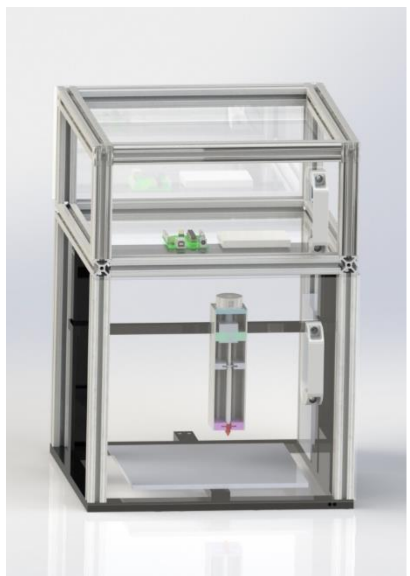
Automated Microarray System



By: Miranda Griffith, Michael D’Arpino, Michael Sherman, and John Handy

Table of Contents

Table of Figures

Introduction

An automated microarray device is used to arrange a pattern of bio-solution onto a substrate, in this case a microscope slide, to then be analyzed. An example of a microarray is shown in Figure 1. The benefit of using an automated microarray device is that it allows simultaneous experiments while under the same conditions. In industry, these machines are extremely expensive, so the goal of the group is to design a device that is much more accessible to those not working in industry by greatly reducing the price.

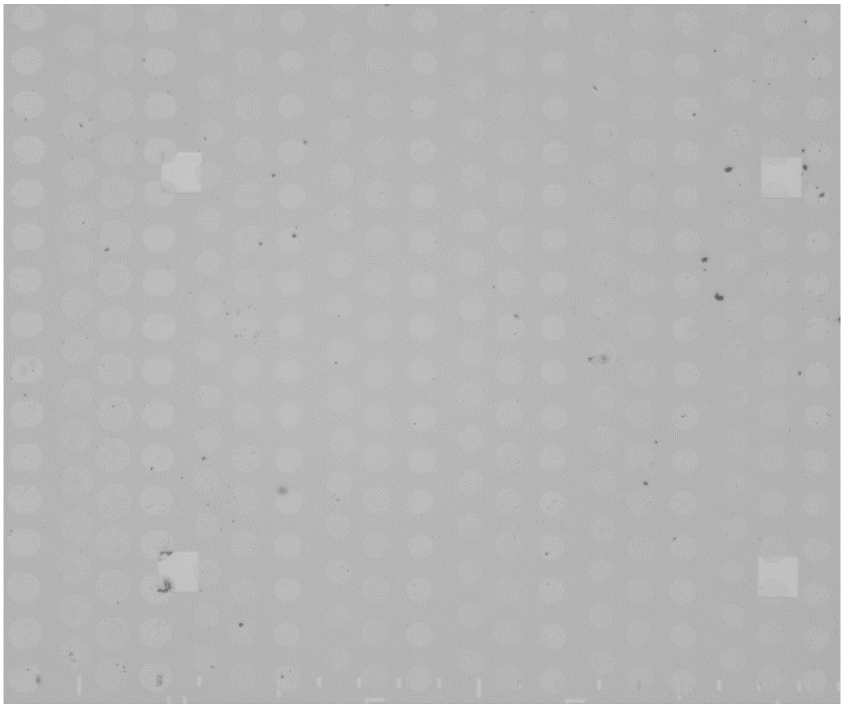


Figure 1: Photo of a microarray.

There are many different uses for microarrays including antibody testing, drug testing, and allergy testing. One specific example is coronavirus antibody testing which is used to determine if a person has been exposed to the coronavirus and has developed antibodies. The first step of that process is taking a blood sample from a patient and spinning it in a centrifuge to separate the parts of the blood by their densities. After that, proteins are extracted from the blood and the sample is treated to remove parts difficult to spot such as red blood cells. Next, the proteins are spotted onto a slide using a microarray system. The microarray is left to dry so that it is no longer a solution and only the proteins stuck to the slide remain. An indicator solution is flowed over the microarray to see if any spots on the slide react. If there is a reaction, coronavirus antibodies are present.

On the market, there is currently a wide variety of automated microarray systems capable of handling similar tasks with distinct built-in features. These manufactured systems can range anywhere from twenty thousand dollars well over fifty thousand dollars and are tailored towards professional lab settings within industry workplaces. Industry standards, such as M2 Automation's iZero seen in Figure 2, are capable of printing arrays within as little as 1µm single increment resolution. This machine's dispensing is pump driven with specialized systems to assure an air free path. The container is also climate controlled to produce optimal conditions.

While these design features assure exact repeatability when testing large amounts of samples, the iZero is so designed towards industry that it is not sold as a single unit. The goal of creating this automated microarray system is to achieve a printing resolution as close as possible to common place machinery that is already being used, while keeping the price tag low enough that all lab settings, including academia, could afford one for testing.

Figure 2: Current solution of a microarray instrument.

Designs

Once the project was broken into different parts specific to each group member, a variety of solutions for each part were explored. By brainstorming multiple solutions, a better idea of the best and worst qualities is gained. In doing so, the group's ideas are conjoined in order to produce a model possessing the best qualities of each group member's designs. This finalized model was then modeled using SolidWorks, however, after beginning the budget stage it became apparent that the decided upon model would not be sufficient.

X, Y, Z Coordinate Movement System

The device planned will move in the x, y, and z directions using a system of stepper motors and pulleys. To allow for different arrangement patterns of the microarray, the system must be programmable using the systems' touchscreen interface. The framing of the system must also be strong enough to support the increased weight of the nozzle fitted with the dispensing mechanism. This movement system must precisely position the droplets in the array to an accuracy of 50 microns. Additionally, the system is required to cover an area that is eight by twelve millimeters.

In brainstorming possible solutions for the x, y, and z coordinate movement system, three original designs were considered. The first of the three was inspired by the common design of many desktop 3D printers. The tray is moving in the x-direction and this movement is provided by a belt and stepper motor. The nozzle moves along a shaft in the y-direction also by the use of a belt and stepper motor. The wiring in this design, as well as the next two, is located underneath the tray. After further consideration, it was concluded that this would not be ideal due to the possibility of spillage onto the circuitry. This design is shown below in Figure 3.

