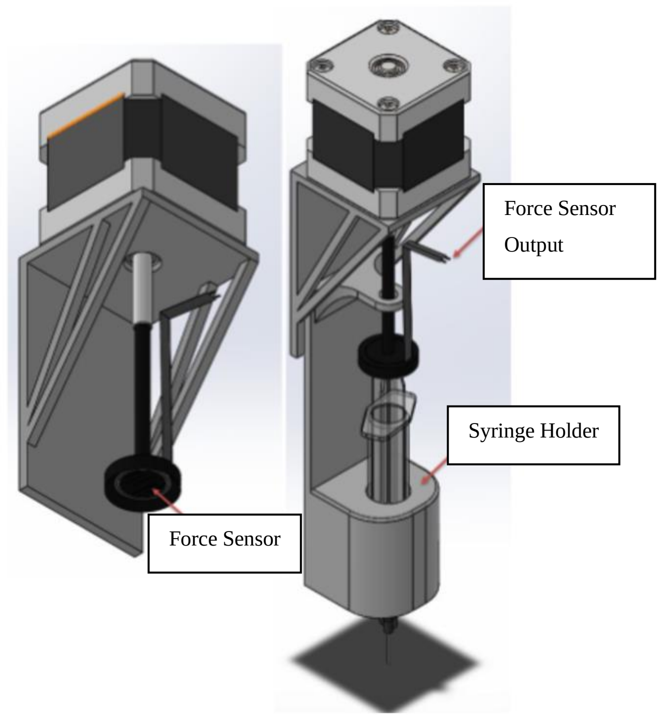


Figure 9.1: Auto-Calibration for Z-Axis CAD Models

A secondary set of goals was to construct a code for each of the scenarios so that the printer could pause or extrude as needed to accommodate for them. This part of the feasibility study was conditional based on whether the force sensor was sufficient to identify the events stated above. The secondary goal was to create a code that performed the following:

1. Stop motor once contact was made
2. Stop motor from over-extruding the syringe
3. Identify air bubbles and communicate possible errors to user

The team established these goals because they were believed to add functionality to the existing SE3D printer. This proposal would be an inexpensive feature that would make the product more user friendly and increase customer satisfaction.

9.3 Auto-Calibration Experimentation

To determine if the force sensor would be able to provide enough information to create a calibration routine, it was necessary to observe how the force changed during an extrusion. The force was expected to change based on the syringe type, material printed, and speed of extrusion; however, the study was intended to determine the code needed to achieve autocalibration. For the trials conducted, lotion was used in the place of biomaterial because it has a similar viscosity to the material used by SE3D. Five trials were run at extrusion speeds of 100, 200, and 300mm/min. This gave the team an idea of how the speed of extrusion affected the force felt at the top of the syringe. The sensor was connected an Arduino and loaded with code that monitored the analog readings from the sensor. The code for this can be found in Appendix H.

The analog data collected from the force sensor was converted to a force value through the conversion found in Appendix H. The force during each extrusion was plotted along with the rate of change of force. Figure 9.4 shows the results for a slow speed rate of 100mm/min.

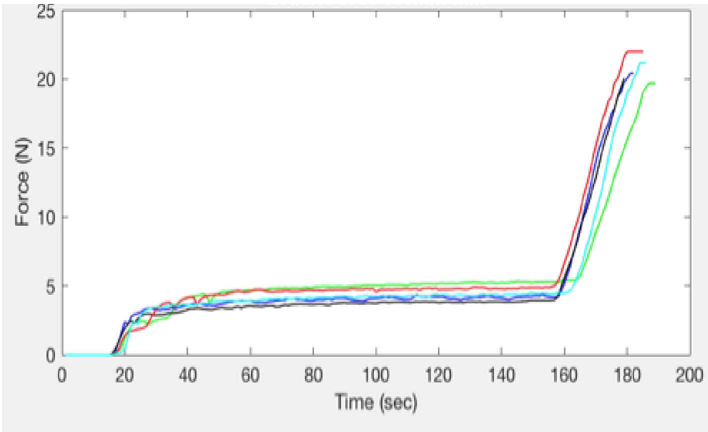


Figure 9.2: Force Readings for 100mm/min

As seen in the force diagram in Figure 9.2, the general shapes of the force curves were similar and followed the same trend. As the motor shaft was lowered, the force began to increase as initial contact was made with the plunger. Once the two were fully in contact, the force leveled off and remained relatively constant during the extrusion until the syringe ran out of material.

At this point, the plunger ran into the bottom of the syringe, and the force rose quickly to about 20 Newtons. The trials at a low extrusion rate were very consistent and clearly indicated the point at which the motor shaft met the plunger and when the syringe had been emptied.

The same procedure was followed for an extrusion speed of 200mm/min and is shown in Figure 9.3. The data collected during the trials taken at 200mm/min generally followed the same trend as the trials taken at a slower rate. However, there was noticeable variation in the constant force values during the main portion of the extrusion. This was unexpected; however, it was believed to be caused by vibrations from the step motor or possibly environmental changes since not all trials were conducted on the same day. Finally, a third set of trials was conducted at the fastest speed of 300mm/min.

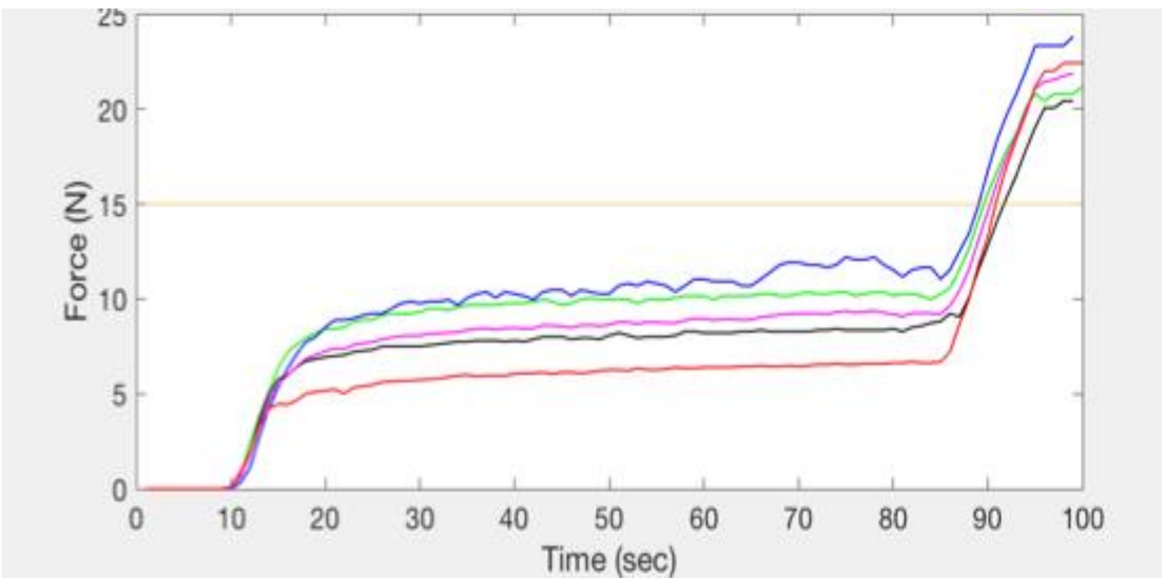


Figure 9.3: Force Readings for 200mm/min

The forces calculated for an extrusion speed of 300mm/min, Figure 9.4, were similar to the previous experiments; however, the slight variation in force values was still present. Because of these deviations between the extrusion rates, the team decided to analyze the rate of change of the force. This data for all three rates can be found in Figure 9.5.

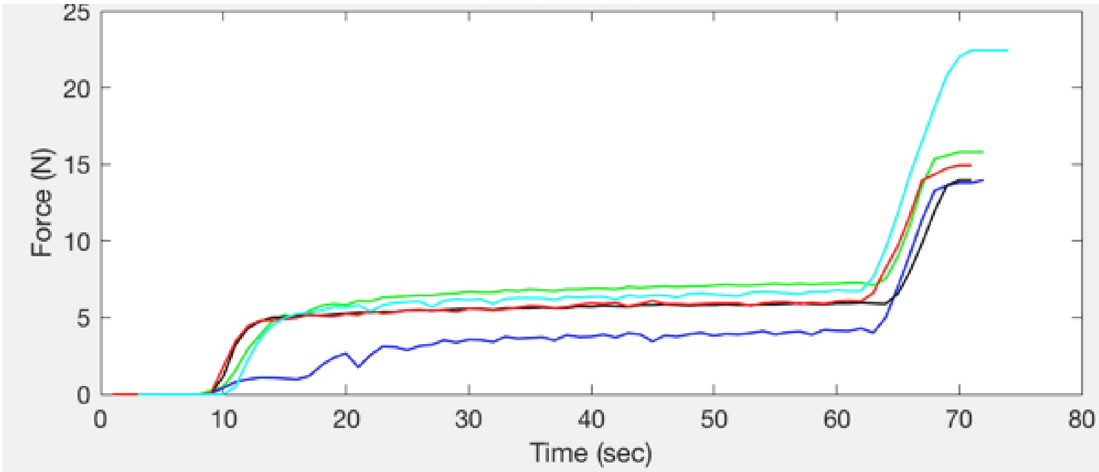
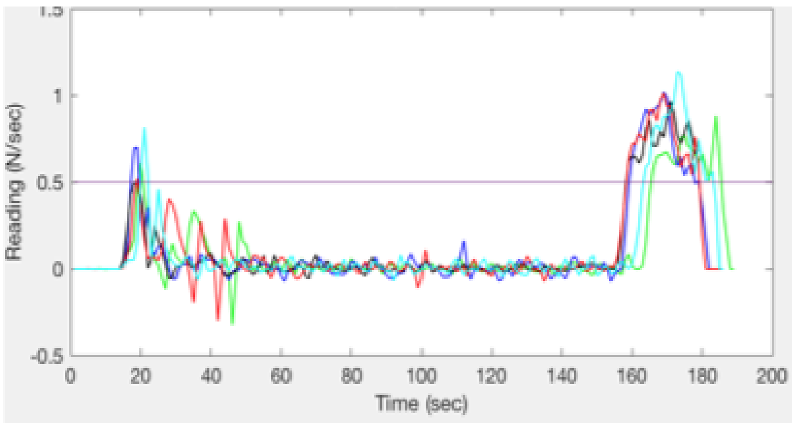
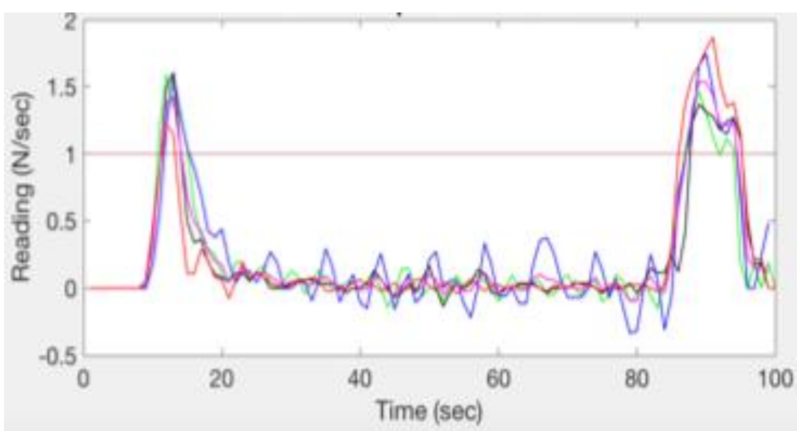


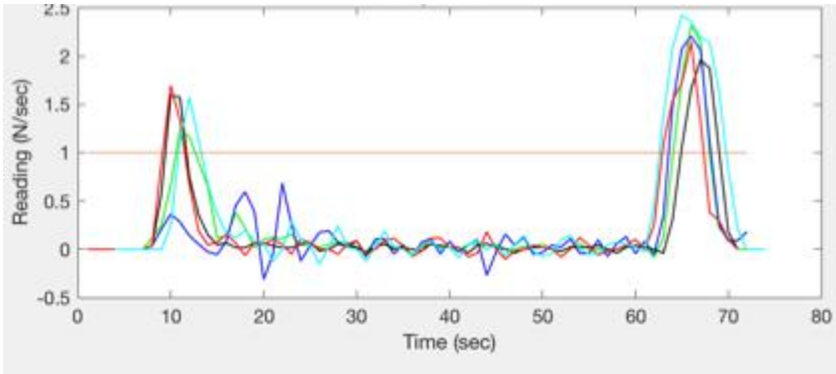
Figure 9.4: Force Readings for 300mm/min



(a)



(b)



(c)

Figure 9.5: Force Rate of Change for Extrusion Rates of (a) 100, (b) 200, and (c) 300mm/min

The rate of change, or slope graphs, were found to be more useful than the force plots themselves because the slopes were much more consistent. Over the majority of the extrusion was when the force was constant, the slope value oscillated around 0 N/sec. However, large peaks in the slope marked both the points at which calibration contact was made and when the syringe ran out of lotion. These features of both the force and slope the plots were distinct enough to create software code that could identify these locations real time and calibrate the system automatically.

9.4 Auto-Calibration Conclusion

Based on the results of the extrusion experiment, the code for the calibration and end of material (the point where all the biomaterial is used up) procedures was written. Given the data from the force sensor, no conclusions could be ascertained in regard to the location of air bubbles, except that large bubbles could interfere with the calibration routine if the syringes were not loaded properly. For this reason, it is important that users understand how to properly load the printer with the biomaterial. The rest of the feasibility study was a success.

The auto-calibration code, found in Appendix H, functions according the flowchart in Figure 9.6

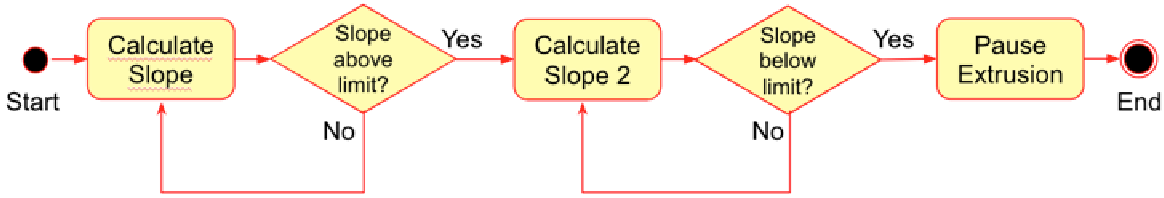


Figure 9.6: Flowchart of Auto-Calibration Code

As shown, the program read the force values real time from the printer and calculated the slope. When it reached a limit specified by the user and the speed of extrusion, the program started to calculate a second slope. For extrusions shown in Figure 9.5c, this first checkpoint occurred around 9 seconds. When the second slope dropped below the limit, at a time of approximately 11 seconds, the extrusion would pause and the calibration would be complete. The software took about a second for each calculation, so a slight delay was expected; however, this error was minimal.

Separate code was created to end the extrusion in the event that the syringe ran out of material. However, this code utilized the force data rather than the slope. A user inputted force threshold was used to communicate when the syringe was emptied. This code can be found in Appendix H.

The code listed above was programmed with an Arduino; however, it was not integrated into the firmware of the existing printer. This was the next step in the project. The testing of the force sensor showed that the data collected was sufficient to implement an automated calibration system. Although it may need fine-tuning, the feasibility study concluded that the proposed method would be possible to incorporate into SE3D's future bioprinter models.

10. Dual Extruder

The other major upgrade to the bioprinter was to introduce a dual extruder design to increase the versatility of the bioprinter as well as produce 3D designs out of biomaterial. Introducing a dual extrusion system allowed the bioprinter to print multiple materials at once, speeding up printing time, as well as enabling easy mixing of different biomaterials for specific classroom experiments. Furthermore, biomaterial is produced as a viscous liquid and needs time to solidify, which means creating a 3D structure is impossible without support. A dual extruder allows a plastic extruder to be mounted alongside the original printer's bio extruder, thereby enabling plastic molds or scaffolding to be 3D printed as a support for the biomaterial as it solidifies into the desired 3D designs. This enhancement was developed to improve the bioprinter's experimental capabilities.

10.1 Design Process

The initial design called for a method of dual or multi extrusion, so the first step in the design process was to decide how many extruders the bioprinter would be implementing and establish the goals those extruders had to fill. Based on limits related to the original bioprinter and bio extruder designs, a side-by-side dual extruder was chosen—this design builds off the original bio extruder design with an additional extruder focusing on plastic extrusion.

This led to the next step in the design process of deciding how to attach the second extruder and which design to utilize. First, based on research, PLA plastic was chosen as scaffolding for the plastic extruder [6]. The article detailed substantial evidence that PLA plastic was bio-friendly, thereby rendering less contamination and material bonding during printing, and was seen in common use among other 3D bioprinters on the market. The website All3DP confirms, "Aside from 3D printing, it is typically used for packaging material, plastic wrap, plastic cups, and plastic water bottles. It is considered to be more ecologically more friendly than ABS – after all, it's made from plants[7]."

During the design process, the team built an open source RepRapPro 3D printer in an attempt to understand the complexities of plastic 3D printing before designing modifications for a 3D bioprinter. Coincidentally, the RepRapPro successfully used PLA plastic extrusion, so the second extruder was built around that design—incorporating a stepper motor, a heat sink, a small fan, and a 3D printed planetary gear feeding system using a tube to push the plastic through the extruder. This design also met the printing precision requirements of $\pm$0.1mm.

Attaching the extruder was slightly difficult, since the original idea was to place the two extruders side by side on a plate. However, it was simpler to stick with the original bio extruder design and extend the plate off of it.

The bio extruder consisted of four 3D printed parts, which can be see assembled in the original design is shown in Figure 10.1. The final product is seen in Figure 10.2.

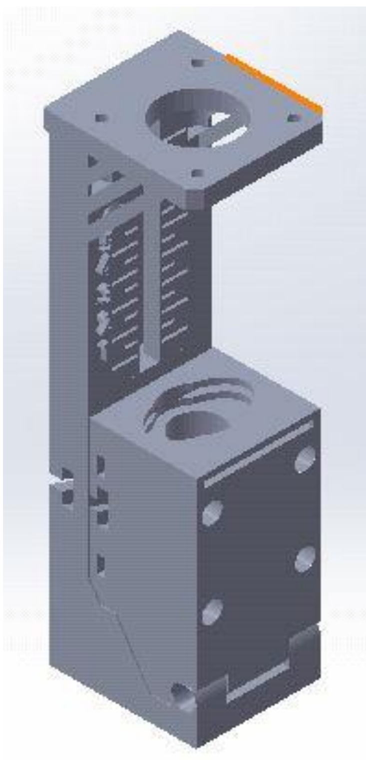


Figure 10.1: Original Bio extruder

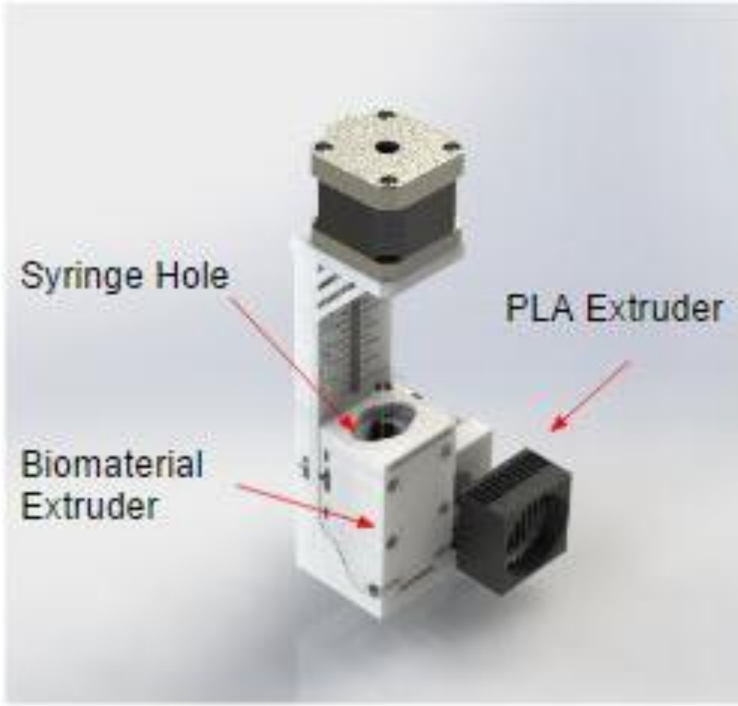


Figure 10.2: Final Dual Extruder Design

Due to the wiring placement, the design attached a plate on the right side of the bio extruder, which holds a module that extrudes PLA. Since the two extruders did not initially have the same height or planar orientation, the plate was extended to properly align the two extruders. Upon implementation, it was noted that the orientation of the PLA extruder resulted in the planetary gear being too large and it hit the heat bed before the extruders, making it impossible to print. The final step in the design process resulted in the removal of the planetary gear in favor of a smaller dual gear system that fit the size constraints.

10.2 Requirements

The main goal of the dual extruder was to enable PLA plastic extrusion alongside the bio extruder to provide the necessary support to enable 3D printing with biomaterial. The accuracy of the original bio extruder when printing was $\pm$ 0.1mm as stated by SE3D, which also meets the specifications of other biomaterial printers on the market. Thus, it was imperative that our PLA extruder be just as precise when printing and be able to function at the same speed as the original bio extruder. Furthermore, since our team was improving upon an existing bioprinter, the size of our dual extruder had to fit within the current design's dimensions.

10.3 Considered Alternative Designs and Prototypes

Initially, many designs were considered for how to implement a dual or multi extruder. One design was to have a number of syringe holders mounted on a ball bearing surrounding a central rod, with the idea of rotating syringes to the front to print the desired material. A second preliminary design focused on using a pipet system, where vials of biomaterial would be lined up on one side of the printer and the bio extruder would automatically load itself, print, and then reload itself when empty. Both of these designs were inadequate as they would require full overhauls of the physical design of the bioprinter and the code to use it.

The next iteration and initial prototype, which is seen in Figure 10.3 below, focused on a plate design based on the Rep Rap Pro, which used a standard plate designed to allow two PLA extruders to be placed side by side.

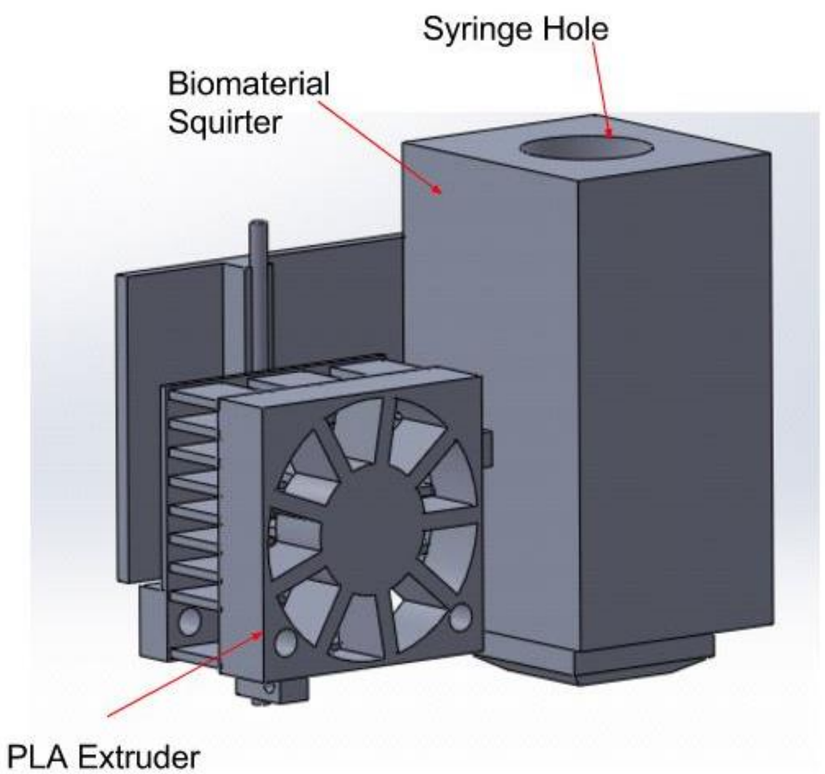


Figure 10.3: Prototype Dual Extruder

Our plan was to further expand this design to create a modular system where extruders could be added, removed, and exchanged easily using a slide-and-lock system, enabling the bio extruder and PLA extruder to be set side-by-side. This would have also enabled customization of the bioprinter depending on which classroom experiment was being used, but ultimately this design would have also resulted in a complete redesign of the bio extruder, and was further limited by special requirements. In the end, this design was limited to improving upon the initial design, which significantly constrained the number of extruders and the methods used to mount them.

10.4 Analysis

According to tests done by SE3D employees, slight variations in orientations between two extruders led to failed prints, even with variation in orientation as small as 5mm. This meant that the two extruders in a dual extrusion system had to be precisely aligned, which the current design did not guarantee due to the modularity of the original bio extruder.

The current design resembled a simple rectangular box with a hole in it, as can be referenced in Figures 10.1, 10.2, and 10.3. This design required a user to manually insert a syringe into the hole to begin printing. However, to accommodate the vast variety of syringes, this hole was larger than the syringe itself, which means that each user inserts the syringe in a slightly different location or orientation, and it's almost impossible for two individual users to set up the system the same. This design was adequate for single material printing, as it was simple to use and user-friendly, plus orientation precision did not cause issue since the variation between users was very small. Additionally, the bio extruder prints in relation to itself, so all errors in orientation were carried throughout the process resulting in a successful print. However, the bio extruder needs to be aligned with the PLA extruder, and the modularity of the bio extruder changes its individual alignment with each reloading and misaligning the system.

In order to continue work on the dual-extruder design, the team created a roadmap of possible fixes to this situation to create a precise system, as can be seen in the flowchart shown in Figure 10.4.

The main question was whether the syringe design be kept or the future design have its own extrusion tip and not rely on the syringe tip. Following the path of keeping the syringe, two basic ideas were formulated with the idea of keeping the initial designs modularity.

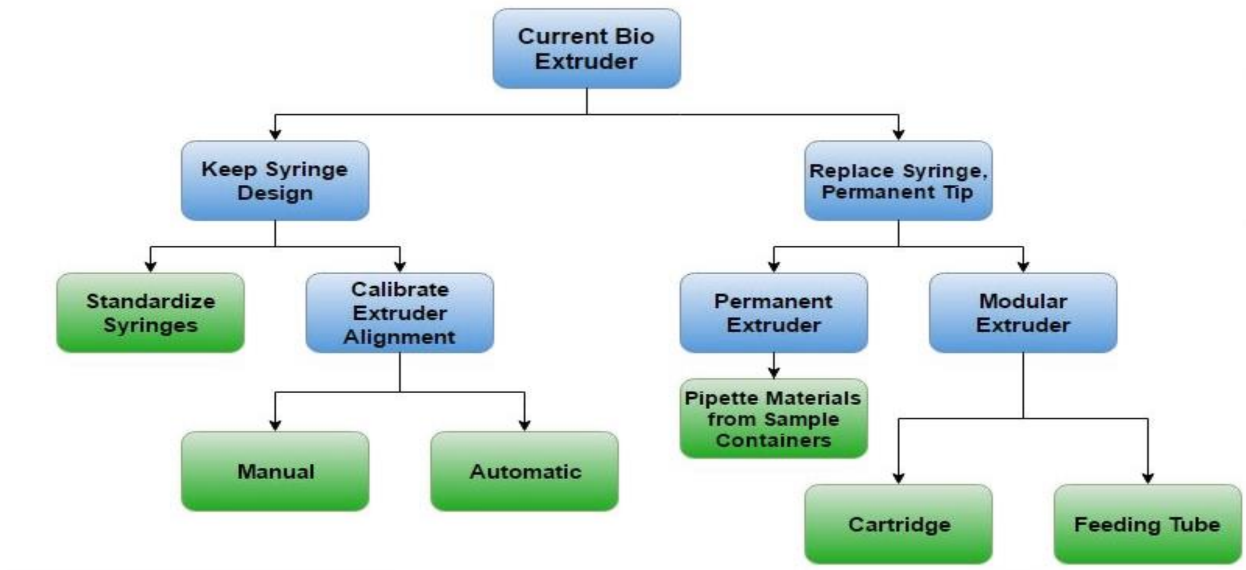


Figure 10.4: Dual Extruder Future Improvements

One is that all syringes are different, both in shape and in needle length. Both of these characteristics relate to precision errors that prevent the success of the dual extruders, so a simple fix of standardizing syringes is a cost effective way of solving the issue. However, the hole may still be too big, so a second idea was to create a calibration device, whether it be a manual or automatic system, to properly align the two extruders.

Taking the other path, which focused on using a permanent tip instead of the syringe tip, but didn't necessarily remove the modularity of the system, another two options were presented. The more complicated response was to reintroduce the pipetting system thereby creating a permanent extruder to guarantee precision while continuing to emulate the modularity. This had drawbacks, such as the complicated process of integration into an existing design. The alternative was to stay with a permanent tip to provide the precision needed for dual extruding, while retaining modularity, either in a cartridge system that replaces the syringe or as a feeding tube design similar to the PLA extruder. Overall, retaining modularity and guaranteeing the required precision was the best choice in proceeding with this design.

10.5 Conclusion

While the dual extruder design was initially successful and both extruders were able to individually print within the precision and time margins, it was noted that the bio extruder relied on a syringe to be manually placed inside an oversized hole rendering it impossible for every user to orient the syringe in the same place. This meant that the two extruders would not be precisely aligned, which would result in a failed print. Further iterations of this design are necessary to provide the precision required of a dual extrusion system. Future goals retain aspects of both a modular and a precise system to provide the best of both designs into the next dual extruder iteration.

Part IV. Professional and Engineering Considerations

11. Team and Project Management

This team was comprised of seven engineers from two different disciplines. This project's success was contingent upon managing the members of the group and making sure that tasks were distributed properly among all members. The team met weekly and communicated daily to ensure that the project was well understood and appropriate goals were set each week for the team.

11.1 Project Challenges and Constraints

The challenge of this project was to take the prototype out of a pure design environment and to design for manufacturability, ease of use, and low-cost. Because the project was intended for commercial use in the education sector, a huge cost constraint was placed on the project. The low cost allows the product to be placed in high school classrooms more quickly than other commercially available products. The team intended to keep costs down by using modular, low cost parts that could be easily integrated into the system without a large overhead. For example, the project made use of a Raspberry Pi microcontroller, which allowed for a user interface and control system to be implemented in one circuit board for under $30.

Another challenge faced by the project was intellectual property, which created a conflict of interest between the company and Santa Clara University. Ultimately, this was avoided as the team did not accept any funding from the startup company; however, this also meant SE3D remained more hands off than initially expected. This gave the team freedom to explore various areas of development for the system and led to the development of a completely new product that SE3D did not originally imagine.

11.2 Timeline

The complete timeline for the fall, winter, and spring quarters can be found in Appendix I.

11.2.1 Fall 2016

During the fall, the team began construction of the RepRapPro 3D printer, which was used by the team to become familiar with the open source 3D printer design. This construction extended to the end of the quarter, and assisted in understanding the structure, mechanics, electric components, and code of a functional 3D printer.

The team also subdivided into taking specific ownership of each facet of the project. On the mechanical side, Emily Takimoto and Franz Plum took primary responsibility for the feasibility study, while Jacob Ososke, Jeff Barone, and Josie Warren took charge of the BETA Box. On the computing side, Max Abrams took primary responsibility of computing systems in the software support on feasibility study and BETA Box, while Cynthia Le managed the front-end systems of the BETA Box. Using this breakdown, proofs of concept for all subsystems were devised by the end of fall quarter.

11.2.2 Winter 2017

For the BETA Box, winter quarter mainly consisted of creating prototype 2 and using it to test the thermal subsystem and the computing subsystem. This quarter held the bulk of the design and experimentation work, which can be seen in detail in Appendix I. The User Interface storyboard was fully developed and preliminary versions were created. At the end of the quarter, the final design for all subsystems—structure, thermal, computing, and others—were established.

The majority of the design and testing on the feasibility study also occurred during winter quarter. For auto-calibration, this entailed the many trials of coding and evaluating the functionality of the force sensor. For the dual extruder, multiple CAD designs were developed and evaluated in physical placement within the printer.

11.2.3 Spring 2017

Spring quarter was used for final testing of all facets of the project and for preparation for the senior design conference. The slides from this presentation can be found in Appendix N.

At the very beginning of spring quarter, prototype 3 of the BETA Box was assembled, and the user interface was installed. This is when student usability testing occurred on campus. Final adjustments to the GUI were made, and then the complete BETA Box was finalized and tested to systems integration success.

For the feasibility study, the printer circuitry was exchanged to support the new additions of the auto calibration and dual extruder. The dual extruder and force sensor were attached to the printer arm. The feasibility study was continued through to the end of spring quarter.

11.3 Risk Management

Table 11.1 outlines three potential risks that this project faced the risk of encountering. Each risk was analyzed for consequences and impact and had associated mitigation strategies. The risk probability was characterized between 0 to 1 where 0 was no probable chance of risk and 1 was certain probable chance of risk. The severity of the risk was categorized between 0-10 where 0 means not severe and 10 means extremely severe. Probability and severity columns were multiplied together to determine impact where 0 means no impact and 10 means significant impact to the project. Mitigation strategies are listed after each risk, which shows how the team has worked to eliminate the risk.

11.4 Budget

At the beginning of the year, the team requested funding from both the Santa Clara University School of Engineering and SE3D. A preliminary breakdown of the requested funds is shown in Table 11.2.

As shown, the proposed funding was designed to allow the team to prototype capabilities that were not part of the grant awarded to SE3D. The funding from the School of Engineering was meant to allow for alternative designs that were not specifically awarded by the grant.

Ultimately, the team was awarded $3,500 from the Santa Clara University School of Engineering; however, the funding from SE3D was not approved. With the existing funds, the team budgeted the money in a more detailed manner according to the main goals of the project. Those goals included approximately 3 prototypes of the incubation unit, the auto-calibration system, and the dual extrusion study. With a 15% contingency, the funding was reallocated as shown in Table 11.3.

Table 11.1: Risk Analysis

Rank	Risk	Consequence	Probability	Severity	Impact	Mitigation Strategies
1	System Integration	- Hardware and software are not properly integrated	0.8	7	5.6	1. Close communication between hardware and software teams 2. Recurrent testing of device
2	Added Cost	- New features will increase cost of system	1	5	5	1. Offer new features in modular design so customers can choose only based on what they need
3	Durability and Lifetime Issues	- System does not survive guaranteed life - Bad reputation - Not for use with target audience (high school students)	0.6	8	4.8	1. Durability redundancy measures 2. Burn-in of electronics to remove infant mortality 3. Develop rigorous testing and certification for system
4	Biomaterial Safety and Compatibility	- Unsafe for use in an open space - Contamination of experiment	0.4	10	4	1. Use of standard biomaterial and safety protocols put in place by the Robotics Systems Lab and Santa Clara Environmental Health and Safety
5	Illness of Team Members	- Will need to re-assign work - May fall behind in timeline	0.7	5	3.5	1. Ensure that teammates get enough sleep 2. Have good hygiene by frequently washing hands and working in a clean environment.
6	Poor Communication Among Team	- Solution may be missing features - May fall behind in timeline	0.4	8	3.2	1. Frequently meet with team members to review design and progress 2. Review timeline assignments
7	Misunderstanding Requirements	- Solution will not meet customer needs - Unnecessary features	0.3	9	2.7	1. Frequently communicate iterations with customer 2. Confirm requirements with team members
8	Market Limitations	- New features will not be competitive in the marketplace - New features will change the target audience of the printer system	0.3	9	2.7	1. Expand market scope to include middle schools and colleges 2. Run trials with potential customers

Table 11.2: Preliminary Budget Breakdown

Undergraduate Programs		
Item Quantity	**Item**	**Total Cost**
1	Heated Print Bed	$200
1	Structural Materials	$700
2	Camera	$300
4	Motors	$300
4	Motor Drivers	$200
1	PCB Material	$250
4	Microcontroller/SBC Boards	$250
10	Sensors and Electronic Components	$300
1	LCD Display and Interface	$200
1	LED Lighting and Controller	$100
1	Air Filtration Unit and Controller	$300
1	Incubator Heater and Controller	$400
SE3D Funding		
10	Biological Supplies	$500
1	Speaker and driver	$100
2	Computer Boards (Raspberry Pi)	$400
1	Modular Unit	$200
1	LCD Display for Modular Unit	$200
2	Motors for Modular Unit	$100
2	Motor Drivers for Modular Unit	$100
1	Network Router for Modular Unit	$200
1	Electronic Components for Modular Unit	$200

Amount requested from the Undergraduate Engineering: **$3,500**

Amount requested from SE3D: **$2,000**

Total project funding requested: **$5,500**

Table 11.3: Subproject Budget Breakdown

Subproject	Total Cost
Incubation Prototype 1	$900
Incubation Prototype 2	$900
Incubation Prototype 3	$900
Auto-Calibration	$200
Dual Extruder	$200
Contingency (15%)	$325
Total	**$3,500**

After all material costs were finalized, the team was well under budget, as seen by the breakdown below.

Budget	$3,500.00
Spent	-$2,508.75
Remaining	$991.25

About $990 remained in the budget after aspects of the project were completed. A complete budget spreadsheet can be found in Appendix J and a full breakdown of all hardware purchased can be found in Appendix K.

11.5 Cost Analysis

A material selection cost analysis was completed to determine the cost for using different plastics. Appendix E details the comparative prices for the top 3 different plastics. Also, a preliminary cost analysis was conducted for the prototype of the BETA Box to determine the overall cost to manufacture the Box. This was the cost for all the materials when bought in very small quantities. Because of this, materials costs, when the parts are bought in bulk from a large-scale manufacturer, will decrease significantly. In Appendix K, it can be seen that the Box costs $325.00 to make, which is well below our $500.00 sell price, allowing for net profit.

11.6 Business Plan for the BETA Box

11.6.1 Business Goals and Objectives

The goal for the BETA Box was to be a cost effective incubator for high school budgets that could provide heating and lighting control needed for the variety of experiments. Furthermore, the BETA Box needed to include a camera for documentation purposes, a large enough area to allow a full classroom of experiments to be run at the same time, and the ability for each individual user to easily set up experiments and extract individual data. Finally, the Box was designed for versatility, so that the teacher can prescribe any setting including lighting, temperature, and duration, optimizing educational impact for students.

11.6.2 Potential Markets

The primary market for the BETA Box involves teachers and/or students looking to keep living experiments at a controlled temperature and/or lighting, while also taking advantage of image capture, all for educational purposes. Therefore, institutions with biology classes are the primary market for the product.

Currently, the target market for the BETA Box is high school biology teachers. While all teachers from all schools- public, private, and charter- are all target consumers, it is most likely that private and charter school teachers will purchase the BETA Box until the selling price can be lowered further.

Long term, the market for the BETA Box has the potential to expand to collegiate classrooms. This identifies the main user as young adults below the age of 22.

The following table shows the current competition for incubators used for educational purposes and that are sold for less than $1,000.

Table 11.4: Educational Incubator Competition Comparison

Company	Product	Selling Price	Strengths	Weaknesses
Carolina	Incubator 701194 [8]	148.50	Low cost, small	No image capture, no lighting, no temperature settings, no independent computer
Repti Pro	6000 Incubator [9]	199.95	Low cost, sleek design, even temperature distribution	No image capture, no lighting, no independent computer
R-Com	Pro 20 with USB Incubator [10]	675.00	Full temperature and humidity control, can track changes via computer	No lighting control, no image capture
Thermo Fisher	Precision Compact Oven PR305225G [11]	802.00	Temperature control ambient-210ºC	Analog display, very large dimensions for same interior size, no lighting control, no image capture

No educational incubators that included lighting control and image capture could be found. The closest product that featured image control was a microscope with an incubator attached, and it sold for over $2,000. In conclusion, there is no explicit competition that sells incubators for educational purposes that also feature lighting control and image capture.

11.6.3 Financial Plan and ROI

Table 11.5 outlines the general financial plan for the BETA Box, including cash needs, debt repayment, and projections.

This financial plan shows that a $16,000 investment allows for two years of development for the business. Yearly profit would be expected in the third year, with the investment fully paid after the fourth year.

Total return on investment after five years is $24,500. After the third year, this ROI would be subject to increase by altering marketing strategies to increase unit sales or by reducing the cost per unit via enhanced manufacturing strategies.

Table 11.5: Five-Year Financial Plan

CASH NEEDS					
	2017	**2018**	**2019**	**2020**	**2021**
Research and Development	$ 5,000.00	$ 1,000.00	$ 500.00	$ 500.00	$ 500.00
Product Cost	$ -	$ 4,500.00	$ 9,000.00	$ 18,000.00	$ 36,000.00
Sales & Marketing	$ 1,000.00	$ 1,000.00	$ 1,000.00	$ 1,000.00	$ 1,000.00
Administration & Other	$ 2,000.00	$ 1,500.00	$ 1,500.00	$ 1,500.00	$ 1,500.00
Total	$ 8,000.00	$ 8,000.00	$ 12,000.00	$ 21,000.00	$ 39,000.00
EXPECTED SALES					
	2017	**2018**	**2019**	**2020**	**2021**
Units Sold	0	15	30	60	120
Gross Per Unit	$ 500.00	$ 500.00	$ 500.00	$ 500.00	$ 500.00
Total	$ -	$ 7,500.00	$ 15,000.00	$ 30,000.00	$ 60,000.00
NET REVENUE	$ (8,000.00)	$ (500.00)	$ 3,000.00	$ 9,000.00	$ 21,000.00

12. Engineering Standards and Constraints

As a team of engineers, the group holds itself to high standards with the products it creates. With all work now and in the future, it is not acceptable to simply ignore ethical concerns. As progress is made, it is crucial to consider the ethical implications of the team's work. Specifically, the implications of The Box system and how the group interacts with the customers must be recognized. The BETA project has addressed the following engineering considerations:

1. Manufacturability
2. Health and Safety
3. Ethical Concerns
4. Social Impact
5. Economic Impact

12.1 Manufacturability

For the final prototype, the manufacturing was specifically tailored to one-off construction. That means that this method for manufacturing the parts is only feasible for making one copy of the BETA Box. All the panels were cut on a CNC laser cutter, and then finished on a CNC router table. The necessity to line up the panels by eye on the router table led to errors in the positioning of certain features on the parts. In addition, the metal extrusion used for the corner joints were machined manually on a CNC mill.

Figure 12.1: BETA Box Manufacturing Process

While the initial design was for the one time prototype, some considerations were taken into account to modify manufacturing as production scales up. The main modification needed is to redesign the HDPE top and side panels to have one cohesive piece of plastic that would be then manufactured using thermoforming. Using this process, the construction of the overall Box frame would be much more repeatable and able to grow with higher production.

12.2 Health and Safety Considerations

The following statement indicates all safety precautions taken in the assembly of the final prototype. All subsequent Boxes produced require the same safety precautions by the manufacturer.

During assembly it was imperative that proper safety regulations were followed. When assembling the power supply and connecting wires to electronics, it was ensured that all soldering were carried out in a well-ventilated area. The solder tip was never directly touched, nor did soldering occur close to the soldering iron. It was ensured that after each use the soldering iron was properly cleaned and placed in the soldering rack. It was then ensured that all components were grounded before attempting to plug the power supply into an outlet. When using machinery or laser cutters to create the walls, floor, and ceiling of the BETA Box, it was ensured that users were properly trained with the machines and followed basic procedure when operating. Many of the systems used to cut the BETA Box were automated, so it was important for the operator to be precise in setting up the machine for use and be aware of emergency protocols to shut down the machine when anything went wrong.

12.2.2 Student Use

Safety is an utmost concern with the project, and the team made sure the device can be safely operated. The subsystems of the BETA Box must never overheat or short circuit under operation. To protect against this, a high quality system was created that includes various fail-safes, such as shutting the system down in case of excessive temperature. Since images can be offloaded to user supplied USB drives, the system is safe when interacting with the customer's devices and must not damage them through use. Standard data transfer processes were used via safe USB removal protocols through the interface. Similarly, adequate instruction will created to educate the users on how to correctly operate the BETA Box, similar to the SEEDs instruction forms used to test the interface, should the Box go into commercial production. Providing clear instructions increases the adoption of the product and helps the users stay safe while working with the Box.

12.3 Ethics

12.3.1 Ethical Justification

Today's technologies have the ability to 3D print biological materials at a reasonable cost and to incubate a wide array of biological substances at a variety of capacities. Nevertheless, none of these technologies are commonly used in high schools, and those that are only available at a high cost. It is in society's best interest to make 3D bioprinting and incubation more accessible because the more students are exposed to this technology, the more likely there will be an increase in adults pursuing STEM. Such success fosters many possibilities, such as the better usage of existing technology for research and medical applications, an increase in developing more advanced technology, and discovering greater medical breakthroughs as more students become interested in the science behind life.

Education is a right in the US, and we must strive to provide students with the best education possible. If we have the capacity to improve learning by maintaining the value of the current biology curriculum, while also augmenting interest in biology through the introduction of career-inspired technology, it is our duty to do so.

12.3.2 Ethical Engineering Practices

During the design process of the dual extruder, force sensor, and BETA Box, the following ethical motivations were considered: safety, social economic consequences, and educational implications. These considerations coincide with the ASME Code of Ethics Canon 1 and Fundamental Principle I, as well as with the ACM Code of Ethics Guidelines 1.2, 1.4, and 1.1, respectively. Safety is always the primary concern when engineering a product. Because of this, all of our materials are bio-friendly and all electronic equipment was sealed. Within the Box, there is circuitry, a large power source, and a 64 sq. in. heating plate. As a result, electrocution and burns are the greatest potential hazards. The wiring and power are all sealed within ¼ in. thick HDPE, and plastic insulator, with the exception of small holes for cooling the electric equipment, all of which were designed so that fingers cannot poke through and touch any of the equipment. The heat plate is also contained in a similar manner, so that it cannot be touched directly. It is also secured in place, so that it will never accidentally move and touch any thermally risky materials.

Although social equality in the distribution of these products cannot be achieved completely, they can be mitigated as much as possible by driving down their costs. Since the dual extruder and force sensor are feasibility studies, they are not included in this ethical discussion; the incubation Box is the subject. Lowered cost for the Box is achieved through extensive research of structural materials, simplification of manufacturability, and use of common, standard parts as much as possible. The team also researched budgeting for high school biology labs, to better understand the spending capacity for the customer. Currently the target cost of materials for the incubator is less than $200, so that the final product after manufacture can be sold for profit at $500 or less.

12.3.3 Ethical Risks of Project: Social Equality

The primary ethical dilemma concerning the development of the 3D bioprinting technology and incubator addresses equality. In developing these pieces of equipment for high schools, they must remain low-cost, but it is currently not possible to make them economically available to all high schools. If only the wealthiest private and public schools will have first access to these technologies, how can we consciously argue that they are still helping the greater good? The immediate social effects of bringing an updated 3D bioprinter and educational incubator to the market are that the schools with students who already have the best chances for success will get yet another opportunity while the schools with students who need the most support will get nothing.

The ethics of social justice counters the introduction of this new bioprinter equipment and incubator, because justice and equality will not be met. In *Theory of Justice,* John Rawls argues for fair equality of opportunity. This would not be accomplished because most students have no choice where they attend school and to what resources their schools have access. A teacher will decide whether his or her students will have this technology in the classroom, and his or her choice will be limited by the school's budget; the student will not independently decide whether or not he or she will have access. Because of this, we are not supporting equal opportunity. Similarly, Ronald Dworkin's "What is Equality? Part 2: Equality of Resources" also asserts that we would not be meeting the requirements for social justice [12]. According to Dworkin, unequal distribution of resources is considered fair only when it results from the decisions and intentional actions of those concerned, and again, the availability of this technology in the classroom will most of the time, if not all of the time, not be a consequence of the student's' choice.

Nevertheless, both virtue ethics and utilitarian ethics justify the introduction of these technologies because of their long-term effects on society and outweigh the argument siding with justice ethics. Virtue ethics stands for actions that allow students to reach their full potential, and in this case, developing these technologies for students to increase their chances of realizing their passion for STEM when they might have not done so otherwise accords with virtue ethics. In addition, a primary goal of the new printer parts and incubator is to further student learning in the classroom; and as understanding of the experiments and curriculum increases, so grows the ability for students to reach their full potential. Finally, the introduction of these technologies will not further inhibit any students directly or indirectly affected by their unequal distribution, and therefore will never decrease anyone's ability to reach their full potential.

Utilitarian ethics also justifies the introduction of these educational pieces because new technology will grow the learning outcomes of science and technology in the classroom. Assuming the balance of autonomous performance and student involvement the students who receive this technology will certainly benefit from using it. Even if only some students get access to this technology, and the equal distribution of it happens inexorably slowly, those first students will still help grow the chances for future STEM breakthroughs. Such breakthroughs will likely also be slow to affect the entire population, however, the entire population can eventually overall benefit from technological and medical advancements long-term as the technology becomes more affordable. The only way to drive down costs of technologies, however, is to first allow the existence of unequal access so that at least *someone* can continue to research and improve them.

12.3.4 Conclusion

In conclusion, the development of additional educational technology in the fields of biology and 3D printing is justified, because it can only help the students who receive it, assuming we address the curricular needs of the student. The inherent value behind this project cannot cause harm to anyone not benefitting from the technology, but it can foster great long-term possibilities for both direct recipients and other members of society. To ensure that these products meet ethical standards, they were designed with safety, cost, and educational value in mind. This technology will help our future technicians, scientists, doctors, and engineers realize their fullest potential, starting with their experience in a high school biology classroom.

12.4 Social Impact: The Effect on Education

As educational technologies, it is necessary to consider how the use of these products will affect the curriculum in the classroom. While technology typically is created to simplify human interaction in accomplishing a task, educational technology must instead only simplify inconsequential tasks and optimize the specific human interactions needed to achieve a learning outcome.

Because of this, the BETA team has identified which aspects of running a biology experiment are inconsequential to learning, necessary to learning, and significant to learning depending on the teacher's beliefs. In accordance with this, exporting file data has been simplified, all electronics are delivered setup and sealed, and the user interface is as simple and friendly as possible (no additional coding and minimal programming is needed). To enhance learning in biology, the camera programming will optimize periodic photo capture so students have a time lapse record of their experiment, and students will get both visual and hands-on experience with a 3D bioprinter and an incubation/lighting/imaging system to better simulate professional practices. The teacher will be able to choose whether or not the experiment is bioprinted or manually extruded by syringe, and if the lighting, temperature, and imaging frequency will be preprogrammed by him or herself or if the students will be programming these aspects as part of learning how to optimize the experiment conditions. By allowing for flexibility in how this technology will be used, the BETA team has ensured that these technologies will not undermine learning outcomes by oversimplifying tasks for students.

12.5 Economic Impact

The team balanced all functional and other criteria with its primary concern: low cost. Since the printer will be marketed to all schools, this quality device needed to be provided at an attractive price. Since teachers will likely need to write proposals to purchase the product with school funding, the team ensured the BETA Box could sell for $500 for greater availability, and that the additional costs to the printer were minimal.

At all phases of the design process, the cost of components and the corresponding value added for the customer were considered. For this reason, many pieces of equipment were chosen because of their low-cost, even though this decreased precision in some cases.

The expected result of the final cost of the BETA Box and modified printer is that most high schools can afford the equipment. The average annual high school biology budget is $5,000.

With the printer selling at $3,300, and the BETA Box selling at $500, both fall below the annual budget, and still leave funding for other experiment costs. Additionally, the printer and the Box can be sold separately, making the BETA Box individually accessible, and allowing for each product to be purchased in different years. While the purchase of these products will negatively affect other purchasing opportunities, the BETA team anticipates that the educational benefits will far exceed the economic costs.

Part V. Conclusion

13. Conclusion

13.1 Completed Work

The objective of the 3D Bioprinting project was to improve biology classroom engagement through the development of technologies for high school biology experiments. The BETA project's first objective was to create a modular incubating Box for the high school classroom. This Box, called the BETA Box was designed with a variety of sensors to allow for custom temperature and lighting environments per experiment. The other objectives included conducting a feasibility study on auto calibration and dual-extrusion for SE3D's existing 3D bioprinter.

The main goals for the BETA Box were to be able to sell to the customer at $500; be capable of handling 36 students; control temperature between 20-50°C with a heat up time of under 30 minutes; have full control over the intensity, uniformity, and color of the interior; and be easily usable by high school and middle school students. The main tasks for the feasibility studies were to have a homing mechanism for the z-axis and to add a PLA extruder to the bioprinter.

Upon completion of this project, the BETA Box was a fully operational product that could immediately be used in the classroom. The BETA Box easily beat the requirement of one picture a second and provided lighting over the full RGB spectrum, meeting the requirements for an experimental recording device. As an incubator, the BETA Box easily beat the desired warm up time of 30 minutes, completing warm up to 37 °C in just 5 minutes. Furthermore, the temperature sensor data proved that the incubation bed was within the desired $\pm$2 °C across the entire area, which proved that the bed was being uniformly heated. The interface was designed to be both intuitive and programmable to run a diverse set of experiments. And, most importantly, the BETA Box achieved complete success while remaining beneath the $500 selling price, making it a viable addition to the high school classroom. While the BETA Box is a fully capable product, improvements to the manufacturing process can be implemented, such as thermoforming most of the Box as a single piece, which will simplify as well as speed up production and assembly.

Both feasibility studies on the bioprinter were successful. The force sensor was able to accurately calibrate the syringe for printing and notify the user when empty while retaining the simple design of the original bio extruder. The dual extruder proved that two materials could be printed simultaneously, which will enable future 3D bioprinting despite lacking precise alignment between the two extruders for this particular design. The team was also able to provide a pathway that will enable a future team to further improve the dual extruder and create SE3D's first 3D biomaterial structure.

13.2 Future Work

Next steps for this project include an in-depth commercialization analysis for the BETA Box. This begins with a 48-hour running safety test to evaluate the BETA Box's proficiency in meeting UL standards. The box will need to be redesigned for manufacturing on a larger scale. A printed circuit board will need to be designed to merge all the electronics onto one board. The thermal distribution can be further improved inside the box.

For the User Interface, an admin only panel was identified as something to be added for teacher use. The software should be able to crop each camera image depending on what dish bed is being used and the memory management of all the photo files can be improved based on cropping the images. It was also identified that the GUI should default back to the dashboard when not being used.

For the 3D bioprinter enhancements, the firmware modifications for dual extrusion and auto calibration needs to be further tested and implemented into the Marlin libraries for easy distribution. The force sensor also needs to be integrated into the software libraries. The parts need to be standardized for the dual extruder addition so they are easy to duplicate with any standard manufacturing practice. The bio-extruder needs to be redesigned by using a fixed extrusion syringe so the height of the syringe does not change during experiments.

Long term goals for the future work would include a complete redesign of the existing bio-extruder for a modular implementation of either a bio-extruder or a PLA extruder. This would include the addition of material plates in the back of the printer so that more material can be used during prints. Ideally, this printer could have four or five extruders of different materials if it used a different type of design. With the addition of more extruders, more materials could be printed, but the extruders would need to be able to clean themselves so they do not contaminate the bio print. Getting the printer and the BETA Box into classrooms around the world is the ultimate goal of this project. The BETA Box can be tweaked for use from middle school all the way up until college and at the $500 price point, the box can be sold to classrooms in less developed countries who cannot afford expensive incubators.