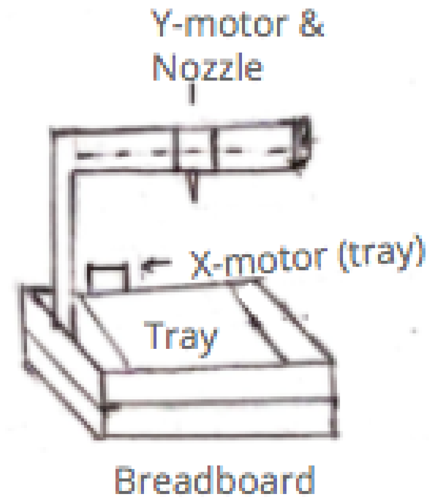


Figure 3: First design idea with a moving tray in x-direction.

The next solution considered was one with a completely stationary tray. An option with a stationary tray was explored due to concern that motion of the tray would affect the accurate placement of the droplets in the array. This model has motors on the top frame that allows for the movement of the nozzle in both the x and y directions. This frame is moved in the z-direction using dual lead screw motors. Using a lead screw motor on each side as opposed to a single motor allowed for more even support of the upper framing, which would have to support the weight of the x and y direction motors, as well as the motor utilized in the dispensing mechanism. Additionally, use of lead screws are best when using them for vertical movement applications due to the resisted tendency to back drive. This sturdier design is labeled and shown below in Figure 4.

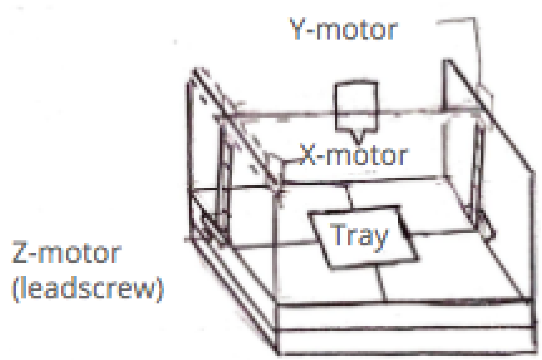


Figure 4: Second design idea with a stationary tray.

The third and final solution considered has mixed qualities from each of the previous solution ideas. The primary difference between this design and the others is that this design features a tray moving in the z-direction. This movement, similarly, to the previous design, is provided by dual lead screw motors to ensure even vertical movement of the tray. Also, similarly to the previous design, the pulley systems allowing for movement in the x and y directions are on the upper frame. In comparison to the previous designs, this model has additional framing to adequately support the upper framing components, as well as the dispensing mechanism. This design is shown below in Figure 5.

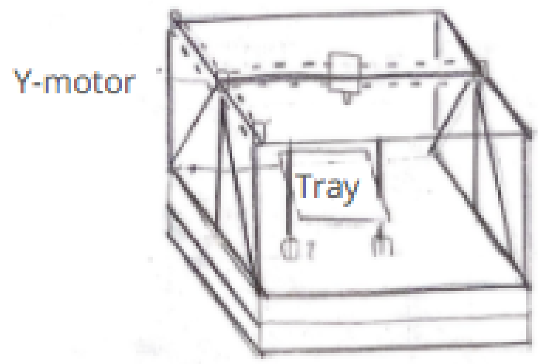


Figure 5: Final design idea with a moving tray in the z-direction.

In consideration of these three solutions for the x, y, and z coordinate movement system, it was concluded that the second of the three designs was best, and it was modeled using SolidWorks. After doing so, it became apparent that, given the allowed budget, purchasing the parts necessary for this movement system individually would not be possible. In reaching this realization, it was decided that the best way to acquire the parts necessary for the movement system, given the budget limitations, would be to buy a pre-existing kit containing the motors, belts, framing, etc. necessary to build the movement system. Additionally, purchasing a disassembled 3D printer kit also would guarantee that the parts would work together flawlessly.

Figure 6: DIY Single Extruder Desktop FDM 3-D Printer.

Upon searching for a disassembled 3D printer kit in the price range of 150 dollars, it was found that the options were relatively limited. Many of the kits started at about 200 dollars, so it was a challenge in itself finding these options. The model decided upon is the Single Extruder Desktop FDM 3D Printer, shown above in Figure 6, was chosen for a number of reasons. Because the movement system must be extremely precise, accuracy of the kit was important to keep in mind when selecting kits. This printer has an accuracy of 0.1 millimeters, which meets the accuracy requirement of the movement system. Dual lead screw motors will provide the stability necessary to accommodate for the large dispensing mechanism. Overall, this kit was evidently the best option for the x, y, z movement system given the budget limitations. In deciding upon this kit specifically, it was modeled in order to gain a better understanding of the size needed for other parts such as the outer casing. This model is shown below in Figure 7.

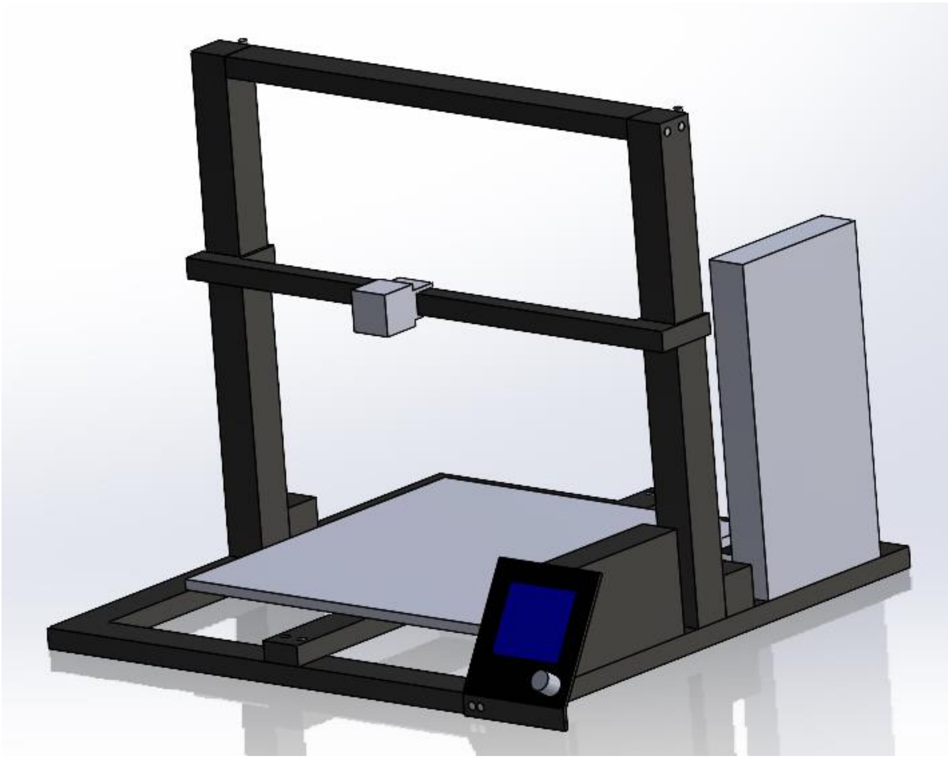


Figure 7: SolidWorks assembly of Single Extruder Desktop FDM 3-D Printer modeled for size reference.