References

[1] ACT, Inc. "National STEM Report - Attainment of College and Career Readiness." *The Condition of STEM 2016—National* (2016): n. pag. *Act.org*. ACT, 2016. Web. 29 Nov. 2016.

[2] Yordanova, Snejana. "Intelligent Approaches For Linear Controllers Tuning With Application To Temperature Control." *Journal of Intelligent & Fuzzy Systems* 27.6 (2014): 2809-2820.

[3] Gao, Qing et al. "Coaxial Nozzle-Assisted 3D Bioprinting with Built-in Microchannels for Nutrients Delivery." *Biomaterials*, vol. 61, 2015, pp. 203–215.

[4] Hergel, Jean, and Sylvain Lefebvre. "Clean Color: Improving Multi-Filament 3D Prints." Computer Graphics Forum 33.2 (2014): 469-478. *Applied Science & Technology Source.* Web.

[5] Sinopoli, B., C. Sharp, L. Schenato, S. Schaffert, and S.s. Sastry. "Distributed Control Applications within Sensor Networks." *Proceedings of the IEEE* 91.8 (2003): 1235-246. Web.

[6] Narayanan, Lokesh Karthik, Pedro Huebner, Matthew B. Fisher, Jeffrey T. Spang, Binil Starly, and Rohan A. Shirwaiker. "3D-Bioprinting of Polylactic Acid (PLA) Nanofiber–Alginate Hydrogel Bioink Containing Human Adipose-Derived Stem Cells." *ACS Publications*. 6 July 2016. Web. 14 Nov. 2016.

[7] Grieser, Franz. "PLA vs ABS: Filaments for 3D Printing Explained & Compared." *All3DP*. 18 Apr. 2017. Web. 23 Apr. 2017.

[8] Carolina Biological Supply Company. "Incubator, Still-Air Picture Window." *Carolina.com*. Carolina Biological Supply Company, 2017. Web. 3 Mar. 2017.

[9] Incubators. "Reptipro 6000 Incubator With One Automatic Egg Turner." *Incubators*. Incubators.org, 2013. Web. 3 Mar. 2017.

[10] Incubators. "R-Com Pro 20 With USB Incubator." *Incubators*. Incubators.org, 2014. Web. 3 Mar. 2017.

[11] Thermo Fisher Scientific. "Precision Compact Oven" *Thermo Fisher.* Thermo Fisher Scientific, 2017. Web. 3 Mar. 3017.

[12] Gosepath, Stefan. "Equality." *Stanford Encyclopedia of Philosophy*. Stanford University, 27 June 2007. Web. 15 Mar. 2017.

Part VI. Appendices

A - Product Design Specifications

Table A-1: Product Design Specification Baseline Deliverables – BETA Box

Performance			
Requirements	**Units**	**Datum**	**Target Range**
Incubation Temperature	Celsius	N/A	±2 from desired temperature
Incubation Humidity	Percent humidity	N/A	25-75
The Box Capacity	Petri dishes	N/A	9
Box Size	Inches	N/A	< 15”x15”x 15”
Incubation Chambers	Number of chambers	0	1
Camera System	MegaPixel	0	4-12
Incubator Lighting Input Power	VDC	N/A	12
Incubator Lighting	Watts	White Light	RGB Spectrum
Total Cost	USD	N/A	500
Lifetime Use	Hours	N/A	1000
Software			
Requirements	**Units**	**Datum**	**Target Range**
Timed Imaging	Images/Interval	Manual	± 1 second response
User Interface		External Computer	Display
Safety			
Requirements	**Units**	**Datum**	**Target Range**
Manual Abort Shutdown – Incubator	Seconds	N/A	20

Table A-2: Product Design Specification Baseline Deliverables - Printer

Performance			
Requirements	**Units**	**Datum**	**Target Range**
Z-Axis Auto-homing	mm	N/A	1 ± 0.1
Air Bubble Error	Percent of biomaterial	N/A	5 ± 0.5%
Accept Dual Extruders	Extruders	N/A	> 2
Accept Plastic Filament	mm	1.75	1.75 ± 0.05, 3.00± 0.05
Power Usage	Watts	105	100±10
Software			
Requirements	**Units**	**Datum**	**Target Range**
Standalone printer		Must be connected to External Desktop Computer	Microcomputer
User Interface		External Computer	Display
Safety			
Requirements	**Units**	**Datum**	**Target Range**
Manual Abort Shutdown - Printer	Seconds	20	10-15

B - Decision Matrices

Project: senior design one
System: The Box
Date: 8-Nov-16

	Criterion	1	2	3	4	5	6	7	8	9	10	11	12	SUM	FACTOR
1	Cost		1	1	1	0.5	1	1						5.5	5
2	Durability	0		1	1	0	1	1						4	4
3	Accuracy	0	0		1	0	0	1						2	4
4	Weight	0	0	0		0	0	0						0	1
5	Safety	0.5	1	1	1		1	1						5.5	3
6	Usability	0	0	1	1	0		1						3	4
7	Size	0	0	0	1	0	0							1	3
8		1	1	1	1	1	1	1						7	
9		1	1	1	1	1	1	1	1					8	
10		1	1	1	1	1	1	1	1	1				9	
11		1	1	1	1	1	1	1	1	1	1			10	
12		1	1	1	1	1	1	1	1	1	1	1		11	

Fill in Purple squares above

Fill in upper triangle of the matrix

Working across each row, determine if the criterion in that row is more important (1), same importance (0.5) or less important (0) than the criterion in that column

Assign weighting factors for each criterion in the Yellow squares

Figure B-1: Weighted Criteria Table

Design Project = senior design one System= The Box Date= 8-Nov-16

CRITERIA	TARGET or FACTOR	1 = Baseline		The Cube		Circular Incubator		Cube String		The Pie		Sliding Door Box		Open Box		Multiple access		Cylindrical rotating		Conveyer Belt Box	
Time – Design	2	2		2		1		3		1		3		4		3		4		5	
Time – Build	10	10		8		10		15		50		10		20		20		30		15	
Time – Test	10	10		5		10		10		100		10		15		10		12		12	
Time Score	10		10		7.67		8.33		13.33		51.67		11.67		18.33		15.00		20.67		17.33
Cost – Prototype	500	$ 500.00		$ 400.00		$ 700.00		$600		########		#######		########		#######		########		########	
Cost – Production	200	$200		$ 200.00		$ 300.00		$400		$ 500.00		#######		$ 400.00		#######		$ 400.00		$ 700.00	
Cost Score	10		10		9.00		14.50		16.00		22.50		10.00		20.00		14.50		25.00		32.50
Cost	5	3	15	5	25	5	25	4	20	2	10	5	25	2	10	3	15	1	5	1	5
Durability	4	3	12	5	20	3	12	4	16	2	8	2	8	2	8	3	12	3	12	1	4
Accuracy	4	3	12	3	12	2	8	2	8	2	8	3	12	1	4	4	16	4	16	2	8
Weight	1	3	3	4	4	2	2	5	5	3	3	4	4	3	3	1	1	2	2	3	3
Safety	3	3	9	3	9	3	9	3	9	2	6	3	9	4	12	2	6	3	9	4	12
Usability	4	3	12	1	4	4	16	3	12	2	8	5	20	2	8	3	12	2	8	2	8
Size	3	3	9	2	6	2	6	4	12	5	15	2	6	3	9	3	9	2	6	1	3
0	0	3	0		0		0				0		0		0		0		0		0
0	0	3	0		0		0				0		0		0		0		0		0
0	0	3	0		0		0				0		0		0		0		0		0
0	0	3	0		0		0				0		0		0		0		0		0
0	0	3	0		0		0				0		0		0		0		0		0
	TOTAL		72.0		83.3		75.2		72.7		3.8		82.3		35.7		61.5		32.3		13.2
	RANK																				
	% MAX		86.4%		100.0%		90.2%		87.2%		4.6%		98.8%		42.8%		73.8%		38.8%		15.8%
	MAX	83.3																			

NOTE: User fills in Purple areas, gold areas are calculated or fixed
Light blue areas filled from prioritizing matrix

BASELINE = The Box

Timescore(i)=Timescore(B)*(TD(i)/TD(B) + TB(i)/TB(B) + TT(i)/TT(B))/3
Costscore(i)=Costscore(B)*(Cprot(i)/Cprot(B) + Cprod(i)/Cprot(B))/2
Total(i) = SUM(Factor(j)*Comparison(i,j)) + (Timescore(B)-Timescore(i)) + (Costscore(B) - Costscore(i))

Comparison(i,j) = 5 if idea "i" is much better than baseline for criteria "j"
Comparison(i,j) = 4 if idea "i" is better than baseline for criteria "j"
Comparison(i,j) = 3 if idea "i" is same as baseline for criteria "j"
Comparison(i,j) = 2 if idea "i" is worse than baseline for criteria "j"
Comparison(i,j) = 1 if idea "i" is much worse than baseline for criteria "j"

Design Idea Descriptions

2 The Cube
3 Circular Incubator
4 Cube String Incubator
5 The Pie
6 Sliding Door Box
7 Open Box
8 Multiple access points box
9 Cylindrical rotating incubator
10 Conveyer Belt Box

Figure B-2: Selection Matrix

C - Detail Drawings

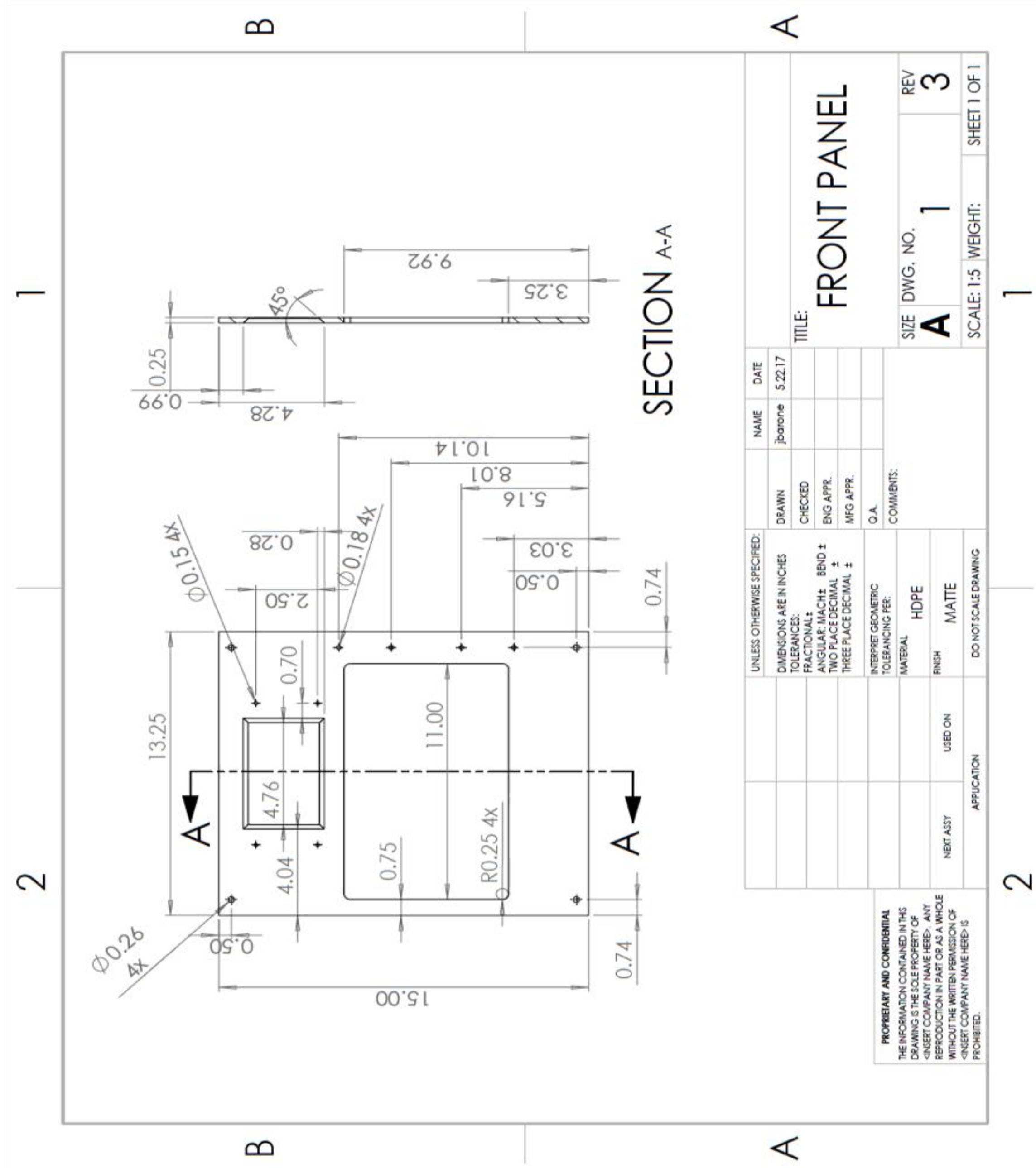


Figure C-1: Front Panel

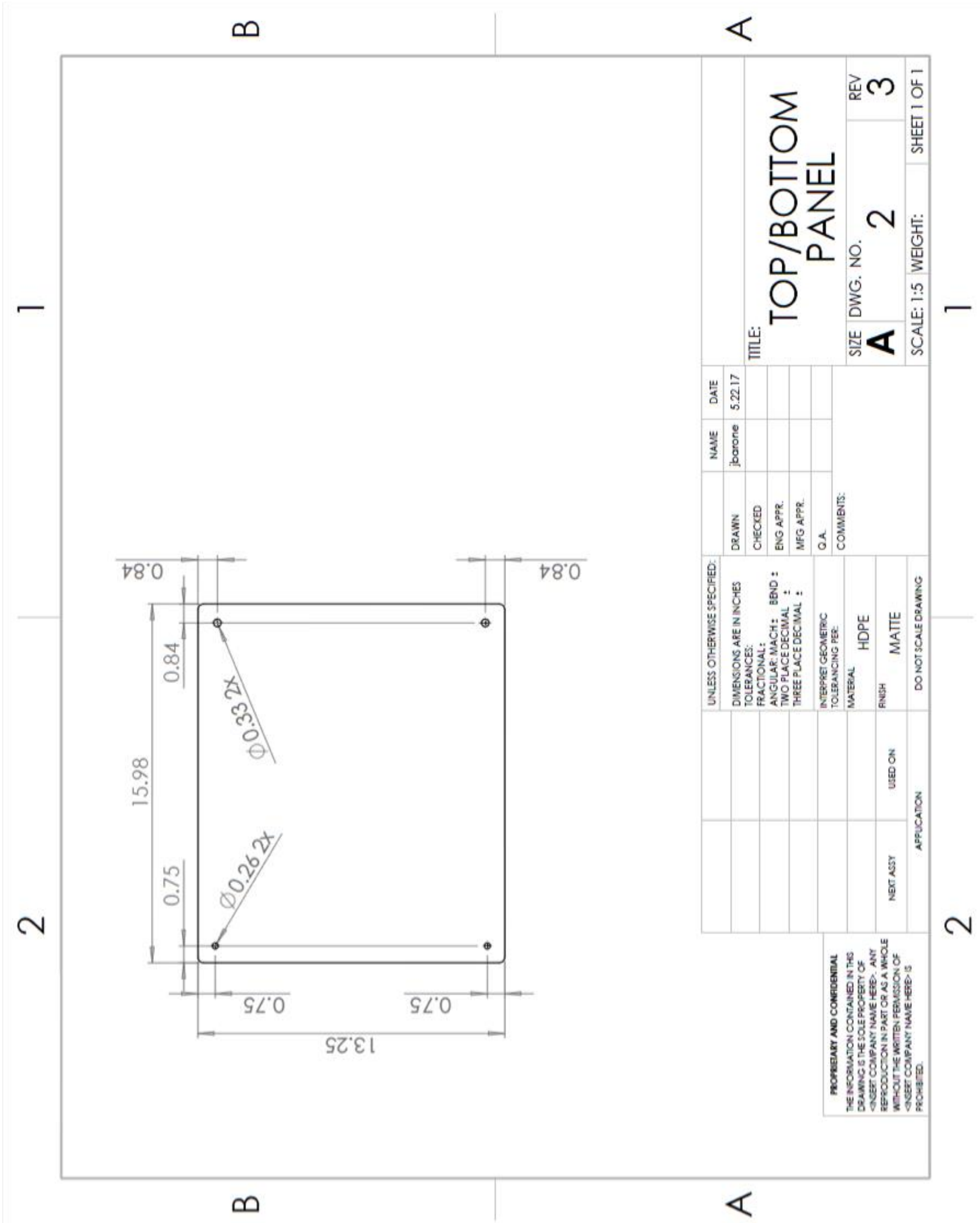


Figure C-2: Top and Bottom Panel

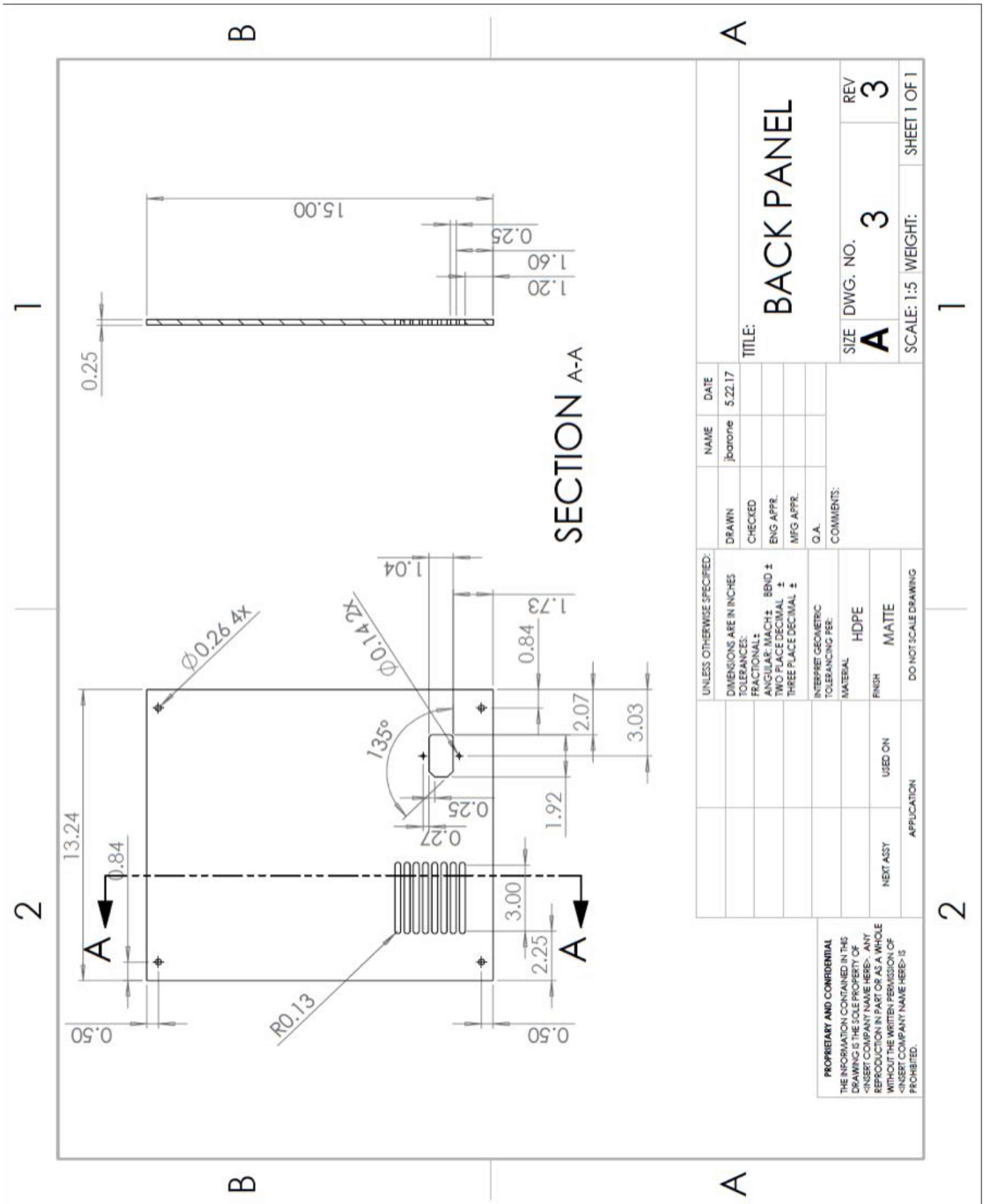


Figure C-3: Back Panel

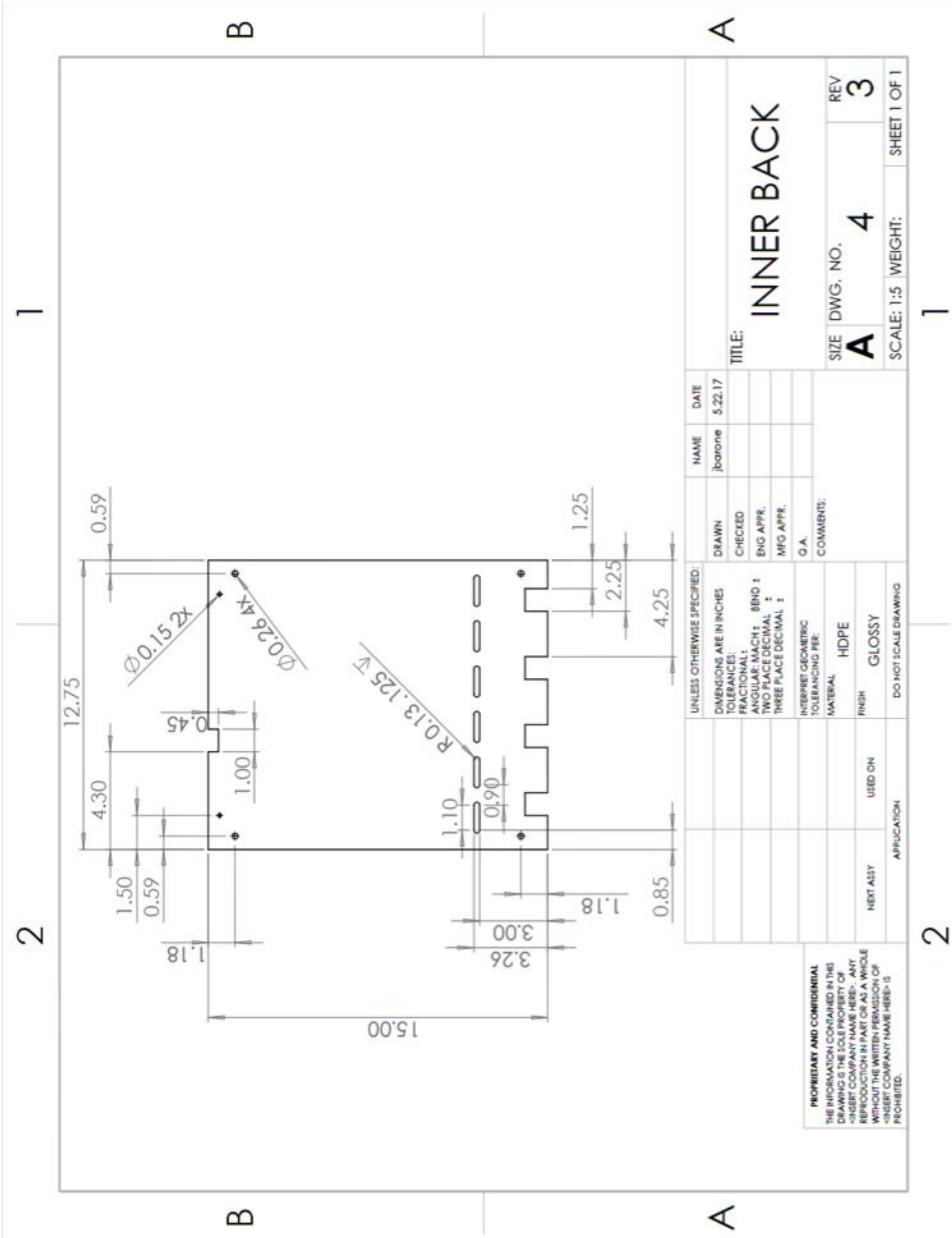


Figure C-4: Inner Back Panel

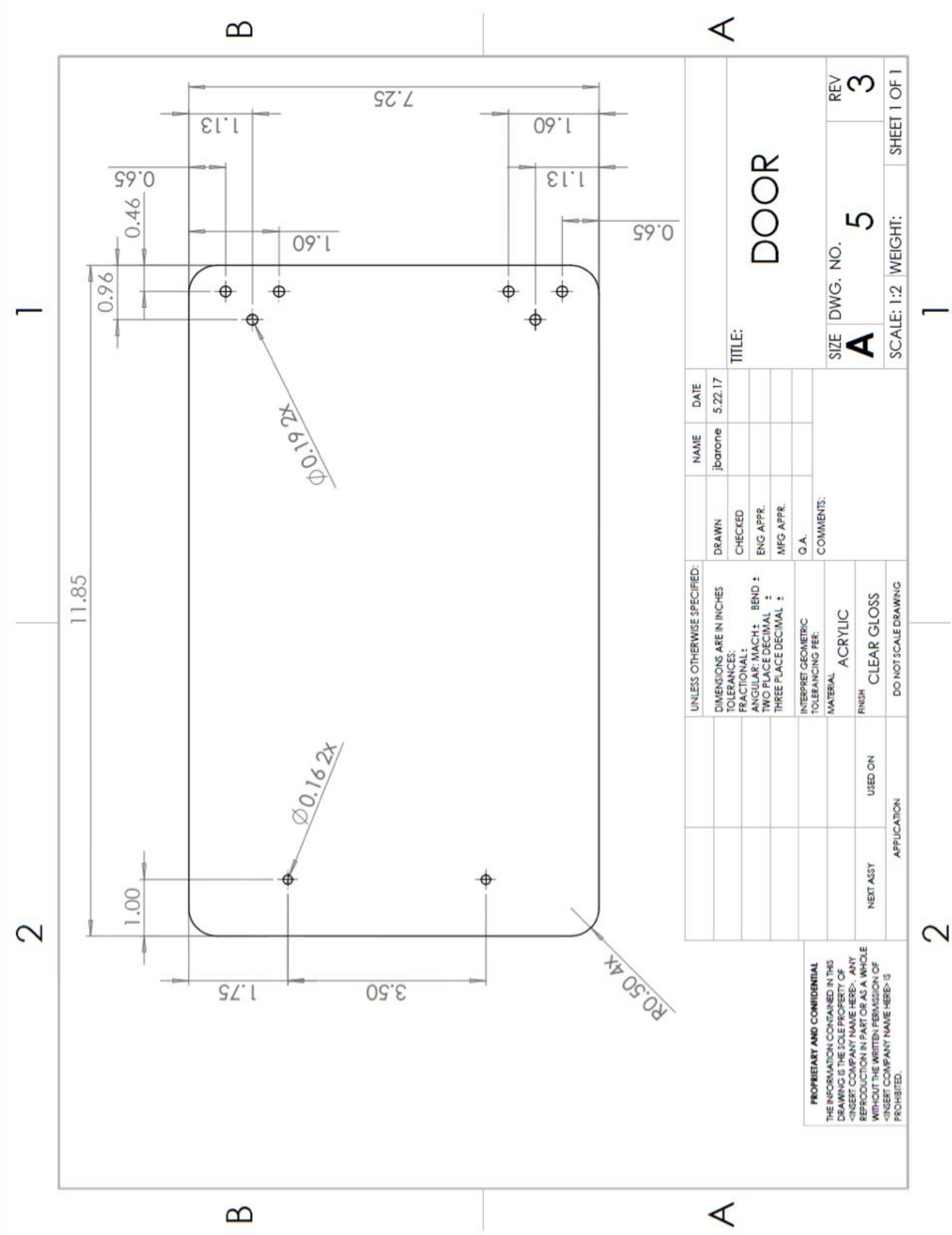


Figure C-5: Door

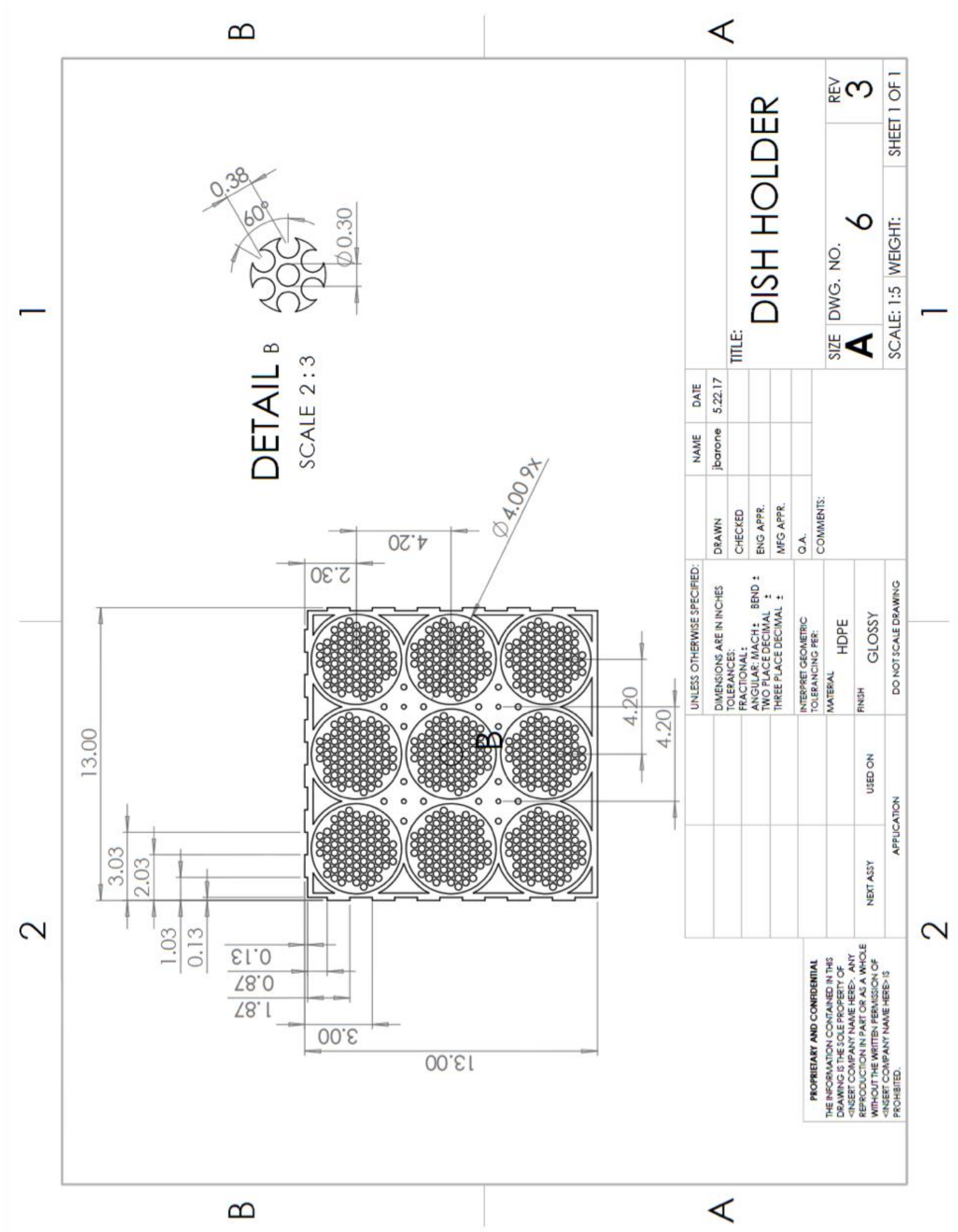


Figure C-6: Dish Holder

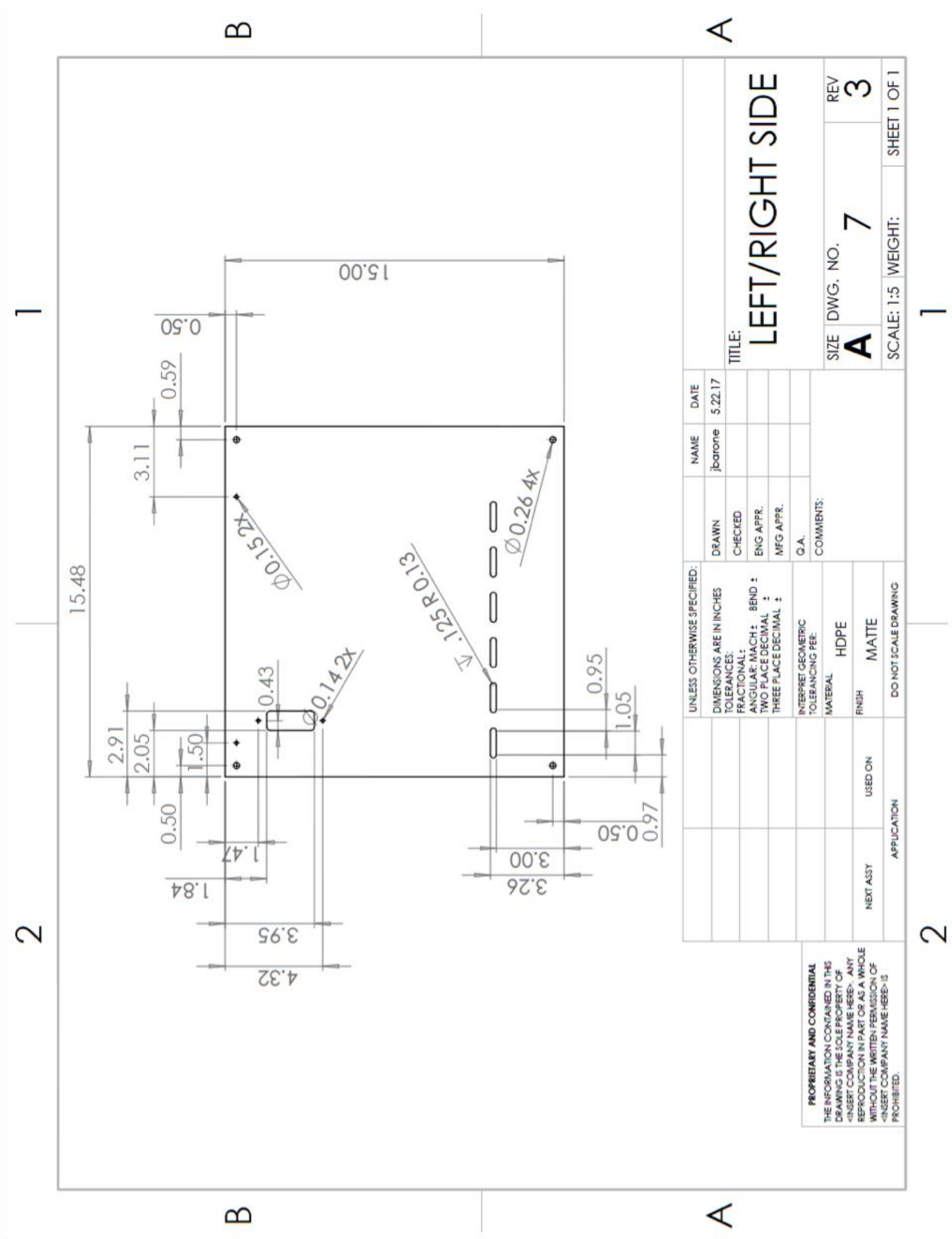


Figure C-7: Left and Right Side Panel

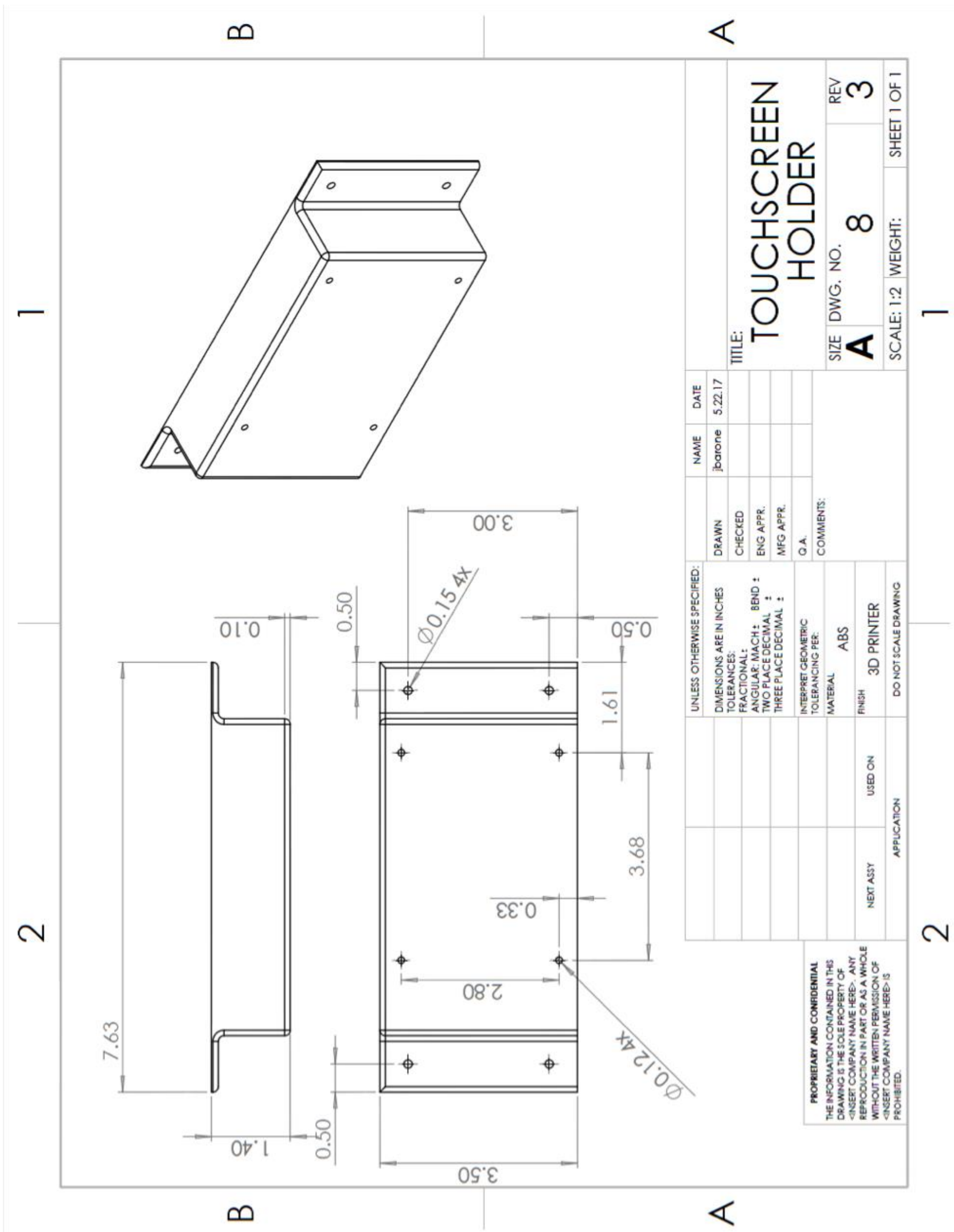


Figure C-8: Touch Screen Holder

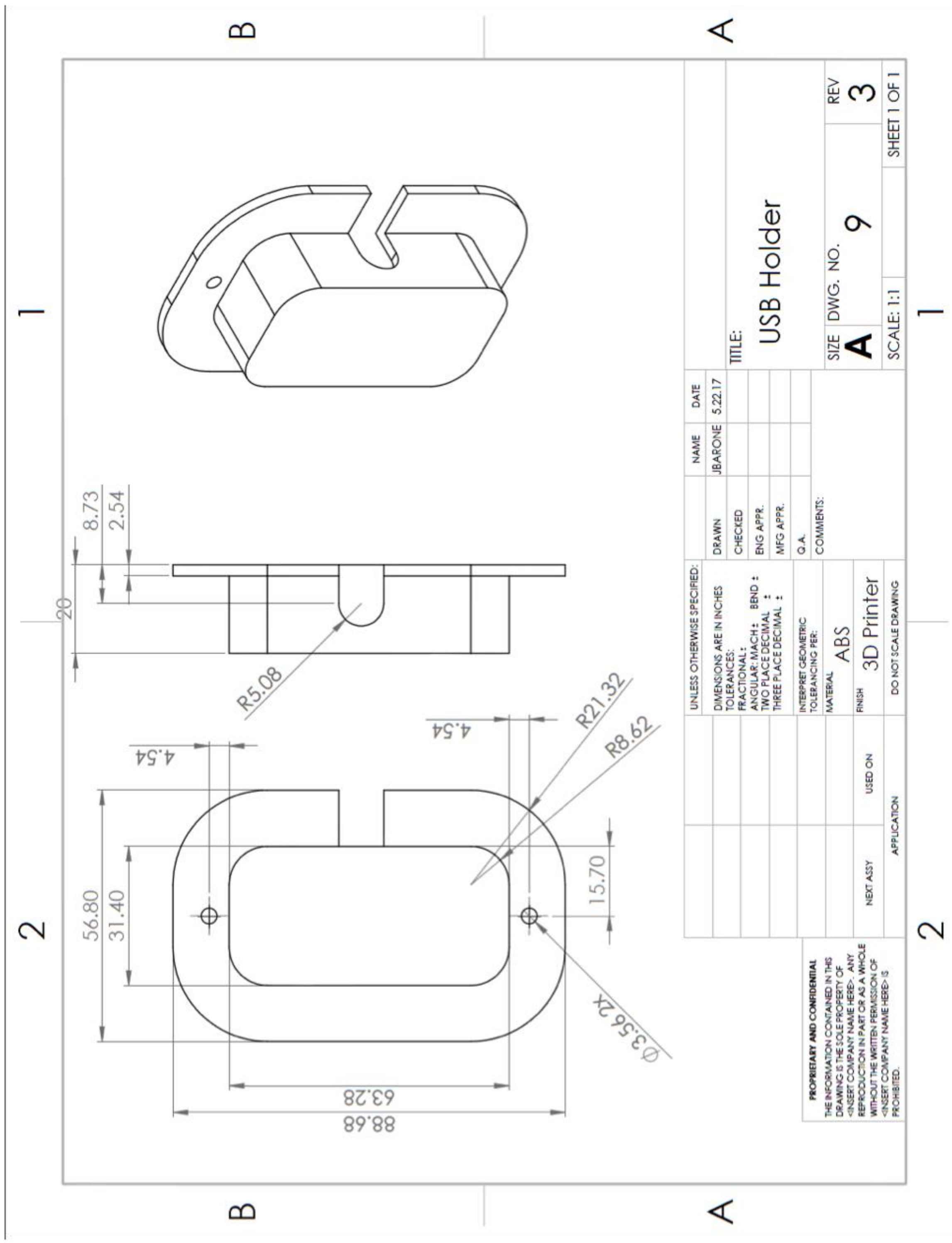


Figure C-9: USB Holder

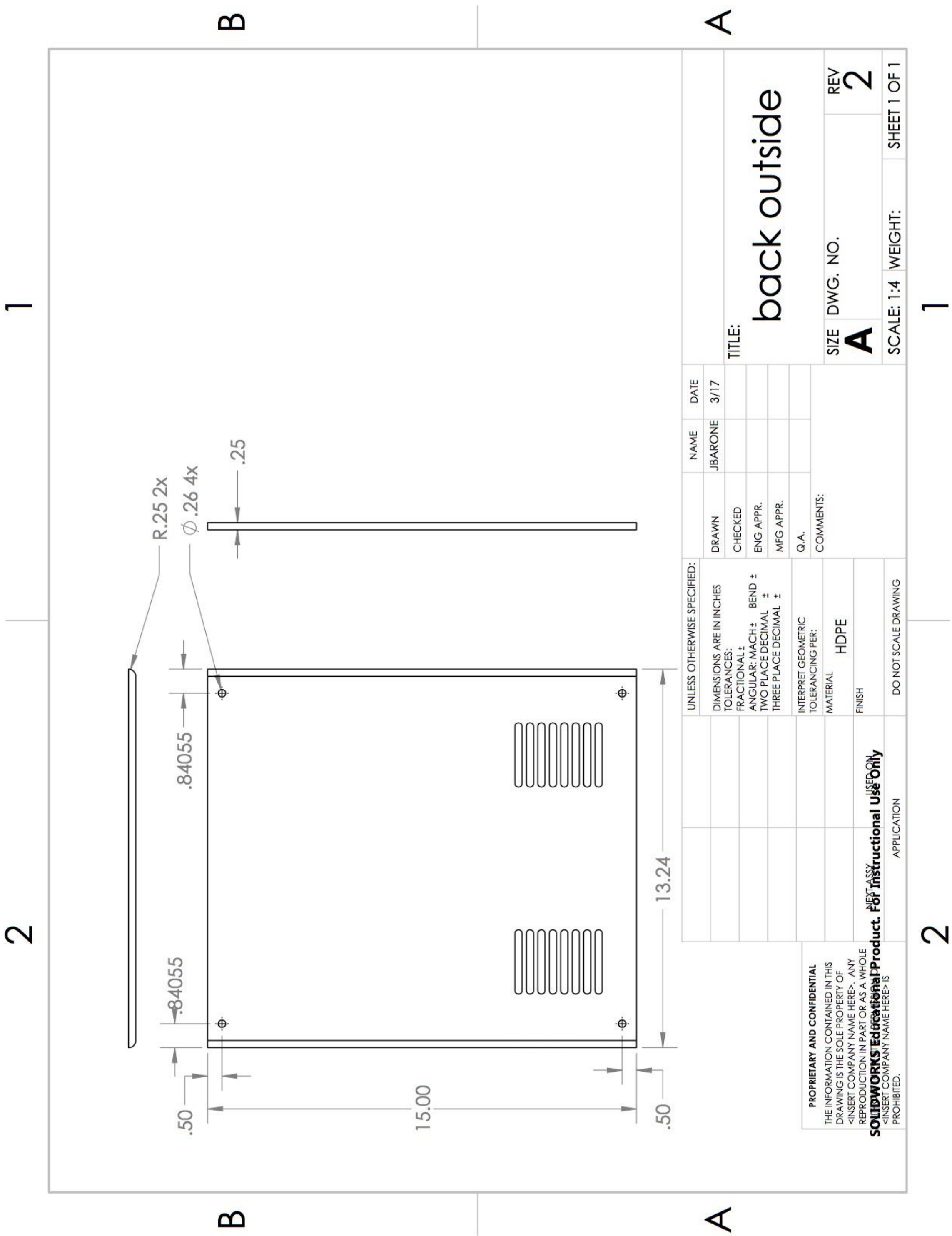


Figure C-10: Back Outside Panel

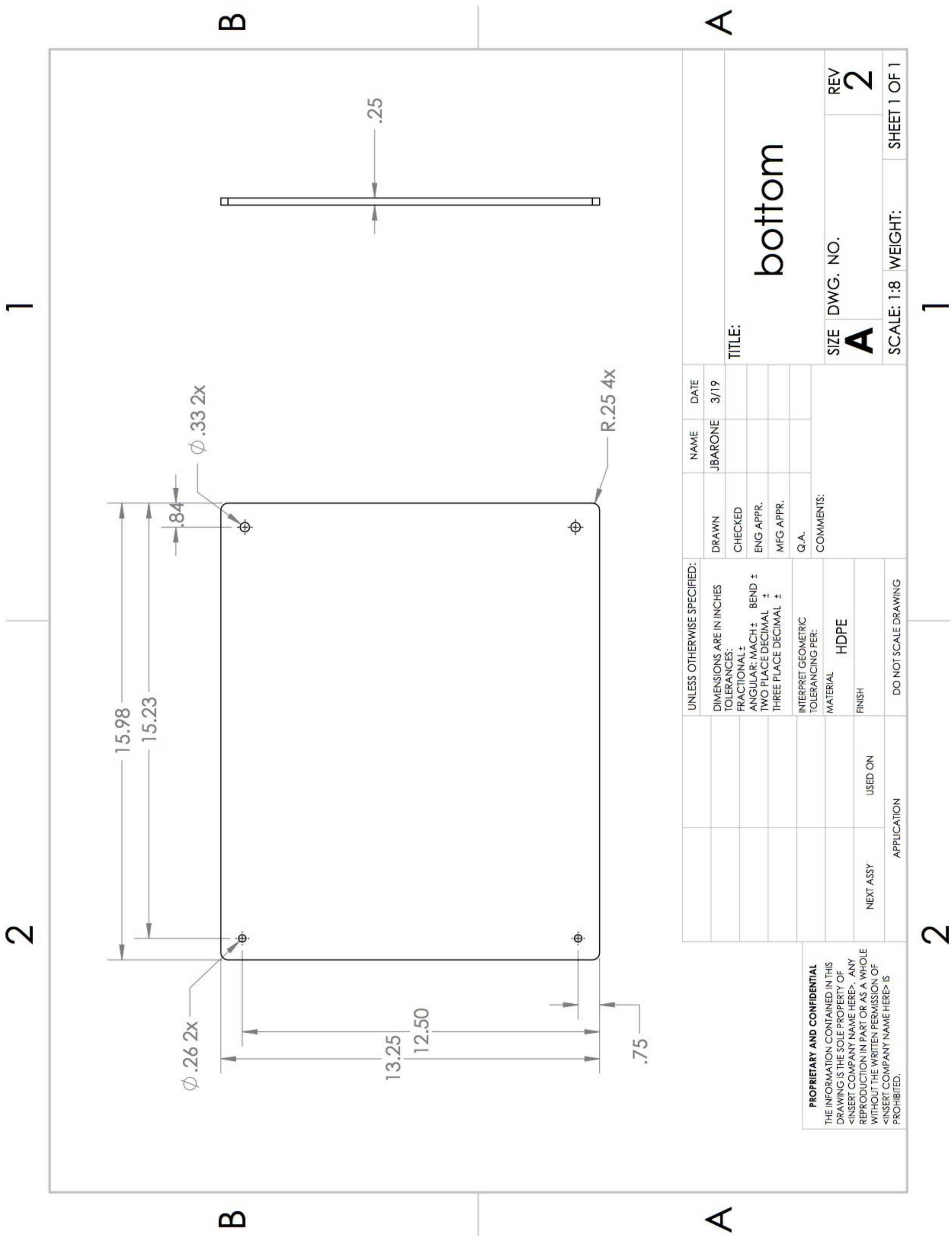


Figure C-11: Bottom Panel

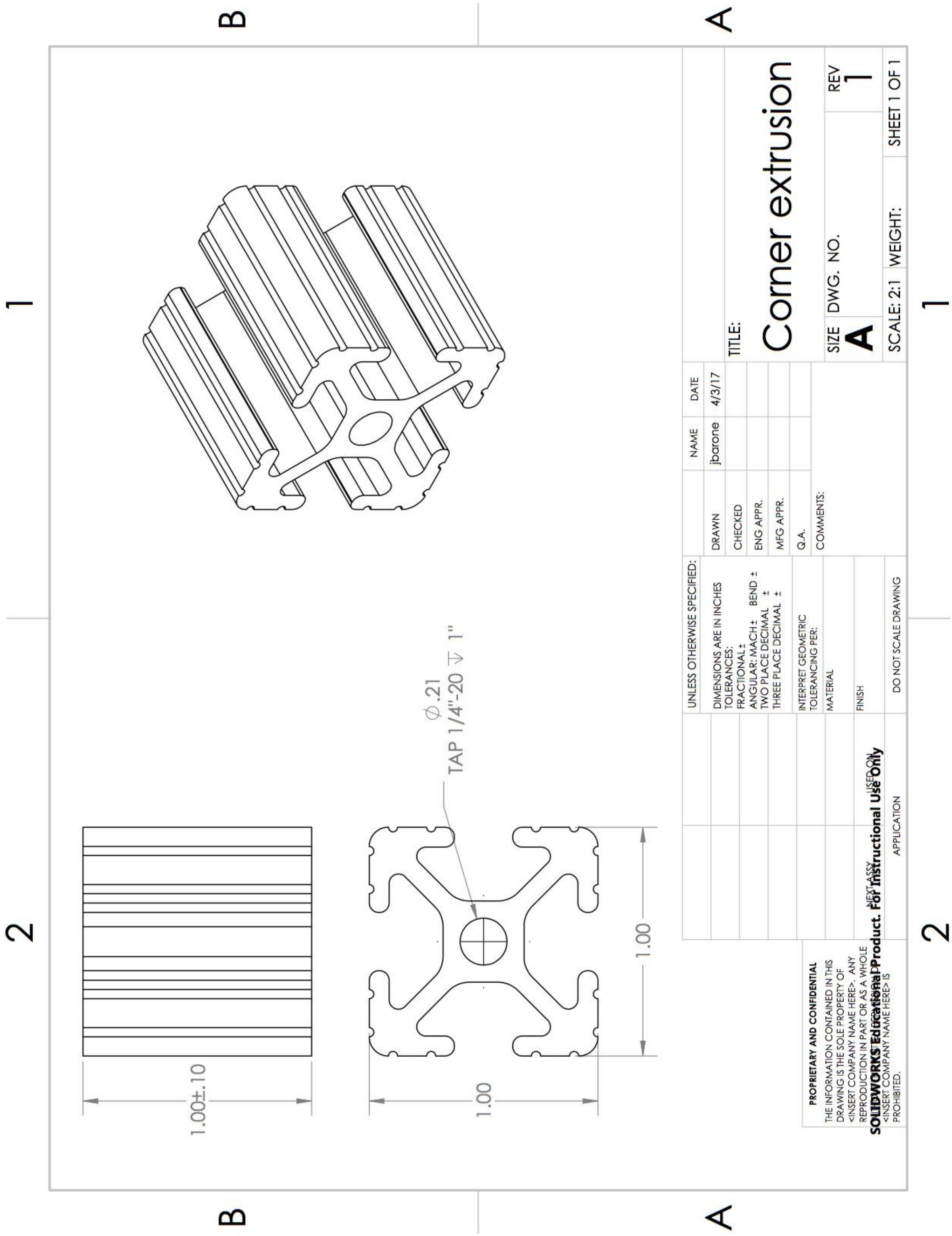


Figure C-12: Corner Extrusion

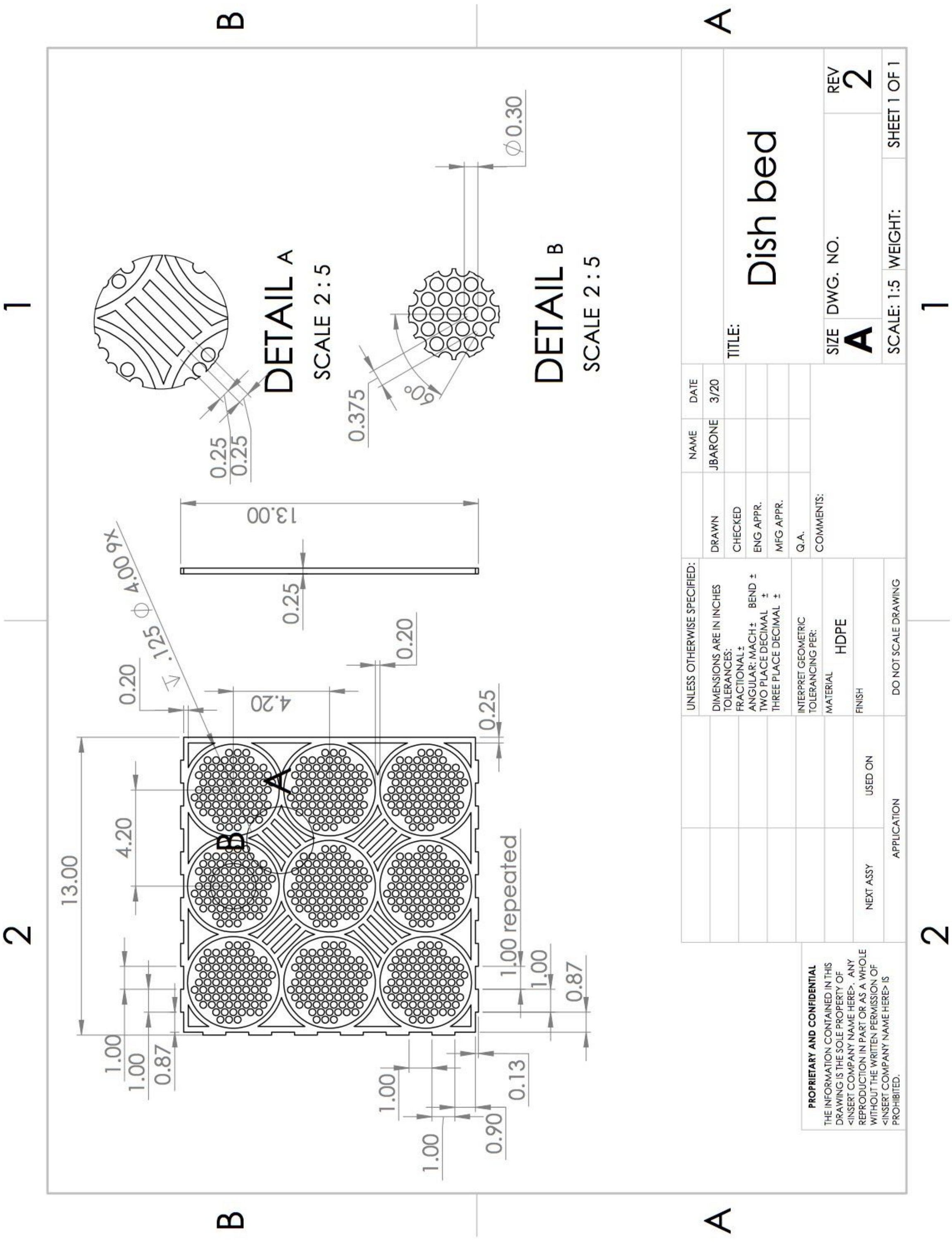


Figure C-13: Dish Base Holder

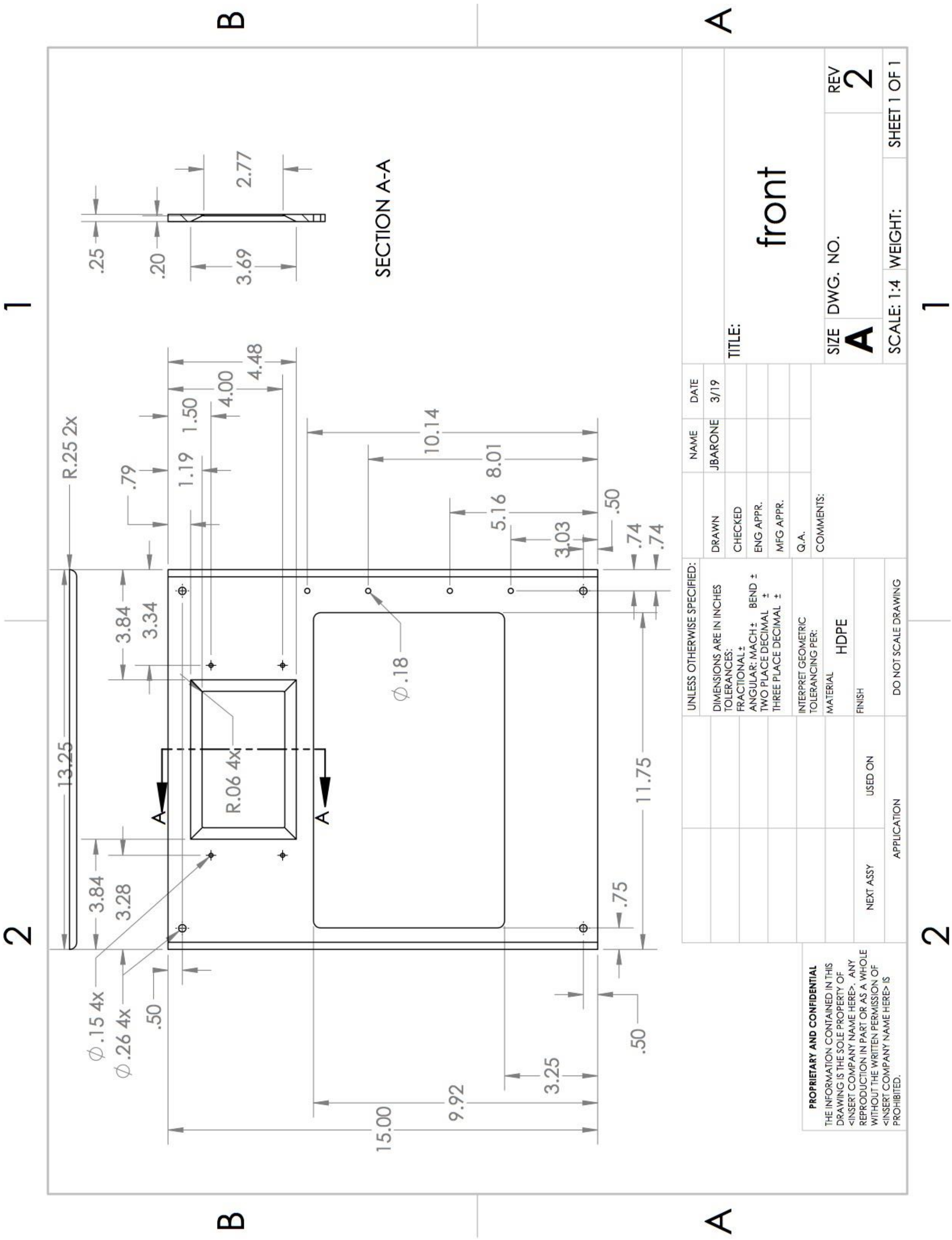


Figure C-14: Front Panel

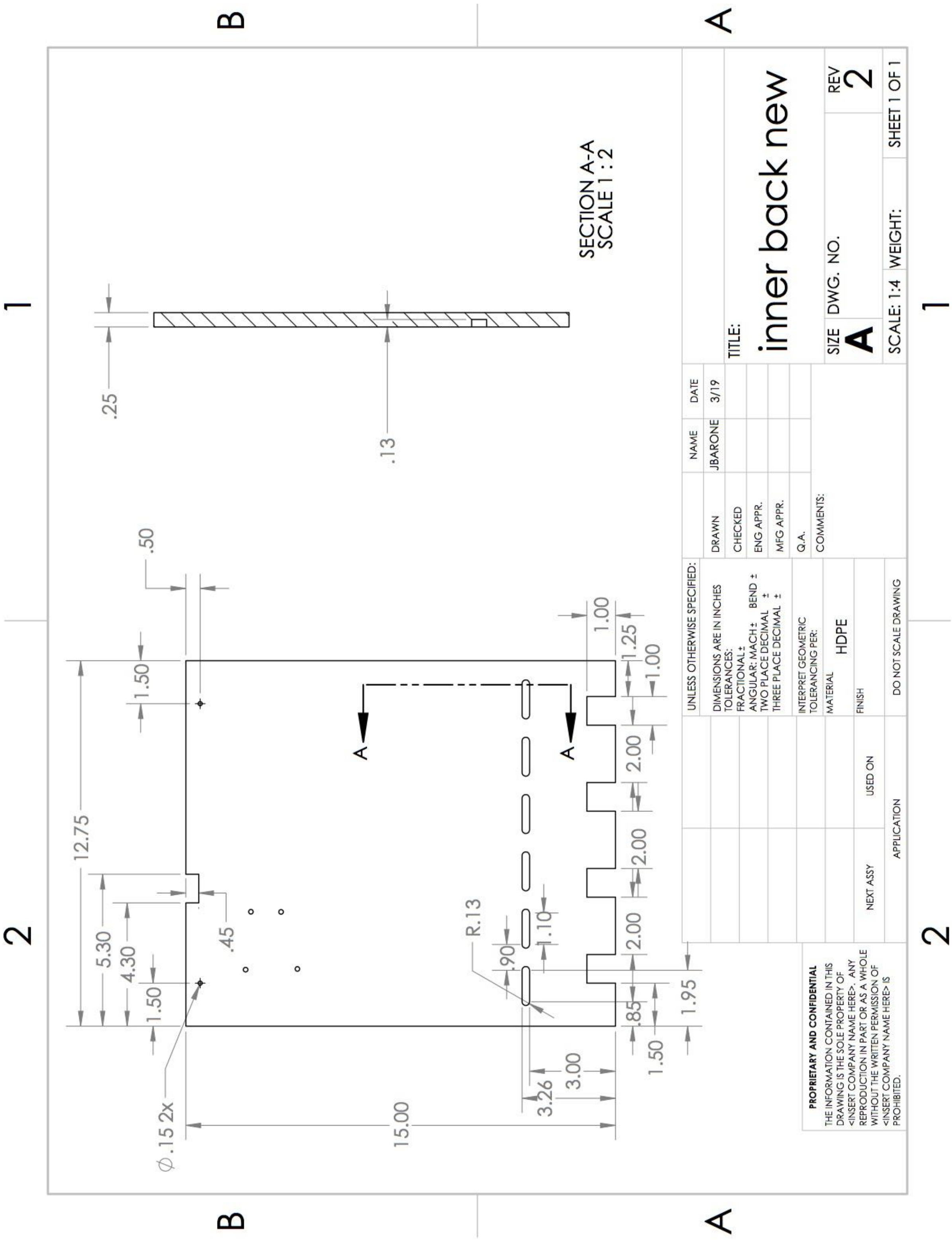


Figure C-15: Inner Back Panel

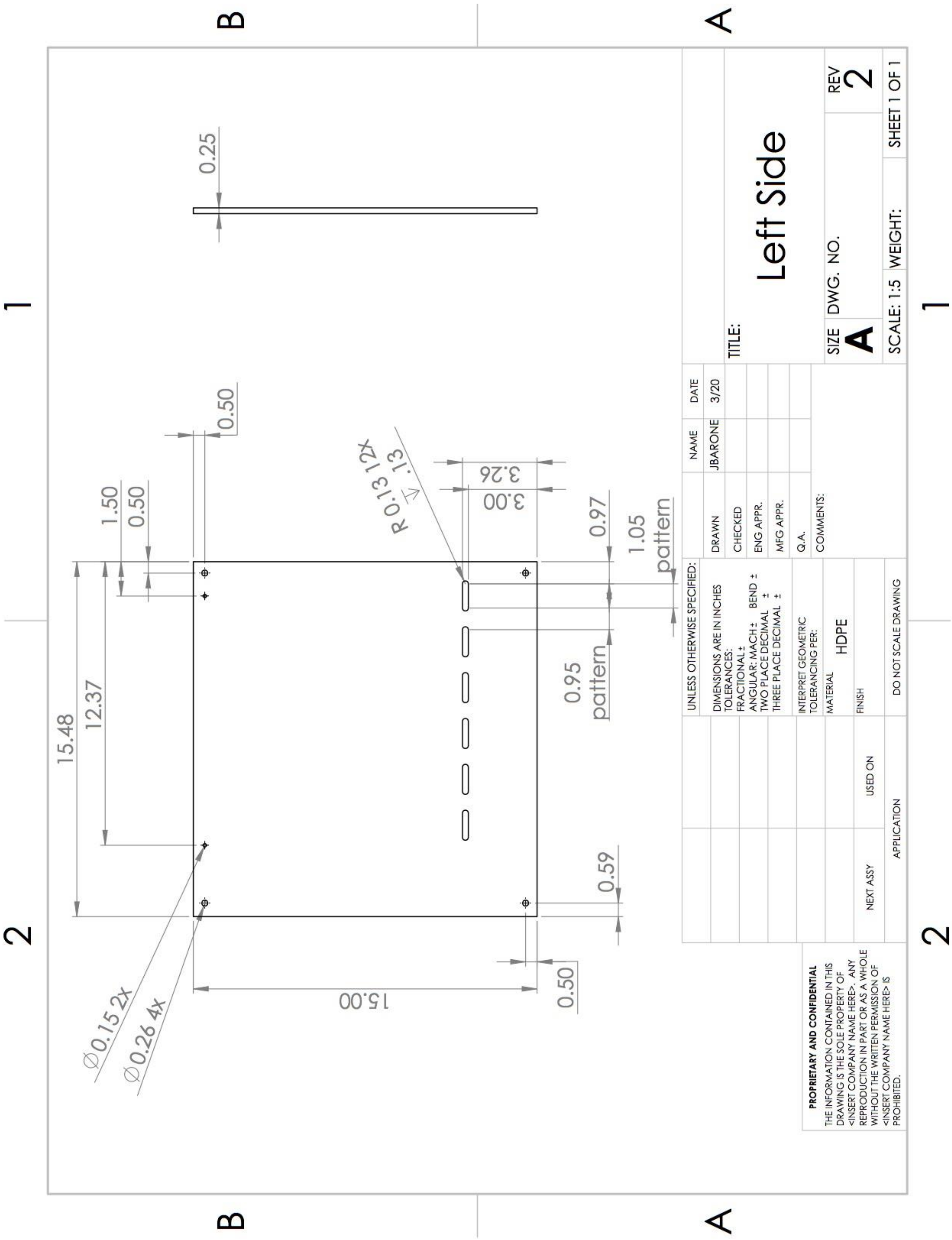


Figure C-16: Left Side Panel

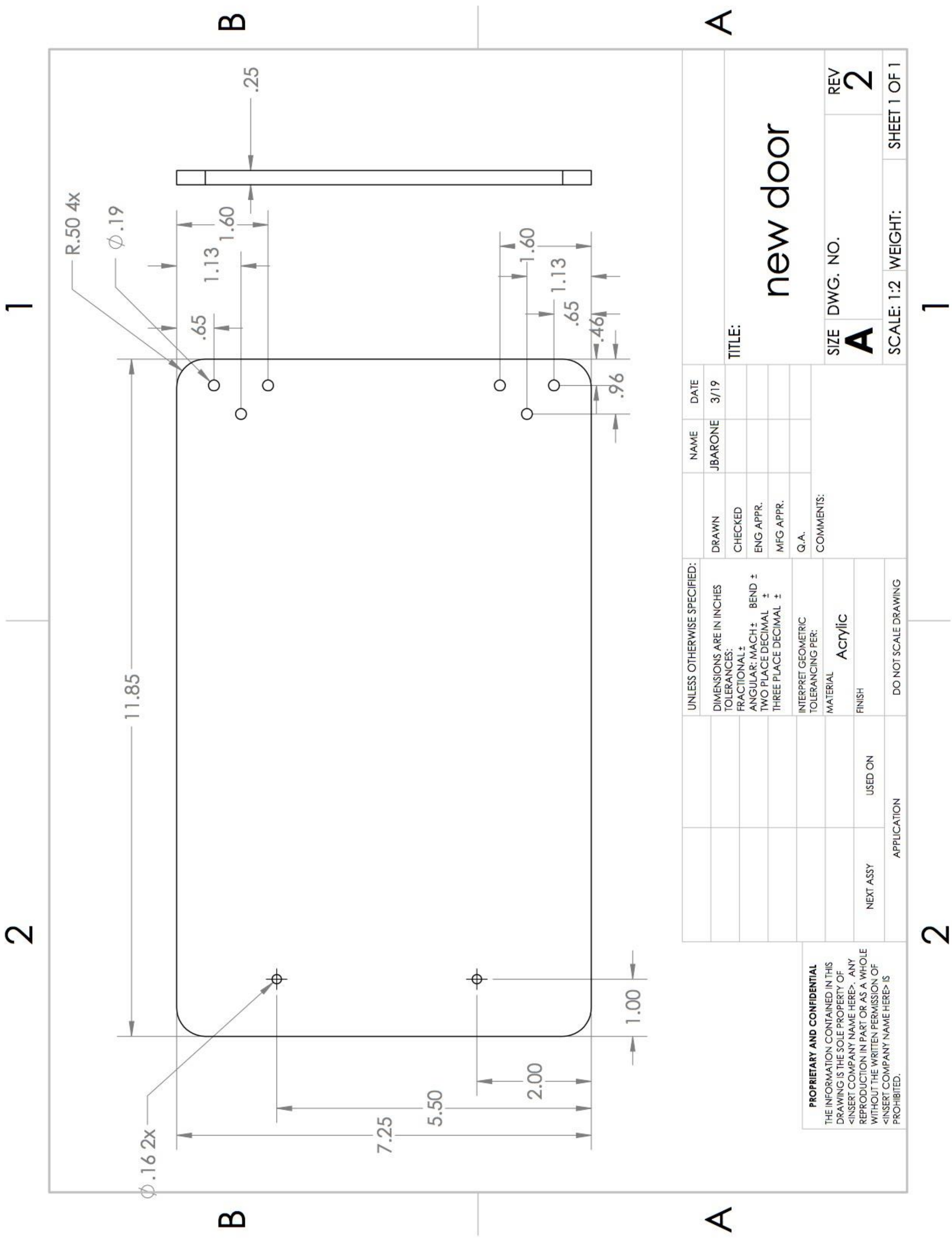


Figure C-17: Door

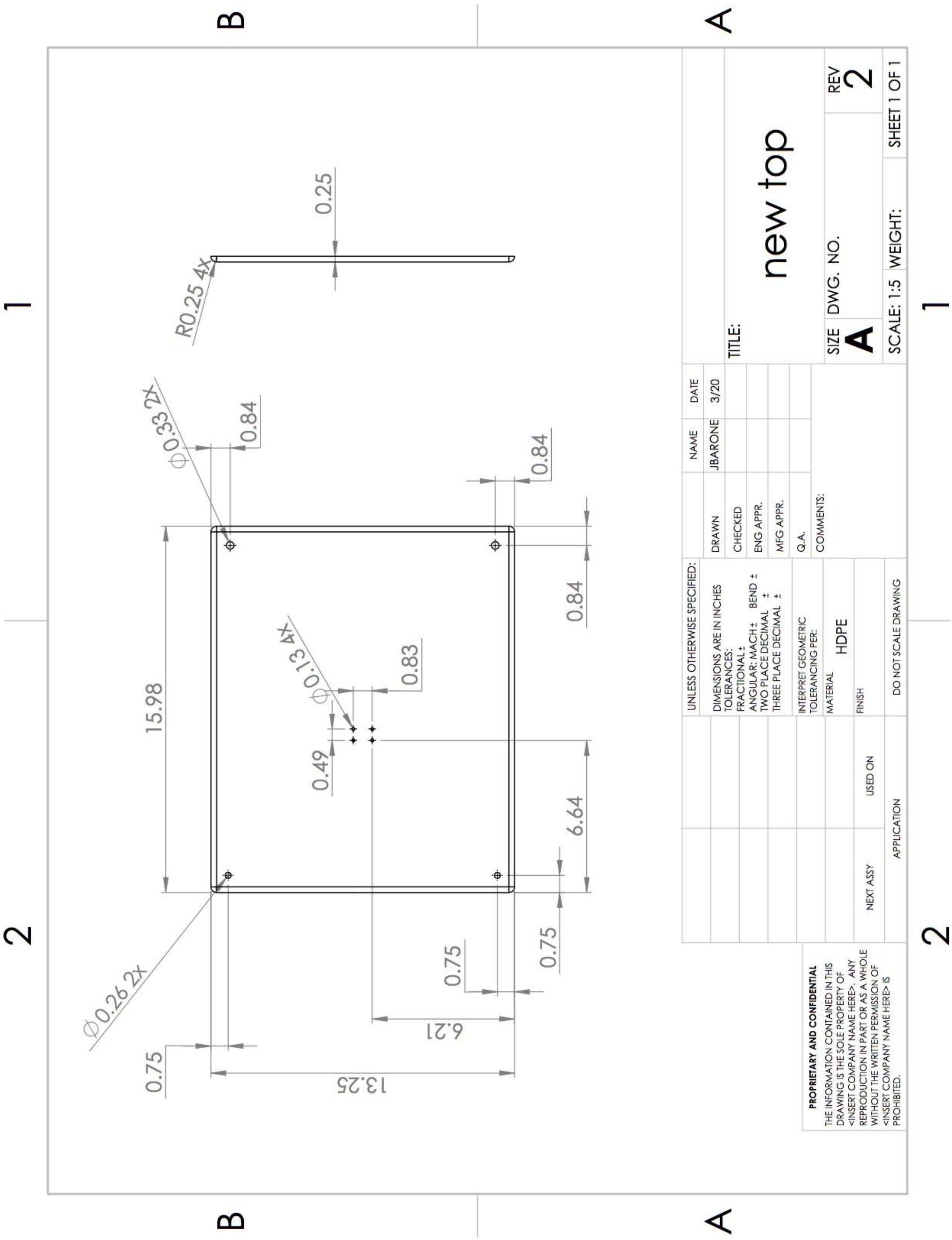


Figure C-18: Top Panel

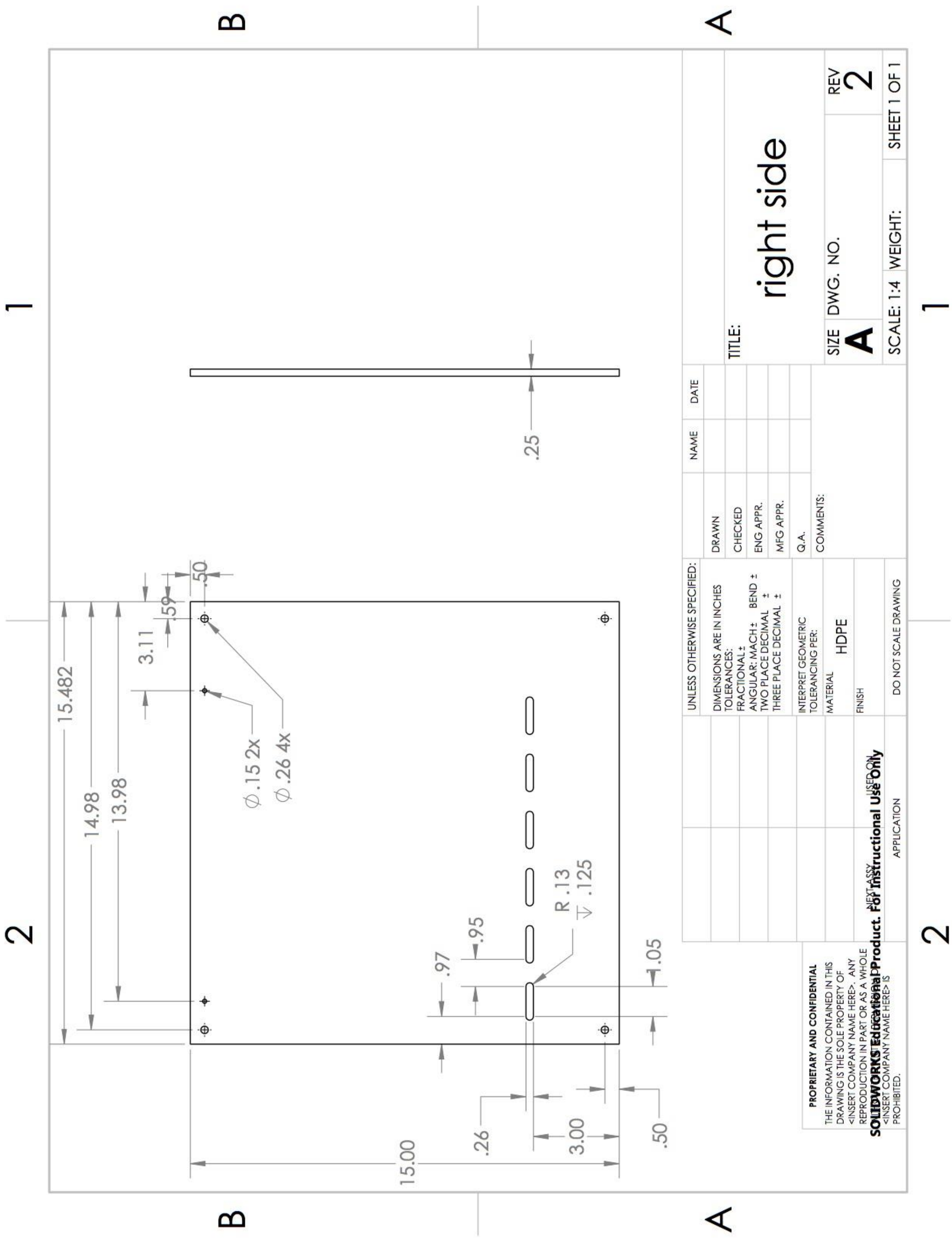


Figure C-19: Right Side Panel

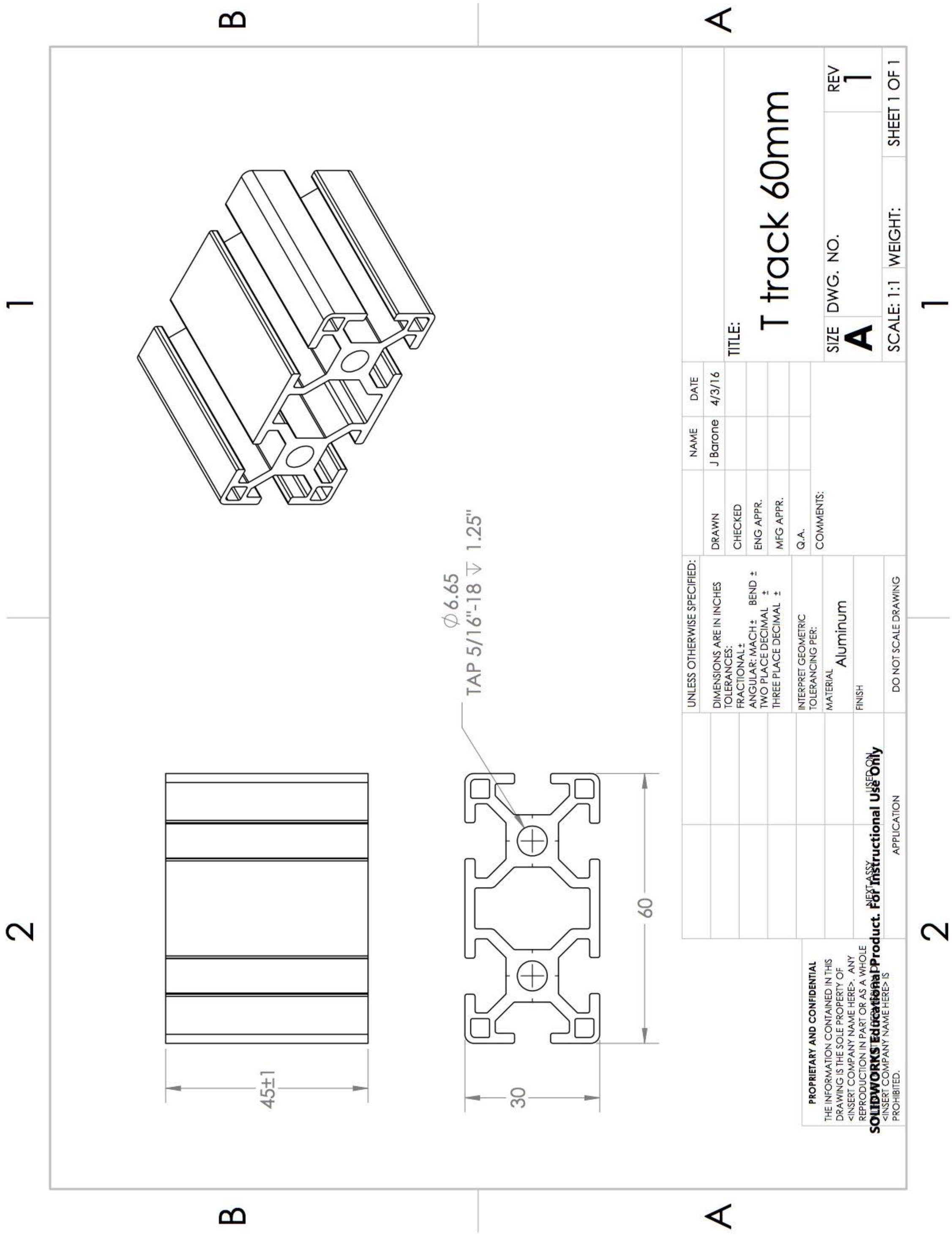


Figure C-20: T-Track

D - Assembly Drawings

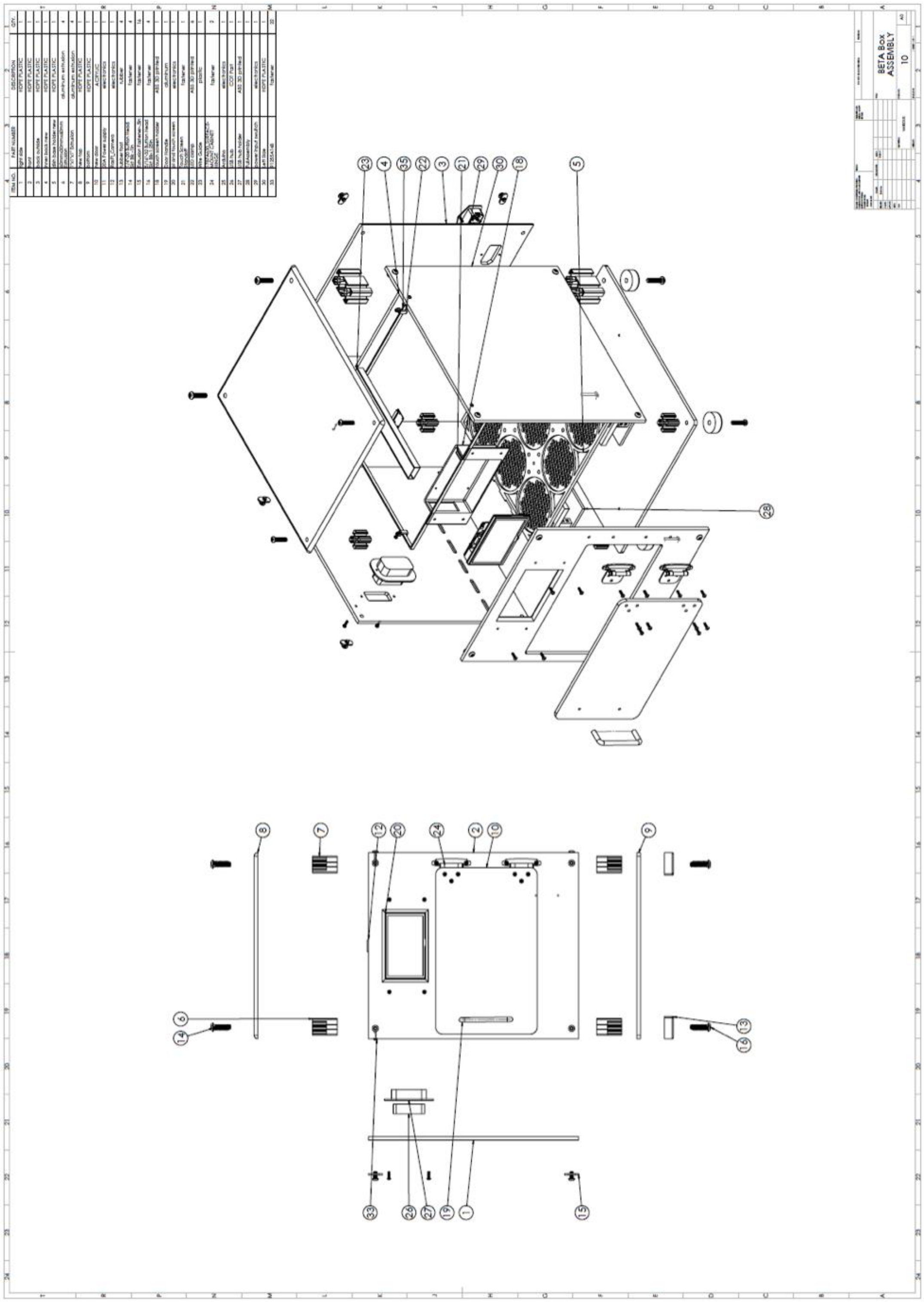


Figure D-1: Assembly Drawing

E - Material Selection Tables

Table E-1: Initial Material Research

Material	Colors Available	Heat Deflection Temp	Tensile Strength	Website	Min. Cost/in^2
~~PVC sheet white 1/8"~~	White	145 F	8,000 psi	Professional Plastics	$0.009
Polystyrene white 1/8"	White	200 F @ 264 psi	7,500 psi	Curbell Plastics	$0.011
~~General Purpose ABS black 1/8"~~	Black	190 F @264 psi	6,160 psi	Curbell Plastics	$0.012
~~Celtec PVC black 1/8"~~	Black	145 F	8,000 psi	Professional Plastics	$0.012
Polystyrene black 1/8"	Black	200 F @ 264 psi	7,500 psi	Curbell Plastics	$0.012
~~General Purpose ABS white 1/8"~~	White	190 F @264 psi	6,160 psi	Curbell Plastics	$0.015
~~ABS general purpose black 1/8"~~	Black	190 F @264 psi	6,160 psi	Professional Plastics	$0.016
~~foamed PVC white 1/8"~~	White			tap plastics	$0.016
Polypropelene black 1/8"	Black	110 F @264	4,800 psi	Curbell Plastics	$0.017
Polypropylene white 1/8"	White	110 F @264	4,800 psi	Professional Plastics	$0.017
LDPE white 1/8"	White	98 F @264	1,363 psi	Professional Plastics	$0.018
~~PVC white 1/8"~~	White			Curbell Plastics	$0.019
LDPE black 1/8"	Black	98 F @264	1,363 psi	Curbell Plastics	$0.020
HDPE black 1/8"	Black	176 F	4,600 psi	Curbell Plastics	$0.021
~~PVC sheet black 1/4"~~	Black			Professional Plastics	$0.021
Cast Acrylic clear 1/8"	Clear	180F	<10,000 psi	Professional Plastics	$0.021
Plexiglass Cast Acrylic clear 1/8"	Clear	180F	<10,000 psi	Professional Plastics	$0.021
UHMW sheet white 1/8"	White			Curbell Plastics	$0.022
Cast Acrylic Clear 1/8"	Clear	180F	<10,000 psi	Curbell Plastics	$0.022
Smooth HPDE white 1/8"	White	176 F	4,600 psi	tap plastics	$0.022

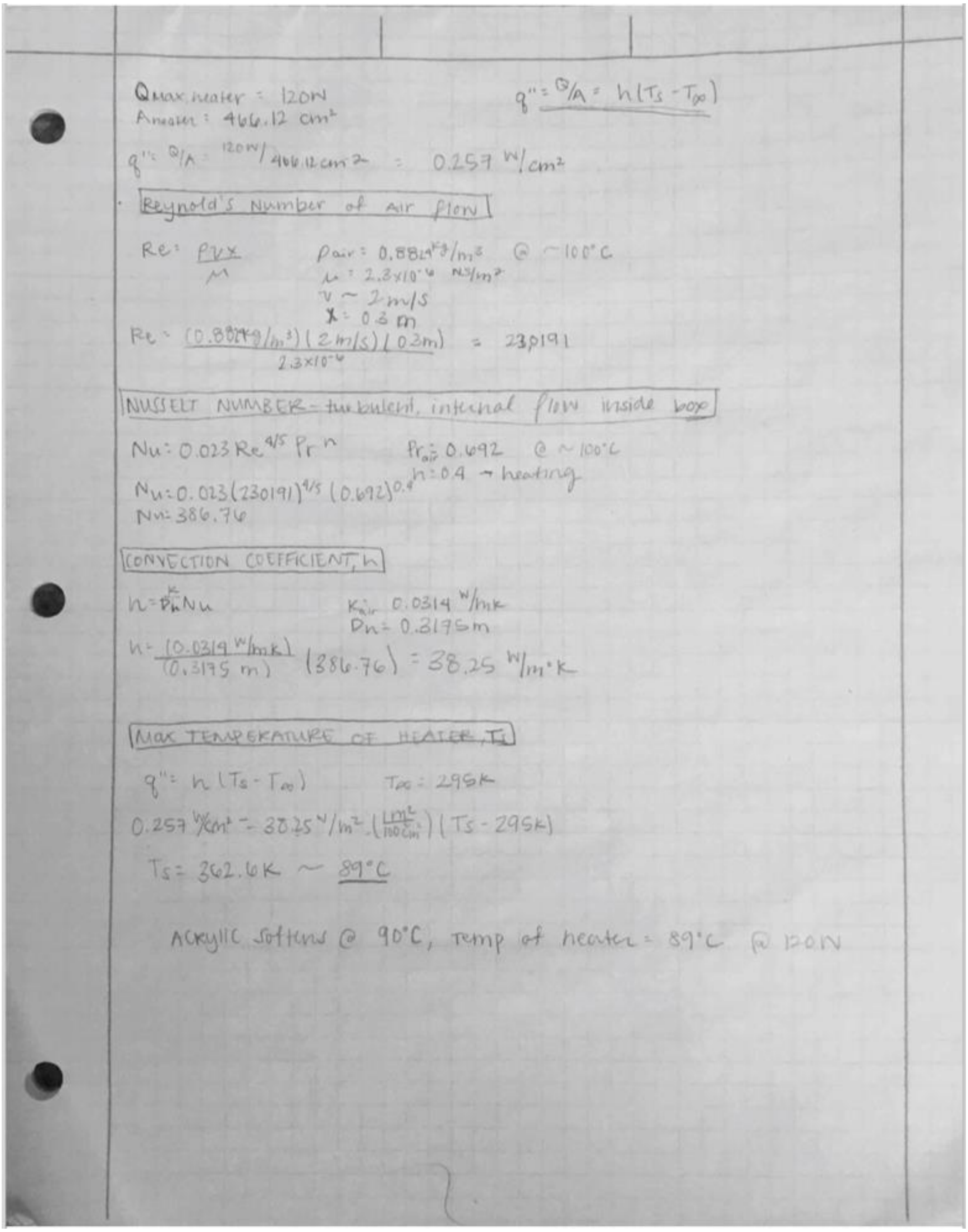
Table E-2: Final Material List Details

Material	Finish	Color	Thickness	Location	Cut Size 1	Cost/ in^2	Cost	Shipping Cost	Taxes	Standardized Cost 18x36	Min. Total Cost
HDPE	stress relieved	natural	1/4"	Curbell	48x96	$0.038	$173.540	24.19	16.45	$24.40	$214.18
HDPE	stress relieved	black	1/4"	Curbell	18x36	$0.041	$190.890	24.19	16.45	$26.84	$231.53
HDPE	smooth	natural	1/4"	TAP	18x28	$0.044	$22.400	0		$28.80	$22.40
HDPE	smooth	black	1/4"	eplastics	24x48	$0.022	$25.160	30	1.95	$14.15	$57.11
HDPE	smooth	natural	1/4"	eplastics	24x48	$0.021	$24.240	30	1.95	$13.64	$56.19
HDPE- seaboard	matte	black	1/4"	eplastics	24x54	$0.028	$36.140			$18.07	$36.14
HDPE-starboard	matte	black	1/4"	TAP	18x42	$0.051	$38.320	0		$32.85	$38.32
HDPE-starboard	matte	white	1/4"	TAP	18x42	$0.051	$38.320	0		$32.85	$38.32
Polystyrene	matte	black	1/4"	Curbell	48x96	$0.024	$108.730	285.43	9.42	$15.29	$403.58
Polystyrene	smooth	black	1/4"	Curbell	48x96	$0.024	$108.730			$15.29	$108.73
Polystyrene	matte	natural	1/4"	Curbell	48x96	$0.024	$108.730			$15.29	$108.73
Polystyrene	smooth	natural	1/4"	Curbell	48x96	$0.024	$108.730			$15.29	$108.73
Acrylic	smooth	clear	1/4"	TAP	12x12	$0.069	$10.000	11		$45.00	$21.00
Acrylic	smooth	clear	1/4"	Professional	12x24					$0.00	$0.00
Acrylic	smooth	clear	1/4"	eplastics	24x48	$0.047	$54.230	30	1.95	$30.50	$86.18
Acrylic	smooth	clear	1/4"	Curbell	48x96	$0.039	$181.120	24.19	16.45	$25.47	$221.76

Table E-3: Material Testing Detailed Weighted Results

MATERIALS SELCTION MATRIX			all scores on scale 1-5														
						Manufacturability				Temperature			Impact/Brittle		Aesthetics		
Material	Cost/in^2	Weight	Rank	weight	rank		weight	rank	notes	weight	rank	notes	weight	rank	weight	rank	Total Score
Acrylic	$ 0.025	5	2	1	4	easy	3	5	pretty!	3	4		2	2	2	5	55
LDPE	$ 0.020	5	3	1	4	can't lasercut	3	0	not allowed	3			2		2	2	23
HDPE- matte	$ 0.021	5	3	1	4	can lasercut	3	4	good	3	5	1/2 in deflection, very insulating	2	4	2	4	62
HDPE- glossy	$ 0.021	5	3	1	1		3			3			2		2		16
Polypropelene	$ 0.017	5	4	1	4	can't lastercut	3	1	melted	3			2		2	2	31
Polystyrene	$ 0.012	5	5	1	1		3			3			2		2		26
														MAXIMUM SCORE			62

F - Finite Element Iterations



$Q_{max,heater} = 120W$
$A_{heater} = 466.12\ cm^2$

$q'' = Q/A = h(T_s - T_\infty)$

$q'' = Q/A = 120W / 466.12\ cm^2 = 0.257\ W/cm^2$

Reynold's Number of Air flow

$Re = \frac{\rho v x}{\mu}$ $\rho_{air} = 0.882\ kg/m^3$ @ ~100°C
$\mu = 2.3\times10^{-6}\ N s/m^2$ [illegible]
$v \sim 2\ m/s$
$x = 0.3\ m$

$Re = \frac{(0.882\ kg/m^3)(2\ m/s)(0.3m)}{2.3\times10^{-6}} = 230191$

NUSSELT NUMBER - turbulent, internal flow inside box

$Nu = 0.023 Re^{4/5} Pr^n$ $Pr_{air} = 0.692$ @ ~100°C
$n = 0.4 \rightarrow$ heating

$Nu = 0.023(230191)^{4/5}(0.692)^{0.4}$
$Nu = 386.76$

CONVECTION COEFFICIENT, h

$h = \frac{k}{D_h} Nu$ $k_{air} = 0.0314\ W/mK$
$D_h = 0.3175m$

$h = \frac{(0.0314\ W/mK)}{(0.3175\ m)}(386.76) = 38.25\ W/m^2 \cdot K$

MAX TEMPERATURE OF HEATER, T_s

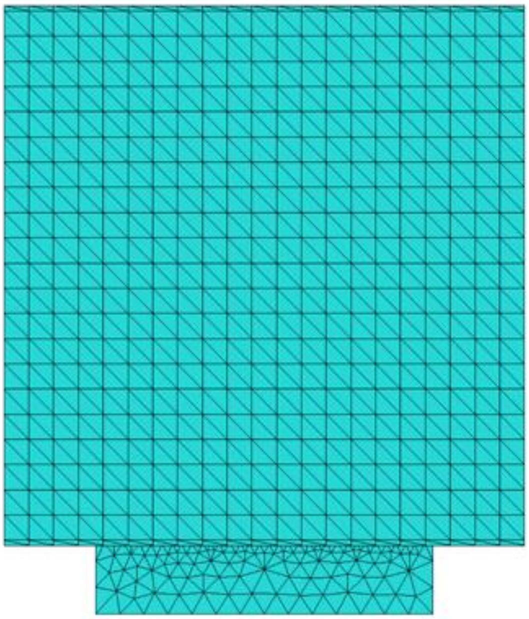
$q'' = h(T_s - T_\infty)$ $T_\infty = 295K$

$0.257\ W/cm^2 = 38.25\ W/m^2 \cdot \left(\frac{1m^2}{100cm^2}\right)(T_s - 295K)$

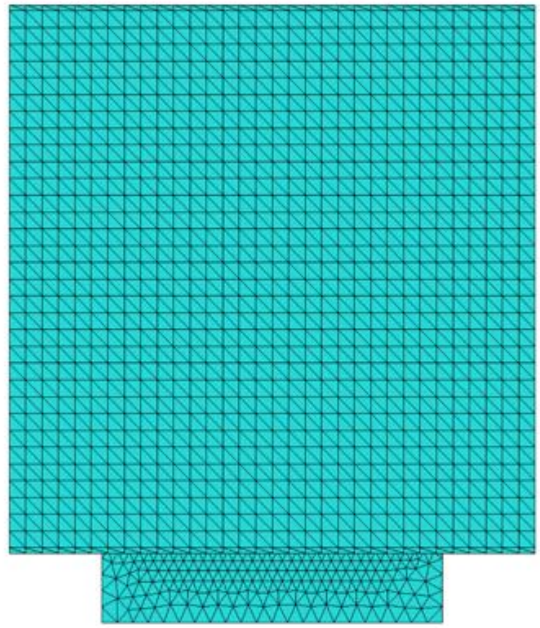
$T_s = 362.6K \sim 89°C$

Acrylic softens @ 90°C, temp of heater = 89°C @ 120W

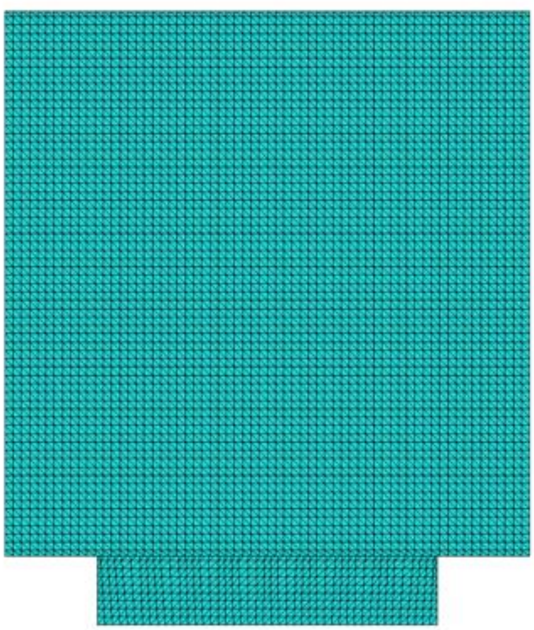
Figure F-1: FEA Hand Calculations



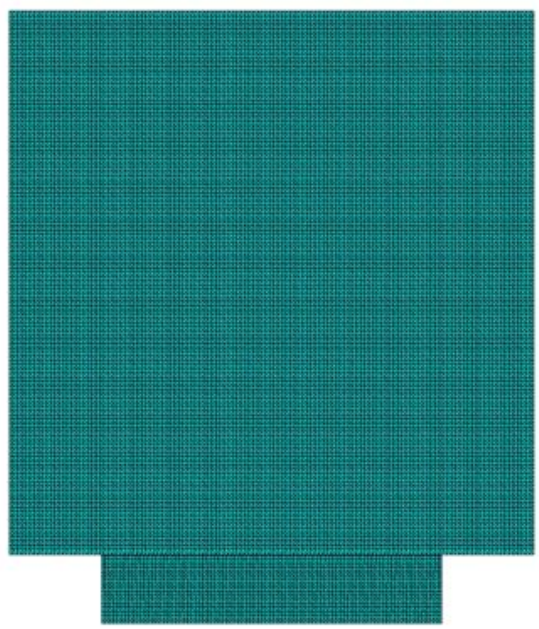
(a)



(b)



(c)



(d)

Figure F-2: Mesh Sizes

Meshes with seed sizes of (a) 1.5, (b) 1.0, (c) 0.5, and (d) 0.25 used to test convergence

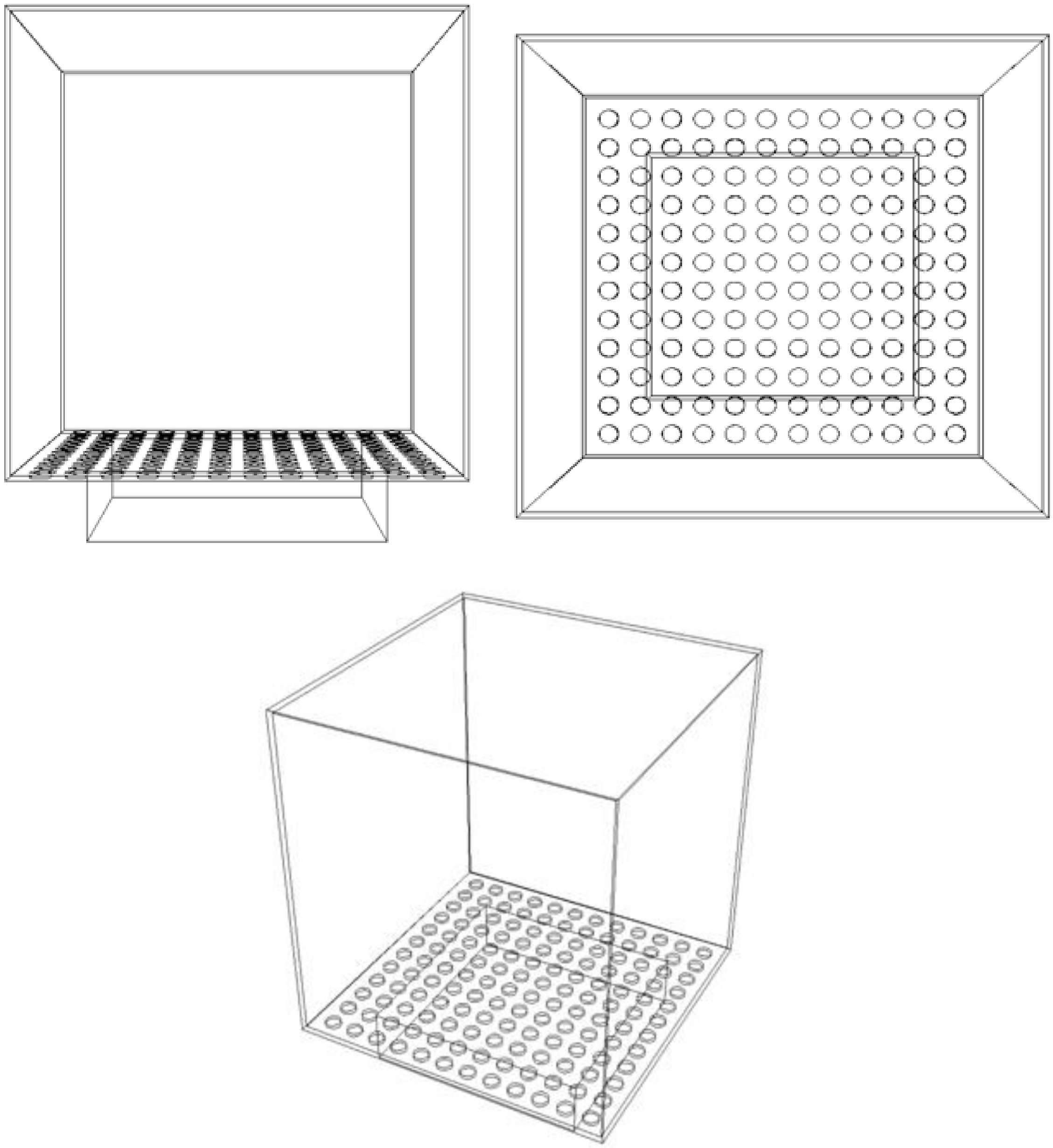
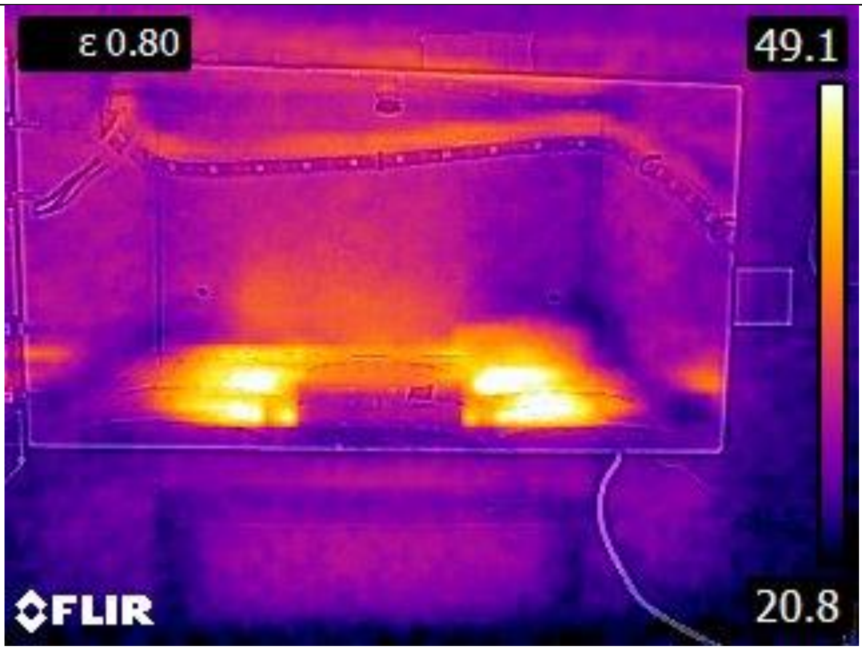
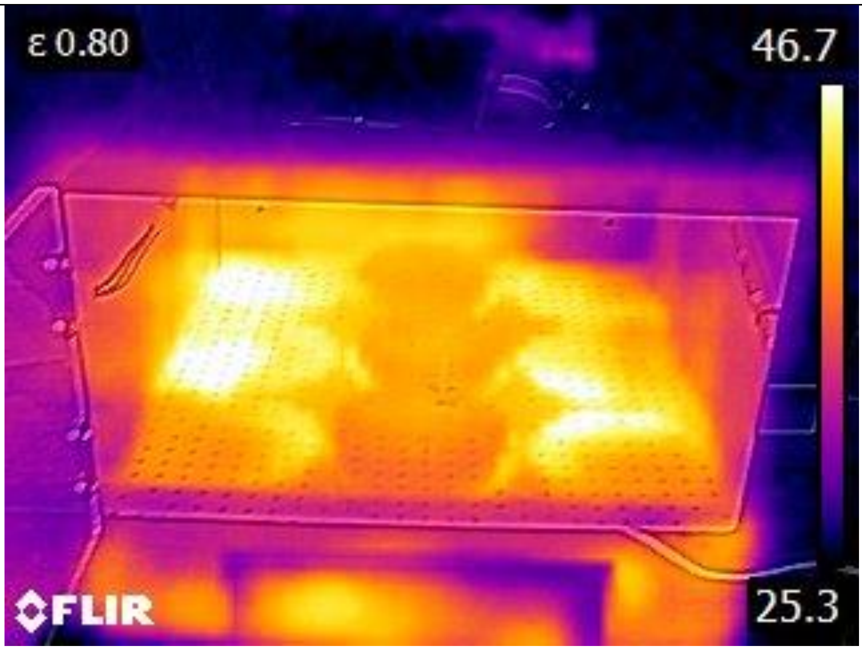