Dispensing Mechanism

The dispensing mechanism must dispense droplets with a consistent volume of 1 microliter or less. The mechanism must be user friendly, meaning that it must be intuitive, easy to clean, and easy to fill with solution. The design should be lightweight and small enough to be moved into position by the x, y, z coordinate mechanism. Physical properties of the bio-solution must be considered in order to avoid air bubbles and produce consistently sized and shaped droplets. If any air bubbles are dispensed or if there are any irregularities in the droplets, the microarray will need to be spotted again.

Three potential designs were considered for the dispensing mechanism. The first design was to use a ballpoint pen to dispense the bio-solution. The ink cartridge of the pen would be removed, cleaned out, and filled with bio-solution. When the tip of the pen made contact with the slide, a small amount of solution would be dispensed. This is a design that has been used in other projects, and if it were to be used in this project it would be a low-cost and simple solution to the problem. The droplets produced using this design were inconsistent in size and shape during testing and it was decided that the work required to fill the cartridge would be too much to ask of a user who is expecting an automated solution. A sketch of this design is shown below in Figure 8.

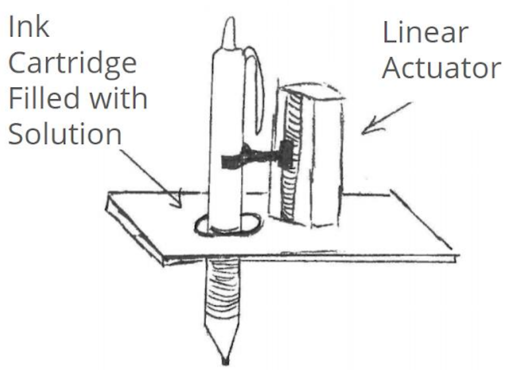


Figure 8: Sketch of ballpoint pen dispensing mechanism design.

The second considered design utilized a dosing pump to create the droplets. Dosing pumps are small pumps capable of dispensing small, controlled volumes of fluid at a steady rate. They are often used to pump measured amounts of chemicals into aquariums. The advantage of this design was that the pump would be small enough to be moved, but the pump also did not need to be moved. It would be possible to have the pump fixed and move only the tube using the x, y, z movement system. A sketch of this design is shown below in Figure 9. The disadvantage of this idea is that dosing pumps are better for continuous flow than they are for creating droplets. The testing of multiple pumps would need to be done to ensure that droplets can be formed before this design could be selected.

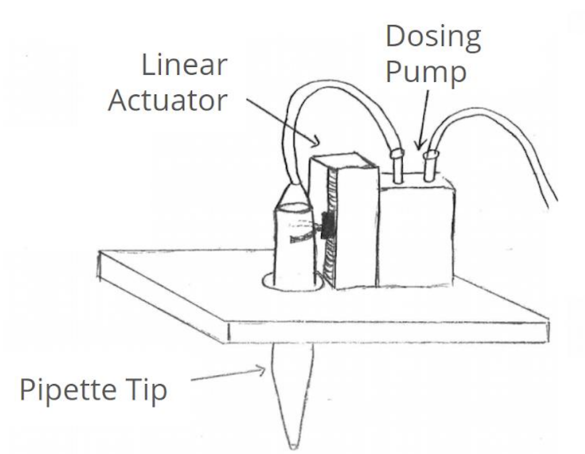


Figure 9: Sketch of dosing pump dispensing mechanism design.

The third considered design involved using a stepper motor attached to a lead screw to control the plunger of a syringe. Because the stepper motor is able to make small and controlled movements, it can cause the syringe to dispense precise amounts of solution consistently. By using a low-volume syringe, only a small amount of solution would be dispensed with each step of the motor. This design is user-friendly because syringes are tools that a person working in a lab is already familiar with. Syringes are readily available in labs, they are also low-cost and dispensable. A sketch of this design is shown below in Figure 10.

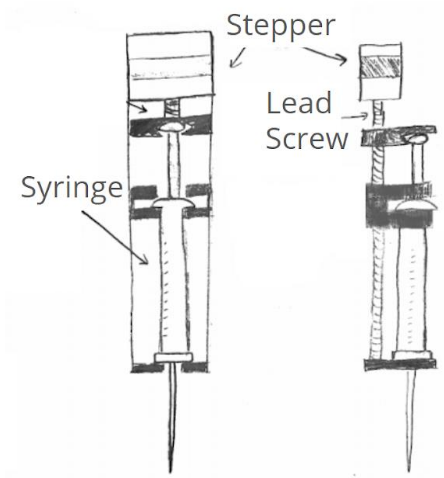


Figure 10: Sketch of stepper motor-controlled syringe design.

The stepper-controlled syringe was determined to be the best choice because it is a robust solution and is the most user-friendly option. The current design is shown in Figure 11.

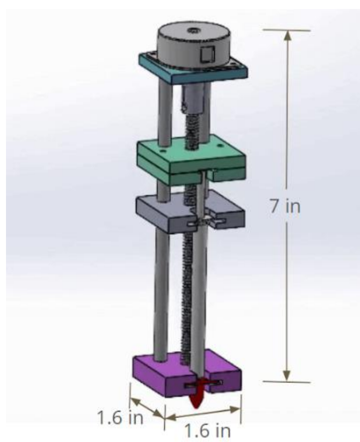


Figure 11: Current dispensing mechanism design. Note that the side panels have been removed for better visibility of the mechanism.