Figure F-3: Front, Top, and Isometric Views of FEA Simplifications

G - Experimental Data

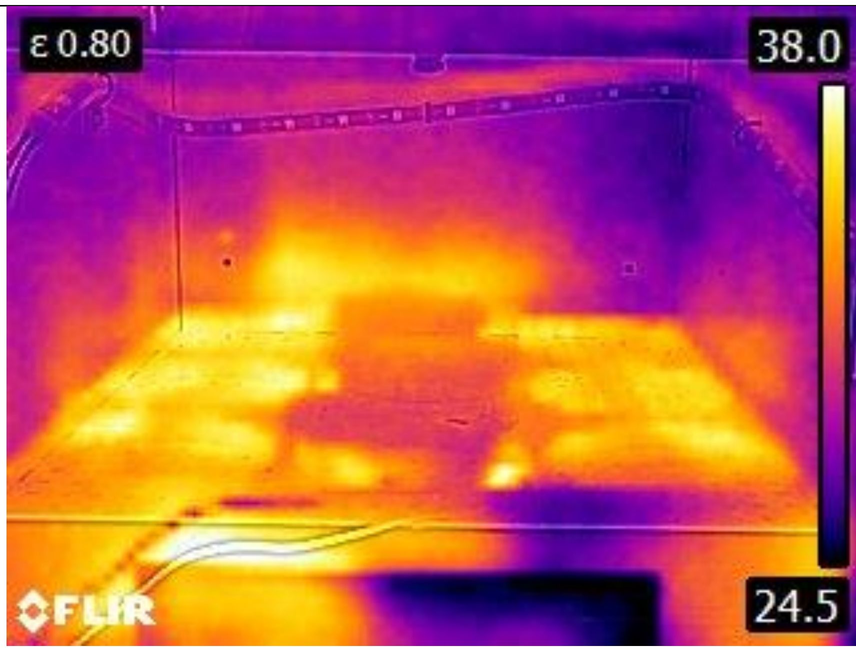


Figures G.1: Thermal Images from Experimentation Trial 1



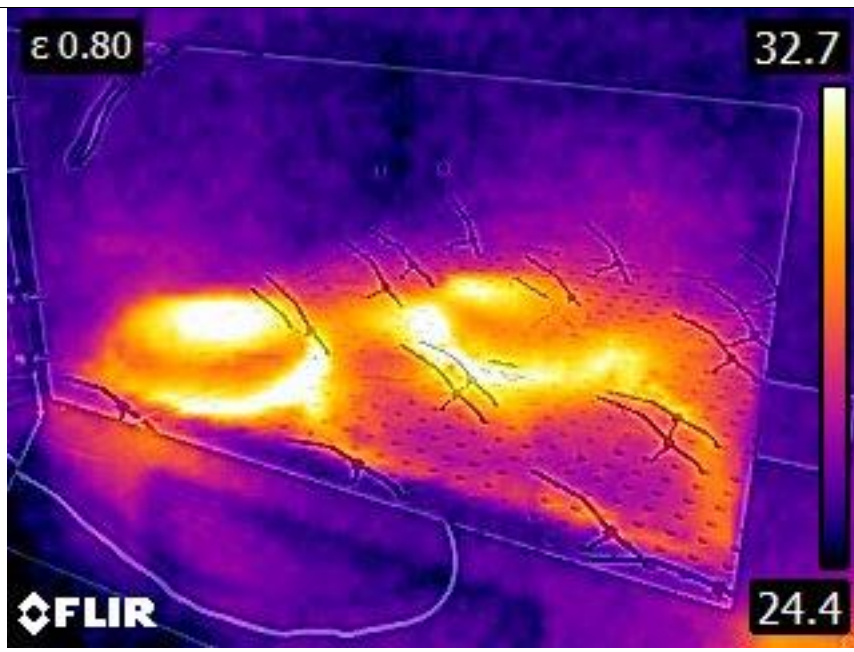
Figures G.2: Thermal Images from Experimentation Trial 2

Figures G.3: Thermal Images from Experimentation Trial 3

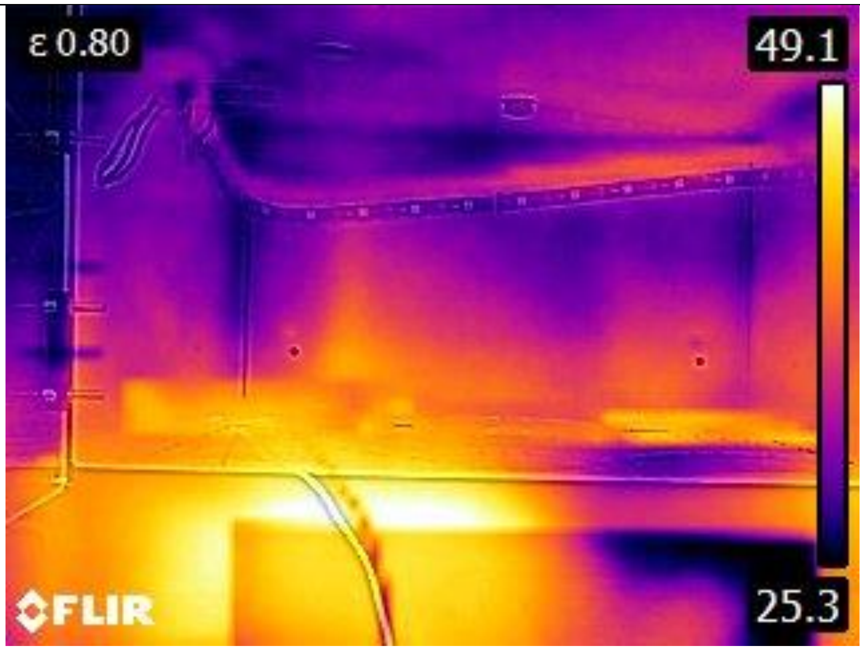


Figures G.4: Thermal Images from Experimentation Trial 4

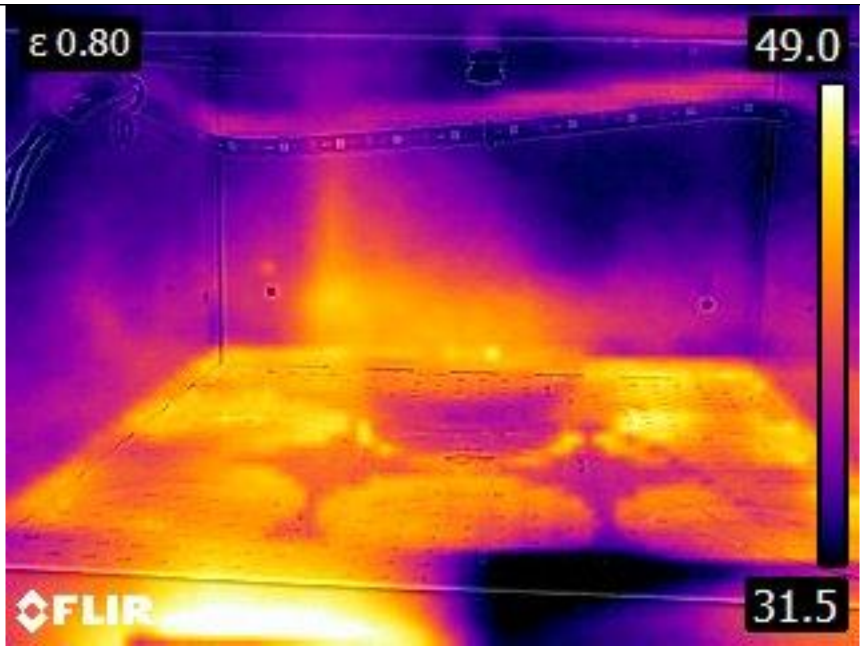
Figures G.5: Thermal Images from Experimentation Trial 5



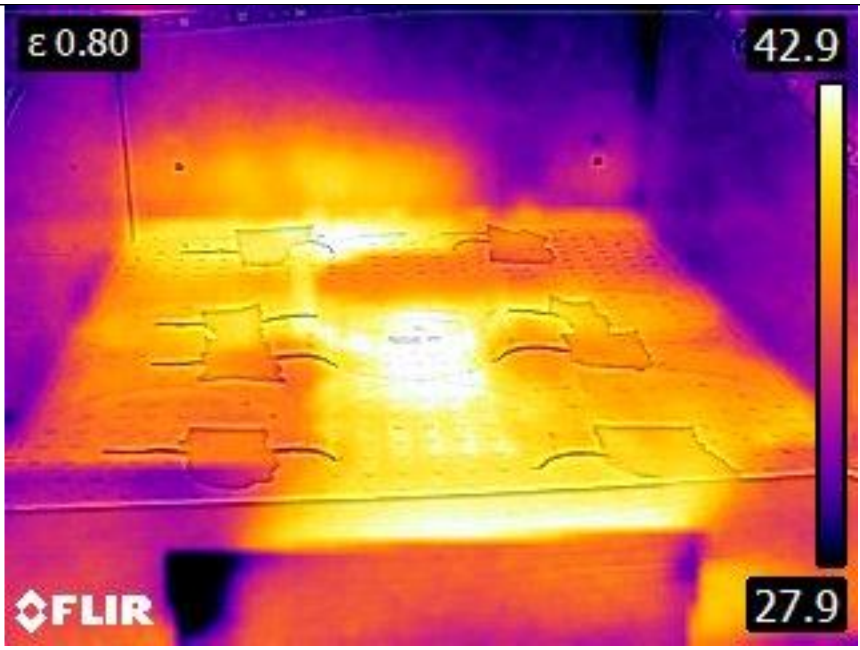
Figures G.6: Thermal Images from Experimentation Trial 6



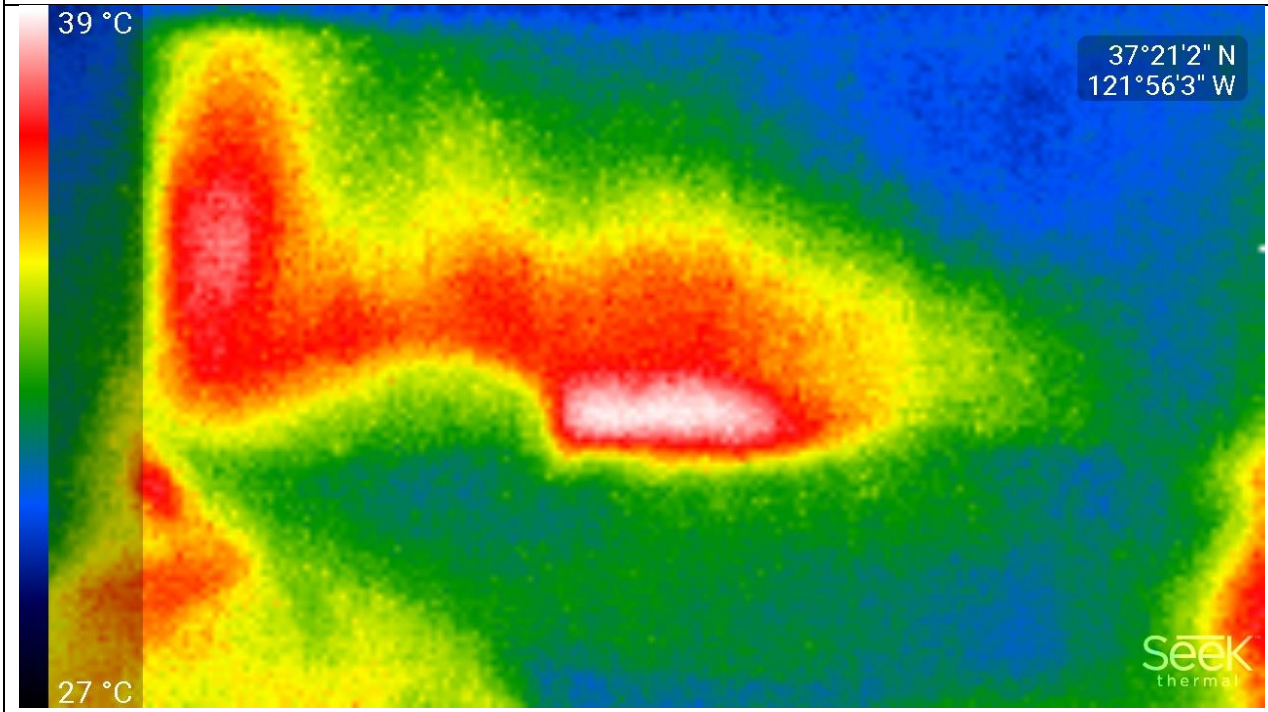
Figures G.7: Thermal Images from Experimentation Trial 7



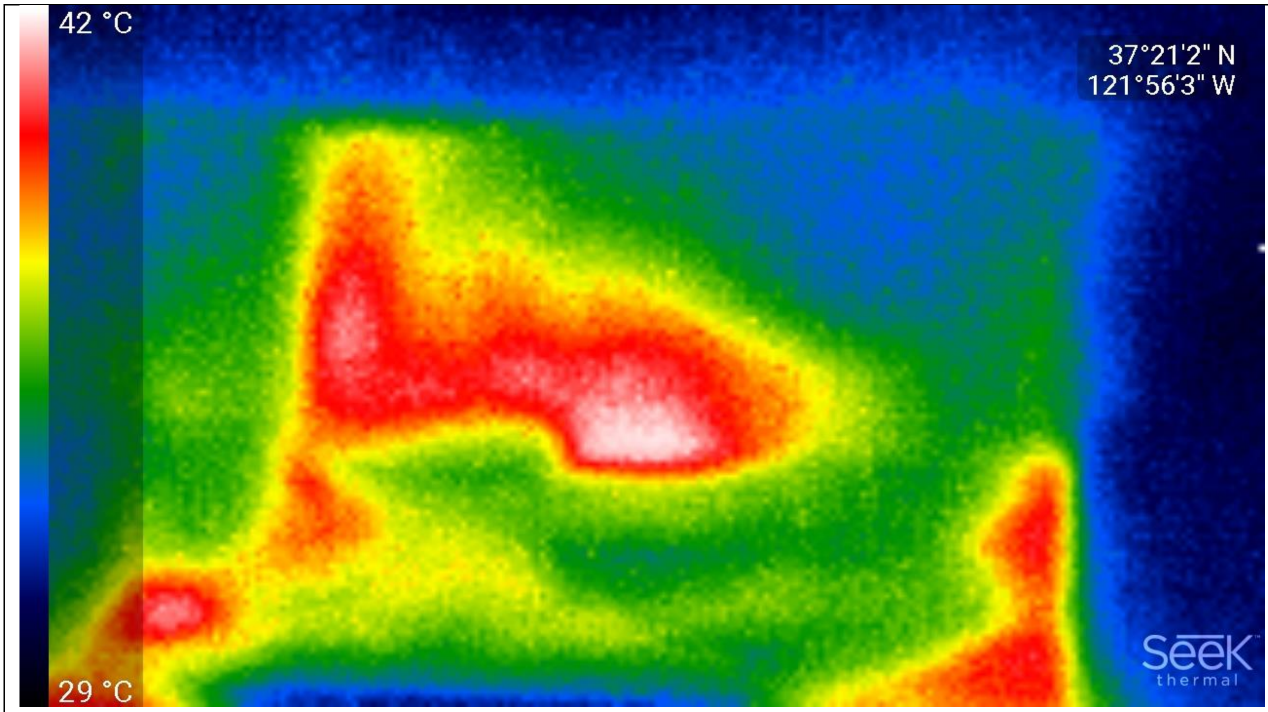
Figures G.8: Thermal Images from Experimentation Trial 8



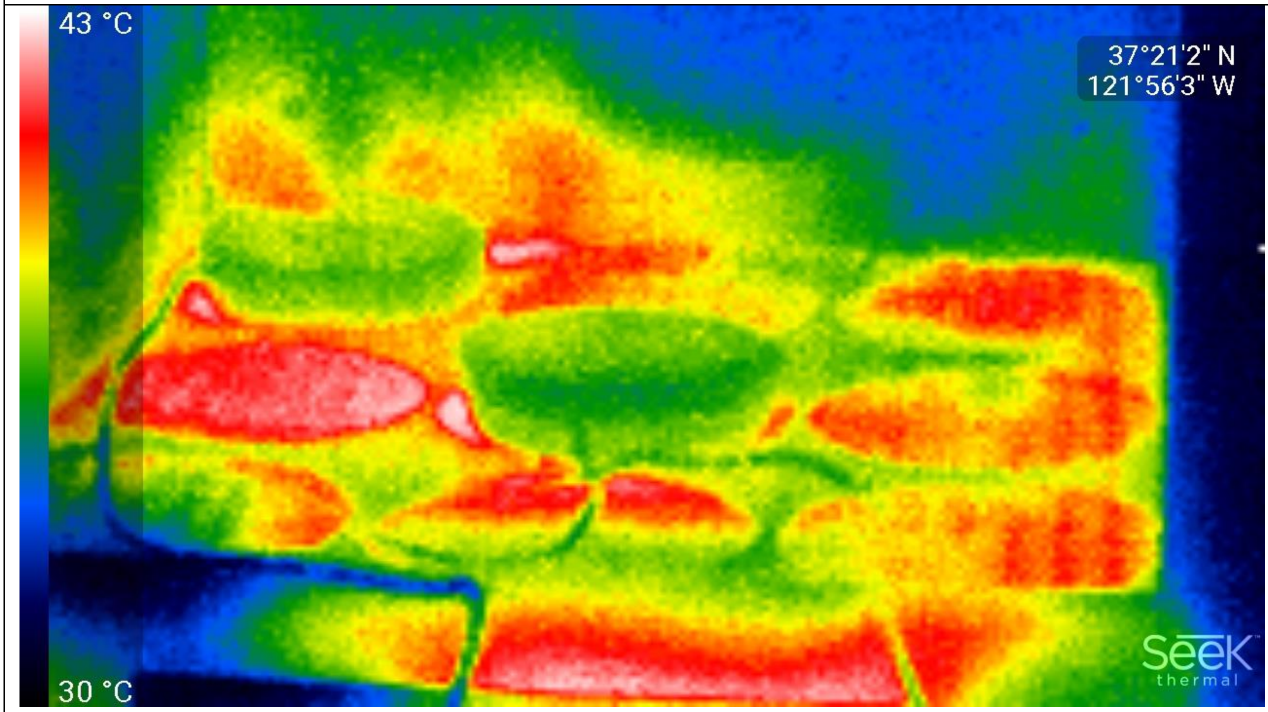
Figures G.9: Thermal Images from Experimentation Trial 9



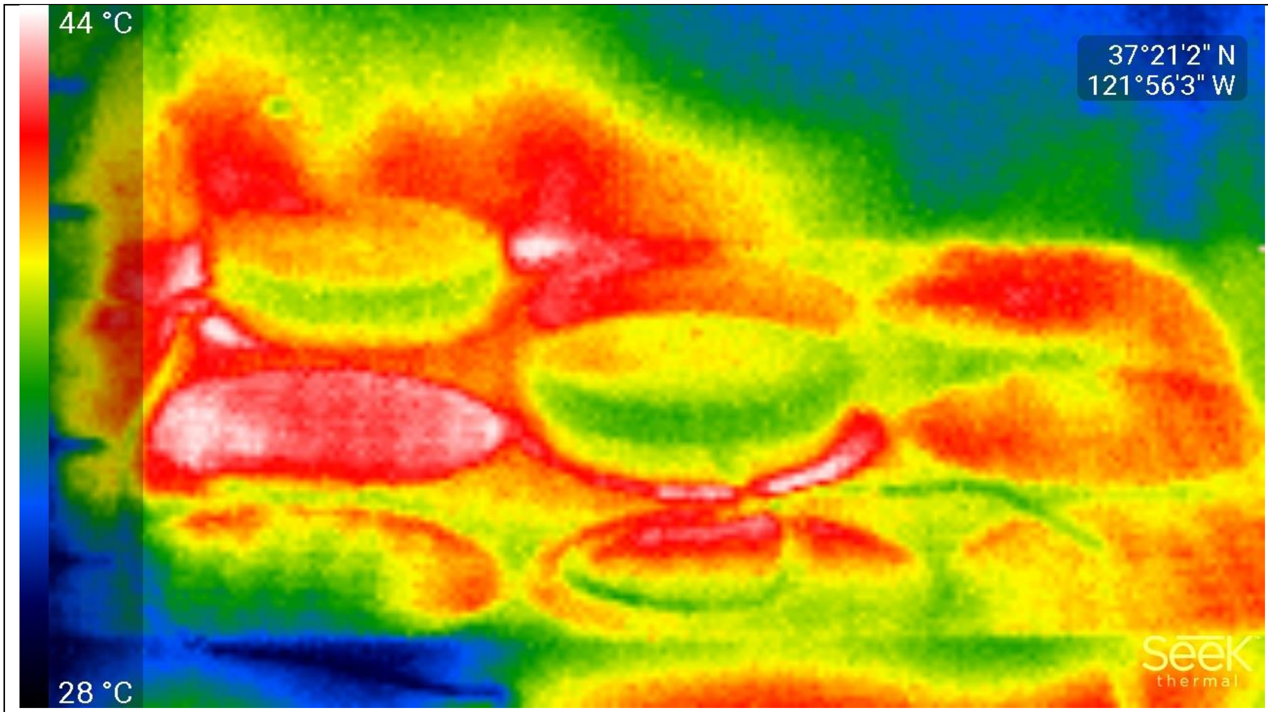
Figures G.10: Thermal Images from Experimentation Trial 10



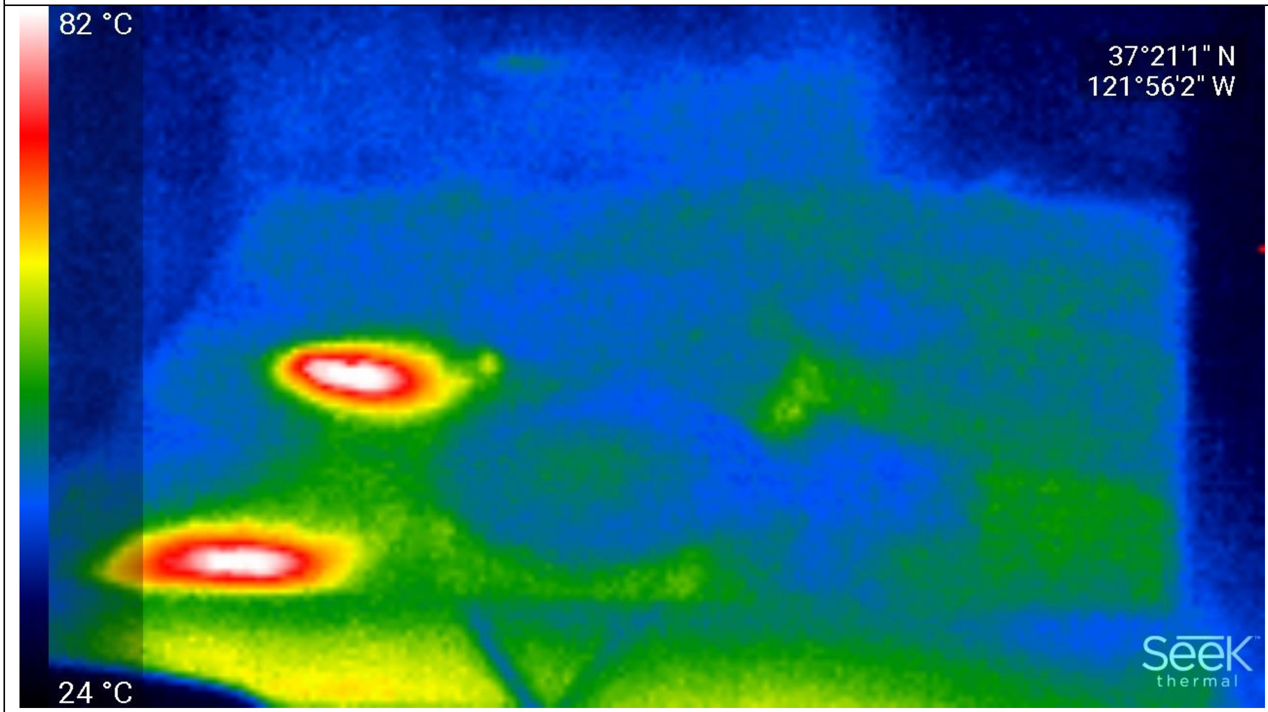
Figures G.11: Thermal Images from Experimentation Trial 11



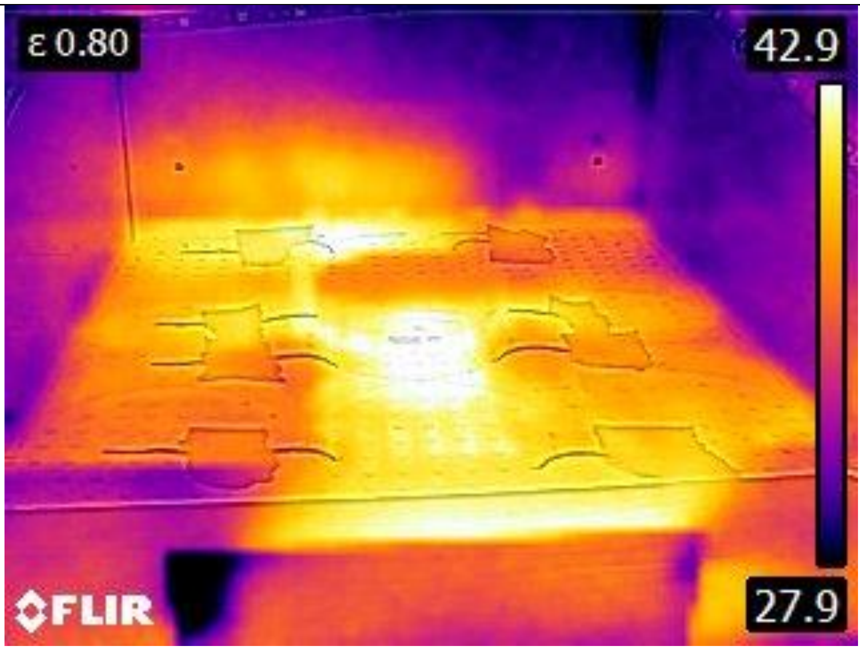
Figures G.12: Thermal Images from Experimentation Trial 12



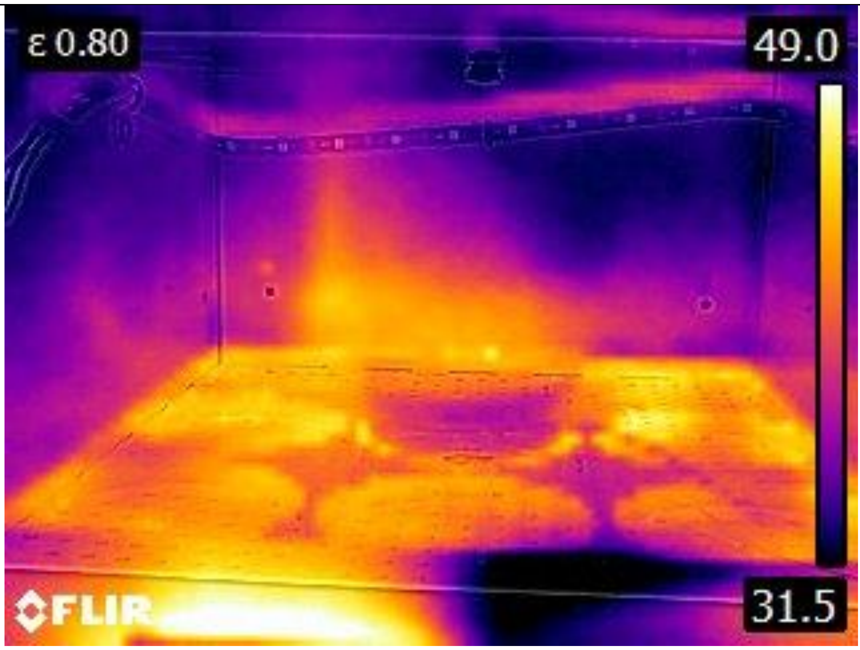
Figures G.13: Thermal Images from Experimentation Trial 13



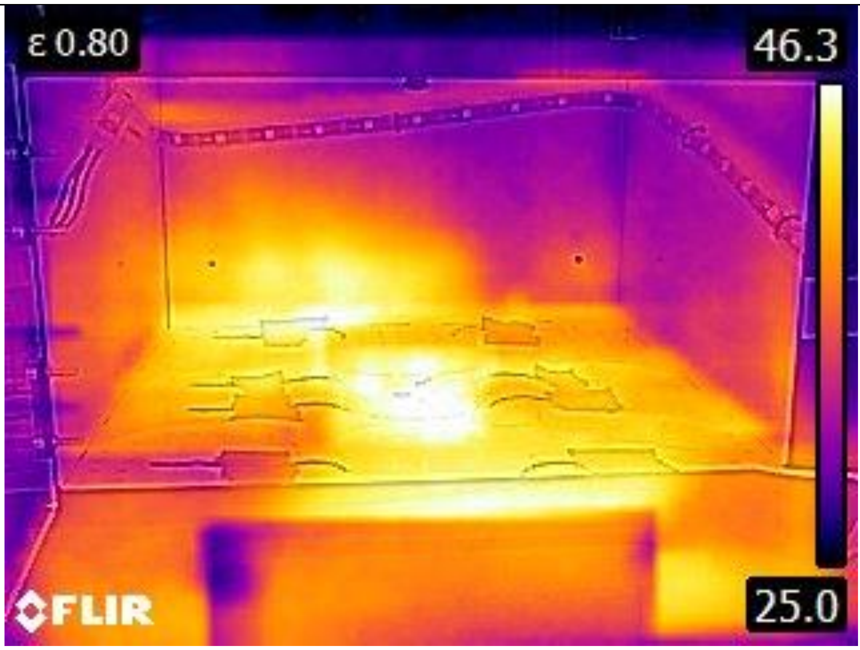
Figures G.14: Thermal Images from Experimentation Trial 14



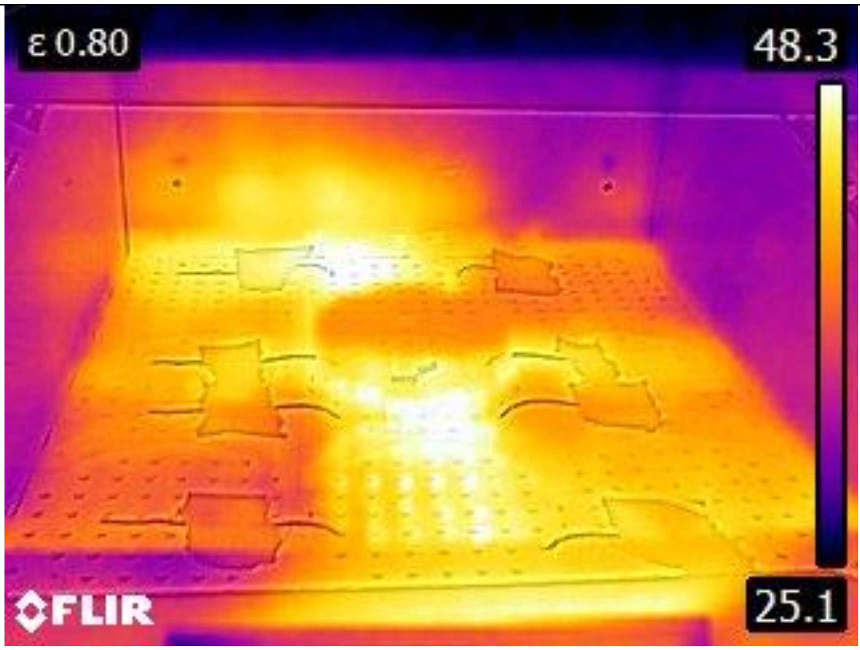
Figures G.15: Thermal Images from Experimentation Trial 15



Figures G.16: Thermal Images from Experimentation Trial 16



Figures G.17: Thermal Images from Experimentation Trial 17



Figures G.18: Thermal Images from Experimentation Trial 18

H - Auto-Calibration Codes

H-1 Code for Collecting Force Data

```
int fsrAnalogPin = 0; // FSR connected to Analog 0
int fsrReading;
unsigned long time;

void setup(void) {
  Serial.begin(9600);
}
void loop(void) {
  fsrReading = analogRead(fsrAnalogPin);
  Serial.print("Analog reading = ");
  Serial.println(fsrReading);

  Serial.print("Time: ");
  time = millis()/1000;
  //prints time since program started
  Serial.println(time);
  delay(1000);
  }
}
```

H-2 Code for Auto-Calibration

```
int fsrAnalogPin = 0;                     // FSR is connected to analog 0
int fsrReading, fsrReading2;              // the analog reading from the FSR resistor divider
const int numReadings = 4;                //number of readings for slope measurements
int readings[numReadings];                // the readings from the analog input
int index = 0;                            // the index of the current reading
double volt[2], resist[2], conduct[2], force[2];
double slope1, slope2, feed_adjust;
int feedrate = 300;                       //input from the users settings
unsigned long time;

void setup(void) {
  Serial.begin(9600);                     // initialize all the readings to 0:
  for (int thisReading = 0; thisReading < numReadings; thisReading++)
    readings[thisReading] = 0;
}

void loop(void) {
//initial calibration should go in Marlin_main.cpp
//get feedrate value input
 feed_adjust = feedrate / 300;   //feed_adjust is the linear difference in slopes that comes from
                                 //different feedrates should be extruding material for calibration
```

```
//calculate slope1
  readings[0] = analogRead(fsrAnalogPin);
  Serial.println(readings[0]);
  volt[0] = map(readings[0], 0, 1023, 0, 5000);         //map voltage to analog
  Serial.println(volt[0]);
  resist[0] = (5000 - volt[0]) * 1000 / volt[0];        //resistance with 10k resistor
  Serial.println(resist[0]);
  conduct[0] = 1000000 / resist[0];                     //conductance in microhms
  Serial.println(conduct[0]);
  force[0] = conduct[0] / 80;                           //force based on data sheet chart
  Serial.println(force[0]);
  readings[2] = millis() / 1000;
  delay(1000);

  readings[1] = analogRead(fsrAnalogPin);
  volt[1] = map(readings[1], 0, 1023, 0, 5000);
  resist[1] = (5000 - volt[1]) * 1000 / volt[1];              //resistance with 10k resistor
  conduct[1] = 1000000 / resist[1];                           //conductance in microhms
  force[1] = conduct[1] / 80;                                 //force based on data sheet chart
  readings[3] = millis() / 1000;                  //if the time is the same to avoid infinite slope
  if (readings[2] == readings[3])
          slope1 = 0;
  else
          slope1 = (float)(force[1] - force[0]) / (readings[3] - readings[2]);

  if (slope1 > feed_adjust)
  {
//if slope1 > 20, do nothing with G-code path but still extrude
    Serial.println("extrude before print");                   //test to see if extruding
          slope2 =  feed_adjust;
          while (slope2 > feed_adjust)
          fsrReading2 = analogRead(fsrAnalogPin);

          //calculate slope2
          readings[0] = analogRead(fsrAnalogPin);
          volt[0] = map(readings[0], 0, 1023, 0, 5000);       //map voltage to analog
          resist[0] = (5000 - volt[0]) * 1000 / volt[0];      //resistance with 10k resistor
          conduct[0] = 1000000 / resist[0];                   //conductance in microhms
          force[0] = conduct[0] / 80;                          //force based on data sheet chart

          readings[2] = millis() / 1000;
          delay(1000);
          readings[1] = analogRead(fsrAnalogPin);
          volt[1] = map(readings[1], 0, 1023, 0, 5000);       //map voltage to analog
          resist[1] = (5000 - volt[1]) * 1000 / volt[1];       //resistance with 10k resistor
          conduct[1] = 1000000 / resist[1];                   //conductance in microhms
```

```
        force[1] = conduct[1] / 80;                          //force based on data sheet chart

        readings[3] = millis() / 1000;
        if (readings[2] == readings[3])
        slope2 = 0;
        else
        slope2 = (float)(force[1] - force[0]) / (readings[3] - readings[2]);
        delay(1000);
  }
  else
        delay(1000);

  //secondary force sensor reading should go in Marlin_main.cpp (around line 600)
  //get feedrate value input
feed_adjust = feedrate / 300;  //feed_adjust is the linear difference in slopes that come from
//different feedrates should be extruding material on G-code path

  //calculate slope1
  readings[0] = analogRead(fsrAnalogPin);
  volt[0] = map(readings[0], 0, 1023, 0, 5000);              //map voltage to analog
  resist[0] = (5000 - volt[0]) * 1000 / volt[0];                     //resistance with 10k resistor
  conduct[0] = 1000000 / resist[0];                          //conductance in microhms
  force[0] = conduct[0] / 80;                                //force based on data sheet chart

  readings[2] = millis() / 1000;
  delay(1000);
  readings[1] = analogRead(fsrAnalogPin);
  volt[1] = map(readings[1], 0, 1023, 0, 5000);              //map voltage to analog
  resist[1] = (5000 - volt[1]) * 1000 / volt[1];             //resistance with 10k resistor
  conduct[1] = 1000000 / resist[1];                          //conductance in microhms
  force[1] = conduct[0] / 80;                                //force based on data sheet chart
  readings[3] = millis() / 1000;
  if (readings[2] == readings[3])
        slope1 = 0;
  else
        slope1 = (float)(force[1] - force[0]) / (readings[3] - readings[2]);
  while (slope1 >  feed_adjust)
  Serial.println("extruding");
        //continue extruding but stop moving in the XY axis
        delay(1000);
}
```

H-3 Code for Ending Print

```
int fsrAnalogPin = 0; // FSR connected to analog 0
int fsrReading;
```

```
double volt[1], resist[1], conduct[1], force[1];
int force_limit = 15; //input from the users settings
unsigned long time;

void setup(void) {
  Serial.begin(9600);
}
void loop(void) {
   Serial.print("Time: ");
   time = millis()/1000;
    //prints time since program started
   Serial.println(time);

  fsrReading = analogRead(fsrAnalogPin);
  Serial.print("Analog reading = ");
  Serial.println(fsrReading);

   // force reading conversion
  readings[1] = analogRead(fsrAnalogPin);
  volt[1] = map(readings[1], 0, 1023, 0, 5000); //map voltage to analog
  resist[1] = (5000 - volt[1]) * 1000 / volt[1]; //resistance with 10k resistor
  conduct[1] = 1000000 / resist[1]; //conductance in microhms
  force[1] = conduct[1] / 80; //force based on data sheet chart
  Serial.println("force1_0 = ");
  Serial.println(force[0]);
delay(1000);

//When the slope increases above the set limit, all the material has been extruded and the printer
//should pause.
  if (fsrReading > force_limit) {
    Serial.println("Pause Print. REFILL SYRINGE");
 }
}
```

I – Timelines

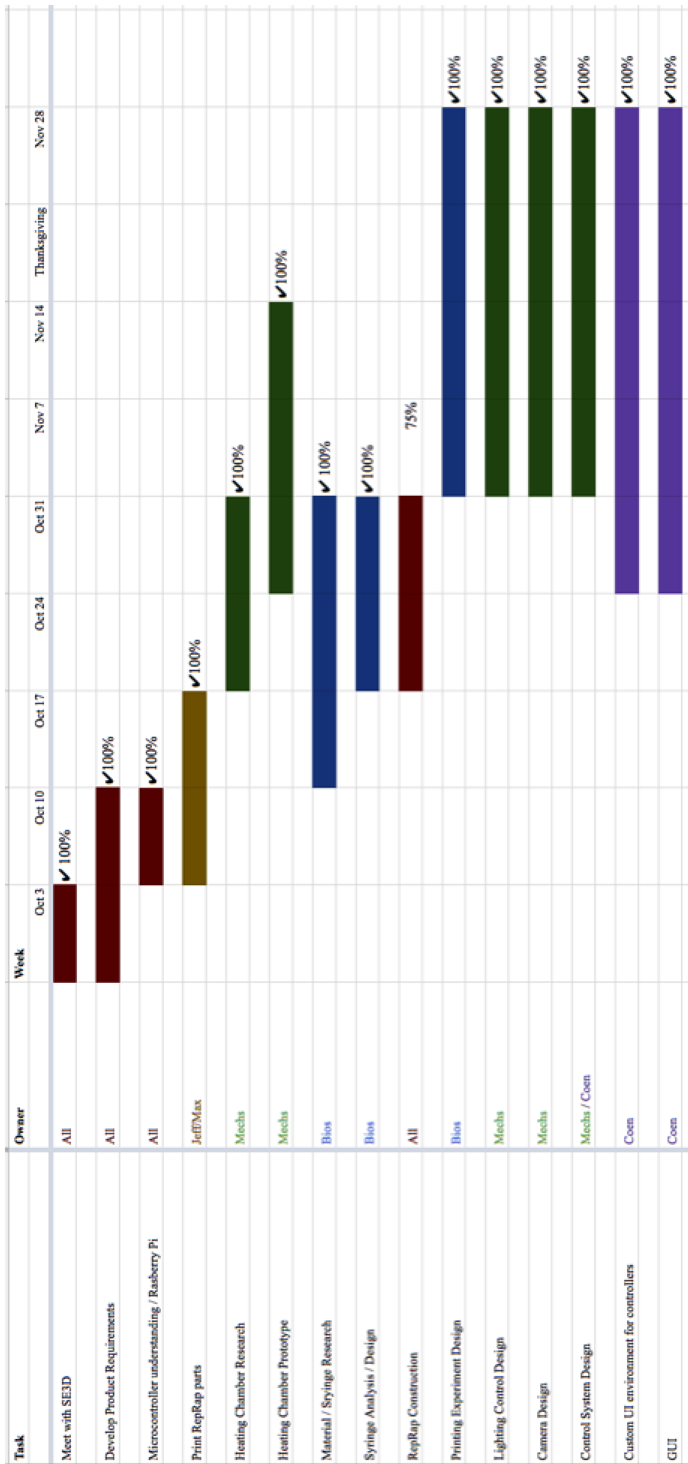


Figure I-1: Fall Gantt Chart

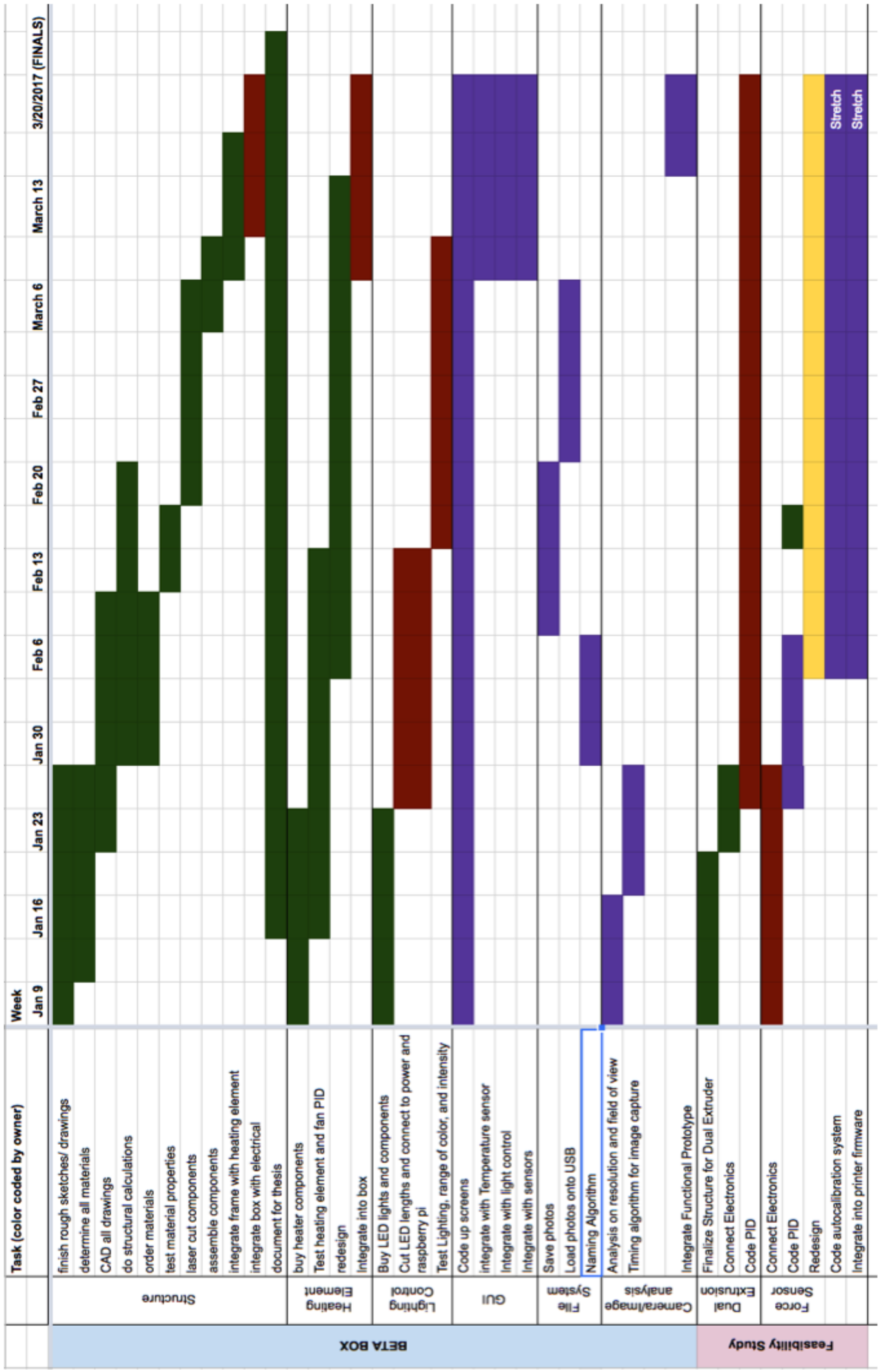


Figure I-2: Winter Gantt Chart

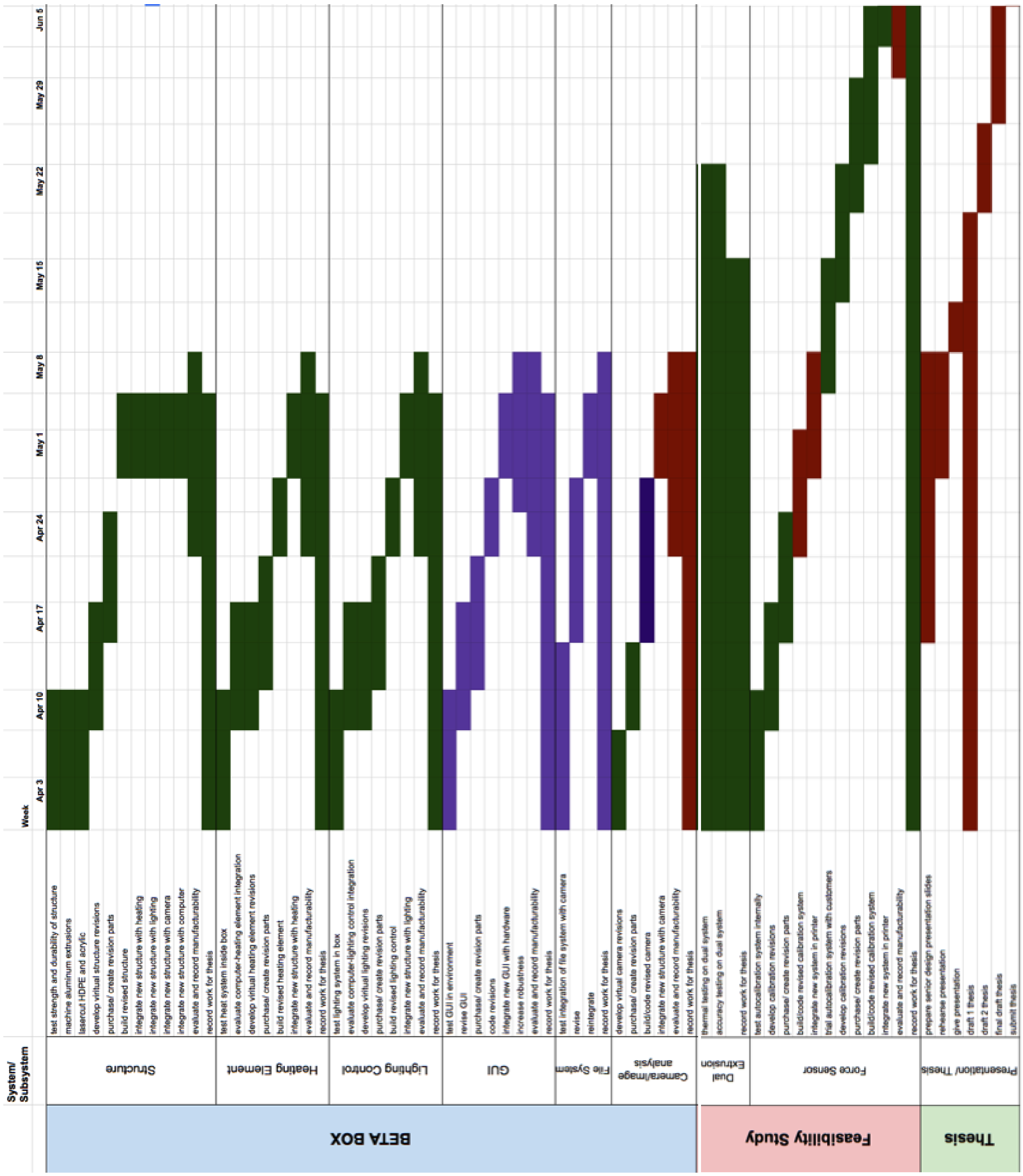


Figure I-3: Spring Quarter Gantt Chart

J - Budgeting Spreadsheet

Table J-1: Budget Spread Sheet

Date	Order #	Vendor	Item	Unit Price	Qty	Price	Tax and Shipping
10/6/2016		Amazon					0.00
			Raspberry Pi	88.86	1	88.86	
10/13/16		Michael's/Lowes					0.00
			Foamcore/Wiring	41.88	1	41.88	
1/17/17		RobotShop					
			Force Sensor	7.00	1	7.00	
1/17/17	103-1728724-97	Amazon					0.00
			10A Power Supply	15.99	1	15.99	
			5A Power Supply	8.99	1	8.99	
1/17/17		Amazon					0.00
			0.10A Fans	12.98	2	25.96	
			12V Heat Plate	11.73	2	23.46	
			Jumper Wires	19.99	1	19.99	
			Stepper Motors	12.99	2	25.98	
			Speaker	6.99	1	6.99	
			Breadboard	7.86	1	7.86	
			Syringes	5.99	1	5.99	
1/17/17	6309	SingleTact					9.21
			Force Sensor	19.95	1	19.95	
1/17/17	1556-8056-3082	Houzz					4.10
			Door Handle	1.89	1	1.89	
1/17/17	1336835	Adafruit					16.20
			LED Lights	16.95	1	16.95	
			2 Pin Plug and Cable	0.75	1	0.75	
			Female Adapter	2.00	1	2.00	
			Force Sensor	7.00	1	7.00	
1/26/17	6320	SingleTact					10.11
			FSR Connector			29.95	
1/26/17		TAP Plastic					53.02
			Black Acrylic	16.09	10	160.90	
			White Acrylic	13.41	8	107.28	
			Orange Acrylic	16.09	2	32.18	
			Clear Acrylic	42.33	1	42.33	
1/26/17		ePlastic					32.18
			Polycarbonate	28.15	1	28.15	
1/26/17	#105320896473	Amazon					0.00
			Door Seal	15.96	1	15.96	
1/26/17	#105975341313	Amazon					6.49
			PLA	99.51	1	99.51	
1/26/17	#105263539702	Amazon					5.95
			12 Gauge Wire	12.98	1	12.98	
			Case Fan	14.99	1	14.99	
			16 Gauage Wire	6.68	1	6.68	
			18 Gauge WIre	6.98	1	6.98	
			10 Gauge Wire	12.36	1	12.36	
			Arduino Kit	36.99	1	36.99	
			Angle Brackets	19.99	1	19.99	
			HDMI	5.99	1	5.99	
			Rubber Feet	301.00	1	301.00	
			Flex Cable	7.93	1	7.93	
			Heatbed	14.94	1	14.94	
			12V Relay	6.68	2	13.36	

1/26/17		McMaster Carr						5.99
			Hinge	1.15	4	4.60		
			100 Screws	2.76	1	2.76		
2/1/17		Albany County Fasteners						6.50
			#10 Screws	3.44	1	3.44		
			Angled Brackets	2.75	1	2.75		
2/1/17	#105254665507!	Amazon						0.00
			Servo Camera	19.99	1	19.99		
2/1/17	#105481039796(	Amazon						1.99
			5V Relay	5.80	2	11.60		
			Pan-Tilt	23.38	1	23.38		
			Canakit	9.99	1	9.99		
2/1/17		McMaster Carr						7.09
			Angled Bracket	0.32	50	16.00		
2/7/17	#116427463599(	Amazon						0.00
			10 Ohm Resistors	6.85	1	6.85		
			SSRs	11.99	3	35.97		
			12V Power Adapter	17.99	1	17.99		
			2 10 Ohm Resistor	4.70	1	4.70		
			Power Supply	18.95	1	18.95		
2/7/17		Lowes						1.68
			Angled Brackets	2.80	6	16.80		
2/9/17	#1168553116961	Amazon						7.10
			Raspberry Pi	39.44	2	78.88		
2/22/17	#116919935026:	Amazon						1.98
			Nonwaterproof LED	28.99	1	28.99		
			MOSFET Transistor	6.36	1	6.36		
			ADC Microchip	16.99	1	16.99		
			Mini Breadboard	8.89	1	8.89		
			Orion Fan	8.95	1	8.95		
2/22/17	#116656320672:	Amazon						1.52
			Cooling Fan	18.99	1	18.99		
3/1/17	#105966643760(	Amazon						0.00
			Power Step Down	11.59	1	11.59		
			USB Cable	9.49	1	9.49		
3/3/17	002-8371940-03	Amazon						3.92
			HDPE-Black 1/8"	8.40	1	8.40		
			Polystyrene	4.50	1	4.50		
3/3/17	002-8371940-03	Amazon						1.72
			Insulation	5.48	1	5.48		
			Polypropylene	10.66	1	10.66		
			LDPE	4.04	1	4.04		
			Textured HDPE	16.00	1	16.00		
3/9/17		Amazon						5.66
			Square containers	5.36	1	5.36		
			IRLZ34N Transistors	4.36	2	8.72		
3/17/17		Tap Plastics						10.40
			Starboard	118.90	1	118.90		
3/23/17		Amazon						
	104-2456184-8521020		USB Hub	9.99				
	104-3598373-1730669		HDMI Cable	2.20				

3/23/17		McMaster						16.00
			Compact End-Feed Fastener	1.85	4	7.40		
			Hex Screw 1-1/4	6.87	1	6.87		
			Hex Screw 1-1/8	7.73	1	7.73		
			Aluminum Extrusion Solid	6.49	1	6.49		
			Aluminum Extrusion Hollow	9.69	1	9.69		
			Hex Screw 1/2"	7.65	1	7.65		
			Round Bumper	6.55	1	6.55		
			Pull Handle	4.39	1	4.39		
			Aluminum Standoff	0.94	4	3.76		
			Hex Screw 1"	10.38	1	10.38		
			Low-Voltage Raceway	12.75	1	12.75		
			Steel Locknut	2.61	1	2.61		
			Foam Rubber Seal	11.60	1	11.60		
			Cabinet Hinge	3.12	1	3.12		
3/23/17	1401817-11115632	Adafruit						5.19
			Analog Temp Sensor	1.50	5	7.50		
			Temp Sensor Board	4.95	5	24.75		
			Header for Pi	6.50	1	6.50		
			Stacking Header	1.95	1	1.95		
			Extender cable	4.95	1	4.95		
4/7/17		Amazon						1.61
			18 Gauge Wire	18.99	1	18.99		
			24 Gauge Wire	17.30	1	17.30		
			Switch	4.99	1	4.99		
			Arduino	21.99	2	43.98		
4/13/17		Amazon						0.00
			Raspberry Pi	38.00	2	76.00		
			Diodes	6.99	1	6.99		
			Heat Shrink	7.99	1	7.99		
4/13/17		Adafruit						4.47
			Temperature Sensors	1.50	5	7.50		
4/19/17		Tap Plastics						3.60
			Starboard	38.97	1	38.97		
4/20/17		Amazon						
			USB A to B	9.99	2	19.98		
4/20/17		Adafruit						15.04
			Transistors	1.58	10	15.80		
						Total Spent		$2,508.75
						Remaining		**$991.25**

K - Cost Analysis

Table K-1: Cost Analysis Spread Sheet

Prototype Design 1- *9 dishes, 7" camera height, touchscreen at top front*						
Part	**Direct Size (in^2)**	**True Material Used (in^2)**	**Material**	**Cost/in^2**	**Direct Cost**	**True Cost**
Inner Left Panel		126.8	1/8" White Acrylic	$0.05	$0.00	$6.30
Inner Right Panel		126.8	1/8" White Acrylic	$0.05	$0.00	$6.30
Inner Back Panel		125	1/8" White Acrylic	$0.05	$0.00	$6.21
Inner Top Panel		161.67	1/8" White Acrylic	$0.05	$0.00	$8.03
Door		89.5776	1/4" Cleary Acrylic	$0.09	$0.00	$7.90
TOTAL		2718.8201			$0.00	$125.91
IF WE USE A DIFFERENT MATERIAL						
Lower Bottom		252.96	1/8" Black Polystyrene	$0.01	$0.00	$3.04
Upper Bottom		252.96	1/8" White Polystyrene	$0.01	$0.00	$2.78
Outer Front Panel		263.5	1/8" Black Polystyrene	$0.01	$0.00	$3.16
Outer Left Panel		230.64	1/8" Black Polystyrene	$0.01	$0.00	$2.77
Outer Right Panel		230.64	1/8" Black Polystyrene	$0.01	$0.00	$2.77
Outer Back Panel		263.5	1/8" Black Polystyrene	$0.01	$0.00	$3.16
Outer Top Panel		252.96	1/8" Black Polystyrene	$0.01	$0.00	$3.04
Thermal Front Panel		18.6	1/8" Orange Acrylic	$0.05	$0.00	$0.92
Lower Bed Filled		161.60625	1/8" White Polystyrene	$0.01	$0.00	$1.78
Upper Bed		161.60625	1/8" White Polystyrene	$0.01	$0.00	$1.78
Inner Left Panel		126.8	1/8" White Polystyrene	$0.01	$0.00	$1.39
Inner Right Panel		126.8	1/8" White Polystyrene	$0.01	$0.00	$1.39
Inner Back Panel		125	1/8" White Polystyrene	$0.01	$0.00	$1.38
Inner Top Panel		161.67	1/8" White Polystyrene	$0.01	$0.00	$1.78
Door		89.5776	1/4" Clear Acrylic	$0.09	$0.00	$7.90
TOTAL					$0.00	$36.00

Prototype Design 1- *9 dishes, 7" camera height, touchscreen at top front*						
Part	**True Material Used (in^2)**	**Material**	**Actual Cost/in^2**	**Manufacture Cost/in^2**	**True Cost**	**Manufacture Cost**
Bottom	212	1/4" Black HDPE	$0.051	$0.028	$10.75	$5.912
Front Panel	198.75	1/4" Black HDPE	$0.051	$0.028	$10.07	$5.542
Left Panel	232.35	1/4" Black HDPE	$0.051	$0.028	$11.78	$6.479
Right Panel	232.35	1/4" Black HDPE	$0.051	$0.028	$11.78	$6.479
Outer Back Panel	198.75	1/4" Black HDPE	$0.051	$0.028	$10.07	$5.542
Outer Top Panel	168.9375	1/4" Black HDPE	$0.051	$0.028	$8.56	$4.711
Dish Bed	169	1/4" White HDPE	$0.044	$0.021	$7.51	$3.556
Inner Back Panel	191.25	1/4" White HDPE	$0.044	$0.021	$8.50	$4.024
Door	91	1/4" Clear Acrylic	$0.088	$0.039	$8.03	$3.577
TOTAL					**$76.30**	**$45.823**

PURCHASING						
Material	**Parts**	**Width (in)**	**Length (in)**	**Cost/in^2**	**Cost**	
1/4" Black HDPE	bottom, left, right	18	46	$0.051	$41.970	
1/4" Black HDPE	front, outer back, top	16	42	$0.051	34.06222222	
1/4" Black HDPE	extra	16	16	$0.051	12.97608466	
1/4" White HDPE	dish bed, inner back, extra	16	42	$0.044	29.86666667	
1/4" Clear Acrylic	door			-	0	
				Total	**$118.874**	

Table K-2: BETA Box Prototype Cost

BETA BOX					$325
Thermal		Electronics		Structure	
heatbed	12	lights	10	HDPE	80
fan	15	raspberry	35	Compact End-Feed Fastener	9.25
temp sens	11	camera	25	Hex Screw 5/16"-1-1/4	0.2
insulation	3	touchscreen	15	Hex Screw 1/4"-1-1/8	0.2
heat shrink	8	wires	20	Aluminum Extrusion Solid	2.2
		power supply	18	Aluminum Extrusion Hollow	4.8
		relays	10	Round Bumper	2
		transistors	8	Pull Handle	4.4
		cable and usb ports	12	Aluminum Standoff	2.8
		switch	5	Hex Screw 1"	1
				Low-Voltage Raceway	1
				Steel Locknut	1
				Foam Rubber Seal	6
				Cabinet Hinge	3
	49		**158**		**117.85**

L – Thermal Control Matlab Simulink Figures

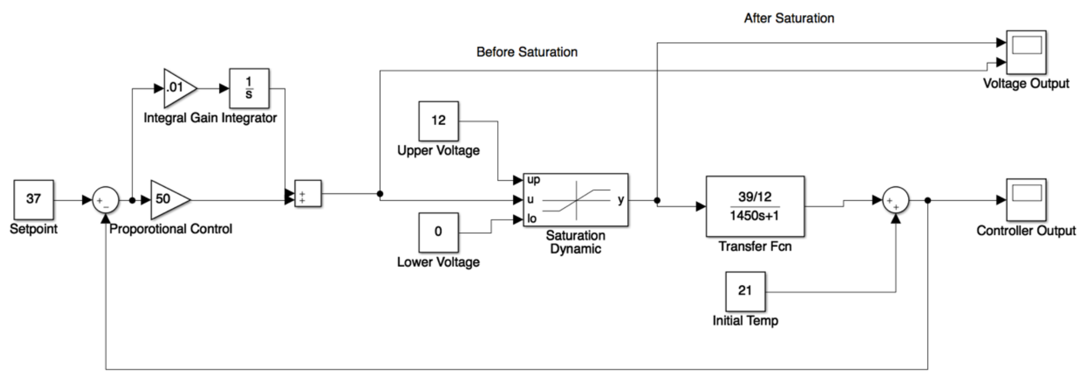


Figure L-1: Control Simulation

M - Copy of Meeting Minutes

3D Bioprinter Meeting Minutes
Senior Design 2017

Location:	RSL	**Meeting Scribe:**	Josie Warren
Meeting Date:	Thursday September 29, 2016	**Meeting Time:**	8:30am

ATTENDANCE:

Team Member	Present	Team Member	Present	Team Member	Present
Kitts	__X_ yes ___ no	Le	__X_ yes ___ no	Warren	t __X_ yes ___ no
Abrams	__X_ yes ___ no	Ososke	__X_ yes ___ no	SE3D	___ yes _X__ no
Barone	__X_ yes ___ no	Plum	__X_ yes ___ no	Other ___________	___ yes ___ no
Bhatnagar	__X_ yes ___ no	Takimoto	__X_ yes ___ no	Other ____________	___ yes ___ no

REPORTS:

Division	Situation	Team Response	Action/Results
MECH:	NA		
BIOE:	NA		
COEN:	NA		
Organization:	Established weekly meeting time of 8:30am Thursdays	All can attend	-
Other:			

ACTION ITEMS (Recently Accomplished):

Who	What	When

NEW BUSINESS:

Member(s) Involved	Discussion	Next Steps
Entire team	Brainstorming what we can do this year with the project, covered whiteboard with sticky notes.	Set up meeting time with SE3D to discuss options and opportunities. This is needed to move forward.
Student team	Need to gain access to RSL	Ososke will submit team names and student ID numbers to Kitts
Student team	-	Plan to meet and gain practice with Arduino

3D Bioprinter Meeting Minutes
Senior Design 2017

Location:	RSL	**Meeting Scribe:**	Emily Takimoto
Meeting Date:	Tuesday, October 4, 2016	**Meeting Time:**	9:00 AM

ATTENDANCE:

Team Member	Present	Team Member	Present	Team Member	Present
Kitts	___ yes _X__ no	Le	__X_ yes ___ no	Warren	___ yes __X_ no
Abrams	___ yes __X_ no	Ososke	__X_ yes ___ no	SE3D	___ yes _X__ no
Barone	__X_ yes ___ no	Plum	_X__ yes ___ no	Other ________	___ yes ___ no
Bhatnagar	__X_ yes ___ no	Takimoto	_X__ yes ___ no	Other _________	___ yes ___ no

REPORTS:

Division	Situation	Team Response	Action/Results
MECH:	N/A		
BIOE:	N/A		
COEN:	N/A		
Organization:	N/A		
Other:	N/A		

ACTION ITEMS (Recently Accomplished):

Who	What	When
Entire Team	Encode ID cards to get access to RSL	By Friday 10/7

NEW BUSINESS:

Member(s) Involved	Discussion	Next Steps
Entire Team	Began work on a project update and problem statement	Continue to edit and revise project summary
Entire Team	Meet with Kevin to understand more about the project and what he has been working on	Follow up with Kevin on Thursday morning meeting time (10/6)

OLD BUSINESS:

Member(s) Involved	Discussion	Next Steps
Entire Team	Set up meeting with SE3D	Waiting to hear from company on meeting time
Entire Team	-	Continue to research project topic and add to the Google Drive

3D Bioprinter Meeting Minutes
Senior Design 2017

Location:	RSL	**Meeting Scribe:**	Warren
Meeting Date:	10/13/16	**Meeting Time:**	8am

ATTENDANCE:

Team Member	Present	Team Member	Present	Team Member	Present
Kitts	MACROBUTTON HTMLDirect ___ yes __X_ no	Le	__X_ yes ___ no	Warren	MACROBUTTON HTMLDirect __X_ yes ___ no
Abrams	__X_ yes ___ no	Ososke	__X_ yes ___ no	SE3D	___ yes _X__ no
Barone	__X_ yes ___ no	Plum	__X_ yes ___ no	Other ___________	___ yes ___ no
Bhatnagar	__X_ yes ___ no	Takimoto	_X__ yes ___ no	Other ____________	___ yes ___ no

REPORTS:

Division	Situation	Team Response	Action/Results
MECH:			
BIOE:			
COEN:	Abrams ordered and brought in raspberry pi and components	Will explore capabilities and opportunites with the raspberry pi for the printer	Still need to finalize understanding of reimbursement process; raspberry pi and components will be property of the project
Organization:			
Other:			

ACTION ITEMS (Recently Accomplished):

Who	What	When
Bhatnagar, Le, Takimoto, Warren	Received maker lab training	10/11/16
Abrams	Submitted proposal for ENGL 181	10/12/16
Bhatnagar	Submitted Oral report in BIOE 194	10/12/16
Ososke	Submitted rough draft of Grant Proposal for School of Engineering to Kitts	10/11/16

NEW BUSINESS:

Member(s) Involved	Discussion	Next Steps
Team	Writing final project proposal for ENGL 181	Accomplish to submit by 10/14/16
Abrams, Barone	Began 3D printing parts to build sample 3D printer	Check on printing progress at end of the day, print more parts this week
MECH Team	Writing report for MECH 194	Complete and submit 10/19/16

3D Bioprinter Meeting Minutes
Senior Design 2017

Location:	RSL	**Meeting Scribe:**	Warren
Meeting Date:	10/13/16	**Meeting Time:**	8am

ATTENDANCE:

Team Member	Present	Team Member	Present	Team Member	Present
Kitts	MACROBUTTON HTMLDirect ___ yes __X_ no	Le	__X_ yes ___ no	Warren	MACROBUTTON HTMLDirect __X_ yes ___ no
Abrams	__X_ yes ___ no	Ososke	__X_ yes ___ no	SE3D	___ yes _X__ no
Barone	__X_ yes ___ no	Plum	__X_ yes ___ no	Other ___________	___ yes ___ no
Bhatnagar	__X_ yes ___ no	Takimoto	_X__ yes ___ no	Other ____________	___ yes ___ no

REPORTS:

Division	Situation	Team Response	Action/Results
MECH:			
BIOE:			
COEN:	Abrams ordered and brought in raspberry pi and components	Will explore capabilities and opportunites with the raspberry pi for the printer	Still need to finalize understanding of reimbursement process; raspberry pi and components will be property of the project
Organization:			
Other:			

ACTION ITEMS (Recently Accomplished):

Who	What	When
Bhatnagar, Le, Takimoto, Warren	Received maker lab training	10/11/16
Abrams	Submitted proposal for ENGL 181	10/12/16
Bhatnagar	Submitted Oral report in BIOE 194	10/12/16
Ososke	Submitted rough draft of Grant Proposal for School of Engineering to Kitts	10/11/16

NEW BUSINESS:

Member(s) Involved	Discussion	Next Steps
Team	Writing final project proposal for ENGL 181	Accomplish to submit by 10/14/16
Abrams, Barone	Began 3D printing parts to build sample 3D printer	Check on printing progress at end of the day, print more parts this week
MECH Team	Writing report for MECH 194	Complete and submit 10/19/16

3D Bioprinter Meeting Minutes
Senior Design 2017

Location:	RSL	**Meeting Scribe:**	Warren
Meeting Date:	10/19/16	**Meeting Time:**	8am, 9am

ATTENDANCE:

Team Member	Present	Team Member	Present	Team Member	Present
Kitts	MACROBUTTON HTMLDirect ___ yes _X__ no	Le	___ yes __X_ no	Warren	MACROBUTTON HTMLDirect __X_ yes ___ no
Abrams	_X__ yes ___ no	Ososke	_X__ yes ___ no	SE3D	_X__ yes ___ no
Barone	__X_ yes ___ no	Plum	__X_ yes ___ no	Other ___________	___ yes ___ no
Bhatnagar	__X_ yes ___ no	Takimoto	__X_ yes ___ no	Other ____________	___ yes ___ no

REPORTS:

Division	Situation	Team Response	Action/Results
MECH:			
BIOE:			
COEN:			
Organization: Team	Need approval for money request from SoE from Kitts	Waiting for approval	Email Kitts
Other:			

ACTION ITEMS (Recently Accomplished):

Who	What	When
Barone, Abrams	3D Printing parts for the 3D Printer kit	This week
MECH	194 report	10/17
Abrams	Aquired control board for RepRap	10/17
Riccomini class	Gave 7 minute presentation	10/16

NEW BUSINESS:

Member(s) Involved	Discussion	Next Steps
Students	Team organization discussion – what are team member strengths and weakness, what is each person excited about and hoping to avoid, what experience does each member have? see responses in file “meeting 10-19-16 outline”	Refer to responses when needed for organization or improving team communication
Students	Recap SE3D Meeting	Wait for scope tasks, evaluate after they come in

SE3D Meeting

Want	Current State	Potential Additions
High Schools	What are the usability opinions right now?	Get feedback to start, Design to enhance high school experience

Priority: LCD Screen	Doesn't exist	
	Reprap pro controller	LCD screen that's compatible already exists
Multiple Extrusion	Can replaces the existing extruder with another system	Would like to do plastic and bio printing
		Dual extrusion system?- not priority, but interested
PCL		Polycaperol lacton- great scaffold
Bigger Syringe	5 mL syringe	
Software	Pronter face, slicing software – slicer, CURA (up to user, keep open source)	LCD panel, chromebooks – requested by teachers, one click and done installation
Biomaterials	Developed in-house, cannot grow any cells or bacteria in RSL, use a lot of alginate in current experiments	Would be interested in trying a new material to print, less interested in improving 3D scaffolding
Color Quantifying Software	Uses open source software supplied by nih - imageJ – turns colors to grayscale and quantifies grayscale – can be ineffective when color change has same concentration on grayscale	Easier software, more accurate software that quantifies color
Mobile App		Create entire program and put on phone for app
Image capturing Software		Incubator camera, camera on printer?- depends on reaction time, lighting inside printer
Incubator	Need more feedback from teachers- cost is a major factor	Leaning towards external, want as separate entity
PRIORITY: Autohoming		Auto-calibration
Unified Control System		System that allows control of all units together – printing, incubator, imaging
Students do research	Biobots, Cell Ink/ Inkredible, Aether, Bio3D Technologies, VisionTech, Regenhu	New assays - Assay printing is SE3D big market differentiation

-will get scope (main 3-4 tasks) by end of today

3D Bioprinter Meeting Minutes
Senior Design 2017

Location:	RSL	**Meeting Scribe:**	Warren
Meeting Date:	10/27/16	**Meeting Time:**	8:30am

ATTENDANCE:

Team Member	Present	Team Member	Present	Team Member	Present
Kitts	MACROBUTTON HTMLDirect __x_ yes ___ no	Le	__x_ yes ___ no	Warren	MACROBUTTON HTMLDirect __x_ yes ___ no
Abrams	__x_ yes ___ no	Ososke	_x__ yes ___ no	SE3D	___ yes _x__ no
Barone	__x_ yes ___ no	Plum	__x_ yes ___ no	Other ___________	___ yes ___ no
Bhatnagar	__x_ yes ___ no	Takimoto	__x_ yes ___ no	Other ____________	___ yes ___ no

REPORTS:

Division	Situation	Team Response	Action/Results
MECH: All	Began building of 3D Printer kit		Continue construction
BIOE: All	" "		Continue construction
COEN: All	" "		Continue construction
Organization: All	Met with Maya, received proposed scope email	We're confused	Kitts met with Maya again

ACTION ITEMS (Recently Accomplished):

Who	What	When
Kitts	Met with Maya, our project might be partially what SE3D needs and partially work/ research on our own part that they may or may not use at the end of the year. Initial thoughts: Doesn't think that color observation analysis/ image processing should be part of our scope, but can be. Temperature control- might not work with original scope, but can do it in a super modular method, keep low cost. Bio Experiments- maybe create one that works best with temperature control Camera- carousel might be over specified- think about the "why" before agreeing with the "how"	10/26/16
Barone	Printed the additional parts that were missing from 3D kit	10/26/16

MEETING WITH KITTS:

Member(s) Involved	What Is Our New Scope?	Proposed Action
Team	Modular Box with lighting and camera control, maybe temperature control	Box- lighting control- are their multiple light levels?, has a camera- what kind of imaging do we need to make the data/pictures quantitatively important- resolution=?, number of photos, rates and samples- timer requirements and memory requirements, supports multiple samples, possibly modular-compatible, add-on temperature control function, (maybe humidity- is it really control? sensed? do you just put a wet sponge inside?) *specify all true requirements
Bhatnagar	New bio experiment	Brainstorm- ideal new experiment would incorporate new additions in Box and with extruder
Team	Multi-arm extruder? problem- current discussion between school and SE3D over intellectual property, solution might be making designs open source	Multi-material- requirements should not imply that this will be implemented, create scaffolding capability (PLA)?- will require a lot of functionality early on in winter quarter and a lot of testing, autohoming, ability to use more material than 5mL, would require changing out the dual board on the printer
Abrams, Le	touch screen- non-plugged in system (they bring and plug in	Box could certainly do, printer might be ad hoc—prioritize making a really nice prototype

	right now)	
Team		Organization- identify "owners" for each aspect of project; still allow for overlap on working across project aspects together, list needs and requirements with an idea of verification methods; create project statement- one sentence "The objective of this project is ___" 3-4 sentences: "To do this, we focused on ___"
Team		For thesis: include project statement- 5 year goals, actual deliverables from this year Ch2: system definition- customer needs, requirements, overview of mechanics, electronics, software, bio; 1st attempt at block diagram, masterview flowchart, bio overview Ch 3-7ish: details on each aspect Integration/Test Results Chapter Professional Issues Chapter: ethics, legal, sustainability, etc. Final Chapter: Summary and Conclusion- lessons learned, etc. Appendices References
Team		DUE next week: Project Statement Paragraph, Customer needs list, a few light Box rough sketches/functionality ideas

NEW BUSINESS:

Member(s) Involved	**Discussion**	**Next Steps**
Team	Brainstorming ideas for Box and extruder	

3D Bioprinter Meeting Minutes
Senior Design 2017

Location: **Meeting Scribe:**

Meeting Date: **Meeting Time:**

ATTENDANCE:

Team Member	**Present**	**Team Member**	**Present**	**Team Member**	**Present**
Kitts	___ yes _X__ no	Le	__X_ yes ___ no	Warren	__X_ yes ___ no
Abrams	___ yes _X__ no	Ososke	__X_ yes ___ no	SE3D	___ yes _X__ no
Barone	__X_ yes ___ no	Plum	_X__ yes ___ no	Other______	___ yes ___ no
Bhatnagar	__X_ yes ___ no	Takimoto	__X_ yes ___ no	Other______	___ yes ___ no

REPORTS:

Division	**Situation**	**Team Response**	**Action/Results**
MECH:	Continue working on 3D printer kit		Continue construction
BIOE:	Continue working on 3D printer kit Project proposal and presentation		Continue construction
COEN:	Continue working on 3D printer kit		Continue construction
Organization:			
Other:			

ACTION ITEMS (Recently Accomplished):

Who	**What**	**When**

NEW BUSINESS:

Member(s) Involved	**Discussion**	**Next Steps**
Team	Project Statement	None
Team	Customer Needs	None
Team	Requirements	None
Team	Baseline Deliverables	None
Team	Stretch Goals	None
Bhatnagar and Takimoto	Project Proposal and Presentation	Continue working
Team	Sketches	Continue working on sketches

3D Bioprinter Meeting Minutes
Senior Design 2017

Location:	RSL	**Meeting Scribe:**	Warren
Meeting Date:	11/3/2016	**Meeting Time:**	8:30

ATTENDANCE:

Team Member	**Present**	**Team Member**	**Present**	**Team Member**	**Present**
Kitts	__x_ yes ___ no	Le	__x_ yes ___ no	Warren	t __x_ yes ___ no
Abrams	__x_ yes ___ no	Ososke	__x_ yes ___ no	SE3D	___ yes _x__ no
Barone	__x_ yes ___ no	Plum	__x_ yes ___ no	Other ___________	___ yes ___ no
Bhatnagar	__x_ yes ___ no	Takimoto	__x_ yes ___ no	Other ____________	___ yes ___ no

REPORTS:

Division	Situation	Team Response	Action/Results
MECH:			
BIOE: Bhatnagar	Meeting with Andy, looking for/ found a molecule that's temp sensitive	Cool- keep going!	Still looking, talking to Andy about possibilities
COEN:			
Organization:			
Other:			

ACTION ITEMS (Recently Accomplished):

Who	What	When
Ososke, Takimoto, Bhatnagar	Project Statement, Preliminary Specifications, Summary of Customer Needs	11/1/16
Warren, Plum, Le, Barone	Worked on construction of 3D Printer- Warren, Plum, and Le on structure and axes, Barone on electronics	11/1/16
MECH Team	Sketches for lightBox, printer	11/3/16
COEN Team	UI sketches and system design sketches and research	11/3/16
Barone	Created specification for syringe for printer	11/1/16

NEW BUSINESS:

Member(s) Involved	Discussion	Next Steps
Team, Kitts	Project Statement- need higher lever first sentence. Add detailed sentences about what we will be developing and exploring on a slightly lower level.	Write a full paragraph, with first sentence encompassing all aspects of what we will be doing.
Team, Kitts	PDS – edits recommended/needed – see written edits by Kitts on printout and online on Google Doc	Modify PDS according to recommendations
Team, Kitts	What does Kitts want to see from us by the end of this quarter? – a very basic prototype concept, show what it does and doesn't do (can be completely made of foam); basic software developments, might be a learning curve; understanding what it takes to run a full existing experiment and use to sketch design for new experiment	DUE: Come in next week with a proposal of what we're going to accomplish by this week
MECH Team	Needs PDS and 10 drawings for 194	Franz- adjust PDS All- scan drawings to turn in 194
Riccomini Team	Need project update, specification	Le- project update w/ minutes, Ososke- update Gannt Chart, reach out to Jeff Barone- CAD Drawing Plum – assist Jeff Warren – edit final writing

3D Bioprinter Meeting Minutes
Senior Design 2017

Location:	RSL	**Meeting Scribe:**	Warren
Meeting Date:	11/10/16	**Meeting Time:**	8:30am

ATTENDANCE:

Team Member	Present	Team Member	Present	Team Member	Present
Kitts	MACROBUTTON HTMLDirect __x_ yes ___ no	Le	__x_ yes ___ no	Warren	MACROBUTTON HTMLDirect _x__ yes ___ no
Abrams	___x yes ___ no	Ososke	__x_ yes ___ no	SE3D	___ yes _x__ no
Barone	__x_ yes ___ no	Plum	__x_ yes ___ no	Other ___________	___ yes ___ no
Bhatnagar	__x_ yes ___ no	Takimoto	__x_ yes ___ no	Other ____________	___ yes ___ no

REPORTS:

Division	Situation	Team Response	Action/Results
MECH:			
BIOE: Bhatnagar	Looked into Chitozan with Andy- doesn't seem as feasible	This was an issue with first senior design team a few years ago	Keep exploring for something that will utilize thermo control
COEN:			
Organization:			
Other:			

ACTION ITEMS (Recently Accomplished):

Who	What	When
Team	2nd revision of the project statement, PDS	11/9/16
Team, Barone	Devised first overall concept for Box, created Solid works of basic Box prototype	11/7/16-11/9/16
Team	Working on printer construction, more time spent on electronic aspects	11/9/16
Bhatnagar	Met with Andy, still trying to figure out experiments, Maya recommended Algae experiments	11/8/16
COEN	Researched Raspberry Pi cameras, setup GIT repository, setup raspberry Pi with correct operating system	11/7/16

NEW BUSINESS:

Member(s) Involved	Discussion	Next Steps
Team, Kitts	Looking at current model of plan and thinking about prototype: material is less important- show all functions: lighting, wiring, LCD, camera (with some capabilities/ features), heating element. We want to see the volume, the physicality, and be able to walk through a scenario. Might even be able to create something that we can walk a teacher or student through, and getting some feedback for the next model.	
Team, Kitts	Camera- someone needs to take charge of finding out ability to take photos from 10" (or whatever) away	
Team, Kitts	Consider function for adding new dishes in the Box- how does it know that a new dish has been added, will the tray have some form that forces locations of the dishes?	
Coen, Kitts	Consider the USB/ photo downloading process with the user interface and quantity of petri dishes that can be	Think about this process further in relation to the pis.

	handled	
Team, Kitts	It would be helpful to think about all the parts that are going to go into the Box- what is each element that needs to be considered?	Create a product breakdown structure
Mech, Kitts	What all should be our final prototypes for the CDR?	Double check with Hight, but most likely just the LightBox prototype will be sufficient if it's quality and shows all the features
Mech, Bios, Kitts	How will we determine the temperature in the Box? Is the Box too big to be heated?	
Kitts	Key questions to consider for senior design: Have we worked out the resolution on the camera? How are we ensuring the temperature of the bio is correct?	Compile this list and make sure all are addressed or answerable by May

3D Bioprinter Meeting Minutes
Senior Design 2017

Location:	RSL	**Meeting Scribe:**	Warren
Meeting Date:	11/17/16	**Meeting Time:**	8:30am; 9am

ATTENDANCE:

Team Member	Present	Team Member	Present	Team Member	Present
Kitts	MACROBUTTON HTMLDirect __x_ yes ___ no	Le	__x_ yes ___ no	Warren	MACROBUTTON HTMLDirect __x_ yes ___ no
Abrams	__x_ yes ___ no	Ososke	__x_ yes ___ no	SE3D	__x_ yes ___ no
Barone	__x_ yes ___ no	Plum	__x_ yes ___ no	Other ___________	___ yes ___ no
Bhatnagar	__x_ yes ___ no	Takimoto	_x__ yes ___ no	Other ____________	___ yes ___ no

REPORTS:

Division	Situation	Team Response	Action/Results
MECH: Team	Presentations week 10, we all need to present for class, but it's good practice to get everyone to present together	Each group has own class presentations	COEN- can present in own class MECH- present in own class
BIOE:			
COEN:			
Organization: Team, Kitts	Sent out When2Meet	Filled out by all	Best time in Thursdays, will need to meet in smaller groups, too

ACTION ITEMS (Recently Accomplished):

Who	What	When
Team, Kitts **Response**	Created a prototype! -like the heating modularity, but how will it achieve? -test dips in temp as door opens and closes	11/15/16
Team	Continuing construction of 3D Printer	11/15/16-11/17/16

NEW BUSINESS:

Member(s) Involved	Discussion	Next Steps
Team, Kitts	Funding- haven't heard from school of engineering yet, but we will get enough funding from Kitts	Wait to hear from School of Engineering
Team	Final reports do in ENGL 181	Keep going, looking good
Team, Kitts	Meeting times for Winter quarter, sent out a doodle, Thursdays 8-9am work best	Established meeting time in Winter Quarter for 8-9am on Thursdays
Team, Kitts	Going to meet with Maya at SE3D	Bring prototype to Maya

MEETING WITH SE3D (MAYA):

Member(s) Involved	Discussion	Next Steps
Team- material	Demonstrated prototype -what plastic will be best, thermal? -they are moving away form acrylic bc of manufacturing	-explore materials heavily, consider manufacturing
MECH, COEN - extruder	Using SE3D's existing extruder for study on additional extrusion. SE3D already has patent for it	
MECH, COEN - autohome	Autohoming, sensor plan is to be on plunger head, concern	Be careful with IP issues, keep all options on

	with sensitivity needs	table for now
SE3D, Team	Left a printer in EDVenture lab for team to play with	
MECH, Kitts – thermal control	Thoughts on thermal control? -to grow cells, you need humidity and CO2 -camera lense fogginess, condensation -condensation with short circuit? -humidity is very nice to have -CO2 could be done later, but think about it -oven door idea? -want to seal, take considerations -keep simple for high school students -confirming that initial build is ok with thermal control	
BIOE, MECH - experiment	Time needed to grow a million cells- 1week to 2 months	
Team - size	Size of Box? -right now can see all 9 images at once -test if you can see well enough with one image for all 9 dishes, concern that won't be enough -experiment with various camera options	-test what we have with image J -likes the ability to do 9 in one Box
Member(s) Involved	**Discussion**	**Next Steps**
BIOE - experiment	Interest in growing cells? Further the algae experiment? -with Incubator, could test culturing -could play with algae optimization with variation of color and brightness	Can explore a lot of options with the existing algae experiment
Kitts - modularity	-possibility of modularity for multiple Boxes -Maya—1 Box of 9 is better than 2 Boxes of 4	
COEN - screen	-LCD for Box and raspberry pi MAYA prefers touch screen over LCD -there is a price limit, though	
Team - cost	Expected unit as a whole cost? -initial cost of goods for printer was $500, labor cost was outrageous -takes 1 wk to build rebel printer -challenging to guess for Box -consider manufacturing and labor when considering cost	
COEN - UI	Demonstrated UI mock-ups to Maya -walked through entire scenario -preloaded experiment settings -settings would just be temperature, image capture, number of dishes in Box -can we get students to set it themselves? -settings for pictures, if camera takes 1 dish at time, will need to coordinate image timing	-think about options of saving a setting, rather than we program the settings
Team - aesthetics	Asthestics? Can put SE3D Logo on it	Have fun with it! Surprise Maya! -white on inside is good to consider with lighting and image capturing -play with surface texture and affect on imaging
Team - safety	Safety, high school students -moving components, add Emergency Stop Botton -safety feature for the heating -look into overheating issues with any heater we choose	

3D Bioprinter Meeting Minutes
Senior Design 2017

Location:	RSL	**Meeting Scribe:**	Warren
Meeting Date:	January 12, 2017	**Meeting Time:**	8:30 am

ATTENDANCE:

Team Member	Present	Team Member	Present	Team Member	Present
Kitts	MACROBUTTON HTMLDirect ___ yes _x__ no	Le	__x_ yes ___ no	Warren	MACROBUTTON HTMLDirect __x_ yes ___ no
Abrams	__x_ yes ___ no	Ososke	__x_ yes ___ no	SE3D	___ yes __x_ no
Barone	__x_ yes ___ no	Plum	__x_ yes ___ no	Other ___________	___ yes ___ no
Bhatnagar	__x_ yes ___ no	Takimoto	__x_ yes ___ no	Other ____________	___ yes ___ no

REPORTS:

Division	Situation	Team Response	Action/Results
MECH:			
BIOE:			
COEN: Abrams	Not yet reimbursed for equipment	Barone sent Abrams email for finances	Will send some emails to get reimbursed
Organization:			
Other:			

ACTION ITEMS (Recently Accomplished):

Who	What	When
MECH	Worked on RepRap electrical	Monday, Wednesday

NEW BUSINESS:

Member(s) Involved	Discussion	Next Steps
Team	Need to order equipment: heating element, Nema Stepper motor or 2, laptop charger, force sensor	Finalize list of parts to order this week and submit within 2 weeks
Abrams	Will need Max soon to help with RepRap	
MECH	Assigned specialization: Barone: Heating Element Plum: Lighting Ososke: Dual Extruder Warren: Structures Takimoto: Force Sensor	Take initiative on individual aspects: create full part list, begin thinking about calculations
COEN	We have a new nice camera, let's test it out!	Goal next 2-3 weeks: due ImageJ analysis with cameras we have and dots on petri dishes, return with brief presentation
Team	Need more time to meet	Meet next week 9-10am on Tuesday, bring finalized list of material orders
BIOE	Will be responsible for all biomaterials we need for experiment	Come up with list of experiment-related materials that need to be ordered now, and a list for what we will need in the future
Team	Registering for design conference, working on title and descript	Come back next Tuesday each with one idea for title/name
MECH	Assignments due for 195	Continue parts list

3D Bioprinter Meeting Minutes
Senior Design 2017

Location:	RSL	**Meeting Scribe:**	Warren
Meeting Date:	1/20/17	**Meeting Time:**	8am

ATTENDANCE:

Team Member	Present	Team Member	Present	Team Member	Present
Kitts	MACROBUTTON HTMLDirect __x_ yes ___ no	Le	__x_ yes ___ no	Warren	MACROBUTTON HTMLDirect __x_ yes ___ no
Abrams	__x_ yes ___ no	Ososke	__x_ yes ___ no	SE3D	___ yes __x_ no
Barone	___ yes _x__ no	Plum	__x_ yes ___ no	Other ___________	___ yes ___ no
Bhatnagar	__x_ yes ___ no	Takimoto	__x_ yes ___ no	Other ____________	___ yes ___ no

REPORTS:

Division	Situation	Team Response	Action/Results
MECH:	RepRap turns on!	But it's not fully working	Get some code going, make it REALLY working
BIOE:			
COEN:			
Organization:			
Other: Kitts	NDA- didn't have a term length	School didn't accept, returned with a request for a term	SE3D should return with new NDA with 3 year term

ACTION ITEMS (Recently Accomplished):

Who	What	When
MECH	Wired reprap, it turns on	This week
COEN	Taken preliminary pictures with camera for evaluation	Last 2 weeks
BioE	Researched biomaterials and living needs	Last 2 weeks
TEAM	Completed detailed master Gantt chart	This week

NEW BUSINESS:

Member(s) Involved	Discussion	Next Steps
MECH	Hardware goals, working on what are these?	Talk to Hight today, work on assignment for Monday this weekend
MECH	Force sensor, is our existing mindset design very feasible? Kitts isn't confident it's going to work	Get working on force sensor NOW
MECH	Dual extrusion, can get working on that soon, as well	Start early on this as well
Bhatnagar	Contacted Asuri to talk about bio materials, hoping to do a bio-luminescent material, now focusing on isolating luciferase, exploring concentration variance	Meeting with Asuri today, wait for SE3D waiver before working there, contact Andy to ask about waiver, don't work in BioE lab at SCU
Kitts	Maya still owes Kitts a waiver to approve working at the SE3D lab	
COEN	Need to finalize hashing of pixilation of camera, angle, resolution—PRIORITY	Come next week with report for team, also make something that can be presented to SE3D, so they can approve—provide final number of pixilation and determine if we think is sufficient and why
MECH, BIOE	Temp for luciferase is 37 degrees Celsius	
COEN	Using imageJ for color analysis- this is an option, but is a	Start with camera resolution, go from there to

	stretch goal	see how far we get
Kitts, COEN	Sees camera as major priority	Run a lot of tests early on
Kitts, BIOE, MECH	We need to evaluate change in temp with the door opening—what is the effect, change operating conditions, change instructions, change design?	This will need to be tested early on- structure and heating must be developed to test
TEAM	Name chosen for design conference: BETA	Register for conference

3D Bioprinter Meeting Minutes
Senior Design 2017

Location:	RSL	**Meeting Scribe:**	Warren
Meeting Date:	1/26/17	**Meeting Time:**	8am

ATTENDANCE:

Team Member	Present	Team Member	Present	Team Member	Present
Kitts	MACROBUTTON HTMLDirect __x_ yes ___ no	Le	__x_ yes ___ no	Warren	MACROBUTTON HTMLDirect __x_ yes ___ no
Abrams	__x_ yes ___ no	Ososke	__x_ yes ___ no	SE3D	___ yes __x_ no
Barone	__x_ yes ___ no	Plum	__x_ yes ___ no	Other ___________	___ yes ___ no
Bhatnagar	__x_ yes ___ no	Takimoto	__x_ yes ___ no	Other ____________	___ yes ___ no

REPORTS:

Division	Situation	Team Response	Action/Results
MECH:			
BIOE:			
COEN:			
Organization:			
Other:			

ACTION ITEMS (Recently Accomplished):

Who	What	When
Ososke, Plum, Barone , Abrams	RepRap is working, need to solder one wire, but it's functioning!	This week
Warren	CAD of thermal frame- want to order parts so can assemble and test	Yesterday
Abrams, Le	Developed calculations for fish eye lense/ camera	Past 2 weeks
Takimoto	Force sensor- got working with Arduino	This week

NEW BUSINESS:

Member(s) Involved	Discussion	Next Steps
Barone, Kitts	Need to get thermal chamber up and running, so we can test it- primary concern	Ignore PID for now, just use a thermostat-- #1
Abrams, Kitts	Presented camera/ pixel info, camera resolution is 5 Megapixels	#2 priority
Warren	What materials do we NEED for the structure—can we substitute materials over polycarbonate, because so expensive	Evaluate other material options, order only what it necessary
Le, Kitts	Materials due for class soon	Send to Kitts
COEN, Kitts	Software for capacity to follow multiple samples put in incubator at different times	Begin working on this software- #3 priority
MECH, COEN	Integration between Box and raspberry pi	
Warren, COEN	Touch screen- what are we dealing with size wise and orientation wise	Touch screen is here- look at and determine how to integrate
MECH, COEN	Power supplies- how do we integrate to make one plug, do we have the best equipment?	Evaluate what we have ordered and is delivered- is it functional to what we need, is it the smallest size for our function/ price
MECH, COEN	Does orientation of dishes in each bed spot matter? Do we need to address a way to ensure that orientation is constant?	Ask Maya her opinion, explore the consequences if not address
TEAM, Kitts	As much interaction as we can have with SE3D, the more we can know if we're going down the correct path	Send Maya an email/ set up a meeting?
Warren, MECH	Want to start iterating structure and evaluating material choices	Go to TapPlastics in person- buy; set up a meeting with Don
Bhatnagar	Going to meet with Andy today	Will determine what SE3D sees as a 'good enough' picture

3D Bioprinter Meeting Minutes
Senior Design 2017

Location:	RSL	**Meeting Scribe:**	Warren
Meeting Date:	2/2/17	**Meeting Time:**	8:30am

ATTENDANCE:

Team Member	Present	Team Member	Present	Team Member	Present
Kitts	MACROBUTTON HTMLDirect ___ yes _x__ no	Le	__x_ yes ___ no	Warren	MACROBUTTON HTMLDirect __x_ yes ___ no
Abrams	__x_ yes ___ no	Ososke	__x_ yes ___ no	SE3D	___ yes __x_ no
Barone	___ yes __x_ no	Plum	__x_ yes ___ no	Other ___________	___ yes ___ no
Bhatnagar	___ yes __x_ no	Takimoto	__x_ yes ___ no	Other ____________	___ yes ___ no

REPORTS:

Division	Situation	Team Response	Action/Results
MECH:			
BIOE:			
COEN:			
Organization:			
Other:			

ACTION ITEMS (Recently Accomplished):

Who	What	When
Warren, Plum, Barone	Submitted drawings to Don yesterday for review, meeting with him today or tomorrow, planning to cut all parts tomorrow or next week	This week
Takimoto, Plum	Made part to place at top of syringe, obtained arduino code to convert sensing to force, force sensor is showing diagnostics	This week
COEN	Thermostat program, bed integration, and temperature sensor hookup, GUI development, touch screen calibration, BETA version from RepRap firmware, camera analysis	This week
Ososke, Barone	Circuitry for heater element for relay and touch screen sensor, camera analysis	This week

NEW BUSINESS:

Member(s) Involved	Discussion	Next Steps
Warren	Acrylic is very expensive- looking into ways to cut down on size or change material to lower cost	Material research
Ososke	Massive distortion with 5MP camera, 8MP camera gives good resolution, but need 12in height	Exploring ways around this- camera, lense, Box sizing

3D Bioprinter Meeting Minutes
Senior Design 2017

Location:	RSL	**Meeting Scribe:**	Warren
Meeting Date:	2/9/17	**Meeting Time:**	8:45am

ATTENDANCE:

Team Member	Present	Team Member	Present	Team Member	Present
Kitts	MACROBUTTON HTMLDirect ___ yes _x__ no	Le	__x_ yes ___ no	Warren	MACROBUTTON HTMLDirect __x_ yes ___ no
Abrams	__x_ yes ___ no	Ososke	_x__ yes ___ no	SE3D	___ yes _x__ no
Barone	__x_ yes ___ no	Plum	__x_ yes ___ no	Other ___________	___ yes ___ no
Bhatnagar	___ yes _x__ no	Takimoto	__x_ yes ___ no	Other ____________	___ yes ___ no

REPORTS:

Division	Situation	Team Response	Action/Results
MECH:			
BIOE:			
COEN:			
Organization:			
Other:			

ACTION ITEMS (Recently Accomplished):

Who	What	When
Barone	CADing 3rd prototype model	This week

NEW BUSINESS:

Member(s) Involved	Discussion	Next Steps
COEN	Do the lights need to be programmed for different colors?	yes
MECH	Still working on temperature characterization	
TEAM	We NEED a full working prototype by next week	
Barone	Finishing CADing 3rd prototype	Finish by Saturday for Abicus
Le	GUI first round	Will implement on Pi by next Thursday
Abrams	GUI work, calibrating on touch screen	
Warren	Will assemble plastic prototype 2, will laser cut prototype 3 , need more material to order, but wait until we get a better handle on budget	Assemble, buy screws potentially
Plum	Assisting as needed	Will be assigned new specific task
Takimoto	Running experiment with force sensor	Can start testing with new force sensor, will set up another meeting with Maxine
Ososke	Running heating element, need to test inside Box	Characterize the transfer function

3D Bioprinter Meeting Minutes
Senior Design 2017

Location:	RSL	**Meeting Scribe:**	Warren
Meeting Date:	2/16/17	**Meeting Time:**	8am

ATTENDANCE:

Team Member	Present	Team Member	Present	Team Member	Present
Kitts	MACROBUTTON HTMLDirect ___ yes __x_ no	Le	__x_ yes ___ no	Warren	MACROBUTTON HTMLDirect __x_ yes ___ no
Abrams	___ yes _x__ no	Ososke	__x_ yes ___ no	SE3D	___ yes _x__ no
Barone	__x_ yes ___ no	Plum	__x_ yes ___ no	Other: Greg Richmond	__x_ yes ___ no
Bhatnagar	___ yes __x_ no	Takimoto	__x_ yes ___ no	Other _________	___ yes ___ no

REPORTS:

Division	Situation	Team Response	Action/Results
MECH:			
COEN:			
Organization: Team	Lots of interested mentors!	We met with Greg Richmond today, a few others who we have been emailing	
Other:			

ACTION ITEMS (Recently Accomplished):

Who	What	When
Warren	Talked to Don about other potential materials, he's waiting to hear back from his laser cutter supplier	This week
Barone	Finished CAD model of next working prototype	This week
Takimoto	Coding the Arduino and testing feasibility study, started getting some force graphs for the sensor	This week
MECHs	Assembled first working prototype- lots and LOTS of time was spent assembling structure, wiring, heating elements	Since Friday
COENs	Loaded new code onto new pi, camera integration to GUI, touch screen is setup, GUI controls the lights	This week

NEW BUSINESS:

Member(s) Involved	Discussion	Next Steps
Team, Richmond	Familiarized with status of our project for mentoring purposes. Presented existing functional prototype. Introduced feasibility study. Control board and dual extruder. NDA/IP issues.	Be careful of technical squirrels- things that are interesting and distracting, but don't help reaching end goal
Team, Richmond	Exploring financial costs, look at what works and where we can cut down on costs	Find out the expected manufacturing cost margin by SE3D
Team, Richmond	Single circuit board is a good idea	Explore the competition, find out more about what other companies have/do
Barone, Richmond	Changes desired for next prototype: -camera height—make 12in above dishes -improve manufacturability—assembly of screws needs to be easier -aluminum extrusion instead of angle brackets in corners -need a mounting mechanism for the lights, changing light strips	-Ososke would like to print a 1/10th scale model in foam core board before making a full scale full material next iteration -Richmond recommends exploring molding for manufacturability, simplify pieces so some sides are the exact same
Team, Richmond	we are not looking for a vast knowledge of the exact	Test the uniformity of the temperature- explore

	temperature everywhere all the time, we are looking for uniform temperature	more thermal testing
Team, Richmond	Speed of existing temperature time is slow- right now we need to preheat	Ask Maya if preheat option is ok
Richmond	His experience is in taking a new product for a market and putting it into production	Get through next generation of changes on the Box, explore manufacturing options, simplify parts wherever possible
Richmond	Think about telling the story-	
Richmond	-What do you really want from me?	-Advice, reference points for exploring manufacturing

3D Bioprinter Meeting Minutes
Senior Design 2017

Location:	RSL	**Meeting Scribe:**	Warren
Meeting Date:	2/21/17	**Meeting Time:**	8am

ATTENDANCE:

Team Member	Present	Team Member	Present	Team Member	Present
Kitts	MACROBUTTON HTMLDirect ___ yes _x__ no	Le	___ yes _x__ no	Warren	MACROBUTTON HTMLDirect __x_ yes ___ no
Abrams	___ yes __x_ no	Ososke	__x_ yes ___ no	SE3D	___ yes __x_ no
Barone	__x_ yes ___ no	Plum	__x_ yes ___ no	Other: Greg Richmond	__x_ yes ___ no
Bhatnagar	___ yes _x__ no	Takimoto	___ yes _x__ no	Other _________	___ yes ___ no

REPORTS:

Division	Situation	Team Response	Action/Results
MECH:			
BIOE:			
COEN:			
Organization:			
Other:			

ACTION ITEMS (Recently Accomplished):

Who	What	When
Team, SE3D	Meet with SE3D, she also wants to avoid acrylic because too brittle	Friday
Richmond	Gave us a book called *Design for Manufacturability* – look at it to explore how to best manufacture the Box	Today
Warren	Created the headlines for draft 1 of thesis, will comment on where everyone can add information and will start to add text from CDR	This week

NEW BUSINESS:

Member(s) Involved	Discussion	Next Steps
Team	Want to try thermal testing with camera from Yuen	Make sure failsafe software is coded, ask for thermal camera
Team	Where can we reduce space to shrink down the Box and save materials	Looking about removing double-paneled sides, how can we compartmentalize the touch screen wiring
Team	Touch screen- we're meeting the requests of SE3D, but are we making the best possible outcome?	Look at all ways for 'touching' the screen
Team	Wanting to plan 24-hr testing	Need to coordinate with Kitts and Anne when/where we can run a 24 hr test
Team, SE3D	Asked Maya about the L-door- she doesn't have a preference	L-door is an option, if truly not-cool with it, we can explore something else
Ososke, Barone	Played with Eagle, looking to make more developments with it after doing hand calculations	Keep learning how to do it
MECH	Analysis due March 6th	Find out exactly what is due, do it
MECH	Maya is happy to share the 3D printed compartments for dual extruder experiments	Maya needs to talk to Kitts about it, first, or vice versa
Plum	Fan layout for heating element needs to be modified- we need more space beneath the fans to pull air	Create a design to lift the fans off the floor of the Box
MECH, Richmond	Chicken test – iterative quick testing; don't worry about which set up is best, do a quick test	If we get the thermal camera- test multiple configurations and just see what changes with different configurations; decide what different configurations we want to try
Ososke, Plum, Richmond	For plate and fan testing, we should try various options, try a baseline of hot plate with no fans	Plum- come up with varying configurations to test: change fan height, plate height, fan

		orientation, plate orientation, fan location, plate location
MECH, Richmond	We’re not looking to move air, we’re looking to stir air and create an even distribution	Explore all fans that we have/that are coming

3D Bioprinter Meeting Minutes
Senior Design 2017

Location:	RSL	**Meeting Scribe:**	Warren
Meeting Date:	2/23/17	**Meeting Time:**	8am

ATTENDANCE:

Team Member	Present	Team Member	Present	Team Member	Present
Kitts	MACROBUTTON HTMLDirect __x_ yes ___ no	Le	__x_ yes ___ no	Warren	MACROBUTTON HTMLDirect __x_ yes ___ no
Abrams	__x_ yes ___ no	Ososke	__x_ yes ___ no	SE3D	___ yes _x__ no
Barone	__x_ yes ___ no	Plum	__x_ yes ___ no	Other: Sam Varney	__x_ yes ___ no
Bhatnagar	___ yes __x_ no	Takimoto	__x_ yes ___ no	Other: Greg Barsky	__x_ yes ___ no

REPORTS:

Division	Situation	Team Response	Action/Results
MECH:			
BIOE:			
COEN:			
Organization: Team	Google doc thesis- can we consider a different format	Latex is an option, Overleaf, group word doc	Stick with Google doc for now as we compile info- switch program after May when we just need to worry about formatting
Other:			

ACTION ITEMS (Recently Accomplished):

Who	What	When
Team, Varney, Barsky	Introduced project and status to mentors, met mentors	today
Ososke, Warren	Met with Yuen yesterday, received permission to borrow thermal camera for testing next week	yesterday
Plum	Ran Maker Bot to print some stands for thermal testing	This morning

NEW BUSINESS:

Member(s) Involved	Discussion	Next Steps
Kitts, COEN	Timing for imaging program- how is the timing planned for multiple samples	Coding that guide this week
Team, Mentors, Kitts	Temperature control in Box- our current temperature sensor is more expensive than necessary, where will be the placement of the sensor, how will the temperature inside the dish be different from the temperature in the Box	Get data on dish and Box in thermal testing
MECH, Kitts	Analysis due in 2 Mondays- can be FEA, can be thermal analysis, control analysis	Try to pick something actually relevant
Team, Barsky	Looking at electrical system and power in Box- Barsky has electrical background	Try switching heat element on and off with oscilloscope- if no effect is seen, don't need additional components, if we see an effect, we need bulk capacitance
Barone, Ososke, Abrams, Barsky	Hoping to use Eagle to make a custom PCB board	
Team, Barsky	Discussion about team organization and leading	Keep motivation strong, have purpose at meetings- know what the goal of each meeting is, record regularly- keep good records as we go
Team, Barsky	Shipping issues- mechanical testing	Before end of year, do drop test

3D Bioprinter Meeting Minutes
Senior Design 2017

Location: RSL | **Meeting Scribe: Jeff**

Meeting Date: 3/2/17 | **Meeting Time: 8:00am**

ATTENDANCE:

Team Member	Present	Team Member	Present	Team Member	Present
Kitts	_X_ yes ___ no	Le	_X_ yes ___ no	Warren	_X_ yes ___ no
Abrams	_X_ yes ___ no	Ososke	_X_ yes ___ no	SE3D	___ yes _X_ no
Barone	_X_ yes ___ no	Plum	_X_ yes ___ no	Other ________	___ yes ___ no
Bhatnagar	___ yes _X_ no	Takimoto	_X_ yes ___ no	Other _________	___ yes ___ no

REPORTS:

Division	Situation	Team Response	Action/Results
MECH:	Analysis doc	Will work together on the presentation tomorrow	Use FEA to test heat distribution
COEN:			
Organization:			
Other:			

ACTION ITEMS (Recently Accomplished):

Who	What	When
Josie, Jake	Thermal heat plate distribution testing	3/1
Jake, Jeff, Kitts	Met with Kitts for proper electronics design for the system electronics	2/27
COENS	File name setup. Algorithm for camera picture taking is being designed.	2/28

NEW BUSINESS:

Member(s) Involved	Discussion	Next Steps
MECHS	More tests for thermal stuff. With different configurations for better heat distribution	Continue testing with the FLIR camera
MECHS	FEA analysis and presentation for the analysis report	Continue working on the thermal simulation and resulting presentation for next Mon.
Kitts, Mechs,	New testing for thermal system	test moving the heat source away from well plates. Try using a space heater element for more heating.
Emily, Mentor	Look at FEA design to get it working for the analysis report	troubleshoot program
Mechs, Kitts	Where to put the temperature sensor to get the best readings. Maybe encase the sensor in a petri dish like casing to mimic temp inside experiments.	Try mounting around the 4 corners of the bed.