The stepper motor that was selected has a stride angle of 7.5 degrees, the selected lead screw has a travel distance per turn of 0.063 inches and a 1 mL syringe will be used. With these factors, the volume of the droplet per step of the motor is calculated to be 0.57 microliters as shown in Equation 1.

$$\left(\frac{1\,mL}{5.8cm}\right)\left(\frac{2.54cm}{1\,in}\right)\left(\frac{0.063\,in}{1\,turn}\right)\left(\frac{1\,turn}{48\,steps}\right) = 0.00057\,mL = 0.57\mu L$$

Equation 1

GUI and Processing System

The overarching intention when devising a graphical user interface and a processing system for the microarrayer was to hone in on compatibility, user friendliness, making the system stand alone, and to complete this all while keeping the circuit elements minimal and confined. The design philosophy was further divided into subcategories that are highlighted in the provided flow chart.

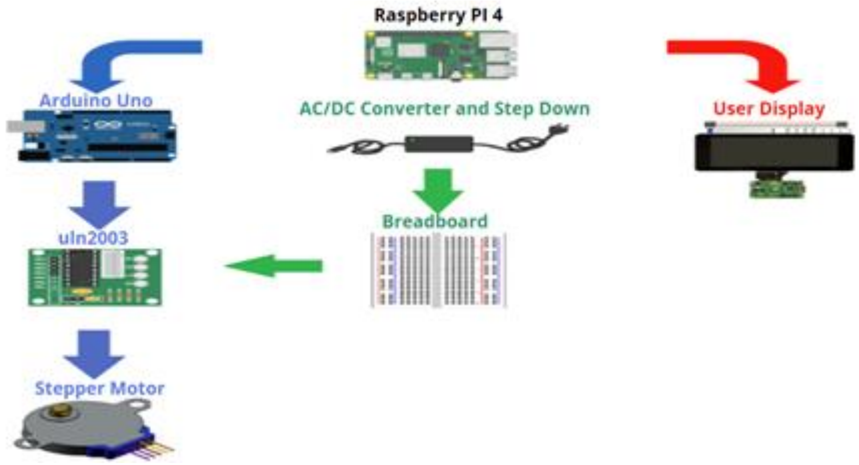


Figure 12: Flow path diagram of electronics system.

Currently, the original processing system contains a Raspberry PI4 as its central unit. As the primary focus for this portion was to create a stand-alone, space confined product, the Raspberry PI4 would function in an identical way to a laptop or desktop counterpart while also maintaining a drastically lower cost at thirty-five dollars. This specific pseudo computer was selected due to its processing power, I/O ports, and its Linux operating system. Its main design features can be seen in Figure 13. This card is where the code for the entire system will be compiled and it is paramount that it has the computing power to perform this action. By running the card off either Raspbian OS or Ubuntu it will be fully capable of installing Arduino's Linux extension of their IDE with no workarounds or limitations. This focal point of the system is what will power both the Arduino as well as the User display.

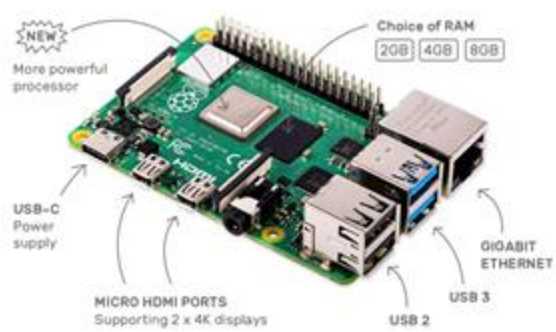


Figure 13: Labeled model of Raspberry PI4.

To avoid creating a large footprint in already limited lab space, the user display is planned to be attached to the exterior structure and allow for touch sensing to limit the use of I/O devices. Although the Raspberry PI4 is capable of responding to both keyboard and mouse plugins, touch screen capabilities add to the seamless user experience and drastically increases portability while decreasing size. The Raspberry PI 7" Touch Display screen can be directly plugged into the Pi's GPIO port for power as well as the ribbon cable can be connected to the DSI port present on the Pi. The adapter board that is included with the display seen in Figure 14 will handle both power and signal conversion allowing for easy plug and play application of the device. This portion of the system is where the user will view and interact with the user interface, updating values within the code as desired.

Figure 14: 7" Raspberry PI touchscreen display with included hardware.

While the C/C++ language does not naturally tend to the straightforward creation of user interfaces, there is a variety of software that exists to remedy this drawback. The most powerful and widely used is MegunoLINK. This software is designed to work directly with the Arduino IDE and allows for drag and drop coding. With built-in capabilities of generating real time graphs and data updates, MegunoLINK would be more than capable of handling the formation of the desired GUI. However, its main drawback is its hefty price tag and while it is capable of forming incredibly complex interfaces, a user interface akin to what is seen in Figure 15 below will suffice.

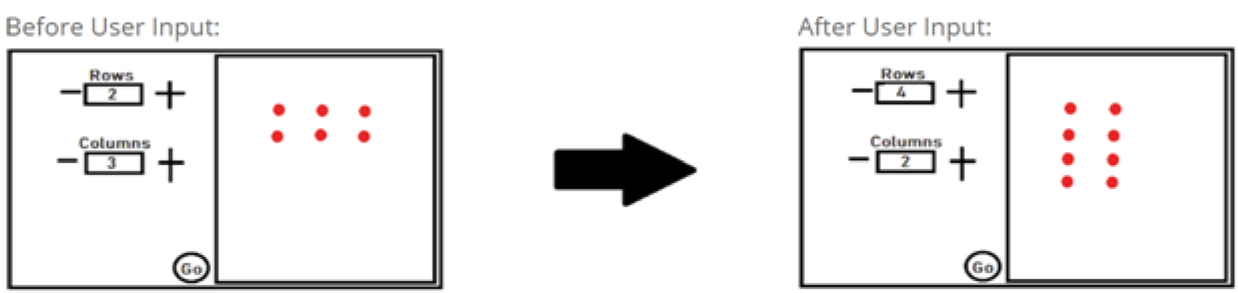


Figure 15: Potential user interface design.