3D Bioprinter Meeting Minutes
Senior Design 2017

Location:	RSL	**Meeting Scribe:**	Warren
Meeting Date:	3/6/17	**Meeting Time:**	8am

ATTENDANCE:

Team Member	Present	Team Member	Present	Team Member	Present
Kitts	MACROBUTTON HTMLDirect ___ yes __x_ no	Le	___ yes _x__ no	Warren	MACROBUTTON HTMLDirect __x_ yes ___ no
Abrams	___ yes __x_ no	Ososke	__x_ yes ___ no	SE3D	___ yes _x__ no
Barone	__x_ yes ___ no	Plum	__x_ yes ___ no	Other: Greg Richmond	__x_ yes ___ no
Bhatnagar	___ yes _x__ no	Takimoto	__(x)_ yes __x_ no	Other ________	___ yes ___ no

REPORTS:

Division	Situation	Team Response	Action/Results
MECH:			
BIOE:			
COEN:			
Organization:			
Other:			

ACTION ITEMS (Recently Accomplished):

Who	What	When
MECH, Richmond	Review of analysis presentation, overall well done, a little lost near the end of the body of the presentation; need further explanation of WHY assumptions and simplifications were made- further analysis of failings	Presented yesterday, review today

NEW BUSINESS:

Member(s) Involved	Discussion	Next Steps
Warren	Wanting to test final material choices- laser cut aesthetics and manufacturability test, impact test, thermal test	Outline and perform these tests on all five sample materials this week- WARREN
MECH	Thermal system options- still need a final setup choice but we don't like anything we've tested. New options: fiberglass/insulation above heat plate above fans; 9 smaller resistors distributed above fans; larger heat plate that covers base underneath perpendicular fans; resistor on back and fans on top	Going to test 4 small resistors in mini array tomorrow, test heat plate in corner to simulate large heat bed, insulated heat bed, upper thermal setup
Richmond, MECH	Recommendations for testing- just simulate, don't build whole thing, write down variables- heat source, orientation, etc.	Run more tests tomorrow- OSOSKE, PLUM
MECH	Temperature sensor and casing—found potential products online	Explore position locations within Box
MECH	auto calibration update- stuck on making a controller through pronterface, can arduino commands be used?	TAKIMOTO- talk to Abrams,
MECH	Dual extruder update, need to coordinate movements, test temperature effects	Print out dual extruder concept tomorrow- BARONE

3D Bioprinter Meeting Minutes
Senior Design 2017

Location:	RSL	**Meeting Scribe:**	Warren
Meeting Date:	3/9/17	**Meeting Time:**	8am

ATTENDANCE:

Team Member	Present	Team Member	Present	Team Member	Present
Kitts	MACROBUTTON HTMLDirect ___ yes _x__ no	Le	__x_ yes ___ no	Warren	MACROBUTTON HTMLDirect __x_ yes ___ no
Abrams	__x_ yes ___ no	Ososke	__x_ yes ___ no	SE3D	___ yes _x__ no
Barone	__x_ yes ___ no	Plum	__x_ yes ___ no	Other: Greg Barsky	__x_ yes ___ no
Bhatnagar	___ yes _x__ no	Takimoto	__x_ yes ___ no	Other _________	___ yes ___ no

REPORTS:

Division	Situation	Team Response	Action/Results
MECH: Ososke	Thermal testing- don't know best way to do upper side heater test	Safety concerns for students and upper heating locations, structural issues	Scratched upper side location test
BIOE:			
COEN:			
Organization:			
Other:			

ACTION ITEMS (Recently Accomplished):

Who	What	When
Warren	Cut HDPE	yesterday
Ososke, Plum	Thermal testing- ran 2 new tests, tried insulation- completely flipped heating curve, did get rid of hot spot	yesterday
COEN	GUI!- showed existing version to us- algorithm is good for having different dishes with different photo needs	This week
COEN	Worked on lights	yesterday

NEW BUSINESS:

Member(s) Involved	Discussion	Next Steps
Warren	Final materials testing- will run tests this week and make selection matrix, waiting for last samples to arrive	Create selection matrix, test materials that are here
Plum	Thermal testing- 4 small resistor matrices	Determine optimal thermal setup based on tests run this week
TEAM, Barsky	Thermal testing discussion- can we just take longer to preheat and reduce voltage? What are other setup options	Barsky recommends to run simulations- test heat cycling, how much heat, setup
TEAM	Look forward to control development	Finish thermal decisions so we can get started on controls
Takimoto, COEN	Needs some help integrating arduino code	Send the link to COENs
Takimoto	Will order more materials this afternoon	
COEN	Only 5 open pins on pi- just a heads up	

3D Bioprinter Meeting Minutes
Senior Design 2017

Location:	RSL	**Meeting Scribe:**	Warren
Meeting Date:	3/16/17	**Meeting Time:**	8am

ATTENDANCE:

Team Member	Present	Team Member	Present	Team Member	Present
Kitts	MACROBUTTON HTMLDirect __x_ yes ___ no	Le	__x_ yes ___ no	Warren	MACROBUTTON HTMLDirect __x_ yes ___ no
Abrams	__x_ yes ___ no	Ososke	__x_ yes ___ no	SE3D	___ yes _x__ no
Barone	__x_ yes ___ no	Plum	__x_ yes ___ no	Other: Greg Richmond	__x_ yes ___ no
Bhatnagar	___ yes __x_ no	Takimoto	__x_ yes ___ no	Other: Greg Barsky	__x_ yes ___ no

REPORTS:

Division	Situation	Team Response	Action/Results
MECH:			
BIOE:			
COEN:			
Organization:			
Other:			

ACTION ITEMS (Recently Accomplished):

Who	What	When
MECH	Determined best thermal setup: 4 66CFM fans perpendicular underneath heat bed with thin film of insulation over heat bed- achieves +/-1ºC across entire dish bed	This week
Ososke	Modeled system with optimal thermal setup, began simulating control on Matlab	yesterday
COEN	Done with file export, dish status, camera capture, lighting RGB slider control, temperature initialized	Past couple weeks

NEW BUSINESS:

Member(s) Involved	Discussion	Next Steps
Ososke, MECH	Working on thermal control, Matlab simulation- how to check that output voltage is constrained to 12V?	Look at feedback equation and reevaluate what the gains signify
Kitts, Richmond, MECH	If current thermal system works, fine, but having concerns that it will not work long term	If having trouble- definitely try a model where the heater is farther away from the dishes- place above- if concerned about acrylic, insulate between heater and acrylic
COEN, Richmond	Current GUI	Go find non-technical students/persons and see how well they can understand it to test effectiveness
COEN	Want to add a section that's authorized user only section, safety addition to touch screen so locked down on occasion	Ask Maya her opinion
TEAM, Richmond	If we are going to test in the field, make sure to do it BEFORE the conference, so we have that to present	GOAL- have something ready to test by preview days when we have to present!
COEN, Richmond, Kitts	Lighting colors for experiment- nice to have some presets	Keep custom option, but include preset options for established experiments
COEN, Barsky, Kitts	Potential user errors- incorrectly loading dish bed,	Test with real users

	accidentally reprogramming another's dish location	
TEAM	Honors stuff	Turn in honors required submissions by tomorrow
Warren	Ethics paper- have someone else edit after done	Finish paper
Le, TEAM	Poster for Preview Days- showed tri-fold outline	Determine who exactly is presenting
Takimoto, Kitts	Force sensor- we can see air bubbles, slowed down sample rate to once per second to mitigate noise in data, G-code is done, so far only with lotion	Going to change feed rate and see what happens, run G-code by COENS, run some tests with water
Plum	Dual extruder- adjusting files from SE3D to work with tolerance on MakerBots,	Determine how to attach PLA extruder to biosquirter mold -Can adjust firmware or ramps board
Barone, TEAM, Barsky, Richmond, Kitts	Most recent CAD model- addresses how to attach final outer panel, all electronics in back, where will all the wires go?	Make a wiring diagram, consider running wires along inside ceiling and just using a cover, consider holes in bottom chunk
Barsky, Abrams, Ososke	Transistors issues from last week- need really fast on/off	Good to go

3D Bioprinter Meeting Minutes
Senior Design 2017

Location:

Meeting Scribe:

Meeting Date:

Meeting Time:

ATTENDANCE:

Team Member	Present	Team Member	Present	Team Member	Present
Kitts	__x_ yes ___ no	Le	_x__ yes ___ no	Warren	_x__ yes ___ no
Abrams	_x__ yes ___ no	Ososke	__x_ yes ___ no	SE3D	___ yes _x__ no
Barone	_x__ yes ___ no	Plum	__x_ yes ___ no	Other ________	___ yes ___ no
Bhatnagar	___ yes _x__ no	Takimoto	__x_ yes ___ no	Other _________	___ yes ___ no

REPORTS:

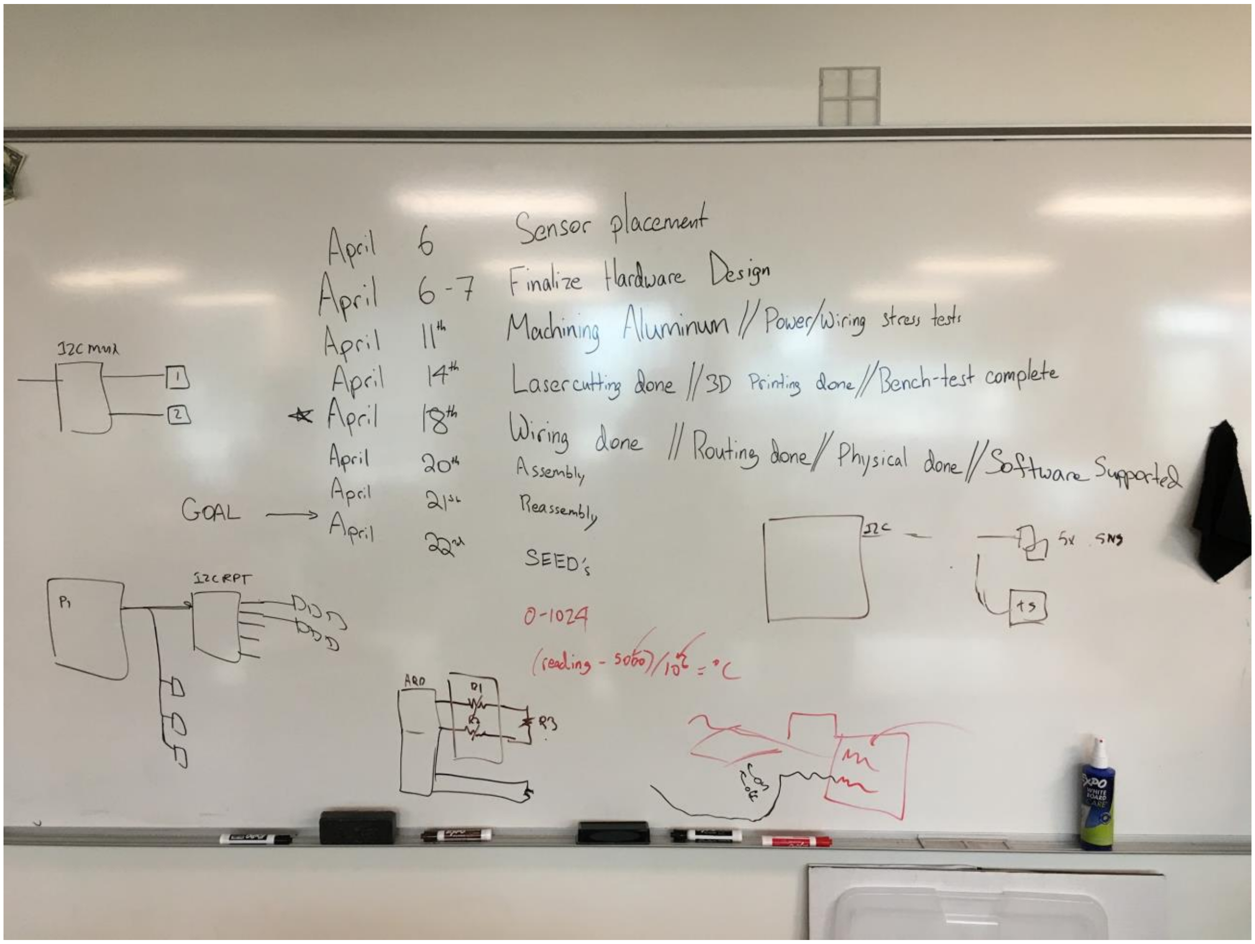
Division	Situation	Team Response	Action/Results
MECH:	Box designed almost done		Will start laser cutting
BIOE:	xxxxxxxxx	xxxxxxxxxxxx	xxxxxxxxxxx
COEN:			
Organization:	xxxxxxxxxxx	xxxxxxxxxxx	xxxxxxxxxxxx
Other:	xxxxxxxxxxxx	xxxxxxxxxxxxxx	xxxxxxxxxxxx

ACTION ITEMS (Recently Accomplished):

Who	What	When
Jake	Electronics schematics	last week
Mechs	Started manufacturing of metal components	yesterday
Coens	All the coding is finished	

NEW BUSINESS:

Member(s) Involved	Discussion	Next Steps
Jeff, Josie	Laser cutting the Box and starting to build the Box	talk to Don
Jake, Franz, Josie	Building the dual extruder CAD design	Finish up designing
Emily, Jake	Making force sensor code to add to the printer firmware.	Adding the code from arduino to the firmware
Cynthia, Max	getting the temp sensors to work on the raspberry pi. DtoA converter.	I2C debugging
Coens	Making GUI look nice and incorporating into Pi	


April 6 Sensor placement
April 6-7 Finalize Hardware Design
April 11th Machining Aluminum // Power/Wiring stress tests
April 14th Lasercutting done // 3D Printing done // Bench-test complete
April 18th Wiring done // Routing done // Physical done // Software Supported
April 20th Assembly
April 21st Reassembly
GOAL April 22nd
SEED's
0-1024
I2C MUX
I2CRPT
Pi
ARD
I2C

3D Bioprinter Meeting Minutes
Senior Design 2017

Meeting Scribe: Jeff

Location: RSL
Meeting Date: 4/3/17

Meeting Time: 7AM

ATTENDANCE:

Team Member	Present	Team Member	Present	Team Member	Present
Kitts	_x__ yes ___ no	Le	__x_ yes ___ no	Warren	_x__ yes ___ no
Abrams	__x_ yes ___ no	Ososke	__x_ yes ___ no	SE3D	___ yes _x__ no
Barone	__x_ yes ___ no	Plum	__x_ yes ___ no	Other ___________	___ yes _x__ no
Bhatnagar	___ yes __x_ no	Takimoto	__x_ yes ___ no	Other ____________	___ yes _x__ no

REPORTS:

Division	Situation	Team Response	Action/Results
MECH:	Team updated PDS and experimental procedures	Working on it	due 4/13
BIOE:	x	x	x
COEN:	x	x	x
Organization:	x	x	x
Other:	x	x	x

ACTION ITEMS (Recently Accomplished):

Who	What	When
Jeff, Josie	Finished machining of the aluminum extrusions	last mon/tues
Jake	Tested buck boost circuit	monday
Jake, Emily	Tested analog temp sensors. Braided the wires. Noticed the sensors were not getting correct voltage in.	Tues
Josie, Jake	Switched to RAMPS board for dual extrusion	Tues
Franz	Dual extruder mount is done and needs to be printed	Wed
Emily	Code has been troubleshot. Found where to put it in the marlin firmware.	Tues
Coens	GIU is coming around. Keyboard code is pretty neat	past week

NEW BUSINESS:

Member(s) Involved	Discussion	Next Steps
Mechs	Wiring, assembly of the final Box	
Franz, Jeff	Print some things	Use 3D printer
Emily	User manual tells how to load the syringe, not coding that fixes air bubbles for you.	Code that stops and throws error when it detects air bubble

3D Bioprinter Meeting Minutes
Senior Design 2017

Meeting Scribe: Jeff

Location: RSL

Meeting Date: 4/20

Meeting Time: 7AM ,bright and early

ATTENDANCE:

Team Member	Present	Team Member	Present	Team Member	Present
Kitts	_x__ yes ___ no	Le	__x_ yes ___ no	Warren	_x__ yes ___ no
Abrams	__x_ yes ___ no	Ososke	__x_ yes ___ no	SE3D	___ yes _x__ no
Barone	__x_ yes ___ no	Plum	__x_ yes ___ no	Other __________	___ yes _x__ no
Bhatnagar	___ yes __x_ no	Takimoto	__x_ yes ___ no	Other ___________	___ yes _x__ no

REPORTS:

Division	Situation	Team Response	Action/Results
MECH:	Commercialization Environmental/ Societal Impact	Will begin soon	
BIOE:	x	x	x
COEN:	x	x	x
Organization:	x	x	x
Other:	x	x	x

ACTION ITEMS (Recently Accomplished):

Who	What	When
Jake, Franz	Wiring done	yesterday
Coens	LEDs are working, wiring looks good, have not looked at the relays	yesterday
Jeff Josie	Laser cutter broke, starting to route instead	yesterday
Coens	Skin for GUI created for aesthetics	yesterday
Jake	Firmware loaded for dual extruder	Monday
Franz	Dual extruder mount implemented on existing bioprinter	Today
Emily	Close to having the firmware updated for force sensor	Today

NEW BUSINESS:

Member(s) Involved	Discussion	Next Steps
Jeff	Begin routing panels, then assembly	router everything
Mech	Temperature sensor refinement	temp sensor testing
All	48 hour incubation test	not priority yet
Coen	refine GUI. Lots of ideas for improving the interface	once we go to get feedback at SEEDs
All	Work on powerpoint slides	Work on slides

Ask maya:
How many pictures needed for 48 hour test: 1 every hour
What preset light colors:cool blue, white, no light, green
Should students be allowed to set/white temp and lights, or want teacher tab: experiment in progress prompt
Ask the fastest rate for picture taking: every 30 seconds

See about adjusting dual extruder tips to the same height

3D Bioprinter Meeting Minutes
Senior Design 2017

Meeting Scribe:

Location:
Meeting Date:

Meeting Time:

ATTENDANCE:

Team Member	Present	Team Member	Present	Team Member	Present
Kitts	_x__ yes ___ no	Le	__x_ yes ___ no	Warren	__x_ yes ___ no
Abrams	___ yes _x__ no	Ososke	__x_ yes ___ no	SE3D	___ yes _x__ no
Barone	__x_ yes ___ no	Plum	__x_ yes ___ no	Other ___________	___ yes _x__ no
Bhatnagar	___ yes __x_ no	Takimoto	__x_ yes ___ no	Other ____________	___ yes _x__ no

REPORTS:

Division	**Situation**	**Team Response**	**Action/Results**
MECH:	Business plan Societal/ environmental impact	Working on it	xxxx
BIOE:	xxxx	xxxx	xxxxx
COEN:	xxxx	xxxx	xxxxx
Organization:	xxxx	xxxx	xxxx
Other:	xxxx	xxxx	xxxxx

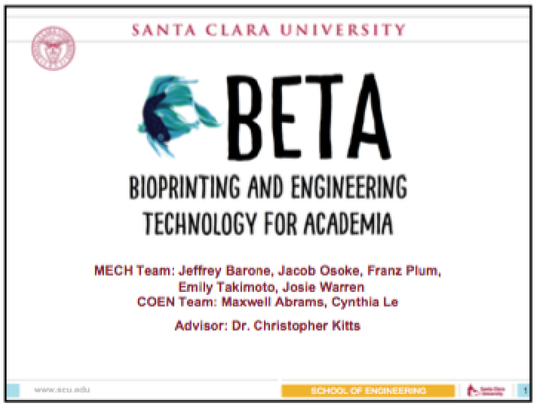
ACTION ITEMS (Recently Accomplished):

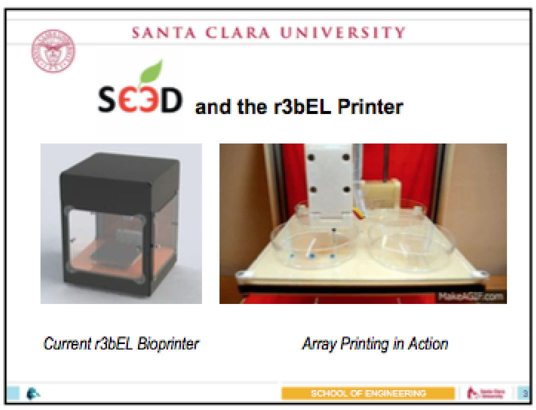
Who	**What**	**When**
Jeff	Box assembly	Tuesday
Jeff, Josie	Tested UI at Seeds and ran a experiment through the Box.	Saturday
Jake, Franz	Have the wiring inserted into the correct positions in the Box.	Wednesday

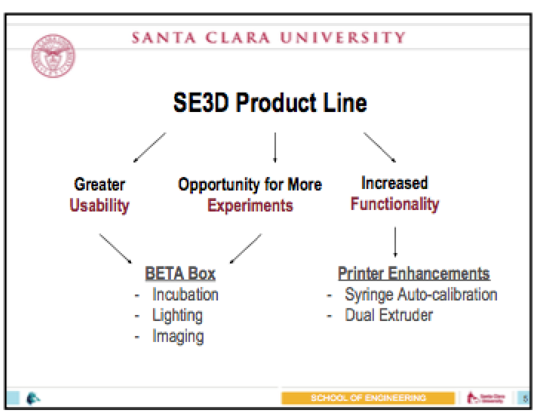
NEW BUSINESS:

Member(s) Involved	**Discussion**	**Next Steps**
COENS	Update the GUI	
Jeff	Make small adjustments to Box asssembly	
All	Have our slides rough draft by monday	

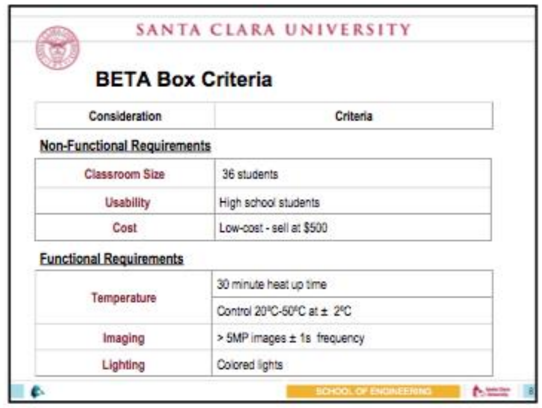
N - Copy of Senior Design Conference Presentation Slides

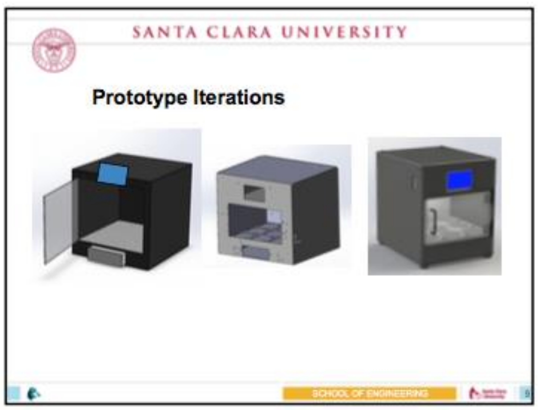


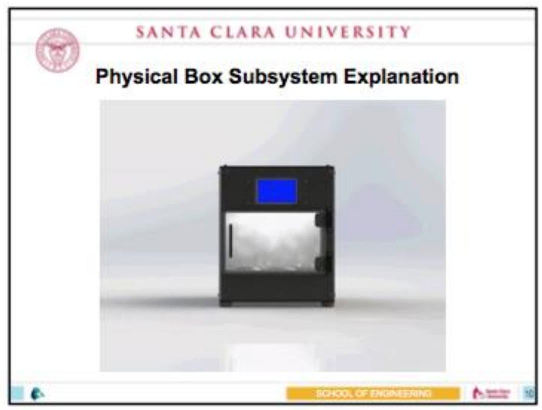


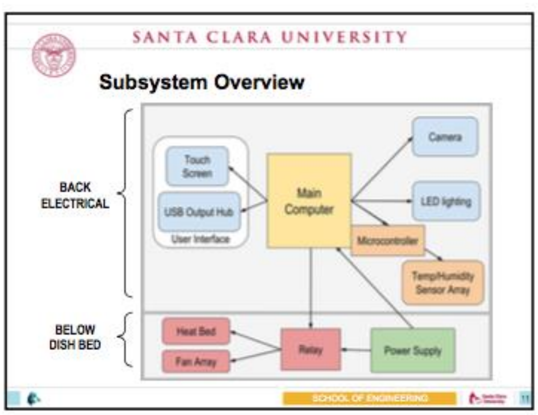



SANTA CLARA UNIVERSITY
WHAT IS BETA BOX?
It's an experimentation box!
Design for SE3D's Production Line
Takes Photos of Color Change
Controls Lighting
Incubates
SCHOOL OF ENGINEERING
7

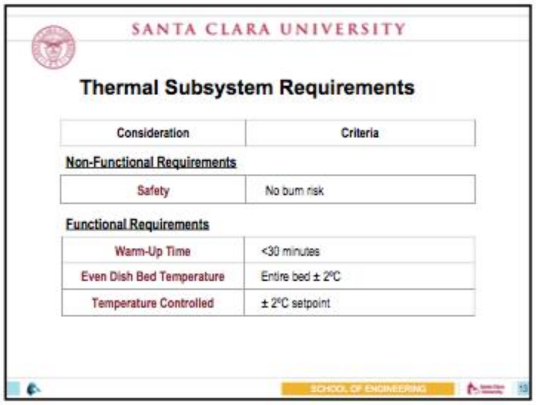

SANTA CLARA UNIVERSITY
BETA Box Criteria
Consideration
Criteria
Non-Functional Requirements
Classroom Size
36 students
Usability
High school students
Cost
Low-cost - sell at $500
Functional Requirements
Temperature
30 minute heat up time
Control 20ºC-50ºC at ± 2ºC
Imaging
> 5MP images ± 1s frequency
Lighting
Colored lights
SCHOOL OF ENGINEERING
8

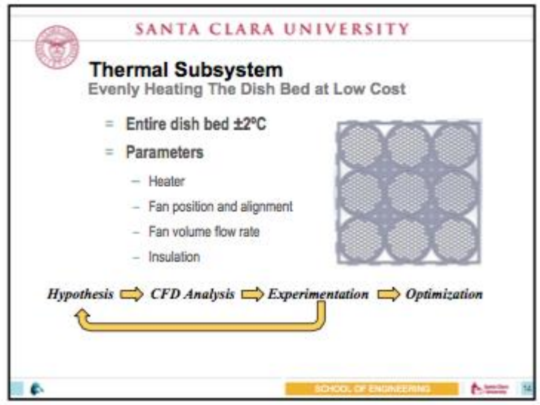

SANTA CLARA UNIVERSITY
Prototype Iterations
SCHOOL OF ENGINEERING
9

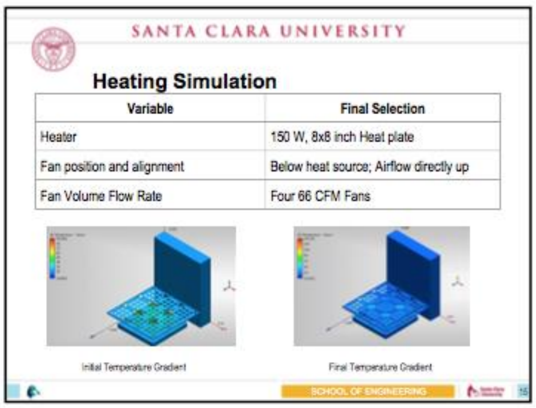

SANTA CLARA UNIVERSITY
Physical Box Subsystem Explanation
SCHOOL OF ENGINEERING
10

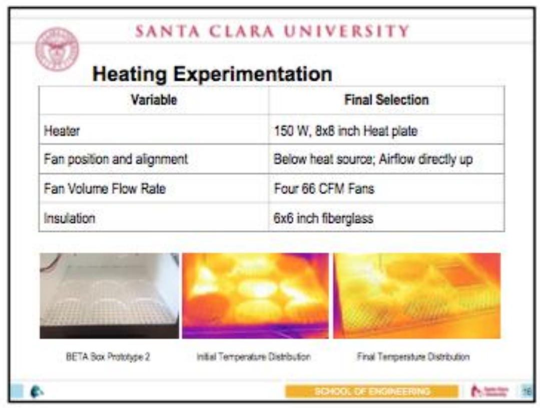

SANTA CLARA UNIVERSITY
Subsystem Overview
BACK ELECTRICAL
BELOW DISH BED
Touch Screen
USB Output Hub
User Interface
Main Computer
Camera
LED lighting
Microcontroller
Temp/Humidity Sensor Array
Heat Bed
Fan Array
Relay
Power Supply
SCHOOL OF ENGINEERING
11

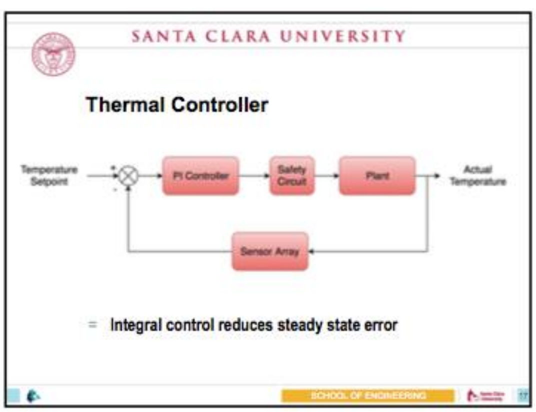

SANTA CLARA UNIVERSITY
Agenda
1. Project Background & Objective
2. BETA Box
a. Thermal Subsystem
b. Computing Subsystem and User Interface
3. Printer Enhancements
4. Key Considerations, Results and Next Steps
5. Questions
SCHOOL OF ENGINEERING
12

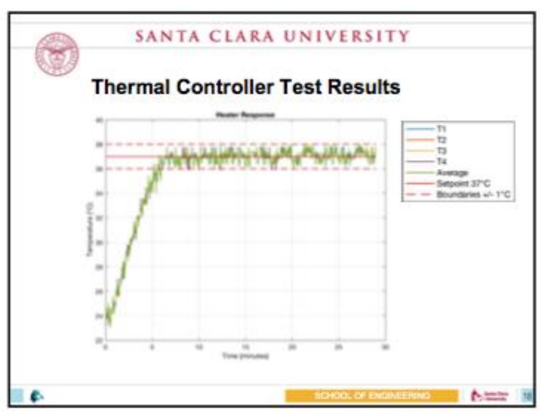

SANTA CLARA UNIVERSITY
Thermal Subsystem Requirements
Consideration
Criteria
Non-Functional Requirements
Safety
No burn risk
Functional Requirements
Warm-Up Time
<30 minutes
Even Dish Bed Temperature
Entire bed ± 2°C
Temperature Controlled
± 2°C setpoint
SCHOOL OF ENGINEERING
13

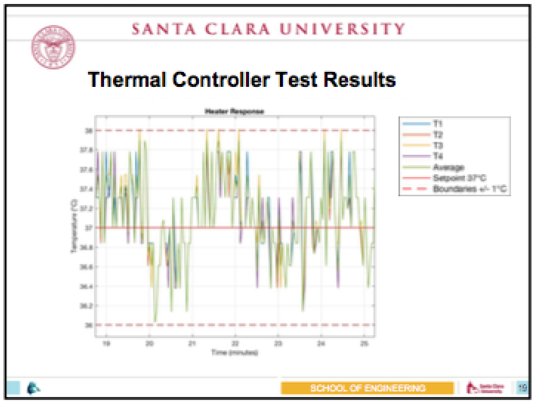

SANTA CLARA UNIVERSITY
Thermal Subsystem
Evenly Heating The Dish Bed at Low Cost
Entire dish bed ±2°C
Parameters
Heater
Fan position and alignment
Fan volume flow rate
Insulation
Hypothesis
CFD Analysis
Experimentation
Optimization
SCHOOL OF ENGINEERING
14

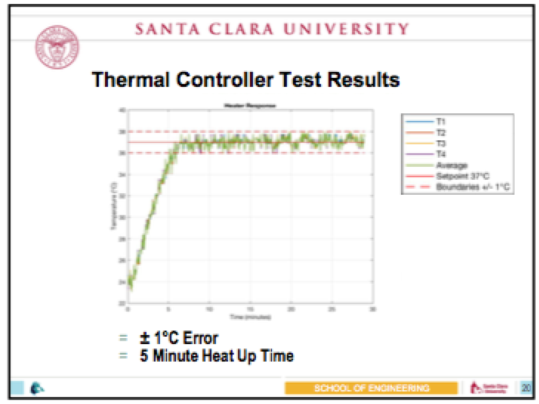

SANTA CLARA UNIVERSITY
Heating Simulation
Variable
Final Selection
Heater
150 W, 8x8 inch Heat plate
Fan position and alignment
Below heat source; Airflow directly up
Fan Volume Flow Rate
Four 66 CFM Fans
Initial Temperature Gradient
Final Temperature Gradient
SCHOOL OF ENGINEERING
15


SANTA CLARA UNIVERSITY
Heating Experimentation
Variable
Final Selection
Heater
150 W, 8x8 inch Heat plate
Fan position and alignment
Below heat source; Airflow directly up
Fan Volume Flow Rate
Four 66 CFM Fans
Insulation
6x6 inch fiberglass
BETA Box Prototype 2
Initial Temperature Distribution
Final Temperature Distribution
SCHOOL OF ENGINEERING
16

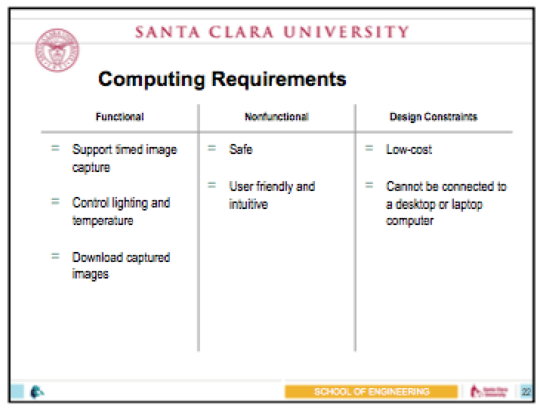

SANTA CLARA UNIVERSITY
Thermal Controller
Temperature Setpoint
PI Controller
Safety Circuit
Plant
Actual Temperature
Sensor Array
Integral control reduces steady state error
SCHOOL OF ENGINEERING
17

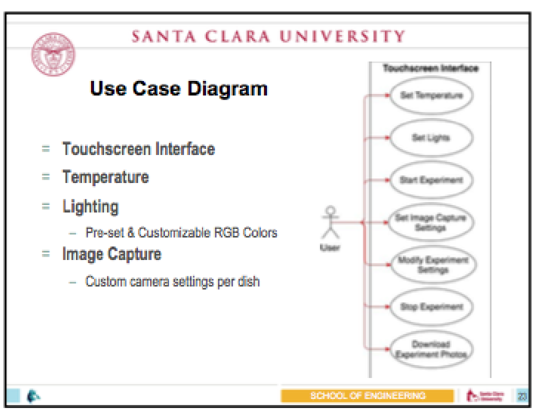

SANTA CLARA UNIVERSITY
Thermal Controller Test Results
Heater Response
T1
T2
T3
T4
Average
Setpoint 37°C
Boundaries +/- 1°C
Time (minutes)
SCHOOL OF ENGINEERING
18

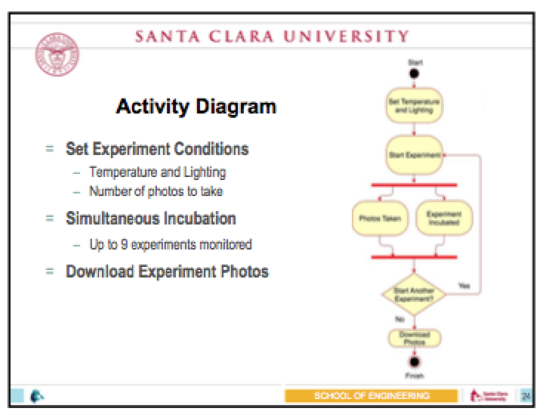

SANTA CLARA UNIVERSITY
Thermal Controller Test Results
Heater Response
T1
T2
T3
T4
Average
Setpoint 37°C
Boundaries +/- 1°C
Temperature (°C)
Time (minutes)
SCHOOL OF ENGINEERING
19

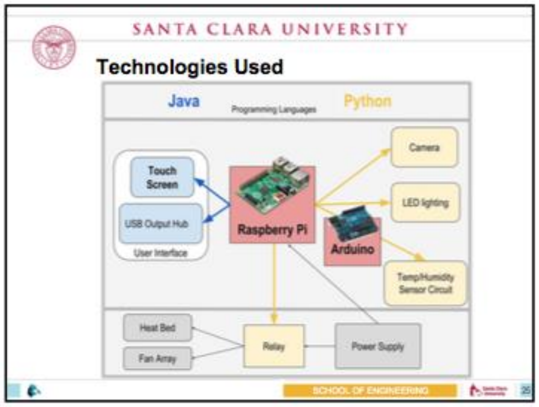

SANTA CLARA UNIVERSITY
Thermal Controller Test Results
Heater Response
T1
T2
T3
T4
Average
Setpoint 37°C
Boundaries +/- 1°C
Temperature (°C)
Time (minutes)
± 1°C Error
5 Minute Heat Up Time
SCHOOL OF ENGINEERING
20

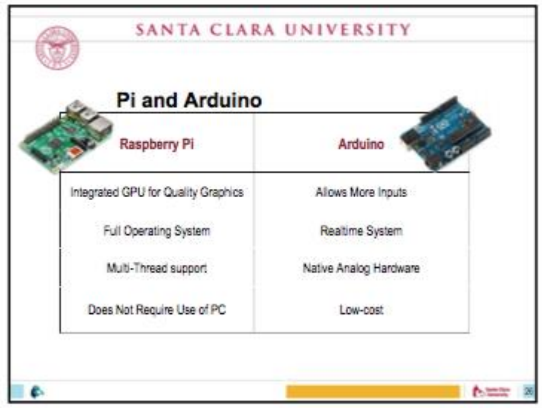

SANTA CLARA UNIVERSITY
Agenda
1. Project Background & Objective
2. BETA Box
a. Thermal Subsystem
b. Computing Subsystem and User Interface
3. Printer Enhancements
4. Key Considerations, Results and Next Steps
5. Questions
SCHOOL OF ENGINEERING
21

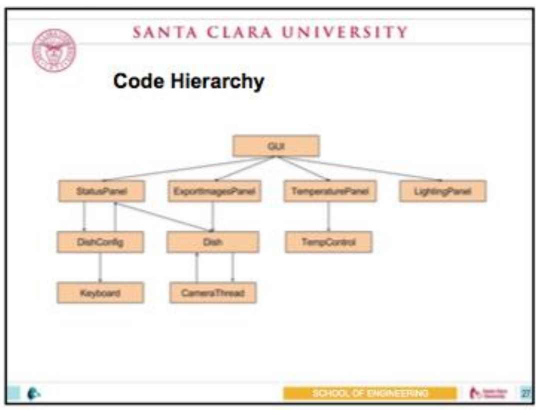

SANTA CLARA UNIVERSITY
Computing Requirements
Functional
Support timed image capture
Control lighting and temperature
Download captured images
Nonfunctional
Safe
User friendly and intuitive
Design Constraints
Low-cost
Cannot be connected to a desktop or laptop computer
SCHOOL OF ENGINEERING
22

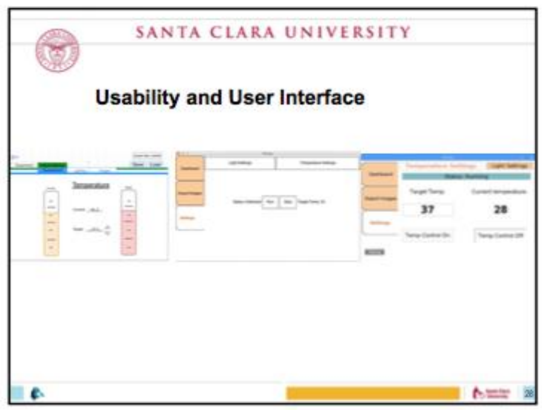

SANTA CLARA UNIVERSITY
Use Case Diagram
Touchscreen Interface
Temperature
Lighting
– Pre-set & Customizable RGB Colors
Image Capture
– Custom camera settings per dish
Touchscreen Interface
Set Temperature
Set Lights
Start Experiment
Set Image Capture Settings
Modify Experiment Settings
Stop Experiment
Download Experiment Photos
User
SCHOOL OF ENGINEERING
23


SANTA CLARA UNIVERSITY
Activity Diagram
Set Experiment Conditions
– Temperature and Lighting
– Number of photos to take
Simultaneous Incubation
– Up to 9 experiments monitored
Download Experiment Photos
Start
Set Temperature and Lighting
Start Experiment
Photos Taken
Experiment Incubated
Start Another Experiment?
Yes
No
Download Photos
Finish
SCHOOL OF ENGINEERING
24

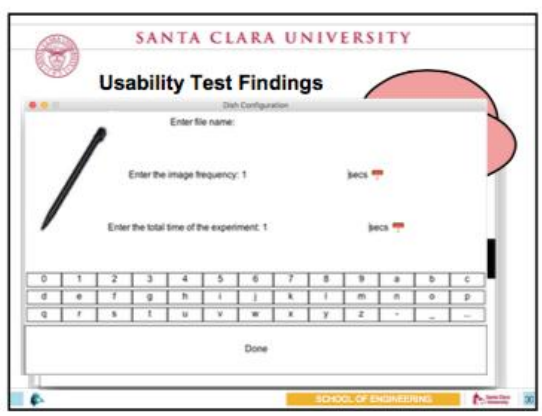

SANTA CLARA UNIVERSITY
Technologies Used
Java
Programming Languages
Python
Touch Screen
USB Output Hub
User Interface
Raspberry Pi
Arduino
Camera
LED lighting
Temp/Humidity Sensor Circuit
Heat Bed
Fan Array
Relay
Power Supply
SCHOOL OF ENGINEERING
25

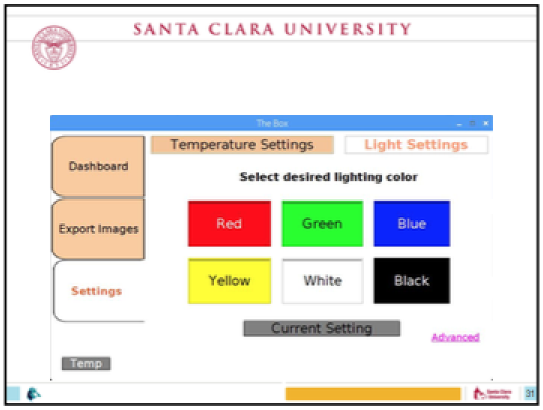

SANTA CLARA UNIVERSITY
Pi and Arduino
Raspberry Pi
Arduino
Integrated GPU for Quality Graphics
Allows More Inputs
Full Operating System
Realtime System
Multi-Thread support
Native Analog Hardware
Does Not Require Use of PC
Low-cost
26

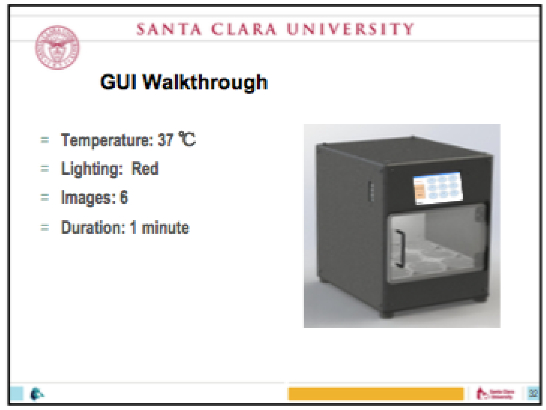

SANTA CLARA UNIVERSITY
Code Hierarchy
GUI
StatusPanel
ExportImagesPanel
TemperaturePanel
LightingPanel
DishConfig
Dish
TempControl
Keyboard
CameraThread
SCHOOL OF ENGINEERING
27

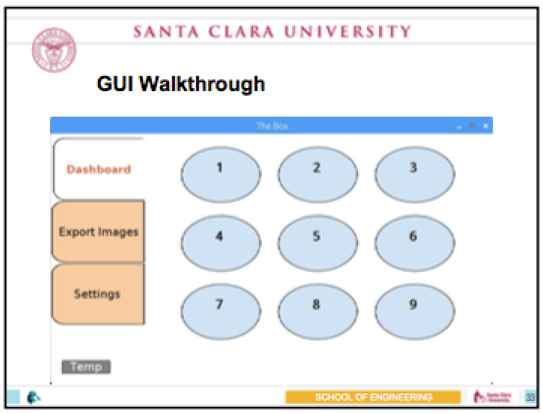

SANTA CLARA UNIVERSITY
Usability and User Interface
37
28
28

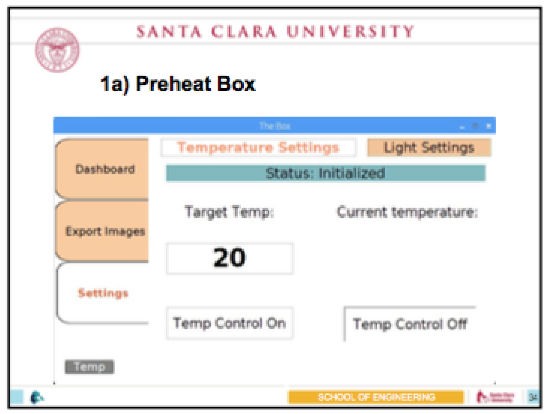

SANTA CLARA UNIVERSITY
Usability Test
WHO
High school sophomores & juniors
HOW
Simulated Experiment Setup
WHERE
SCU campus
Average Score on first GUI iteration
4/5
SCU High School Outreach, scu.edu
SCHOOL OF ENGINEERING

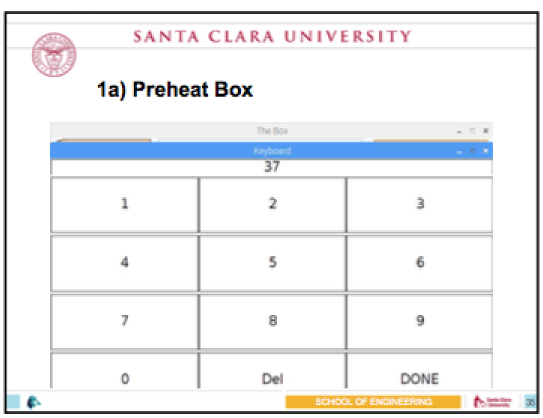

SANTA CLARA UNIVERSITY
Usability Test Findings
Dish Configuration
Enter file name:
Enter the image frequency: 1
secs
Enter the total time of the experiment: 1
secs
Done
SCHOOL OF ENGINEERING
30

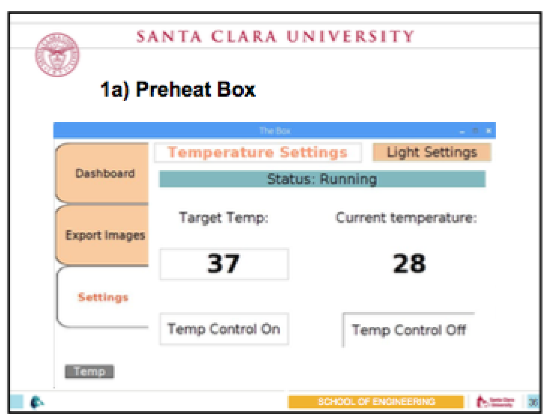

SANTA CLARA UNIVERSITY
The Box
Dashboard
Export Images
Settings
Temp
Temperature Settings
Light Settings
Select desired lighting color
Red
Green
Blue
Yellow
White
Black
Current Setting
Advanced
31

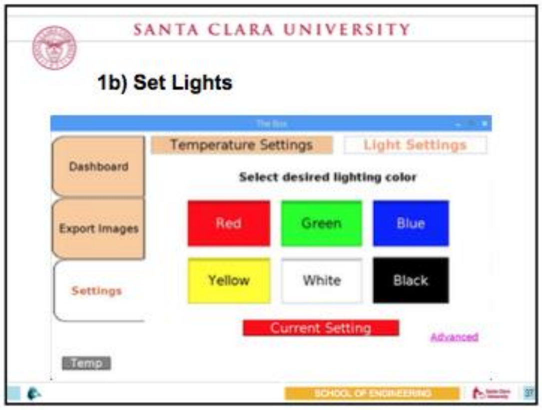

SANTA CLARA UNIVERSITY
GUI Walkthrough
Temperature: 37 ℃
Lighting: Red
Images: 6
Duration: 1 minute
32


SANTA CLARA UNIVERSITY
GUI Walkthrough
The Box
Dashboard
Export Images
Settings
Temp
1
2
3
4
5
6
7
8
9
SCHOOL OF ENGINEERING
33


SANTA CLARA UNIVERSITY
1a) Preheat Box
The Box
Dashboard
Export Images
Settings
Temp
Temperature Settings
Light Settings
Status: Initialized
Target Temp:
Current temperature:
20
Temp Control On
Temp Control Off
SCHOOL OF ENGINEERING
34


SANTA CLARA UNIVERSITY
1a) Preheat Box
The Box
Keyboard
37
1
2
3
4
5
6
7
8
9
0
Del
DONE
SCHOOL OF ENGINEERING
35

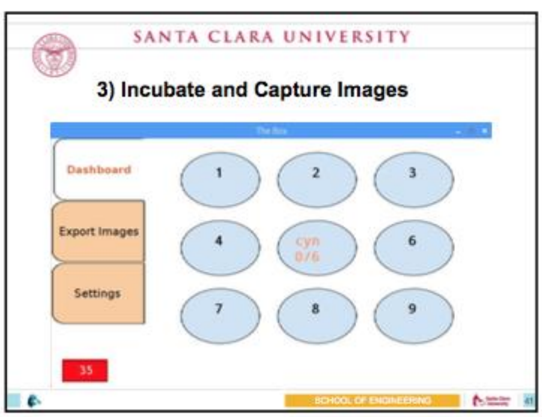

SANTA CLARA UNIVERSITY
1a) Preheat Box
The Box
Dashboard
Export Images
Settings
Temp
Temperature Settings
Light Settings
Status: Running
Target Temp:
Current temperature:
37
28
Temp Control On
Temp Control Off
SCHOOL OF ENGINEERING
36

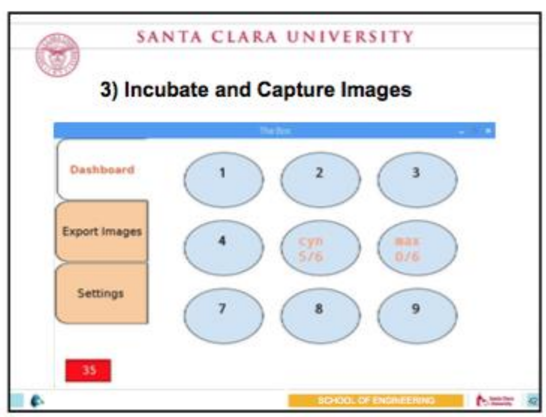

SANTA CLARA UNIVERSITY
1b) Set Lights
Temperature Settings
Light Settings
Dashboard
Select desired lighting color
Export Images
Red
Green
Blue
Settings
Yellow
White
Black
Current Setting
Advanced
Temp
SCHOOL OF ENGINEERING
37

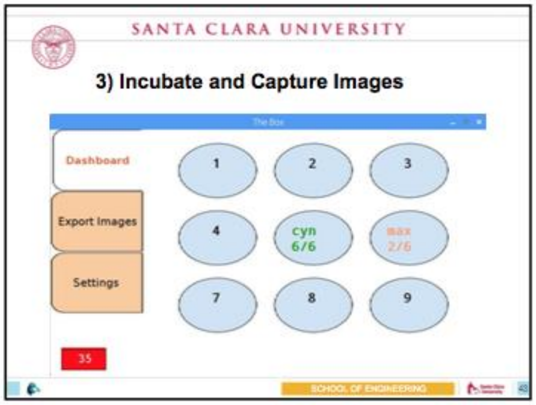

SANTA CLARA UNIVERSITY
1b) Set Lights
Current Setting
Red:
255
255
Green:
0
0
Blue:
232
232
Done
SCHOOL OF ENGINEERING
38

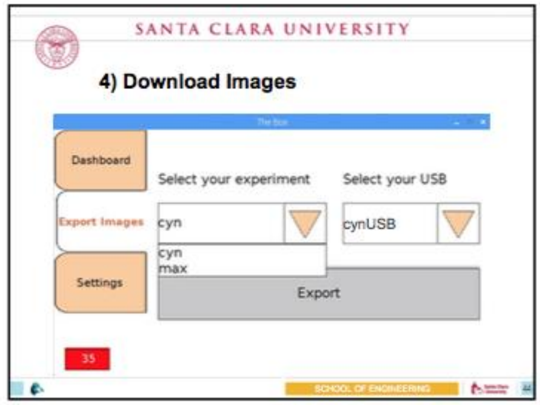

SANTA CLARA UNIVERSITY
2) Begin your Experiment
Dashboard
1
2
3
Export Images
4
5
6
Settings
7
8
9
28
SCHOOL OF ENGINEERING
39


SANTA CLARA UNIVERSITY
2a) Set Image Capture
Enter file name: cyn
Enter the image frequency: 10 secs
Enter the total time of the experiment: 1 mins
0 1 2 3 4 5 6 7 8 9 a b c
d e f g h i j k l m n o p
q r s t u v w x y z - _ <-
Done
SCHOOL OF ENGINEERING
40

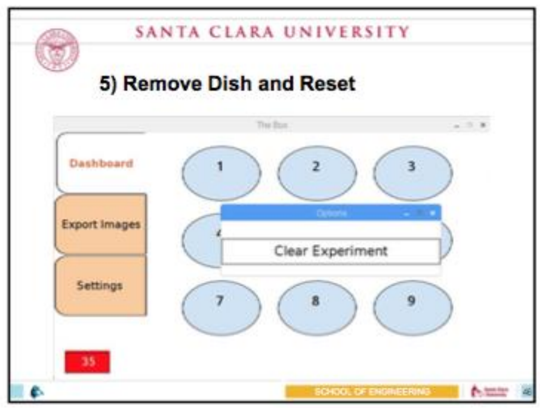

SANTA CLARA UNIVERSITY
3) Incubate and Capture Images
Dashboard
1
2
3
Export Images
4
cyn 0/6
6
Settings
7
8
9
35
SCHOOL OF ENGINEERING
41


SANTA CLARA UNIVERSITY
3) Incubate and Capture Images
Dashboard
1
2
3
Export Images
4
cyn 5/6
max 0/6
Settings
7
8
9
35
SCHOOL OF ENGINEERING
42

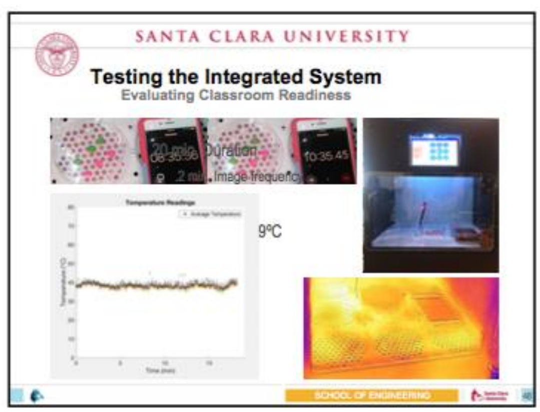

SANTA CLARA UNIVERSITY
3) Incubate and Capture Images
Dashboard
Export Images
Settings
1
2
3
4
cyn 6/6
max 2/6
7
8
9
35
SCHOOL OF ENGINEERING


SANTA CLARA UNIVERSITY
4) Download Images
Dashboard
Export Images
Settings
Select your experiment
Select your USB
cyn
cynUSB
cyn
max
Export
35
SCHOOL OF ENGINEERING


SANTA CLARA UNIVERSITY
5) Remove Dish and Reset
Dashboard
Export Images
Settings
1
2
3
4
cyn 6/6
max 6/6
7
8
9
35
SCHOOL OF ENGINEERING

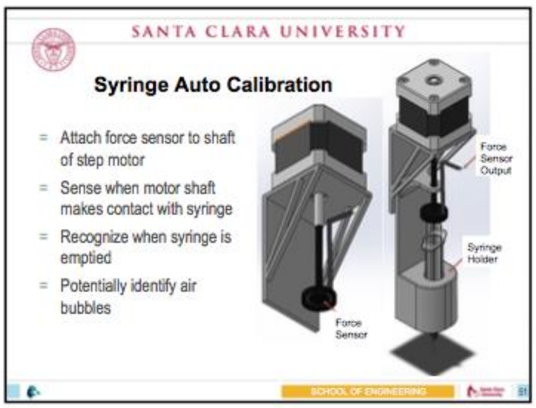

SANTA CLARA UNIVERSITY
5) Remove Dish and Reset
Dashboard
Export Images
Settings
1
2
3
Clear Experiment
7
8
9
35
SCHOOL OF ENGINEERING

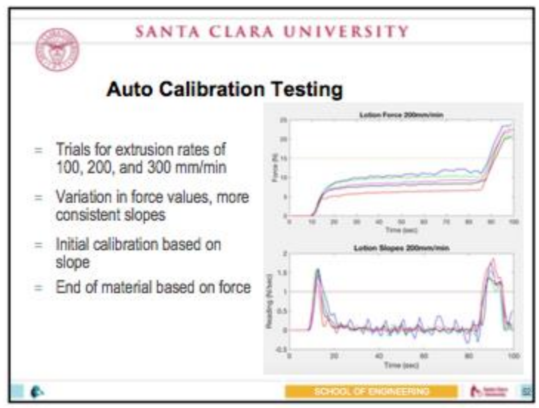

SANTA CLARA UNIVERSITY
5) Remove Dish and Reset
Dashboard
Export Images
Settings
1
2
3
4
5
max 6/6
7
8
9
35
SCHOOL OF ENGINEERING

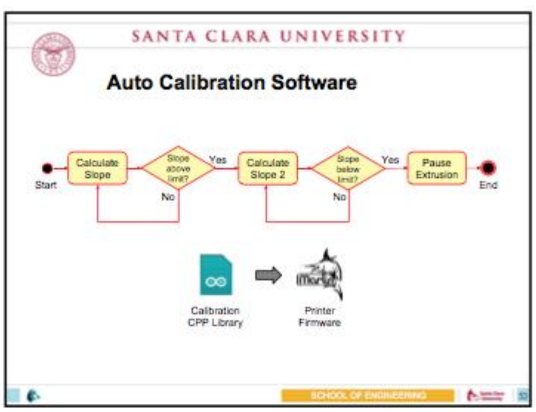

SANTA CLARA UNIVERSITY
Testing the Integrated System
Evaluating Classroom Readiness
Duration
Image frequency
9ºC
SCHOOL OF ENGINEERING

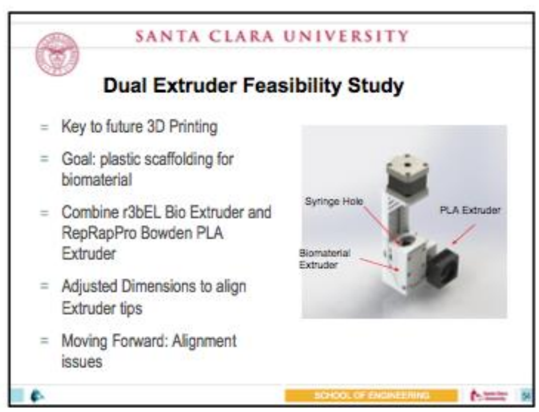

SANTA CLARA UNIVERSITY
Agenda
1. Project Background & Objective
2. BETA Box
3. Printer Enhancements
4. Key Considerations, Results and Next Steps
5. Questions
SCHOOL OF ENGINEERING
49


SANTA CLARA UNIVERSITY
Printer Redesign Goals
Goal
Status
Calibrate position of motor shaft
Auto-calibration of stepper motor to syringe verified and coded
Print multiple materials
Dual extruder platform for biomaterial and PLA plastic integrated into printer
r3bEL Prototype Used for Study (SE3D Education)
SCHOOL OF ENGINEERING
50


SANTA CLARA UNIVERSITY
Syringe Auto Calibration
Attach force sensor to shaft of step motor
Sense when motor shaft makes contact with syringe
Recognize when syringe is emptied
Potentially identify air bubbles
Force Sensor Output
Syringe Holder
Force Sensor
SCHOOL OF ENGINEERING
51


SANTA CLARA UNIVERSITY
Auto Calibration Testing
Trials for extrusion rates of 100, 200, and 300 mm/min
Variation in force values, more consistent slopes
Initial calibration based on slope
End of material based on force
Lotion Force 200mm/min
Time (sec)
Lotion Slopes 200mm/min
Time (sec)
SCHOOL OF ENGINEERING
52


SANTA CLARA UNIVERSITY
Auto Calibration Software
Start
Calculate Slope
Slope above limit?
Yes
No
Calculate Slope 2
Slope below limit?
Yes
No
Pause Extrusion
End
Calibration CPP Library
Printer Firmware
SCHOOL OF ENGINEERING
53


SANTA CLARA UNIVERSITY
Dual Extruder Feasibility Study
Key to future 3D Printing
Goal: plastic scaffolding for biomaterial
Combine r3bEL Bio Extruder and RepRapPro Bowden PLA Extruder
Adjusted Dimensions to align Extruder tips
Moving Forward: Alignment issues
Syringe Hole
PLA Extruder
Biomaterial Extruder
SCHOOL OF ENGINEERING
54

SANTA CLARA UNIVERSITY

Dual Extruder Path Moving Forward

SCHOOL OF ENGINEERING 55

SANTA CLARA UNIVERSITY

Agenda

1. Project Background & Objective
2. BETA Box
3. Printer Enhancements
4. Key Considerations, Results and Next Steps
5. Questions

SCHOOL OF ENGINEERING 56

SANTA CLARA UNIVERSITY

Ethical Implications

Considerations in product development for students

Area of Impact	Relation to BETA	Response/Justification
Economic: Social Equality	Cost affects accessibility	- Low-cost materials - Simplified manufacturability - Standard biology budget = $5000
Effect on Education	Learning must be bettered, not diminished	Learning — Simplification Learning — Optimization

57

SANTA CLARA UNIVERSITY

Results

	Required Spec	Solution Met
BETA Box Cost	Sell at $500	✓
BETA Box Capacity	36 students	✓
BETA Box Temperature	20°C-50°C at ± 2°C → 5 min warm-up	✓
BETA Box Image Capture	> 5MP images ± 1s frequency → ± 0.1s	✓
BETA Box Lighting	Colored lights	✓
BETA Box Usability	High school students	✓
Printer Auto-Calibration	Syringe Z-axis homing	✓
Printer Dual Extruder	Additional PLA extruder	✓

58

SANTA CLARA UNIVERSITY

Next Steps

BETA Box	User Interface	Printer Enhancements
Safety Test	Admin. Only	Test Firmware
Scaling	Image Splicing	Integrate Force Sensor
PCB	File Management	Standardize Parts
	UI Feedback	Permanent Extruders

59

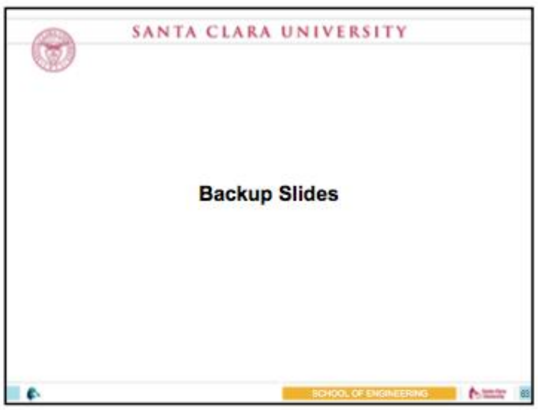
SANTA CLARA UNIVERSITY

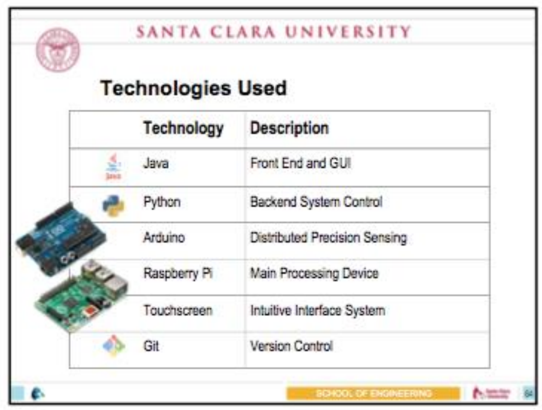
Conclusion: Education for Tomorrow

Partnership with SE3D
- BETA Box added to product line
- Printer enhancements shared
- Products go into the classroom

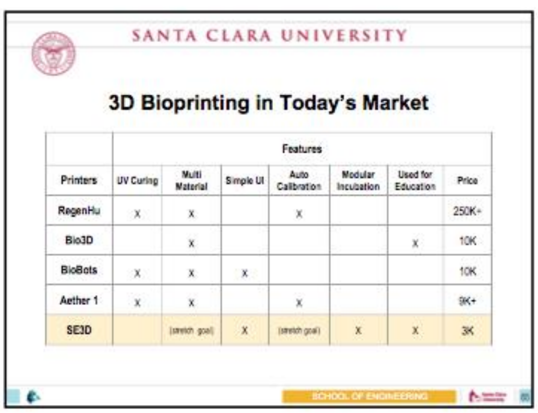
Next generation of STEM students

60

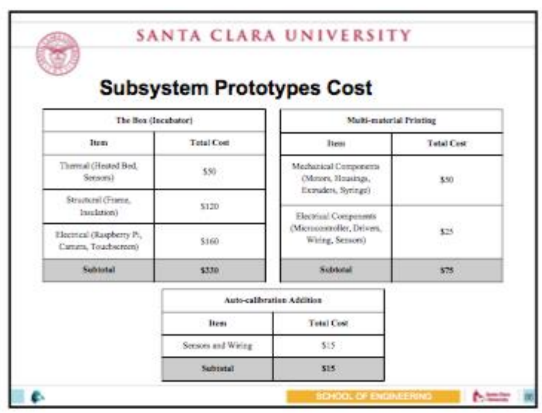




Technology	Description
Java	Front End and GUI
Python	Backend System Control
Arduino	Distributed Precision Sensing
Raspberry Pi	Main Processing Device
Touchscreen	Intuitive Interface System
Git	Version Control



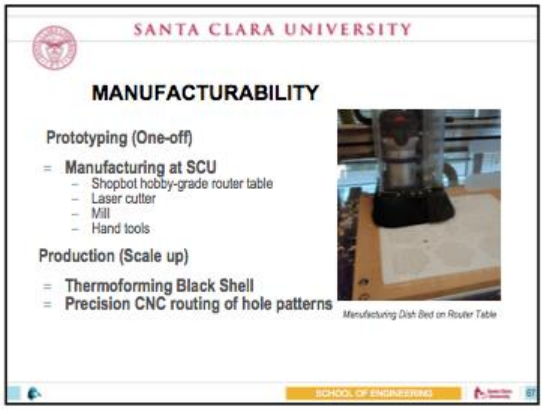
	Features						
Printers	UV Curing	Multi Material	Simple UI	Auto Calibration	Modular Incubation	Used for Education	Price
RegenHu	X	X		X			250K+
Bio3D		X				X	10K
BioBots	X	X	X				10K
Aether 1	X	X		X			9K+
SE3D		(stretch goal)	X	(stretch goal)	X	X	3K

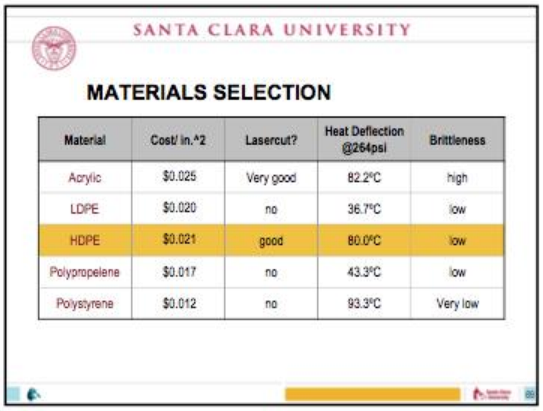


The Box (Incubator)	
Item	Total Cost
Thermal (Heated Bed, Sensors)	$50
Structural (Frame, Insulation)	$120
Electrical (Raspberry Pi, Camera, Touchscreen)	$160
Subtotal	$330

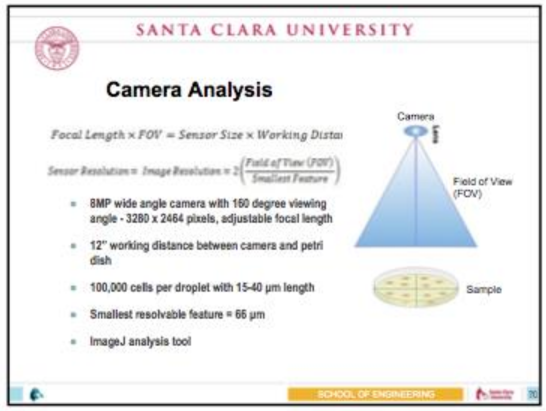
Multi-material Printing	
Item	Total Cost
Mechanical Components (Motors, Housings, Extruders, Syringe)	$50
Electrical Components (Microcontroller, Drivers, Wiring, Sensors)	$25
Subtotal	$75

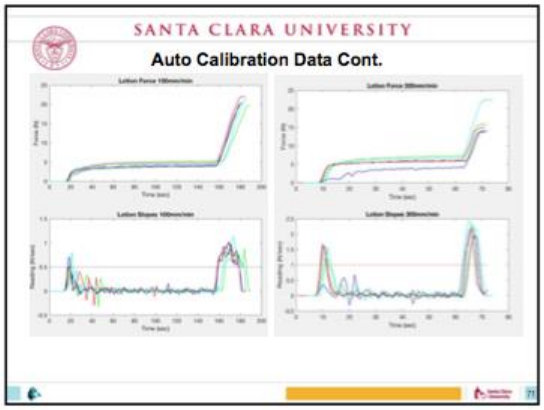
Auto-calibration Addition	
Item	Total Cost
Sensors and Wiring	$15
Subtotal	$15

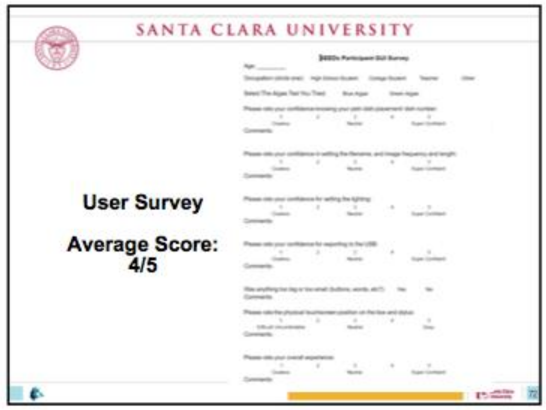




Material	Cost/ in.^2	Lasercut?	Heat Deflection @264psi	Brittleness
Acrylic	$0.025	Very good	82.2ºC	high
LDPE	$0.020	no	36.7ºC	low
HDPE	$0.021	good	80.0ºC	low
Polypropelene	$0.017	no	43.3ºC	low
Polystyrene	$0.012	no	93.3ºC	Very low







O – Usability Testing Survey and Feedback

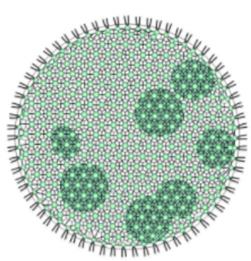
Green Algae Photosynthesis Experiment

Introduction:
Algae are a very diverse group of predominantly aquatic photosynthetic organisms that account for almost 50% of the photosynthesis that takes place on Earth. Algae have a wide range of antenna pigments to harvest light energy for photosynthesis giving different types of algae their characteristic colour.[1] The algae used in this experiment will help to understand the rate at which photosynthesis occurs in green algae under optimal conditions.

Materials:
-algae stock -growth media -hydrogel -petri dish

Incubation Procedure:

1. Place your petri dish in an open dish bed position; limit the time that the incubation door is open.
2. Using the touchscreen, ensure that your experiment is running according to the following conditions:
   a. Enter the file name as your initials.
   b. The unit number of images is 1 seconds per image. The total time of the experiment is 15 seconds.
   c. Lighting is the pre-set blue.
   d. Target temperature is 20ºC.
3. Wait for the touchscreen to signal that your experiment is complete.
4. Export your images to the connected USB and remove your petri dish from the incubator.
   a. Use the USB titled 'null'
5. Finish your experiment by clicking your completed dish.
   a. Exit pop-up box by clicking 'ok'



[1]http://www.els.net/WileyCDA/ElsArticle/refId-a0000322.html

Figure O-1: Instructions for Usability Activity

SEEDs Participant GUI Survey

Age: _________

Occupation (circle one): High School Student College Student Teacher Other

Select The Algae Test You Tried: Blue Algae Green Algae

Please rate your confidence knowing your petri dish placement/ dish number:

1	2	3	4	5
Clueless		Neutral		Super Confident

Comments:

Please rate your confidence in setting the filename, and image frequency and length:

1	2	3	4	5
Clueless		Neutral		Super Confident

Comments:

Please rate your confidence for setting the lighting:

1	2	3	4	5
Clueless		Neutral		Super Confident

Comments:

Please rate your confidence for exporting to the USB:

1	2	3	4	5
Clueless		Neutral		Super Confident

Comments:

Was anything too big or too small (buttons, words, etc?) Yes No

Comments:

Please rate the physical touchscreen position on the box and stylus:

1	2	3	4	5
Difficult/ Uncomfortable		Neutral		Easy

Comments:

Please rate your overall experience:

1	2	3	4	5
Clueless		Neutral		Super Confident

Comments:

Figure O-2: GUI Usability Survey

Table O-1: Scores from Usability Survey

Age	confidence in dish placement	confidence in setting name, image frequency	confidence in lighting	confidence in export	anything too big/small	touchscreen position	overall experience
16	4	4	2	-	no	5	4
17	4	4	5	3	yes	3	4
	5	3	5	2	no	3	4
17	5	5	4	5	no	5	5
17	4	4	4	4	no	3	4
Avg	4.4	4	4	3.5		3.8	4.2
	Average of all scores						
	3.983333333						

P – Code

GUI.java:

```
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.ItemEvent;
import java.awt.event.ItemListener;
import java.io.IOException;
import java.text.ParseException;
import java.util.ArrayList;

import javax.swing.BorderFactory;
import javax.swing.Icon;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JTabbedPane;
import javax.swing.JTextField;
import javax.swing.UIManager;
import javax.swing.UnsupportedLookAndFeelException;
import javax.swing.WindowConstants;
import javax.swing.border.Border;
import javax.swing.border.EtchedBorder;
import javax.swing.event.ChangeEvent;
import javax.swing.event.ChangeListener;
import javax.swing.plaf.synth.SynthLookAndFeel;

public class GUI {
	// Screen size
	private final static int SCREEN_WIDTH = 480;
	private final static int SCREEN_HEIGHT = 800;
	private static Color orangeSorbet= new Color(244,195,149);
	private static Color darkOrange= new Color(255,151,108);
	private static ArrayList<String> dishList = new ArrayList<String>();
	private static int FAN_PIN = 26;
	private static final String PATH_TO_RELAY_ON = "/home/pi/py/relay_on.py"; // Path

	// Main runnable class to create GUI
	public static void main(String[] args) throws ParseException,
			UnsupportedLookAndFeelException {

		SynthLookAndFeel laf = new SynthLookAndFeel();
		laf.load(GUI.class.getResourceAsStream("laf.xml"), GUI.class);
		UIManager.setLookAndFeel(laf);
```

```
JFrame frame = new JFrame("The Box"); // Make a frame tabbed pane
JTabbedPane mainPane = new JTabbedPane(JTabbedPane.LEFT,
                    JTabbedPane.WRAP_TAB_LAYOUT);

// Panel for Settings Tab with light and temperature subtabs
JPanel settings = new JPanel();
settings.setLayout(new GridBagLayout());
GridBagConstraints c = new GridBagConstraints();

c.insets = new Insets(0, 10, 0, 10);

c.gridy = 0;
c.weightx = .5;

c.fill = GridBagConstraints.HORIZONTAL;
final JButton lightingButton = new JButton("Light Settings");
final JButton tempButton = new JButton("Temperature Settings");
lightingButton.setFocusable(false);
tempButton.setFocusable(false);
lightingButton.setBackground(orangeSorbet);
tempButton.setBackground(Color.WHITE);
tempButton.setForeground(darkOrange);
tempButton.setFont(new Font("Ariel", Font.BOLD, 26));

Border etched = BorderFactory.createEtchedBorder(EtchedBorder.RAISED);
lightingButton.setBorder(etched);
tempButton.setBorder(etched);

c.gridx = 0;
c.anchor = GridBagConstraints.EAST;
settings.add(tempButton, c);
c.gridx = 1;
c.anchor = GridBagConstraints.WEST;
settings.add(lightingButton, c);

final JPanel setterPane = new JPanel();
final CardLayout cl = new CardLayout();
setterPane.setLayout(cl);
final LightingPanel lightPanel = new LightingPanel();
final TemperaturePanel tempPanel = new TemperaturePanel();

setterPane.add(tempPanel.getPanel(), "Temps");
setterPane.add(lightPanel.getPanel(), "Lights");

c.anchor = GridBagConstraints.CENTER;
c.gridx = 0;
```

```
c.gridy = 1;
c.gridwidth = 2;
c.weighty = 1.0;
c.weightx = 1.0;
c.insets = new Insets(0, 20, 0, 20);
settings.add(setterPane, c);

//show lighting panel when click lighting button
lightingButton.addActionListener(new ActionListener() {
	@Override
	public void actionPerformed(ActionEvent arg0) {
		// TODO Auto-generated method stub
		System.out.println("hello");
		cl.last(setterPane);
		tempButton.setBackground(orangeSorbet);
		tempButton.setForeground(Color.BLACK);
		tempButton.setFont(new Font("Ariel", Font.PLAIN, 24));


		lightingButton.setBackground(Color.WHITE);
		lightingButton.setForeground(darkOrange);
		lightingButton.setFont(new Font("Ariel", Font.BOLD, 24));
		// lightingButton.setBackground(Color.ORANGE);
		// tempButton.setBackground(Color.WHITE);
	}
});

//show temp panel when click temp button
tempButton.addActionListener(new ActionListener() {
	@Override
	public void actionPerformed(ActionEvent arg0) {
		// TODO Auto-generated method stub
		System.out.println("bye");
		cl.first(setterPane);
		tempButton.setBackground(Color.WHITE);
		lightingButton.setBackground(orangeSorbet);

		lightingButton.setForeground(Color.BLACK);
		lightingButton.setFont(new Font("Ariel", Font.PLAIN, 24));


		tempButton.setBackground(Color.WHITE);
		tempButton.setForeground(darkOrange);
		tempButton.setFont(new Font("Ariel", Font.BOLD, 24));
		// lightingButton.setBackground(Color.WHITE);
		// tempButton.setBackground(Color.ORANGE);
```

```
			}
		});

		/*
		* Can use this to load an Icon ImageIcon icon =
		* createImageIcon("images/icon.gif");
		*/

		StatusPanel statusPanel = new StatusPanel(dishList);

		mainPane.addTab(statusPanel.getPanelName(), statusPanel.getPanel());

		/*
		* CameraPanel cameraPanel = new CameraPanel();
		* mainPane.addTab(cameraPanel.getPanelName(), cameraPanel.getPanel());
		* mainPane.setTabComponentAt(1, changeTab(cameraPanel.getPanelName()));
		* // tab index, jLabel
		*/

		// LightingPanel lightPanel = new LightingPanel();
		// settingPane.addTab(lightPanel.getPanelName(), lightPanel.getPanel());

		// TemperaturePanel temperaturePanel = new TemperaturePanel();
		//
settingPane.addTab(temperaturePanel.getPanelName(),temperaaturePanel.getPanel());

		final ExportImagesPanel eip = new ExportImagesPanel(dishList);
		mainPane.addTab(eip.getPanelName(), eip.getPanel());

		mainPane.addTab("Settings \n\n\n\n\n", settings);
		// settingPane.getComponentAt(0).setName("statusTab");
		// System.out.println("tab"+settingPane.getTabCount());
		// settingPane.getTabComponentAt(0);

		final JLabel tempPV = new JLabel(tempPanel.getTempOutput());
		final JButton lightPV = new JButton("Temp");
		Font buttonFont = new Font("Ariel", Font.PLAIN, 20);
		lightPV.setFont(buttonFont);
		lightPV.setForeground(Color.WHITE);
		lightPV.setPreferredSize(new Dimension(30, 50));
		lightPV.setMaximumSize(new Dimension(30, 50));

		lightPV.setBackground(lightPanel.getLightColor());
		// lightPV.setEnabled(false);
		lightPV.setFocusable(false);
```

```
ChangeListener changeListener = new ChangeListener() {
	public void stateChanged(ChangeEvent changeEvent) {
		// every time tab changed, check for new file and usb names
		JTabbedPane sourceTabbedPane = (JTabbedPane) changeEvent
				.getSource();
		int index = sourceTabbedPane.getSelectedIndex();
		System.out.println("Tab changed to: "
				+ sourceTabbedPane.getTitleAt(index));

		// check if any values have updated so can use as dashboard
		lightPV.setBackground(lightPanel.getLightColor());
		lightPanel.setButtonForeground(lightPV);

		if (tempPanel.getTempOutput() != "") {
			if (tempPanel.getTempOutput() != "Error") {
				lightPV.setText(tempPanel.getTempOutput() + "
C");

			}
			lightPV.setText(tempPanel.getTempOutput());
		}
		// System.out.println(lightPanel.getLightColor().getRGB());
		// tempPV.setText(tempPanel.getTempOutput());
		eip.refreshFileNames();
		eip.refreshUsbNames();
	}
};

mainPane.addChangeListener(changeListener);
// mainPane.add(new JPanel());
frame.setLayout(new GridBagLayout());
GridBagConstraints r = new GridBagConstraints();
r.anchor = GridBagConstraints.CENTER;
r.gridx = 0;
r.gridy = 0;
r.gridwidth = 1;
r.weighty = 1.0;
r.weightx = 1.0;
r.fill = GridBagConstraints.HORIZONTAL;
// c.insets=new Insets(0,20,0,20);

frame.add(mainPane, r); // Add the tabbed pane to the larger frame

r.gridy = 1;
r.gridx = 0;
r.ipadx = 30;
r.insets = new Insets(0, 20, 10, 0);
```

```
                r.fill = GridBagConstraints.NONE;

                r.anchor = GridBagConstraints.WEST;
                frame.add(lightPV, r);

frame.setDefaultCloseOperation(WindowConstants.DO_NOTHING_ON_CLOSE);
                frame.setSize(new Dimension(SCREEN_HEIGHT, SCREEN_WIDTH)); // Set to

                                // screen

                                // resolution
                frame.setResizable(false);
                frame.setVisible(true); // Set the frame to be visible

                // Start fans running
                try {
                        Process fOn = Runtime.getRuntime().exec(
                                        "sudo python " + PATH_TO_RELAY_ON + " " +
FAN_PIN);
                } catch (IOException e) {
                        System.out.println("Error could not turn on");
                }

        }

}
```

CameraThread.java

```
import java.util.Date;

public class CameraThread implements Runnable {
  private static final String CAMERA_SCRIPT = "/home/pi/py/camera.py";
  Dish[] dishes;
  static final String SECS = "secs";
  static final String MINS = "mins";
  static final String HOURS = "hours";
  static final String DAYS = "days";
  static final int POLL_RATE = 1000; //loop every 1000 ms. This will depend on how fast
camera is at taking photos.

  public CameraThread(Dish[] dishes){
     this.dishes = dishes;
  }

  @Override
  public void run() {
```

```
    Date currTime;
    boolean hasTakenPhoto;
    while(true){ //Loop forever in background

      currTime = new Date(); //For now, just take time at start of loop. Will modify after testing
if needed
      hasTakenPhoto = false; //Reuse photo if taken in the same loop
      for(Dish expr : dishes){
        if(expr.isEnabled()){ //Check if dish is "enabled" for taking photos
          double timeVal = Double.MAX_VALUE; //Fake value to override

          switch(expr.getCaptureMetric()){
            case SECS:
              timeVal = expr.getCaptureRate() * 1000;//1000 to convert to seconds, rate for
num of seconds
              break;
            case MINS:
              timeVal = expr.getCaptureRate() * 60 * 1000;
              break;
            case HOURS:
              timeVal = expr.getCaptureRate() * 60 * 60 * 1000;
              break;
            case DAYS:
              timeVal = expr.getCaptureRate() * 24 * 60 * 60 * 1000;
              break;
          }

          if(currTime.getTime() - expr.timeOfLastPic().getTime() >= timeVal){
            //Time to save another photo. Check to see if we can reuse one from a previous
dish
            if(hasTakenPhoto){
              //Reuse photo
              reusePhoto(expr);
            }else{
              //First dish to take photo this loop
              hasTakenPhoto = true;
              takePhoto(expr);
            }

          }
          //Stop taking pictures for the dish once the experiment is over
          if(expr.getPicsTaken()== expr.getTotalImagesNeeded()){
            expr.setEnabled(false);
            expr.setFinished(true);
          }
        }//end of check for if dish is enabled
```

```
        }//end of for loop for all dishes

        try {
            //Sleep for the poll rate
            Thread.sleep(POLL_RATE);
        } catch (InterruptedException e) {
            System.out.println("Error! Photo thread could not sleep");
        }

    }//end of while loop (runs for duration of program)
  }

  private void takePhoto(Dish dish){
      Date currT = new Date();
      System.out.println("sudo python " +CAMERA_SCRIPT + "
/home/pi/Desktop/"+dish.getFileName()+"/"+dish.getFileName()+dish.getPicsTaken()+".jpg");
//Simple message for now

      try{

          Process takePic = Runtime.getRuntime().exec("sudo python " +CAMERA_SCRIPT + "
/home/pi/Desktop/"+dish.getFileName()+"/"+dish.getFileName()+dish.getPicsTaken()+".jpg");
      }catch(Exception e){
          System.out.println("Error could not turn take a picture");
      }
      dish.setTimeOfLastPic(currT);
      dish.incrPicsTaken();
  }

  private void reusePhoto(Dish dish){
      //This will just "copy" the photo previously taken to a file with this dish's name
      System.out.println("Photo resued! " + dish.getFileName());
      dish.setTimeOfLastPic(new Date());
      dish.incrPicsTaken();
  }

}
```

CameraPanel.java:

```
import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import java.io.IOException;

import javax.swing.JButton;
```

```
import javax.swing.JPanel;

public class CameraPanel {
	private JPanel cameraPanel; // Panel to make modifications to
	private String cameraName; // Name for panel
	private JButton captureButton;
	private final String CAMERA_BUTTON_TEXT = "Capture Image";
	private final String CAMERA_SCRIPT_PATH = "/home/pi/py/camera.py";

	public CameraPanel() {
		// Initialize components
		cameraPanel = new JPanel();
		cameraPanel.setLayout(new GridLayout(1, 2)); // Add a layout manager to

	// align buttons as we

	// resize
		cameraName = "Camera Test"; // Assign name
		// Setup and add buttons
		captureButton = new JButton(CAMERA_BUTTON_TEXT);
		captureButton.addActionListener(new ActionListener() {

			@Override
			public void actionPerformed(ActionEvent e) {
				// TODO Auto-generated method stub
				try {
					Process pOn = Runtime.getRuntime().exec(
							"sudo python " +
CAMERA_SCRIPT_PATH);
				} catch (IOException e1) {
					e1.printStackTrace();
				}

			} // Add action listener to respond to button

		});

		cameraPanel.add(captureButton);
	}

	public JPanel getPanel() {
		return this.cameraPanel;
	}

	public String getPanelName() {
		return cameraName;
```

```
	}
}
```

TemperaturePanel.java:

```
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.FocusEvent;
import java.awt.event.FocusListener;
import java.awt.event.KeyEvent;

import javax.swing.BorderFactory;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JTextField;
import javax.swing.border.EtchedBorder;

public class TemperaturePanel {
	private final static String STOP_BUTTON_TEXT = "Temp Control Off";
	private final static String START_BUTTON_TEXT = "Temp Control On";
	private final static String TEMP_PANEL_NAME = "Temperature";
	private JButton startExperiment;
	private JButton stopExperiment;
	private JLabel statusLabel = new JLabel("Status: Initialized");
	private TempControl controlThread;
	private JTextField targetField;
	private JTextField thresholdField;
	private int target = 20;
	private int threshold = 2;
	private JPanel mainPanel;
	private JLabel output;
	private JLabel tmpLabel = new JLabel("Target Temp:");
	private boolean targetFocus;

	public TemperaturePanel(){

		mainPanel = new JPanel();
		mainPanel.setLayout(new GridBagLayout()); //Add a layout manager to align
buttons as we resize
		output = new JLabel();
		//Start button
		startExperiment = new JButton(START_BUTTON_TEXT);
		startExperiment.setName("ExpButton");
		startExperiment.addActionListener(new ActionListener(){
```

```
			@Override
			public void actionPerformed(ActionEvent e) { //Add Action listener to
respond to button
				updateStatus("Running");
				stop(); //Make sure no existing thread
				controlThread = new TempControl(statusLabel, output);

	controlThread.updateThreshold(Double.parseDouble(thresholdField.getText()));

	controlThread.updateTarget(Double.parseDouble(targetField.getText()));
				new Thread(controlThread).start();
				//startExperiment.setEnabled(false);

	//startExperiment.setBorder( BorderFactory.createLoweredBevelBorder());

	//stopExperiment.setBorder( BorderFactory.createRaisedBevelBorder());
				//stopExperiment.setEnabled(true);
			}
		});

		//Stop button
		stopExperiment = new JButton(STOP_BUTTON_TEXT);
		stopExperiment.setName("ExpButton");
		//stopExperiment.setEnabled(false);
		//stopExperiment.setBorder( BorderFactory.createLoweredBevelBorder());

		stopExperiment.addActionListener(new ActionListener(){
			@Override
			public void actionPerformed(ActionEvent e) { //Add Action listener to
respond to button
				//updateStatus("Stopped");
				stop();
			}
		});

		//Temp Input
		targetFocus = false;
		targetField = new JTextField(3);
		targetField.setText(""+target);
		targetField.addFocusListener(new FocusListener(){

			@Override
			public void focusGained(FocusEvent e) {
				if(targetFocus == false){ //TODO: improve click options
					targetFocus = true;
```

```
                    new Keyboard("0123456789", targetField, 4, 3,
mainPanel);
                }else{
                    targetFocus = false;

                    //Send enter key for update
                    /*
                    try {
                      Robot robot = new Robot();
                      robot.keyPress(KeyEvent.VK_ENTER);
                    } catch (AWTException ee) {
                            ee.printStackTrace();
                    } */
                }
            }

            @Override
            public void focusLost(FocusEvent e) {


            }});

        targetField.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                if(controlThread!=null){

    controlThread.updateTarget(Double.parseDouble(targetField.getText())); //If enter key
is required, can make this a function
                }
             }
            });

        //Threshold input (deleted once testing is done)
        thresholdField = new JTextField(3);
        thresholdField.setText(""+threshold);
        thresholdField.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                if(controlThread!=null){

    controlThread.updateThreshold(Double.parseDouble(thresholdField.getText()));
                }
            }
            });

        GridBagConstraints c = new GridBagConstraints();
        c.weightx=1;
```

```
        c.fill= GridBagConstraints.BOTH;
        c.gridx = 0;
        c.gridy = 1;
        c.ipady=80;
        c.insets=new Insets(0,100,0,0);
        c.anchor= GridBagConstraints.CENTER;
        tmpLabel.setHorizontalAlignment(JLabel.CENTER);
        mainPanel.add(tmpLabel,c);

        c.ipady=0;
        c.ipadx=100;
        c.gridy = 2;
        c.gridx = 0;
        c.anchor= GridBagConstraints.CENTER;
        targetField.setFont(new Font("Ariel", Font.BOLD, 45 ));
        targetField.setHorizontalAlignment(JTextField.CENTER);

targetField.setBorder(BorderFactory.createEtchedBorder(EtchedBorder.RAISED));
        mainPanel.add(targetField,c);

        JLabel blank= new JLabel("   ");
        c.ipady=30;
        c.anchor=GridBagConstraints.NORTH;
        c.gridx=1;
        c.gridy=3;
        c.ipadx=40;
        mainPanel.add(blank,c);

        c.fill= GridBagConstraints.HORIZONTAL;

        c.gridy = 4;
        c.gridx = 0;
        c.ipadx=12;
        c.ipady=25;

startExperiment.setBorder(BorderFactory.createEtchedBorder(EtchedBorder.RAISED));
        mainPanel.add(startExperiment,c);

        c.ipadx=500;
        c.gridy = 0;
        c.gridx = 0;
        c.ipady=0;
        c.gridwidth=4;
        statusLabel.setBackground(new Color(127,183,190));
        statusLabel.setFont(new Font("Ariel", Font.PLAIN, 25));
        statusLabel.setHorizontalAlignment(JLabel.CENTER);
```

```
		c.anchor= GridBagConstraints.NORTH;
		mainPanel.add(statusLabel,c);

		c.anchor=GridBagConstraints.CENTER;
		c.gridwidth=1;
		c.fill=GridBagConstraints.NONE;
		c.gridy = 4;
		c.gridx = 2;
		c.ipadx=30;
		c.ipady=30;

	stopExperiment.setBorder(BorderFactory.createEtchedBorder(EtchedBorder.RAISED));
		mainPanel.add(stopExperiment,c);

		c.ipadx=0;
		c.ipady=0;
		c.gridy = 1;
		c.gridx = 2;
		//mainPanel.add(thresholdField); DO NOT USE THRESHOLD FIELD AFTER
BETA TESTING. Set to specified value
		JLabel outLabel= new JLabel("Current temperature: ");
		c.anchor= GridBagConstraints.CENTER;
		mainPanel.add(outLabel,c);

		c.gridy = 2;
		c.gridx = 2;
		c.anchor= GridBagConstraints.CENTER;
		output.setFont(new Font("Ariel", Font.BOLD, 45 ));
		output.setText("" );
		mainPanel.add(output,c);
		stopExperiment.setBorder( BorderFactory.createLoweredBevelBorder());

	}

	public String getTempOutput(){
		return output.getText();
	}

	public JPanel getPanel(){
		return mainPanel;
	}

	public String getPanelName(){
		return TEMP_PANEL_NAME;
	}
```

```
	private void updateStatus(String newStatus){
		statusLabel.setText("Status: " + newStatus);
	}

	private void stop(){
		if(controlThread != null){
			controlThread.shutdown();
			controlThread = null;
		}
	}

	public void paint(Graphics g){
		//Draw a Box to show color
		g.drawRect(100, 100, 40, 40);
		g.setColor(Color.BLACK);
	}

}
```

CloseDialog.java:

```
import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.FlowLayout;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.GridBagConstraints;
import java.awt.GridBagLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import javax.swing.BoxLayout;
import javax.swing.JButton;
import javax.swing.JDialog;
import javax.swing.JPanel;

public class CloseDialog extends JDialog implements ActionListener {

	JButton button;
	JPanel panel;

	public CloseDialog(String dialogTitle, JPanel p, JButton closeButton, int width, int
height) {
		this.setTitle(dialogTitle);
		JPanel buttonPane = new JPanel();
		buttonPane.setLayout(new GridBagLayout());
```

```
        GridBagConstraints c = new GridBagConstraints();
        GridBagConstraints b = new GridBagConstraints();

        // buttonPane.setPreferredSize(new Dimension(800,480));
        buttonPane.setPreferredSize(new Dimension(width, height));
        this.setResizable(false);
        button = closeButton;
        panel = p;

        button.addActionListener(new ActionListener() {

                @Override
                public void actionPerformed(ActionEvent arg0) {
                        // TODO Auto-generated method stub
                        dispose();

                }

        });
        button.setPreferredSize(new Dimension(200, 50));

        c.gridx = 0;
        c.gridy = 0;
        c.fill= GridBagConstraints.BOTH;
        buttonPane.add(panel, c);

        b.gridx = 0;
        b.gridy = 1;
        buttonPane.add(button, b);

        add(buttonPane);
        pack();
        setVisible(true);
}

@Override
public void actionPerformed(ActionEvent e) {
        dispose();

};

// public void paint(Graphics g) {
// Graphics2D ga = (Graphics2D)g;
// ga.setPaint(Color.red);
// ga.drawOval(150,150,100,100);
//
```

```
	// }
}
```

Dish.java:

```
import java.util.Date;

public class Dish {
	private int captureRate;
	private String captureMetric;
	private String fileName;
	private Date timeStart;
	private Date lastPictureTaken;
	private int picsTaken;
	private boolean isEnabled;
	private int experimentTime;
	private String experimentMetric;
	private int totalImagesNeeded;
	public boolean isFinished;
	private int dishNum;
	private boolean isCleared;

	private String dishString;
	// default picsPerMinute=1
	public Dish() {
		captureRate = 1;
		fileName = "";
		picsTaken = 0;
		captureMetric= "mins";
		isEnabled = false;
		isFinished=false;
		setCleared(false);
	}
	public Dish(String string) {
		dishString=string;
		captureRate = 1;
		fileName = "";
		picsTaken = 0;
		captureMetric= "mins";
		isEnabled = false;
		isFinished=false;
		setCleared(false);
	}

	public void reset(){
		captureRate = 1;
		fileName = "";
```

```
		picsTaken = 0;
		captureMetric= "mins";
		isEnabled = false;
		isFinished=false;
	}
	public Date timeOfLastPic(){
		return lastPictureTaken;
	}

	public void setTimeOfLastPic(Date newDate){
		this.lastPictureTaken = newDate;
	}
	public int getDishNum(){
		return dishNum;
	}
	public void setDishNum(int num){
		dishNum=num;
	}
	public String getDishString(){
		return dishString;
	}
	// every time take a picture, increment picsTaken count by 1
	public void takePic() {
		picsTaken += 1;
	}

	public boolean isEnabled(){
		return isEnabled;
	}

	public void setEnabled(boolean enabled){
		isEnabled = enabled;
	}

	public boolean isFinished(){
		return isFinished;
	}
	public void setFinished(boolean finished){
		isFinished=finished;
	}
	public int getCaptureRate() {
		return captureRate;
	}

	public void setCaptureRate(int captureRate) {
		this.captureRate = captureRate;
```

```
}

public String getFileName() {
	return fileName;
}

public void setFileName(String fileName) {
	this.fileName = fileName;
}

public String getCaptureMetric() {
	return captureMetric;
}

public void setCaptureMetric(String captureMetric) {
	this.captureMetric= captureMetric;
}

public Date getTimeStart() {
	return timeStart;
}

public void setTimeStart(java.util.Date timeStamp) {
	this.timeStart = timeStamp;
}

public int getPicsTaken() {
	return picsTaken;
}

public void incrPicsTaken(){
	this.picsTaken++;
}

public void setPicsTaken(int picsTaken) {
	this.picsTaken = picsTaken;
}
public int getExperimentTime() {
	return experimentTime;
}
public void setExperimentTime(int experimentTime) {
	this.experimentTime = experimentTime;
}
public String getExperimentMetric() {
	return experimentMetric;
}
```

```
	public void setExperimentMetric(String experimentMetric) {
		this.experimentMetric = experimentMetric;
	}
	public int getTotalImagesNeeded() {
		return totalImagesNeeded;
	}
	public void setTotalImagesNeeded(int totalImagesNeeded) {
		this.totalImagesNeeded = totalImagesNeeded;
	}
	public boolean isCleared() {
		return isCleared;
	}
	public void setCleared(boolean isCleared) {
		this.isCleared = isCleared;
	}

}
```

DishConfig.java:

```
import java.awt.List;
import java.io.*;
import java.nio.charset.Charset;
import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.Paths;
import java.sql.Timestamp;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.Calendar;
import java.io.*;
import java.util.Date;
import java.text.*;
import javax.swing.JLabel;


public class TempControl implements Runnable {
   private volatile boolean running = true;
   private double threshold = .1;
   private double target = 20;
   private boolean isRelayOn = false;
   private static int RELAY_PIN = 26;
 //  private static int TEMP_PIN = 20;
 //  private GPIO_Pin relay;
  // private GpioController gpio;
  // private GpioPinDigitalOutput pin;
   private static final String PATH_TO_TEMP = "/home/pi/py/temp_bridge.py";
   private static final String PATH_TO_RELAY_OFF = "/home/pi/py/relay_off.py";
```

```
    private static final String PATH_TO_RELAY_ON = "/home/pi/py/relay_on.py";
    JLabel output;
    JLabel status;
    PrintWriter outFile;
    DateFormat dateFormat;

    public TempControl(JLabel status, JLabel label){
        this.status = status;
        output = label;
        // Write temps to a file
        dateFormat = new SimpleDateFormat("MM/dd/yyyy HH:mm:ss");
        try{
        outFile = new PrintWriter("temps.txt");
    }catch(Exception io){
        System.out.println("Temp file error");
    }

        //relay = new GPIO_Pin(RELAY_PIN);
       /* gpio = GpioFactory.getInstance();
        pin = gpio.provisionDigitalOutputPin(RaspiPin.GPIO_18, "Relay", PinState.LOW);
        pin.setShutdownOptions(true, PinState.LOW);
        */
    }

    @Override
    public void run() {

        while (running) {
            double currTemp = getTemp();
            //Timestamp currentTimestamp = new
java.sql.Timestamp(Calendar.getInstance().getTime().getTime());
            System.out.println("curr Temp: "+ currTemp);

            if(currTemp==Integer.MIN_VALUE){
                System.out.println("error temp sensor output!");
                output.setText("Error");
                status.setText("Status: Error!");
                heaterOff();
                running = false; //THIS WILL KILL THE THREAD
            }else{
                status.setText("Status: Running");
                if(currTemp< target - threshold){
                     heaterOn();
                }else{ //if(currTemp >= target)
                    heaterOff();
                }
```

```
            try {
               Thread.sleep(1000);
            } catch (InterruptedException e) {
               System.out.println("Could not sleep!");
            }
            output.setText("" + Math.round((currTemp*100.0)/100.0));
         }
      }
      heaterOff();
      outFile.close();
      System.out.println("done");
      status.setText("Stopped");
     }

   public void shutdown() {
      running = false;
   }


   private double getTemp(){
      try{
      Process p = Runtime.getRuntime().exec("sudo python " + PATH_TO_TEMP );//+ " 2302 "
+ TEMP_PIN);
      BufferedReader stdInput = new BufferedReader(new
InputStreamReader(p.getInputStream()));

      String consoleOutput = stdInput.readLine();
      System.out.println(consoleOutput);
      if(consoleOutput == null){
         System.out.println("Invalid input from temp sensor");
         return Integer.MIN_VALUE;
      }

      String tempVals[] = consoleOutput.split(",");
      double tempAvg = 0;
      try{
      if(tempVals.length == 7){
         if(Double.parseDouble(tempVals[5]) > 100){ //Saftey check at 100c
            //Force off the thread
            System.out.println("FAILURE! Temperature is too hot!! Shutting down.");
            heaterOff();
            running = false;
            return Integer.MIN_VALUE;
         }
         String tmps = "";
```

```
        for(int i = 1; i < 5; i++){
          tmps += tempVals[i] + ",";
          tempAvg += Double.parseDouble(tempVals[i]);
        }
        tempAvg = tempAvg / 4.0;
        try{
          outFile.println(dateFormat.format(new Date())+","+tmps);
        }catch(Exception ioE){
          System.out.println("Could not write to file");
        }
        return tempAvg;
      }else{
        System.out.println("Error could not read temp from bridge");
        return Integer.MIN_VALUE;
      }
    }catch(Exception ex){
      return getTemp();
    }

    }catch(IOException e){
      System.out.println("Error could not read temp");
      return Integer.MIN_VALUE;
    }
    }

    private void heaterOn(){
      if(!isRelayOn){
        //Turn relay on
        System.out.println("Heater is on");
        //relay.setHIGH();
        //pin.high();
      try{
        Process pOn = Runtime.getRuntime().exec("sudo python " + PATH_TO_RELAY_ON +
" " + RELAY_PIN);
      }catch(IOException e){
           System.out.println("Error could not turn on");
      }
        isRelayOn = true;
      }
    }

    private void heaterOff(){
      if(isRelayOn){
        //turn Relay off
        System.out.println("Heater is off");
        //relay.setLOW();
```

```
        //pin.low();
        try{
           Process pOff = Runtime.getRuntime().exec("sudo python " +
PATH_TO_RELAY_OFF + " " + RELAY_PIN);
         }catch(IOException e){
            System.out.println("Error could not turn off");
         }
        isRelayOn = false;
      }

   }

   public void updateTarget(double newTarget){
      target = newTarget;
      System.out.println("new target:" + target);
   }

   public void updateThreshold(double newThreshold){
      threshold = newThreshold;
      System.out.println("new threshold:" + threshold);
   }

}
```