Due to the overall simplicity desired, Processing IDE software is an invaluable asset to achieving the final product. Processing IDE is a free software that is compatible with the Linux operating system run on the Raspberry PI4. Although the code must be entirely handwritten and it does not include the bells and whistles of MegunoLINK, it is possible to create a simple, comprehensible user interface that is capable of updating variable values and executing the code on a button push. This directly aligns with the expected user interface design seen above. If the user were to press either the plus or minus signs next to the row or column box, the value within the box will update accordingly as well as update the amount of steps taken by the stepper motors within the code. Once the user presses the “Go” button, the code will compile and execute on the Arduino Uno board.

The Arduino Uno board is what handles the execution of the code after it is updated by the user and compiled. The board is capable of handling both digital outputs and inputs as well as their analog counterparts. These numbered ports, seen at the top of the board in Figure 16, are what will send the appropriate signal to the distinct stepper motor, telling it how many steps it must make before completion. The Arduino Uno was chosen due to its Linux compatibility as well as its ability to be directly powered by the Raspberry PI4 through a USB A to USB B cable. This means that the board would not require the use of an external power supply, saving both money and space.

Figure 16: Arduino Uno board layout showing pin labels and connections.

As the Arduino Uno executes the code, it will provide digital output to a total of four separate stepper motor driver boards identical to the uln2003 motor module in Figure 17. These modules negate the need for the use of H-Bridges to correctly program the amount of steps taken by each stepper motor. If an H-bridge were to be used, factors such as step delay, phase order, and chip temperature would have to be taken into consideration in the code. By using the modules each motor is assured to receive the proper amount of current for functionality and Arduino IDE’s built-in .step() function can be utilized to step the appropriate amount of steps, similar to the example code provided in Figure 18. Unlike the Arduino, the modules require the implementation of an external power supply to assure they can receive their required current. In general, the digital output and input ports on the Arduino are only capable of powering low current drawing elements such as a piezo buzzer and provide digital signals for larger circuit elements. Because of this an external power supply had to be planned.

Figure 17: Uln2003 motor module with labeled input connections.

```
// Arduino stepper motor control code

#include <Stepper.h> // Include the header file

// change this to the number of steps on your motor
#define STEPS 32

// create an instance of the stepper class using the steps and pins
Stepper stepper(STEPS, 8, 10, 9, 11);

int val = 0;

void setup() {
 Serial.begin(9600);
 stepper.setSpeed(200);
}

void loop() {

 if (Serial.available()>0)
 {
  val = Serial.parseInt();
  stepper.step(val);
  Serial.println(val); //for debugging
 }
```

Figure 18: Example code to properly run a stepper motor via an Arduino.

Harping on the stand-alone element it was decided that the motors would be powered with a laptop charger. By using a laptop charger, the power from a wall outlet can be converted from AC to DC and stepped down to the desired 12V output. To remedy no circuit testing being done within this initial semester, a breadboard has been included in this plan in case the need of additional circuit elements may arise, and it is entirely likely that this component can be transferred to a printed circuit board once the optimal solution is reached.

Contrary to the other subcategories of the overall design, both the hardware and software components saw little overhaul through the initial design process. Rather than being created separate from the hardware chosen by other team members, the system was designed in response to decisions made when tackling their section. This allowed for little alteration and led to drastic additions being made instead. When the group had made the major decision to purchase a DIY

3D printer for construction parts rather than pursuing the build from scratch it had to be decided whether to use the supplied circuitry or the already crafted model. The motherboard included with the single extruder desktop FDM 3D printer provided all the required ports as well as enough built-in motor modules to drive the four total stepper motors that would be used. This motherboard is shown below in Figure 19.

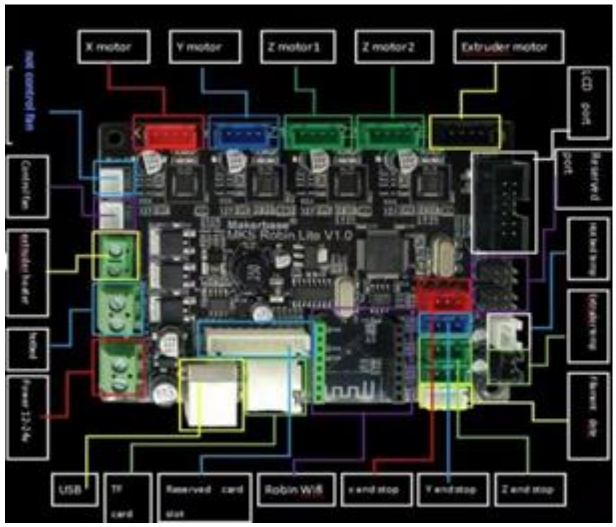


Figure 19: Single Extruder Desktop FDM 3-D Printer included motherboard with labeled input and output ports.

Even though this motherboard contains the required ports to run a 3D printer it is not evident how the group could easily override the board to produce the output goal. This, accompanied with low-cost of the group made alternative, led to the selection of continuing to pursue the above methodology for generating a working system with a graphical user interface.

Supporting Structure and Safety Enclosure

After building the x, y, z coordinate movement system and the dispensing mechanism, there needs to be a supporting structure and covering of the instrument and its processing system. The supporting structure and covering has specific design constraints due to the budget and the small footprint. The small footprint is so the structure can fit in a small area like under the cabinets, not taking up a lot of space in a small laboratory. This design also needs to have a low cost as the total cost for the instrument is to be around 500 dollars. In the original allocation, the supporting structure and covering was estimated to be about 100 dollars. This structure must have a separate area to house the processing system to ensure the electronics will not be damaged. In addition, the enclosure needs to be reliable and durable.

There were three different considered approaches for the supporting structure originally. The first option is the closest option to the final choice. This design is completely covered in acrylic with 80/20 1in x 1in T-slotted single rails as the supports. The top level was going to have a tray with a drip tray that slid in and out for ease of insertion of the slide. Then the bottom was going to be used to house the electronics. There is also a spot to hook on the LCD screen for the GUI. This design on both the top and bottom level was to have double doors. The whole structure itself is a little bigger than 1ft x 1ft. This design is shown below in Figure 20.

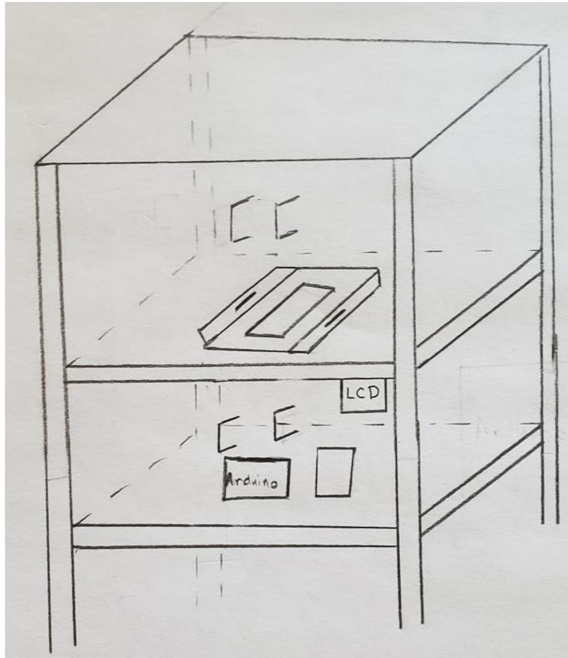


Figure 20: First design idea for the supporting structure.

The second option is the same as the first option but only has one level as shown in Figure 21. It still has the use of the double doors, the placement of the LCD screen, and the tray for the slide. The big difference is where the electronics would be. This option gives no need for the full lower level but there would have to be a separate housing for the processing system on the side.

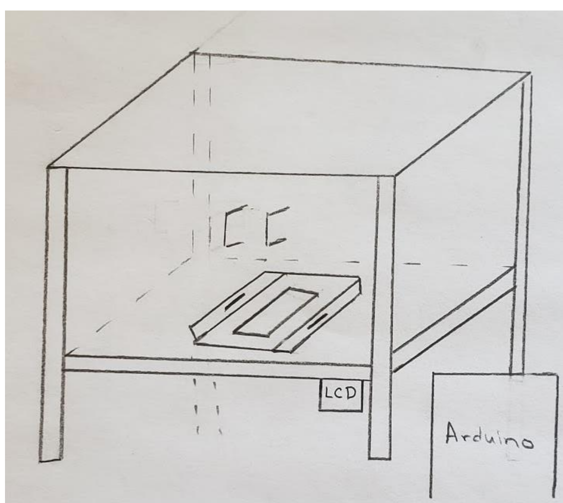


Figure 21: Second design idea for the supporting structure.

The last option is completely different. This is an over-the-top coverage to try to use less material. There would still be a need for the processing system to have a separate housing. This design would still be made out of 80/20 single rails and acrylic with a 3D printed handle. The sketch below in Figure 22 shows this design.

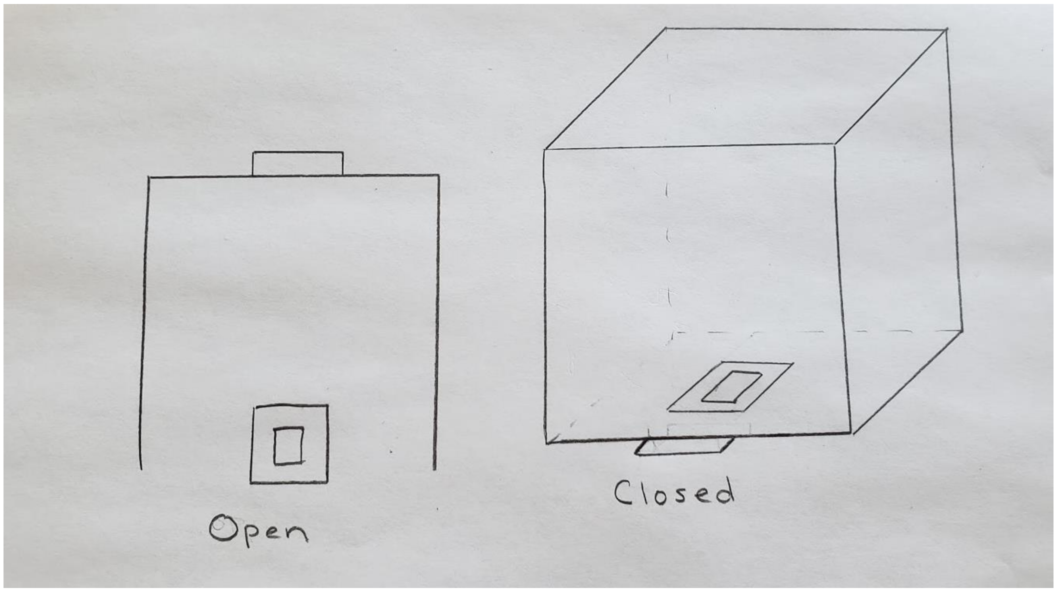


Figure 22: Third design idea for the supporting structure.

The current chosen option was most similar to the first considered approach. With a choice of using a 3D printer for the x, y, z coordinate system, this supporting structure encompasses the use of those parts. This is still subject to change once the 3D printer is received. The supporting structure will be supported by 1in x 1in 80/20 T-slotted single rails using the 3D printer structure as extra support. It will be covered completely in acrylic. These T-slotted rails and acrylic will keep the structure durable. The top level is going to house the electronics. This decision was made so the electronics have no way of getting wet and causing damage to them. This will keep the electronics safe, out of the way and gives the bottom level a lower height. The bottom level will house the x, y, z movement system, a tray for the bio-slide, and the dispensing mechanism. This whole structure was changed to have single doors. The single door will have a magnetic shut to ensure the door is completely closed for safety reasons. This was chosen for the ease of the operator inserting the slides. The operator can open the door with one hand, not have another door in the way, and then put the bio-solution down with the other hand. The use of one door handle for the top and bottom levels also saves money. These handles are going to be 3D printed to save money as well as the corner brackets. The whole structure is also meant for a tabletop so there was no use of the legs. This again saves money because of less materials and it also makes it so it can fit nicely under cabinets and is not extra tall which helps the user. The whole structure is only slightly bigger than the 3D printer to guarantee the small footprint and accessibility of this instrument to smaller laboratories. A SolidWorks model is shown below in Figure 23.