StatusPanel.java:

```
import java.awt.Color;
import java.awt.Dimension;
import java.awt.Font;
import java.awt.Frame;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.GridBagConstraints;
import java.awt.GridBagLayout;
import java.awt.Insets;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.File;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.Date;
import java.util.Timer;
import java.util.TimerTask;

import javax.swing.BorderFactory;
import javax.swing.JButton;
```

```
import javax.swing.JCheckBox;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.SwingConstants;

public class StatusPanel implements ActionListener {
	private static Color darkOrange= new Color(255,151,108);

	private JPanel statusPanel; // Panel to make modifications to
	private String panelName; // Name for panel
	private Font checkboxFont = new Font("Arial", Font.BOLD, 24);
	private String timeStampString = new SimpleDateFormat("yyyy.MM.dd.HH.mm.ss")
			.format(new Date());
	private Date timeStamp;

	private JCheckBox dish_0, dish_1, dish_2, dish_3, dish_4, dish_5, dish_6,
			dish_7, dish_8;
	private Dish exp0, exp1, exp2, exp3, exp4, exp5, exp6, exp7, exp8;

	private Frame frame = new Frame();
	private static String[] times = { "secs", "mins", "hours", "days" };

	private String captureMetric = "mins";
	private CameraThread cameraThread; // Save as global for now in case we need
	// to interact with it later

	static final String SECS = "secs";
	static final String MINS = "mins";
	static final String HOURS = "hours";
	static final String DAYS = "days";

	static final String PIC_PARENT_DIR = "/home/pi/Desktop/";

	private ArrayList<String> dishNames;
	String PRESPACE = "      ";
	Dish[] dishes;
	Dish[] namedExp;

	JCheckBox[] dishCheckBoxes;

	// store of all experiments that have been created by users
	ArrayList<Dish> namedExperiments;

	public StatusPanel(ArrayList<String> dishList) {

		namedExperiments = new ArrayList<Dish>();
```

```
dishNames = dishList;

// Initialize components
statusPanel = new JPanel();
// Initialize Grid bag layout
statusPanel.setLayout(new GridBagLayout());
GridBagConstraints c = new GridBagConstraints();
c.fill = GridBagConstraints.HORIZONTAL;

panelName = "Dashboard"; // Assign name

// Setup checkboxes
dish_0 = new JCheckBox("<html><pre>     1  <br/> </pre></html>");
dish_1 = new JCheckBox("<html><pre>     2  <br/> </pre></html>");
dish_2 = new JCheckBox("<html><pre>     3  <br/> </pre></html>");
dish_3 = new JCheckBox("<html><pre>     4  <br/> </pre></html>");
dish_4 = new JCheckBox("<html><pre>     5  <br/> </pre></html>");
dish_5 = new JCheckBox("<html><pre>     6  <br/> </pre></html>");
dish_6 = new JCheckBox("<html><pre>     7  <br/> </pre></html>");
dish_7 = new JCheckBox("<html><pre>     8  <br/> </pre></html>");
dish_8 = new JCheckBox("<html><pre>     9  <br/> </pre></html>");

dish_0.addActionListener(this);
dish_1.addActionListener(this);
dish_2.addActionListener(this);
dish_3.addActionListener(this);
dish_4.addActionListener(this);
dish_5.addActionListener(this);
dish_6.addActionListener(this);
dish_7.addActionListener(this);
dish_8.addActionListener(this);

dish_0.setFont(checkboxFont);
dish_1.setFont(checkboxFont);
dish_2.setFont(checkboxFont);
dish_3.setFont(checkboxFont);
dish_4.setFont(checkboxFont);
dish_5.setFont(checkboxFont);
dish_6.setFont(checkboxFont);
dish_7.setFont(checkboxFont);
dish_8.setFont(checkboxFont);

// Add checkboxes to an array so that we can identify the dishes based
// on index
JCheckBox[] cb = { dish_0, dish_1, dish_2, dish_3, dish_4, dish_5,
```

```
                dish_6, dish_7, dish_8 };
dishCheckBoxes = cb;

// create Dishes
exp0 = new Dish("dish_0");
exp1 = new Dish("dish_1");
exp2 = new Dish("dish_2");
exp3 = new Dish("dish_3");
exp4 = new Dish("dish_4");
exp5 = new Dish("dish_5");
exp6 = new Dish("dish_6");
exp7 = new Dish("dish_7");
exp8 = new Dish("dish_8");

// Add dishes to an array for camera thread processing
Dish[] d = new Dish[] { exp0, exp1, exp2, exp3, exp4, exp5, exp6, exp7,
                exp8 };
dishes = d;
System.out.println(dishes);
System.out.println("======");

cameraThread = new CameraThread(dishes);
new Thread(cameraThread).start();

int dim = 140;

//print dishes on screen
for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
                // i rows, j cols
                c.gridx = j;
                c.gridy = i;
                c.ipady = 55;
                c.ipadx = 55;
                c.insets= new Insets(5,5,5,5);

                System.out.println((3 * i) + j);
                statusPanel.add(cb[(3 * i) + j], c);

        }
}

Timer t = new Timer();
t.schedule(new TimerTask() {
        @Override
        public void run() {
```

```
                    checkPending();

                }
            }, 0, 1000);
    }

    /*
    * Accessor for Status Panel's JPanel
    *
    * @returns JPanel
    */
    public JPanel getPanel() {
        return this.statusPanel;
    }

    /*
    * Accessor for Status Panel's name
    *
    * @returns String
    */
    public String getPanelName() {
        return this.panelName;
    }

    /*
    * Function to be called by Dish Config when user presses "done" on pop-up
    * GUI. Example call of how to bring up my GUI, whenever you need to bring
    * up the pop-up (on selection) use: new DishConfig(4, this);
    */
    public void setDishConfig(boolean isValid, int dishNum, String fileName,
                String numPics, String picMetric, String numTotal,
                String totalMetric) {
        if (isValid) {
                // DO DISH SETUP HERE. Using print statement now to check
everything
                // is working
                System.out.println("Dish config start:");
                System.out.println(dishNum);
                System.out.println(fileName);
                System.out.println(numPics);
                System.out.println(picMetric);
                System.out.println(numTotal);
                System.out.println(totalMetric);
                // figure out which dish is to be changed
                Dish dish = dishes[dishNum];
```

```
			if (createFolder(fileName) == false) { // Create folder if it has

	// not been made before. Add

	// to global array
				System.out.println("Error! File name already used for folder: "
						+ fileName);
				JCheckBox temp = dishCheckBoxes[dishNum];
				temp.setSelected(false);
				return;
			}

			int captureRate = Integer.parseInt(numPics);

			if (captureRate < 1) {
				captureRate = 1;
			}
			dish.setFileName(fileName);
			System.out.println("Dish Name Input: " + dish.getFileName());

			dish.setCaptureRate(captureRate);
			dish.setCaptureMetric(picMetric);
			dish.setExperimentTime(Integer.parseInt(numTotal));
			dish.setExperimentMetric(totalMetric);
			dish.setDishNum(dishNum);

			System.out.println("Capture Rate: 1 image every "
					+ dish.getCaptureRate() + " " + dish.getCaptureMetric());
			System.out.println("Total exp time: " + dish.getExperimentTime()
					+ " " + dish.getExperimentMetric());

			// Figure out how many pictures total are needed for the experiment
			// if same metric, easy
			if (dish.getCaptureMetric() == dish.getExperimentMetric()) {
				dish.setTotalImagesNeeded(dish.getExperimentTime()
						/ dish.getCaptureRate());
				System.out.println("Total # images taken: "
						+ dish.getTotalImagesNeeded());
			}
			else {
				int capTimeSec = Integer.MAX_VALUE; // Fake value to
override;
				int expTimeSec = Integer.MAX_VALUE;// Fake value to
override;

				switch (dish.getCaptureMetric()) {
```

```
                case SECS:
                    capTimeSec = dish.getCaptureRate() * 1000;
                    break;
                case MINS:
                    capTimeSec = dish.getCaptureRate() * 60 * 1000;
                    break;
                case HOURS:
                    capTimeSec = dish.getCaptureRate() * 60 * 60 * 1000;
                    break;
                case DAYS:
                    capTimeSec = dish.getCaptureRate() * 24 * 60 * 60 *
1000;

                    break;
                }
                switch (dish.getExperimentMetric()) {
                case SECS:
                    expTimeSec = dish.getExperimentTime() * 1000;
                    break;
                case MINS:
                    expTimeSec = dish.getExperimentTime() * 60 * 1000;
                    break;
                case HOURS:
                    expTimeSec = dish.getExperimentTime() * 60 * 60 * 1000;
                    break;
                case DAYS:
                    expTimeSec = dish.getExperimentTime() * 24 * 60 * 60 *
1000;

                    break;
                }
                dish.setTotalImagesNeeded(expTimeSec / capTimeSec);
                System.out.println("Total # images taken: "
                        + dish.getTotalImagesNeeded());
            }

            // start time after file name is input
            timeStampString = new SimpleDateFormat("yyyy.MM.dd.HH.mm.ss")
                    .format(new Date());

            timeStamp = new Date();

            System.out.println("Selected:" + " " + timeStampString);
            dish.setTimeStart(timeStamp);
            // later. Fine for now.
            dish.setTimeOfLastPic(timeStamp);
            dish.setEnabled(true);// last thing to set to start taking photos
            System.out.println("Time Start: " + dish.getTimeStart());
```

```
				dishCheckBoxes[dishNum].setForeground(darkOrange);
				if (fileName.length() > 5) {
					// to avoid resizing circle
					dishCheckBoxes[dishNum].setLabel("<html>"
							+ fileName.substring(0, 5) + "<br/>0/"
							+ dish.getTotalImagesNeeded() + "</html>");

				} else {
					dishCheckBoxes[dishNum].setLabel("<html>" + fileName
							+ "<br/>0/" + dish.getTotalImagesNeeded() +
"</html>");

				}
				namedExperiments.add(dish);

			}

			else {
				dishCheckBoxes[dishNum].setSelected(false);
				System.out.println("Invalid Input");
			}
			System.out.println("Dish config end");

		}

		@Override
		public void actionPerformed(ActionEvent e) {
			Object source = e.getSource();

			int counter = 0;
			for (JCheckBox x : dishCheckBoxes) {
				if (source == x) {
					if (x.getModel().isSelected() == true) {
						DishConfig dc = new DishConfig(counter, this);
					} else {
						// The experiment is already started for this dish
						System.out.println("deselect");
						editDish(counter);
						// options should be to download photos, stop experiment,
						// edit experiment
					}
				}
				counter += 1;
			}
		}
```

```
private boolean createFolder(String fileName) {
	if (dishNames.contains(fileName)) {
		return false;
	}

	File dir = new File(PIC_PARENT_DIR + fileName);
	/*
	* if(dir.exists()){//If it exists in OS and we don't have it saved in
	* global, delete (can be modified later) dir.delete(); }
	*/
	dir.mkdirs();
	dishNames.add(fileName);
	return true;
}

public void editDish(int dishNum) {
	// if dish is already selected, pull up an options menu- edit name,
	// image capture, exit experiment, download photos
	// don't uncheck the box unless they exit experiment

	showDishMenu(dishes[dishNum], dishNum);

	timeStampString = new SimpleDateFormat("yyyy.MM.dd.HH.mm.ss")
			.format(new Date());
	timeStamp = new Date();

	System.out.println("Deselected" + " " + timeStampString);
	System.out.println("Dish Deselected: " + dishes[dishNum].getFileName());

	/*
	* TODO: once "deselect" bug is fixed, just use: dish.isEnabled(false)
	* to stop photos from being taken
	*/
}

public void showDishMenu(final Dish dish, final int dishNum) {
	dishCheckBoxes[dishNum].setSelected(true);

	JPanel myPanel = new JPanel();
	// Stop the experiment- use if want to stop image capture
	final JButton stop = new JButton("Clear Experiment");
	stop.addActionListener(new ActionListener() {
		@Override
		public void actionPerformed(ActionEvent arg0) {
			dish.setEnabled(false);
			dishCheckBoxes[dishNum].setSelected(false);
```

```
                                        dish.reset();
                                        dishPending(dish);

          dishCheckBoxes[dishNum].setLabel("<html><pre>     "+(dishNum + 1)+"  <br/>
</pre></html>");
                              }
                    });
                    myPanel.add(stop);

                    // pop up dialog
                   CloseDialog x = new CloseDialog( dish.getFileName()+ " Options", new JPanel(),
stop, 200, 100);
                    dishCheckBoxes[dishNum].setForeground(Color.BLACK);

          }

          public void checkPending() {
                    for (Dish d : dishes) {
                              dishPending(d);
                    }
          }

          public void dishPending(Dish d) {
                    // If experiment is finished, change label to Done
                    // System.out.println( namedExperiments);
                    // System.out.println(dishes);
                    if (namedExperiments.contains(d) && d.isFinished() && !d.isCleared()) {
                              d.setCleared(true);
                              dishes[d.getDishNum()] = new Dish();
                              dishCheckBoxes[d.getDishNum()].setBorderPainted(false);
                              dishCheckBoxes[d.getDishNum()].setForeground(new Color(0,150,0));
                              if (d.getFileName().length() > 5) {

          dishCheckBoxes[d.getDishNum()].setLabel("<html><pre>   "
                                                            + d.getFileName().substring(0, 5) +
"<br/>   "
                                                            + d.getPicsTaken() + "/" +
d.getTotalImagesNeeded()
                                                            + "</pre></html>");

                              } else {

          dishCheckBoxes[d.getDishNum()].setLabel("<html><pre>   "
                                                  +  d.getFileName() + "<br/>   "
                                                            + d.getPicsTaken() + "/" +
d.getTotalImagesNeeded()
```

```
						+ "</pre></html>");
			}
			System.out.println("Done" + d.getDishNum());

		}
		// if experiment is not finished, update number of photos taken
		else if (namedExperiments.contains(d) && !d.isCleared()) {
			System.out.println(d.getFileName() + "  dfsd");
			String fn = d.getFileName();
			if (fn.length() > 5) {
				// to avoid resizing circle

	dishCheckBoxes[d.getDishNum()].setLabel("<html><pre>   "
						+ fn.substring(0, 5) + "<br/>   " +
d.getPicsTaken() + "/"
						+ d.getTotalImagesNeeded() + "</pre></html>");

			} else {

	dishCheckBoxes[d.getDishNum()].setLabel("<html><pre>   " + fn +
"<br/>   "
						+ d.getPicsTaken() + "/" +
d.getTotalImagesNeeded()
						+ "</pre></html>");

			}
		}
	}

}
```

<u>LightingPanel.java:</u>

```
import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Dimension;
import java.awt.Font;
import java.awt.GridBagConstraints;
import java.awt.GridBagLayout;
import java.awt.Insets;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.FocusEvent;
import java.awt.event.FocusListener;
import java.awt.event.MouseListener;
import java.io.IOException;
import java.util.ArrayList;
```

```
import java.util.Arrays;

import javax.swing.BorderFactory;
import javax.swing.BoxLayout;
import javax.swing.JButton;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JSlider;
import javax.swing.JSpinner;
import javax.swing.JTextField;
import javax.swing.SpinnerNumberModel;
import javax.swing.border.Border;
import javax.swing.border.EtchedBorder;
import javax.swing.event.ChangeEvent;
import javax.swing.event.ChangeListener;

public class LightingPanel extends JPanel {
  private JPanel lightPanel; // Panel to make modifications to
  private String lightName; // Name for panel
  private JButton previewButton;
  private JButton slidePreview;
  private static final String PATH_TO_LED_SCRIPT = "/home/pi/py/strip.py";
  private static final int RED_PIN = 16; // GPIO Pin
  private static final int GREEN_PIN = 20; // GPIO Pin
  private static final int BLUE_PIN = 21; // GPIO Pin
  private JSlider redSlider;
  private JSlider blueSlider;
  private JSlider greenSlider;
  private JSpinner redBox;
  private JSpinner blueBox;
  private JSpinner greenBox;
  private final String RED_LABEL_TEXT = "Red:   ";
  private final String GREEN_LABEL_TEXT = "Green:";
  private final String BLUE_LABEL_TEXT = "Blue:  ";
  private JButton advancedOptions;
  boolean redBoxFocus=false;

  private int REDval;
  private int GREENval;
  private int BLUEval;

  public LightingPanel() {

     try {
        Runtime.getRuntime().exec("sudo pigpiod");
```

```
} catch (IOException error) {
   System.out.println("Error! Could not start LED Service");
}
// Initialize components
lightPanel = new JPanel();
// lightPanel.setLayout(new GridLayout(4, 1)); // Add a layout manager
// to align components

// Initialize Grid bag layout
lightPanel.setLayout(new GridBagLayout());
GridBagConstraints c = new GridBagConstraints();
c.fill = GridBagConstraints.HORIZONTAL;

lightName = "Lighting"; // Assign name

// Setup sliders
redSlider = new JSlider(0, 255);
blueSlider = new JSlider(0, 255);
greenSlider = new JSlider(0, 255);

// Set paint components
redSlider.setPaintTicks(true);
blueSlider.setPaintTicks(true);
greenSlider.setPaintTicks(true);

redSlider.setMinorTickSpacing(5);
blueSlider.setMinorTickSpacing(5);
greenSlider.setMinorTickSpacing(5);

redSlider.setMajorTickSpacing(10);
blueSlider.setMajorTickSpacing(10);
greenSlider.setMajorTickSpacing(10);

// Setup spinners
redBox = new JSpinner(new SpinnerNumberModel(redSlider.getValue(), 0,
      255, 1)); // value, min, max, step
blueBox = new JSpinner(new SpinnerNumberModel(blueSlider.getValue(), 0,
      255, 1)); // Stick to slider value to avoid API changes
greenBox = new JSpinner(new SpinnerNumberModel(greenSlider.getValue(),
      0, 255, 1));

redBox.setBorder(BorderFactory
      .createEtchedBorder(EtchedBorder.RAISED));
blueBox.setBorder(BorderFactory
      .createEtchedBorder(EtchedBorder.RAISED));
greenBox.setBorder(BorderFactory
```

```
        .createEtchedBorder(EtchedBorder.RAISED));

    // if want to add keyboard
//    JSpinner.DefaultEditor editor = (JSpinner.DefaultEditor)redBox.getEditor();
//    final JTextField textField = editor.getTextField();
//
//    textField.addFocusListener(new FocusListener(){
//
//        @Override
//        public void focusGained(FocusEvent e) {
//            if(redBoxFocus == false){
//                redBoxFocus = true;
//                new Keyboard("0123456789", textField, 4, 3, lightPanel);
//            }else{
//                redBoxFocus = false;
//            }
//        }
//
//        @Override
//        public void focusLost(FocusEvent e) {
//
//
//        }});

    // box for color preview based on RGB values set with sliders
    previewButton = new JButton("Current Setting");
    previewButton.setPreferredSize(new Dimension(40, 40));
    previewButton.setBackground(new Color(redSlider.getValue(), greenSlider
        .getValue(), blueSlider.getValue()));
    previewButton.setOpaque(true);
    previewButton.setName("Preview");

    slidePreview = new JButton("Current Setting");
    slidePreview.setPreferredSize(new Dimension(40, 40));
    slidePreview.setBackground(new Color(redSlider.getValue(), greenSlider
        .getValue(), blueSlider.getValue()));
    slidePreview.setOpaque(true);
    slidePreview.setName("Preview");

    // advanced settings button
    advancedOptions = new JButton("<html><u>Advanced</u></html>");
    c.gridx = 2;
    c.gridy = 2;
    c.ipadx = 10;
    c.insets = new Insets(60, 10, 0, 0);
    c.anchor= GridBagConstraints.NORTH;
```

```
        advancedOptions.setFont(new Font("Arial", Font.PLAIN, 18));
        advancedOptions.setBorder(BorderFactory.createEmptyBorder());
        advancedOptions.setForeground(Color.MAGENTA);
        advancedOptions.setName("AdvancedButton");
        lightPanel.add(advancedOptions, c);

        advancedOptions.addActionListener(new ActionListener() {
           @Override
           public void actionPerformed(ActionEvent arg0) {
              new CloseDialog("Advanced Lighting Settings" ,advancedSettings(), new
JButton("Done"), 800,
                    430);
           }
        });

        JLabel inst = new JLabel("Select desired lighting color");
        inst.setFont(new Font("Arial", Font.BOLD, 20));
        // c.weightx = 1.0;
        // c.gridwidth = 1;
        c.gridx = 1;
        c.gridy = 0;
        c.anchor = GridBagConstraints.EAST;
        c.insets = new Insets(30, 150, 20, 0);
        lightPanel.add(inst, c);

        // Preset colors
        JPanel colorPresets = new JPanel();
        colorPresets.setLayout(new GridBagLayout());

        ArrayList<Color> colors = new ArrayList<Color>(Arrays.asList(new Color(
              255, 0, 0), new Color(0, 255, 0), new Color(0, 0, 255),
              new Color(255, 255, 0), new Color(255, 255, 255), new Color(0,
                 0, 0)));


        ArrayList<String> colorNames= new ArrayList<String>(Arrays.asList("Red","Green",
"Blue", "Yellow", "White", "Black" ));
        Color [] foreground= {Color.white, Color.black, Color.white, Color.black, Color.black,
Color.white};

        int count = 0;
        for (final Color col : colors) {
           // label each color with word
           final JButton Preset = new JButton();
           Preset.setText(colorNames.get(count));
           Preset.setForeground(foreground[count]);
```

```
    Preset.setPreferredSize(new Dimension(60, 60));
    Preset.setOpaque(true);
    Preset.setBackground(col);
    Preset.setFocusable(false);
    Preset.setBorder(BorderFactory
            .createEtchedBorder(EtchedBorder.RAISED));

    Preset.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent a) {

            REDval=col.getRed();
            BLUEval=col.getBlue();
            GREENval=col.getGreen();

            redSlider.setValue(REDval);
            blueSlider.setValue(BLUEval);
            greenSlider.setValue(GREENval);
            //setLights();
            previewButton.setBackground(new Color(redSlider.getValue(),
                    greenSlider.getValue(), blueSlider.getValue()));
            setButtonForeground(previewButton);

        }
    });

    if (count > 2) {
        c.gridx = count % 3;
        c.gridy = 1;
    } else {
        c.gridx = count;
        c.gridy = 0;
    }

    c.ipady = 50;
    c.ipadx = 70;
    c.insets = new Insets(10, 10, 10, 10);
    c.gridwidth = 1;
    count++;
    colorPresets.add(Preset, c);
}
c.insets = new Insets(0, 0, -20, 0);

c.gridx = 1;
c.gridy = 1;
c.weighty = 1.0;
```

```
    c.gridwidth = 2;
    c.ipady = 0;
    c.fill = GridBagConstraints.VERTICAL;
    c.anchor = GridBagConstraints.SOUTH;
    // colorPresets.setBackground(Color.BLUE);
    lightPanel.add(colorPresets, c);
    // Add individual panels to main panel
    c.gridwidth = count;

    c.gridx = 0;

    // add preview Button
    c.gridy = 2;
    c.ipady = 0;
    c.ipadx = 100;
    c.insets = new Insets(40, 0, 20, -10);
    lightPanel.add(previewButton, c);
    advancedSettings();
}

public JPanel advancedSettings() {
    // Add update listeners to sliders
    redSlider.addChangeListener(new ChangeListener() {

        @Override
        public void stateChanged(ChangeEvent e) {
            REDval=redSlider.getValue();

            redBox.setValue(redSlider.getValue());
            previewButton.setBackground(new Color(redSlider.getValue(),
                    greenSlider.getValue(), blueSlider.getValue()));
            slidePreview.setBackground(new Color(redSlider.getValue(),
                    greenSlider.getValue(), blueSlider.getValue()));
            setButtonForeground(previewButton);
            setButtonForeground(slidePreview);
            //setLights();
        }
    });

    blueSlider.addChangeListener(new ChangeListener() {

        @Override
        public void stateChanged(ChangeEvent e) {
            BLUEval=blueSlider.getValue();
            blueBox.setValue(blueSlider.getValue());
            previewButton.setBackground(new Color(redSlider.getValue(),
```

```
            greenSlider.getValue(), blueSlider.getValue()));
        slidePreview.setBackground(new Color(redSlider.getValue(),
            greenSlider.getValue(), blueSlider.getValue()));
        setButtonForeground(previewButton);
        setButtonForeground(slidePreview);
        //setLights();
    }
});

greenSlider.addChangeListener(new ChangeListener() {

    @Override
    public void stateChanged(ChangeEvent e) {
        GREENval=greenSlider.getValue();
        greenBox.setValue(greenSlider.getValue());
        previewButton.setBackground(new Color(redSlider.getValue(),
            greenSlider.getValue(), blueSlider.getValue()));
        slidePreview.setBackground(new Color(redSlider.getValue(),
            greenSlider.getValue(), blueSlider.getValue()));
        setButtonForeground(previewButton);
        setButtonForeground(slidePreview);
        //setLights();
    }
});

// Add listeners to box
redBox.addChangeListener(new ChangeListener() {
    @Override
    public void stateChanged(ChangeEvent e) {
        REDval=(int)redBox.getValue();
        redSlider.setValue((int) (redBox.getValue()));
        previewButton.setBackground(new Color(redSlider.getValue(),
            greenSlider.getValue(), blueSlider.getValue()));
        slidePreview.setBackground(new Color(redSlider.getValue(),
            greenSlider.getValue(), blueSlider.getValue()));
        setButtonForeground(previewButton);
        setButtonForeground(slidePreview);
        setLights();
    }
});

blueBox.addChangeListener(new ChangeListener() {

    @Override
    public void stateChanged(ChangeEvent e) {
        BLUEval=(int)blueBox.getValue();
```

```
        blueSlider.setValue((int) (blueBox.getValue()));
        previewButton.setBackground(new Color(redSlider.getValue(),
                greenSlider.getValue(), blueSlider.getValue()));
        slidePreview.setBackground(new Color(redSlider.getValue(),
                greenSlider.getValue(), blueSlider.getValue()));
        setButtonForeground(previewButton);
        setButtonForeground(slidePreview);
        setLights();
    }
});

greenBox.addChangeListener(new ChangeListener() {

    @Override
    public void stateChanged(ChangeEvent e) {
        GREENval=(int)greenBox.getValue();
        greenSlider.setValue((int) (greenBox.getValue()));
        previewButton.setBackground(new Color(redSlider.getValue(),
                greenSlider.getValue(), blueSlider.getValue()));
        slidePreview.setBackground(new Color(redSlider.getValue(),
                greenSlider.getValue(), blueSlider.getValue()));
        setButtonForeground(previewButton);
        setButtonForeground(slidePreview);
        setLights();
    }
});

Font boldFont = new Font("Ariel", Font.BOLD, 26);

// Make separate panels for easily adding /arranging to main panel
JPanel redPanel = new JPanel(new BorderLayout());
JLabel rl = new JLabel(RED_LABEL_TEXT);
rl.setForeground(new Color(255, 0, 0));
rl.setFont(boldFont);
redPanel.add(rl, BorderLayout.WEST);
redPanel.add(redSlider, BorderLayout.CENTER);
redPanel.add(redBox, BorderLayout.EAST);

JPanel greenPanel = new JPanel(new BorderLayout());
JLabel gl = new JLabel(GREEN_LABEL_TEXT);
gl.setForeground(new Color(0, 210, 0));
gl.setFont(boldFont);
greenPanel.add(gl, BorderLayout.WEST);
greenPanel.add(greenSlider, BorderLayout.CENTER);
greenPanel.add(greenBox, BorderLayout.EAST);
```

```
    JPanel bluePanel = new JPanel(new BorderLayout());
    JLabel bl = new JLabel(BLUE_LABEL_TEXT);
    bl.setForeground(new Color(0, 0, 255));
    bl.setFont(boldFont);
    bluePanel.add(bl, BorderLayout.WEST);
    bluePanel.add(blueSlider, BorderLayout.CENTER);
    bluePanel.add(blueBox, BorderLayout.EAST);

    final JPanel lightingSliders = new JPanel();
    lightingSliders.setLayout(new GridBagLayout());
    GridBagConstraints f= new GridBagConstraints();

    f.gridx=0;
    f.ipady=20;
    f.ipadx=400;
    f.insets= new Insets(10,10,10,10);
    f.fill=GridBagConstraints.HORIZONTAL;

    f.gridy=1;
    lightingSliders.add(redPanel,f);

    f.gridy=2;
    lightingSliders.add(greenPanel,f);

    f.gridy=3;
    lightingSliders.add(bluePanel,f);

    f.gridy=0;

    f.insets= new Insets(-10,0,-10,0);
    f.anchor= GridBagConstraints.SOUTH;
    lightingSliders.add(slidePreview,f );

    return lightingSliders;

}

// public void paintComponent(Graphics g){
// System.out.println("hello");
// super.paintComponent(g);
// //Draw a Box to show color
// g.drawRect(100, 300, 50, 50);
// g.setColor(Color.BLACK);
// g.fillRect(25, 55, 200, 100);
// g.drawString("hello", 40, 40);
// }
```

```
public JPanel getPanel() {
   return this.lightPanel;
}

public String getPanelName() {
   return lightName;
}

public Color getLightColor() {
   return new Color(redSlider.getValue(), greenSlider.getValue(),
         blueSlider.getValue());
}

// tell if the light color is darker or lighter
// used to tell if writing on top of color should be white or black
public boolean lightOrDark() {
   // light= T
   // dark = F

   // Special case for solid green- green is so light on its own
   if (greenSlider.getValue() == 255 && redSlider.getValue() == 0
         && blueSlider.getValue() == 0) {
      return true;
   }
   if (redSlider.getValue() + greenSlider.getValue()
         + blueSlider.getValue() > 383) {
      return true;
   }
   return false;
}

public void setButtonForeground(JButton button) {
   if (lightOrDark() == false) {
      // dark background so need light writing
      button.setForeground(Color.WHITE);
   } else {
      button.setForeground(Color.BLACK);
   }
}

public void setLights() {
   try {
      Runtime.getRuntime().exec(
            "sudo python " + PATH_TO_LED_SCRIPT + " " + RED_PIN + " "
                  + GREEN_PIN + " " + BLUE_PIN + " "
                  + String.valueOf(REDval) + " "
```

```
                    + String.valueOf(GREENval) + " "
                    + String.valueOf(BLUEval));
          System.out.println("setting lights");
          System.out.println(""+ redBox.getValue()+ " " + greenBox.getValue()+" "+
blueBox.getValue());
      } catch (IOException error) {
          System.out.println("Error! Could not set LED levels");
      }
   }
}
```

Keyboard.java:

```
import java.awt.AWTException;
import java.awt.BorderLayout;
import java.awt.Dimension;
import java.awt.GridLayout;
import java.awt.Robot;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.KeyEvent;
import java.util.ArrayList;

import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.JTextField;
import javax.swing.WindowConstants;

public class Keyboard extends JFrame {
        private final static int SCREEN_WIDTH = 480;
        private final static int SCREEN_HEIGHT = 800;
        String alphabet;
        JTextField destination;
        JTextField popUp;
        int numRow;
        int numCol;
        JPanel keyboardPanel;
        ArrayList<JButton> buttons;
        JPanel retPanel;

        public Keyboard(String alphabet, JTextField destination, int numRow, int numCol,
JPanel returnPanel){
                super("Keyboard");
                this.alphabet = alphabet;
                this.destination = destination;
                this.numRow = numRow;
```

```
        this.numCol = numCol;
        this.retPanel = returnPanel;
        setLayout(new BorderLayout());
        popUp = new JTextField(destination.getText());
        popUp.setHorizontalAlignment(JTextField.CENTER);
        add(popUp, BorderLayout.NORTH);
        createKeyboard();
        setSize(new Dimension(SCREEN_HEIGHT, SCREEN_WIDTH));
        setDefaultCloseOperation(WindowConstants.DO_NOTHING_ON_CLOSE);
        setResizable(false);
        setVisible(true);
}

public Keyboard(String alphabet, JTextField destination){
        super("Keyboard");
        this.alphabet = alphabet;
        this.destination = destination;
        setLayout(new BorderLayout());
        this.keyboardPanel = createKeyboard(3, 10);
        pack();
        setVisible(false);
}

public JPanel getPanel(){
        return keyboardPanel;
}

private void createKeyboard(){

        JPanel buttonPannel = new JPanel();
        buttonPannel.setLayout(new GridLayout(this.numRow, this.numCol));
        //Auto populate with given keys
        for (int i = 0; i < this.alphabet.length(); i++) {
                JButton addButton = new JButton(this.alphabet.substring(i, i + 1));
    addButton.addActionListener(new ActionListener(){

                        @Override
                        public void actionPerformed(ActionEvent e) {

                                String action = e.getActionCommand();
                                destination.setText(destination.getText() + action);
                                popUp.setText(popUp.getText() + action);

                        }
                });
    buttonPannel.add(addButton);
```

```
    }

        //Add seperate panel for formatting
        //JPanel stdButtonPanel = new JPanel();
        //stdButtonPanel.setLayout(new GridLayout(2, 1));

        //Add a delete button
        JButton deleteButton = new JButton("Del");
        deleteButton.addActionListener(new ActionListener(){

                @Override
                public void actionPerformed(ActionEvent e) {
                    if(destination.getText().length() > 0){
                        //Only remove one if there is one to remove

    destination.setText(destination.getText().substring(0, destination.getText().length() - 1));
                        popUp.setText(popUp.getText().substring(0,
popUp.getText().length() - 1));
                    }else{
                        System.out.println("cannot print");
                    }
                }
        });
        //stdButtonPanel.add(deleteButton);
        buttonPannel.add(deleteButton, BorderLayout.CENTER);

        //Add done button to close keyboard
        JButton doneButton = new JButton("DONE");
        doneButton.addActionListener(new ActionListener(){

                @Override
                public void actionPerformed(ActionEvent e) {
                    if(popUp.getText().isEmpty()){
                        popUp.setText("0");
                        destination.setText("0");
                    }
                    dispose();
                    retPanel.requestFocusInWindow();

                }
            });
        //stdButtonPanel.add(doneButton); //Always add a "done" button to exit

        //Add to main panel
        //add(stdButtonPanel, BorderLayout.SOUTH);
```

```
                buttonPannel.add(doneButton, BorderLayout.CENTER);
                add(buttonPannel, BorderLayout.CENTER);
        }

private JPanel createKeyboard(int row, int col){

                JPanel buttonPannel = new JPanel();
                buttonPannel.setLayout(new GridLayout(row, col));
                buttons = new ArrayList<JButton>();
                //Auto populate with given keys
                for (int i = 0; i < this.alphabet.length(); i++) {
                        JButton addButton = new JButton(this.alphabet.substring(i, i + 1));
              addButton.addActionListener(new ActionListener(){

                                @Override
                                public void actionPerformed(ActionEvent e) {

                                        String action = e.getActionCommand();
                                        destination.setText(destination.getText() + action);

                                }
                        });
              buttons.add(addButton);
              buttonPannel.add(addButton);
          }


                //Add a delete button
                JButton deleteButton = new JButton("<-");
                deleteButton.addActionListener(new ActionListener(){

                                @Override
                                public void actionPerformed(ActionEvent e) {
                                        if(destination.getText().length() > 0){
                                                //Only remove one if there is one to remove

        destination.setText(destination.getText().substring(0, destination.getText().length() - 1));
                                        }else{
                                                //System.out.println("cannot print");
                                        }
                                }
                });
                buttonPannel.add(deleteButton);

                //Let frame that contains keyboard create its own done Button
                return buttonPannel;
```

```
	}

	public void setDestination(JTextField newField){
		this.destination = newField;
	}

	public void setAlphaKeys(boolean enabled){
		if(enabled){
			for(JButton button : buttons){
				button.setEnabled(true);
			}
		}else{
			//Disable alphabetic buttons
			for(JButton button : buttons){
				if(Character.isDigit(button.getText().charAt(0)) == false){
					button.setEnabled(false);
				}
			}
		}
	}
}
```

ImagePanel.java:

```
import java.awt.Font;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import javax.swing.GroupLayout;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JSpinner;
import javax.swing.SpinnerModel;
import javax.swing.SpinnerNumberModel;

public class ImagePanel {

	private JPanel imagePanel; // Panel to make modifications to
	private String panelName; // Name for panel
	private JButton statusButton;
	private final String STATUS_BUTTON_TEXT = "Images1";

	public ImagePanel() {
		// Initialize components
		imagePanel = new JPanel();
```

```
// Initialize Group Layout object
GroupLayout layout = new GroupLayout(imagePanel);
imagePanel.setLayout(layout); // Add a layout manager to align elements

// specify automatic gap insertion
layout.setAutoCreateGaps(true);
layout.setAutoCreateContainerGaps(true);

panelName = "Image"; // Assign name

// Setup buttons
statusButton = new JButton(STATUS_BUTTON_TEXT);
statusButton.addActionListener(new ActionListener() { // Add action

        // listener to

        // respond to

        // button

                public void actionPerformed(ActionEvent e) {
                        System.out.println("Checking system status!"); //
Example

                // action

                }

        });

// Setup text labels
JLabel capture = new JLabel("Capture(Number of Images)");
JLabel duration = new JLabel(
                "Total time of Experiment (hours |minutes)");
JLabel title = new JLabel("Image Capture");

capture.setFont(new Font("Arial", Font.PLAIN, 18));
duration.setFont(new Font("Arial", Font.PLAIN, 18));
title.setFont(new Font("Arial", Font.PLAIN, 20));

// Setup number selectors
SpinnerModel captureModel = new SpinnerNumberModel(0, 0, 60, 1);
JSpinner captureSpinner = new JSpinner(captureModel);

SpinnerModel minuteModel = new SpinnerNumberModel(0, 0, 60, 1);
```

```
		JSpinner minuteSpinner = new JSpinner(minuteModel);

		SpinnerModel hourModel = new SpinnerNumberModel(0, 0, 60, 1);
		JSpinner hourSpinner = new JSpinner(hourModel);

		captureSpinner.setFont(new Font("Arial", Font.PLAIN, 36));
		minuteSpinner.setFont(new Font("Arial", Font.PLAIN, 36));
		hourSpinner.setFont(new Font("Arial", Font.PLAIN, 36));

		// Add components
		// 3x3 matrix
		layout.setHorizontalGroup(layout
				.createSequentialGroup()
				.addGroup(
						layout.createParallelGroup(
								GroupLayout.Alignment.LEADING)

	.addComponent(capture).addComponent(duration))
				.addGroup(
						layout.createParallelGroup(
								GroupLayout.Alignment.LEADING)
								.addComponent(title)
								.addComponent(captureSpinner, 100,
100, 100)
								.addComponent(hourSpinner, 100,
100, 100)

				)
				.addGroup(
						layout.createParallelGroup(

	GroupLayout.Alignment.LEADING).addComponent(
								minuteSpinner, 100, 100, 100))

		);

		layout.setVerticalGroup(layout
				.createSequentialGroup()
				.addComponent(title)
				.addGroup(
						layout.createParallelGroup(
								GroupLayout.Alignment.LEADING)
								.addComponent(capture)
								.addComponent(captureSpinner, 100,
100, 100))
				.addGroup(
```

```
                                        layout.createParallelGroup(
                                                  GroupLayout.Alignment.LEADING)
                                                  .addComponent(duration)
                                                  .addComponent(hourSpinner, 100,
100, 100)
                                                  .addComponent(minuteSpinner, 100,
100, 100)));

          // imagePanel.add(statusButton);
          // imagePanel.add(numSpinner);
     }

     /*
     * Accessor for Image Panel's JPanel
     *
     * @returns JPanel
     */
     public JPanel getPanel() {
          return this.imagePanel;
     }

     /*
     * Accessor for Image Panel's name
     *
     * @returns String
     */
     public String getPanelName() {
          return this.panelName;
     }
}
```

HomePanel.java:

```
import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.JButton;
import javax.swing.JPanel;

public class HomePanel {
     private JPanel homePanel; //Panel to make modifications to
     private String panelName; //Name for panel
     private JButton statusButton;
     private JButton newExperimentButton;
     private final String STATUS_BUTTON_TEXT = "<html><u>Current System
Status</u></html>";
```

```
	private final String NEW_EXPERIMENT_BUTTON_TEXT = "Begin New
Experiment";

	public HomePanel(){
		//Initialize components
		homePanel = new JPanel();
		homePanel.setLayout(new GridLayout(1,2)); //Add a layout manager to align
buttons as we resize
		panelName = "Home"; //Assign name
		//Setup and add buttons
		statusButton = new JButton(STATUS_BUTTON_TEXT);
		statusButton.addActionListener(new ActionListener(){ //Add action listener to
respond to button

			public void actionPerformed(ActionEvent e) {
				System.out.println("Checking system status!"); //Example action

			}

		});
		newExperimentButton = new JButton(NEW_EXPERIMENT_BUTTON_TEXT);
		newExperimentButton.addActionListener(new ActionListener(){

			public void actionPerformed(ActionEvent e) { //Add Action listener to
respond to button
				System.out.println("Starting new experiment!"); //Example action

			}

		});
		homePanel.add(statusButton);
		homePanel.add(newExperimentButton);
	}

	/*
	* Accessor for Home Panel's JPanel
	* @returns JPanel
	*/
	public JPanel getPanel(){
		return this.homePanel;
	}

	/*
	* Accessor for Home Panel's name
	* @returns String
	*/
```

```
	public String getPanelName(){
		return this.panelName;
	}
}
```

ExportImagesPanel.java:

```
import java.awt.Color;
import java.awt.Font;
import java.awt.Frame;
import java.awt.GridBagConstraints;
import java.awt.GridBagLayout;
import java.awt.GridLayout;
import java.awt.Insets;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.ItemEvent;
import java.awt.event.ItemListener;
import java.io.File;
import java.io.IOException;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;

import javax.swing.ComboBoxModel;
import javax.swing.GroupLayout;
import javax.swing.JButton;
import javax.swing.JCheckBox;
import javax.swing.JComboBox;
import javax.swing.JFileChooser;
import javax.swing.JFormattedTextField;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JSpinner;
import javax.swing.SpinnerModel;
import javax.swing.SpinnerNumberModel;
import javax.swing.filechooser.FileNameExtensionFilter;
import javax.swing.filechooser.FileSystemView;
import javax.swing.text.MaskFormatter;

public class ExportImagesPanel  {
	private JPanel exportImagesPanel; // Panel to make modifications to
	private String panelName; // Name for panel
	private Font checkboxFont = new Font("Arial", Font.BOLD, 20);
	private JButton dish_1, dish_2, dish_3, dish_4, dish_5, dish_6, dish_7,
			dish_8, dish_9;
```

```
	private Dish exp1, exp2, exp3, exp4, exp5, exp6, exp7, exp8, exp9;
	private Frame frame = new Frame();
	private ArrayList<String> dishNames;
	JComboBox folderNames;
	JComboBox UsbNames;
	JLabel statusLabel = new JLabel();

	public ExportImagesPanel(ArrayList <String> dishes) {
		dishNames=dishes;
		// Initialize components
		exportImagesPanel = new JPanel();

		// Initialize Grid bag layout
		exportImagesPanel.setLayout(new GridBagLayout());
		GridBagConstraints c = new GridBagConstraints();
		c.fill = GridBagConstraints.HORIZONTAL;


		panelName = "Export Images"; // Assign name

		UsbNames=new JComboBox();
		UsbNames.setVisible(false);

		JButton UsbButton= new JButton("Export");
		UsbButton.addActionListener(new ActionListener(){

			@Override
			public void actionPerformed(ActionEvent e) { //Add Action listener to
respond to button
				coppyFiles();
			}

		});

		folderNames=  new JComboBox(dishes.toArray());
		JLabel  dishLabel= new JLabel("Select your experiment");
		JLabel USBLabel= new JLabel("Select your USB");
		statusLabel.setText("Waiting to coppy...");

		c.gridx=0;
		c.gridy=0;
		c.ipady=50;
		c.ipadx=60;
		exportImagesPanel.add(dishLabel,c);

		c.gridx=1;
```

```
		c.gridy=0;
		exportImagesPanel.add(USBLabel,c);

		c.gridx=0;
		c.gridy=1;
		c.insets= new Insets(0,0,0,30);
		exportImagesPanel.add(folderNames,c);



		c.gridy=1;
		c.gridx=1;
		c.insets= new Insets(0,0,0,0);
		exportImagesPanel.add(UsbNames,c);

		c.gridx=0;
		c.gridy=3;
		c.gridwidth=4;
		c.insets= new Insets(40,0,0,0);
		exportImagesPanel.add(UsbButton,c);

		exportImagesPanel.add(statusLabel, c);
		UsbButton.setBackground(new Color(200,200,200));

		System.out.println(folderNames.getSelectedIndex());
	}

	private void coppyFiles(){
		statusLabel.setText("copying in progress... do not interrupt");
		try {
			System.out.println("sudo cp -a
~/Desktop/"+folderNames.getSelectedItem() + " /media/pi/"+UsbNames.getSelectedItem()+"/");
			Process p = Runtime.getRuntime().exec("sudo cp -a
~/Desktop/"+folderNames.getSelectedItem() + " /media/pi/"+UsbNames.getSelectedItem()+"/");
		} catch (IOException e) {
			System.out.println("Error! Could not copy!");
			statusLabel.setText("Copy error!");
		}

	}


	public void refreshFileNames(){
		folderNames.removeAllItems();
		for(String str: dishNames){
			folderNames.addItem(str);
```

```
		}
	}

	public void refreshUsbNames(){
		File[] paths;
		FileSystemView fsv = FileSystemView.getFileSystemView();
		UsbNames.removeAllItems();
		File dev = fsv.getChild(fsv.getRoots()[0], "dev/disk/by-label/");
		for(String path: dev.list()){
			if(!path.equals("boot")){
				System.out.println(path);
				UsbNames.addItem(path);
			}
		}
		UsbNames.setVisible(true);
	}
	/*
	* Accessor for Status Panel's JPanel
	*
	* @returns JPanel
	*/
	public JPanel getPanel() {
		return this.exportImagesPanel;
	}

	/*
	* Accessor for Status Panel's name
	*
	* @returns String
	*/
	public String getPanelName() {
		return this.panelName;
	}
}
```

DishConfig.java:

```
import java.awt.BorderLayout;
import java.awt.Dimension;
import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.FocusEvent;
import java.awt.event.FocusListener;

import javax.swing.BorderFactory;
```

```
import javax.swing.JButton;
import javax.swing.JComboBox;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JTextField;
import javax.swing.WindowConstants;
import javax.swing.border.Border;
import javax.swing.border.EtchedBorder;

public class DishConfig extends JFrame {
	private final static int SCREEN_WIDTH = 480;
	private final static int SCREEN_HEIGHT = 800;
	private int dishNum;
	private String alphabet = "0123456789abcdefghijklmnopqrstuvwxyz-_";
	private JTextField fileName;
	private JTextField numPics;
	private JTextField totalTime;
	private Keyboard keyboard;
	private static String[] times = { "secs", "mins", "hours", "days" };
	private JComboBox picMetrics;
	private JComboBox totalMetrics;
	private StatusPanel callPanel;

	public DishConfig(int dishNumber, StatusPanel callingPanel){
		super("Dish Configuration");
		dishNum = dishNumber;
		setLayout(new GridLayout(5,1));
		callPanel = callingPanel;

		Border etched=BorderFactory.createEtchedBorder(EtchedBorder.RAISED);

		//create smaller panels
		Dimension fieldDimension = new Dimension(180,45);
		JPanel filePanel = new JPanel();
		JLabel fileLabel = new JLabel("Enter file name:");
		fileName = new JTextField();
		fileName.setBorder(etched);

		fileName.setPreferredSize(fieldDimension);
		filePanel.add(fileLabel);
		filePanel.add(fileName);

		JPanel picPanel = new JPanel();
		JLabel picLabel = new JLabel("Enter the image frequency:");
		numPics = new JTextField("1");
```

```
numPics.setPreferredSize(fieldDimension);
numPics.setBorder(etched);
numPics.setHorizontalAlignment(JTextField.CENTER);

picMetrics = new JComboBox(times);
picPanel.add(picLabel);
picPanel.add(numPics);
picPanel.add(picMetrics);

JPanel totalPanel = new JPanel();
JLabel totalLabel = new JLabel("Enter the total time of the experiment:");
totalTime = new JTextField("1");
totalTime.setBorder(etched);
totalTime.setHorizontalAlignment(JTextField.CENTER);
totalTime.setPreferredSize(fieldDimension);
totalMetrics = new JComboBox(times);
totalPanel.add(totalLabel);
totalPanel.add(totalTime);
totalPanel.add(totalMetrics);

//create keyboard
keyboard = new Keyboard(alphabet, fileName); //set filename to default

//add listeners AFTER keyboard
fileName.addFocusListener(new FocusListener(){

    @Override
    public void focusGained(FocusEvent e) {
        keyboard.setDestination(fileName);
        keyboard.setAlphaKeys(true);
    }

    @Override
    public void focusLost(FocusEvent e) {

    }});

numPics.addFocusListener(new FocusListener(){

    @Override
    public void focusGained(FocusEvent e) {
        keyboard.setDestination(numPics);
        keyboard.setAlphaKeys(false);
    }

    @Override
```

```
			public void focusLost(FocusEvent e) {

			}});

		totalTime.addFocusListener(new FocusListener(){

			@Override
			public void focusGained(FocusEvent e) {
				keyboard.setDestination(totalTime);
				keyboard.setAlphaKeys(false);
			}

			@Override
			public void focusLost(FocusEvent e) {

			}});

		JButton doneButton = new JButton("Done");
		doneButton.addActionListener(new ActionListener(){

			@Override
			public void actionPerformed(ActionEvent e) {
				callPanel.setDishConfig(checkFields(), dishNum,
fileName.getText(), numPics.getText(), (String) picMetrics.getSelectedItem(),
totalTime.getText(), (String) totalMetrics.getSelectedItem());
				dispose();
			}

		});

		//add everything to frame
		add(filePanel);
		add(picPanel);
		add(totalPanel);
		add(keyboard.getPanel());
		add(doneButton);
		setSize(new Dimension(SCREEN_HEIGHT, SCREEN_WIDTH));
		setDefaultCloseOperation(WindowConstants.DO_NOTHING_ON_CLOSE);
		setResizable(false);
		setVisible(true);
	}

	private boolean checkFields(){
		if(fileName.getText().isEmpty() || numPics.getText().isEmpty() ||
totalTime.getText().isEmpty()){
			return false;
```

```
        }else{
            return true;
        }
    }

}
```

lookandfeel.xmf:

```
<synth>

    <!-- TAB -->

    <style id="LabelStyle">
        <font name="Arial" size="16" style="bold"/>

    </style>
    <bind style="LabelStyle" type="region" key="Label" />


    <!-- SLIDER -->
    <style id="SliderThumbStyle">
        <imagePainter id="SliderThumb_Normal"    sourceInsets="1 1 1 1"
paintcenter="false" path="images/arrowOrange.png" />
        <property key="SliderThumb.icon" type="idref" value="SliderThumb_Normal"/>
        <opaque value="TRUE"/>

    </style>
    <bind style="SliderThumbStyle" type="region" key="SliderThumb" />


    <style id="SliderTrackStyle">
        <opaque value="TRUE"/>
        <state>
            <color type="BACKGROUND" value="ORANGE"/>
        </state>
    </style>
    <bind style="SliderTrackStyle" type="region" key="SliderTrack" />




    <!-- TAB -->
    <style id="tabStyle">
        <state>
            <imagePainter id="Tab_Normal"    sourceInsets="0 0 0 0"
paintcenter="false" path="images/icon1.png" />
            <property key="Tab.icon" type="idref" value="Tab_Normal"/>
```

```
                        <opaque value="true"/>
                        <!--      <imagePainter id="Tab_Normal" sourceInsets="0 0 10 10"
paintcenter="false" path="dishGreen.png"/> -->

                        <font name="Arial" size="20"  />
                        <color value="BLACK" type="TEXT_FOREGROUND" />
                        <insets top="43" bottom="43" right="6" left="20"/>
                </state>
                <state value="SELECTED">
                        <insets top="43" bottom="43" right="6" left="20"/>

                        <imagePainter id="Tab_Selected"      sourceInsets="0 0 0 0"
paintcenter="false" path="images/tabWhite.png" />
                        <property key="Tab.icon" type="idref" value="Tab_Selected"/>

                        <font name="Arial" size="19" style="BOLD" />

                        <color value="#FFFFFF" type="BACKGROUND" />
                        <color value="#ea5a21" type="TEXT_FOREGROUND" />
                </state>

        </style>
        <!-- Bind tabStyle to all tabs -->

        <bind style="tabStyle" type="region" key="TabbedPaneTab" />
<!--  -->

        <!-- CHECKBOX -->
        <style id="checkBoxStyle">
                <!-- Shift the text one pixel when pressed -->
                <font name="Arial" size="8"/>

                <insets top="-5" left="-5" right="-5" bottom="-5"/>
                <imagePainter id="Checkbox_Normal"      sourceInsets="0 0 0 0"
paintcenter="false" path="images/dish.png" />
                <property key="CheckBox.iconTextGap" type="integer" value="60"/>
                <!-- <property key="CheckBox.textShiftOffset" type="integer" value="10"/> -->


                <!-- <state value="ENABLED"> -->
                        <property key="CheckBox.icon" type="idref"
value="Checkbox_Normal"/>
                <!-- </state> -->
        </style>

        <bind style="checkBoxStyle" type="region" key="CheckBox"/>
```

```
	<style id="ComboBoxStyle">
		<opaque value="TRUE"/>
		<imagePainter id="cb" sourceInsets="50 50 50 50" paintcenter="false" 
path="images/clearBox.png" />
		<property key="ComboBox.icon" type="idref" value="cb"/>
	</style>

	<bind style="ComboBoxStyle" key="ComboBox" type="region"/>


	<style id="TextFieldStyle">
		<opaque value="TRUE"/>
		<imagePainter id="TextFieldBox" sourceInsets="50 50 50 50" 
paintcenter="false" path="images/clearBox.png" />
		<property key="TextField.icon" type="idref" value="TextFieldBox"/>
	</style>

	<bind style="TextFieldStyle" key="TextField" type="region"/>

	<style id="arrowButton">
		<imagePainter id="arrowButton_Normal" sourceInsets="3 3 3 3" 
paintcenter="false" path="images/arrowOrange.png" />
		<property key="arrowButton.icon" type="idref" value="arrowButton_Normal"/>
	</style>

	<bind style="arrowButton" key="ComboBox.arrowButton" type="name"/>


	<!-- arrow buttons on spinners -->
	<style id="downArrow">
		<imagePainter id="downArrow_norm" sourceInsets="3 3 3 3" paintcenter="false" 
path="images/downSpinArrow.png" />
		<property key="arrowButton.icon" type="idref" value="downArrow_norm"/>
	</style>

	<bind style="downArrow" key ="Spinner.previousButton" type="name"/>

	<style id="upArrow">
		<imagePainter id="upArrow_norm" sourceInsets="3 3 3 3" paintcenter="false" 
path="images/upSpinArrow.png" />
		<property key="arrowButton.icon" type="idref" value="upArrow_norm"/>
	</style>
	<bind style="upArrow" key ="Spinner.nextButton" type="name"/>
```

```
	<!-- <style id="PVB">
		<opaque value="TRUE"/>
		<font name="Arial" size="16"/>

		<color value="BLACK" type="TEXT_FOREGROUND" />
		<state>
			<insets top="20" left="20" right="20" bottom="20"/>
		</state>
		<state value="DISABLED">
			<insets top="20" left="20" right="20" bottom="20"/>
			<imagePainter id="previewButton"	sourceInsets="50 50 50 50"
paintcenter="false" path="images/tabOrange.png" />
			<property key="Button.icon" type="idref" value="previewButton"/>
		</state>

	</style>
	<bind style="PVB" type="name" key="Preview"/>  -->

	<style id="buttonStyle">
		<opaque value="TRUE"/>
		<font name="Arial" size="11"/>

		<color value="BLACK" type="TEXT_FOREGROUND" />
		<state>
			<!-- <insets top="10" left="10" right="10" bottom="10"/> -->

			<imagePainter id="buttonImage"	sourceInsets="2 2 2 2"
paintcenter="false" path="images/clearBox.png" />
			<property key="Button.icon" type="idref" value="buttonImage"/>

		</state>
		<!-- <state value="DISABLED">
			<insets top="20" left="20" right="20" bottom="20"/>
			<imagePainter id="buttonImage_disabled"	sourceInsets="10 10 10 10"
paintcenter="false" path="images/clearBoxNoBorder.png" />
			<property key="Button.icon" type="idref"
value="buttonImage_disabled"/>
		</state>
		-->
		<state value="PRESSED">
			<opaque value="true"/>
		</state>

	</style>
	<!-- Bind  to all JButtons -->
	<bind style="buttonStyle" type="region" key="button"/>
```

```
<!--
	<style id="startExpButtonStyle">
		<opaque value="TRUE"/>
		<state>
			<imagePainter id="SEB_norm" sourceInsets="3 3 3 3" paintcenter="false"
path="images/edgeButtonLight.png" />
			<property key="Button.icon" type="idref" value="SEB_norm"/>
		</state>
		<state value = "DISABLED">
			<imagePainter id="SEB_disabled" sourceInsets="3 3 3 3"
paintcenter="false" path="images/edgeButton.png" />
			<property key="Button.icon" type="idref" value="SEB_disabled"/>
		</state>
	</style>
	<bind style="startExpButtonStyle" type="name" key ="ExpButton"/>
-->

	<!-- no border for advanced button -->
	<style id="advButtonStyle">
		<opaque value="TRUE"/>
		<font name="Arial" size="11"/>

		<color value="BLACK" type="TEXT_FOREGROUND" />
		<state>
			<imagePainter id="advButtonImage"	sourceInsets="2 2 2 2"
paintcenter="false" path="images/clearBoxNoBorder.png" />
			<property key="Button.icon" type="idref" value="advButtonImage"/>
		</state>
	</style>

	<!-- Bind buttonStyle to advanced button -->
	<bind style="advButtonStyle" type="name" key="AdvancedButton"/>


	<!-- Style that all regions will use -->
	<style id="backingStyle">
		<!-- Make all the regions that use this skin opaque-->
		<opaque value="TRUE"/>
		<font name="Arial" size="24"/>
		<state>
			<!-- Provide default colors -->
			<color value="#FFFFFF" type="BACKGROUND"/>
			<color value="BLACK" type="FOREGROUND"/>
		</state>
	</style>
	<bind style="backingStyle" type="region" key=".*"/>
```

</synth>

camera.py:

```
import picamera #Pi Camera Library
from time import sleep #Sleep Library

camera = picamera.PiCamera() #New Camera Obj
camera.resolution = (3280, 2464);
camera.start_preview() #Fill screen with camera preview for demo

sleep(5) #Wait 5 seconds

camera.stop_preview() #End preview
cameraPath = '/home/pi/Desktop/8mp.jpg'
camera.capture(cameraPath)
camera.stop_preview()
```

temp_bridge.py:

```
import serial

ser = serial.Serial('/dev/ttyACM0', 9600)

message = ser.readline()
print(message)
```

relay_on.py:

```
import RPi.GPIO as GPIO

pin = sys.argv[1]
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)

GPIO.setup(pin, GPIO.OUT)
GPIO.output(pin, GPIO.HIGH)
```

relay_off.py:

```
import RPi.GPIO as GPIO

pin = sys.argv[1]
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)

GPIO.setup(pin, GPIO.OUT)
GPIO.output(pin, GPIO.LOW)
```

strip.py:

```
import sys
import pigpio

pi = pigpio.pi()
pi.set_PWM_dutycycle(float(sys.argv[1]),float(sys.argv[4]))
pi.set_PWM_dutycycle(float(sys.argv[2]),float(sys.argv[5]))
pi.set_PWM_dutycycle(float(sys.argv[3]),float(sys.argv[6]))
```