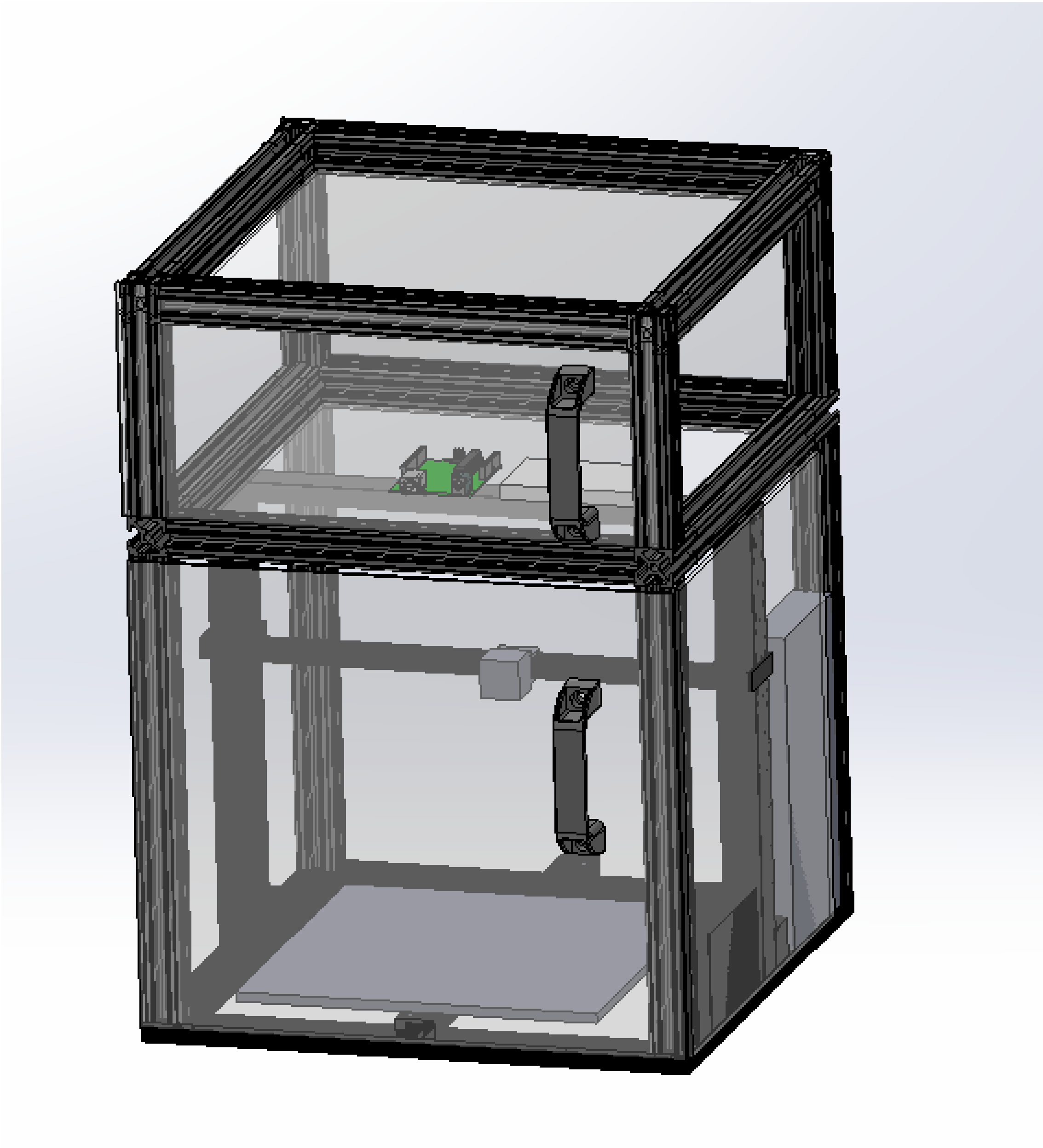


Figure 23: Current design idea for the enclosure including supports from the single extruder desktop 3D printer.

Testing

During the semester, some tests were performed in order to gain information that would improve the overall design of the microarray system. One test determined whether a 3D printed nozzle could improve the performance of the syringe. The other test was used to determine whether or not 3D printed brackets could be used to save money.

Nozzle

The 1 mL syringe that will be used with the design does not have a needle or any sort of tip attached to it. Because of this, the opening on the hub of the syringe has a diameter of about 3mm. This produces droplets that are too large in area to fit into the small array that will be achieved. An image of the hub of the syringe is shown below in Figure 24.

Figure 24: Hub of 1mL syringe.

A nozzle was designed, and 3D printed to fit snugly over the hub of the syringe and restrict the flow of solution to create a droplet with a much smaller footprint. The size of the opening of the nozzle was limited by the resolution of the Monoprice Ultimate 3D printer used. An image of the nozzle attached to the syringe is shown below in Figure 25. In order to verify that the nozzle worked properly to consistently produce smaller droplets than having nothing attached to the syringe, an experiment was conducted.

Figure 25: 3D printed nozzle attached to 1 mL syringe.

Hand-spotting droplets of solution onto a piece of glass, both end types were tested to compare droplet formation. The solution used was a 1:1 ratio of water to skim milk. Multiple attempts were used for both cases to make the best 4x4 array of droplets. Attempts were judged based on how consistent the droplets were in size and shape, how close the droplets were placed together, and overall size of the array. The results of the test are shown in Figure 26 below.

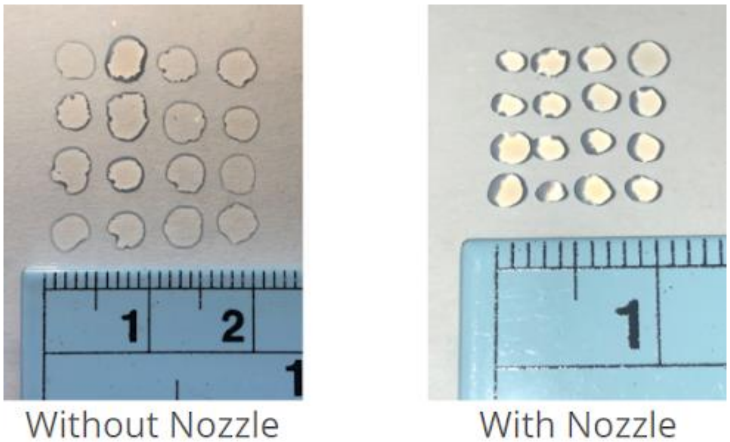


Figure 26: Hand spots on glass without and with nozzle attached.

It is clear from Figure 26 that the nozzle improved the performance of the syringe. The syringe with nothing attached to the hub produced the grid in an area of about 4 cm^2 while the syringe with the nozzle produced the grid in an area of about 1 cm^2. More tests in the future will be used to optimize the nozzle further and compare it with other designs.

Supporting Brackets

The supporting brackets used with 80/20 compatible T-slotted framing cost about 5 dollars for the L-shaped and 10 dollars each for the triangular. For the current supporting structure idea, there is a top and bottom level that are completely covered. This causes there to be a need for 32 brackets which would cost over half of the budget. To save money, 3D printing the brackets is an option. The L-shaped support brackets shown below in Figure 27 were the first brackets to be tested.

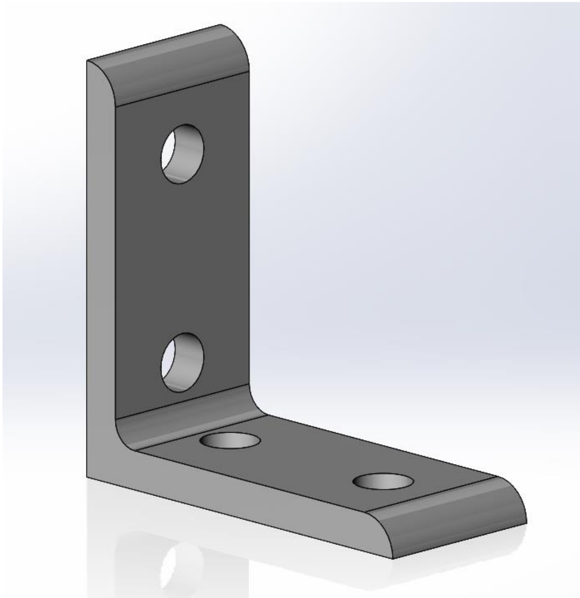


Figure 27: L-shaped support bracket compatible with 80/20 T-slotted rails.

The L-shaped brackets easily broke down the middle where the holes are and cracked at the hinge. Although it was only 20% infill, a decision to go with the triangular support was made so the structure would be more durable. The triangular support is shown below in Figure 28.

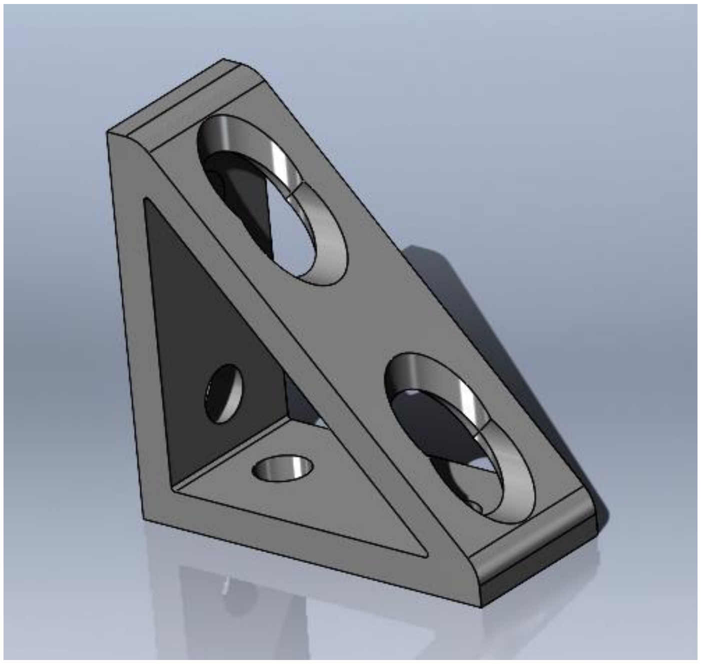


Figure 28: Triangular support bracket compatible with 80/20 T-slotted rails.

These triangular support brackets were printed in ABS. The first bracket was printed with a 20% infill and was printed on its side. In this first iteration, the holes were printed with it. This test showed that more infill was needed and that its weak points are at the holes due to how it broke when given force as seen in Figure 29.

Figure 29: Photo of a 3D printed bracket with a 20% infill with printed holes. Note that this bracket split down the middle where the holes were.

The next bracket that was printed has a 100% infill and included filling in the holes as seen in Figure 30. This produces a stronger support that is durable and reliable for this structure. The plan is to drill the holes in the 3D printed part to keep as much material as possible to ensure the strength of the part. The current problem is that this print took 3 hours so it will take 96 hours over the course of next semester to print these with a 100% infill. This bracket also still needs to be tested after drilling holes into it.

Figure 30: Photo of a 3D printed bracket with a 100% Infill with no printed holes. Note that the holes will be drilled through the bracket.

Currently, based on the budget, it seems like it is necessary to 3D print these brackets. As of now, the brackets appear durable but will be tested more when the 3D printer and other materials are purchased. Testing infills and other aspects of these support brackets will continue next semester.

Technical Plan

Based on the original Gantt chart seen in Appendix A, the group was on track this semester. The research and design aspects of the project were completed and purchasing materials is going to be looked over during winter break. Therefore, a 3D printer will be available to modify when next semester starts in February. The budget amounted to 496 dollars which is below the 500 dollar goal. The approved budget can be seen in Appendix B. Notice that money was saved through using readily available parts and 3D printed parts. For next semester, the main plan is to test and fabricate the final design after purchasing all materials. The future plans can also be seen in Appendix A. Below are the future plans for each individual part of the project.

GUI and Processing System

Since the creation of the circuit powering the entire system is largely dependent on components chosen and required by the other portions of the project, the initial step will be to assess the parts provided in the DIY 3D printer and add circuit elements accordingly. If it is possible to salvage the provided motherboard then that would negate the need for the majority of the planned processing system. However, this will most likely not be the case and the devised processing system provides a sound backup option. Once all the electronic components receive proper power and inputs, machine function code will be created before pursuing a GUI to make sure the group can get the automated microarray system to function as intended. The final stage will be to form an interface for the user that allows for input.

X, Y, Z Coordinate Movement System

As the next semester approaches, the very first thing that needs to be done is to buy the 3D printer kit. After doing so, there will be a better understanding on the next steps necessary to begin assembly of the movement system. Once this movement system is assembled, the other parts can be integrated into the design such as the GUI processing, dispensing mechanism, and the supporting structure. Acquisition of the printer as soon as possible will ensure that there is enough time to complete assembly of the system.

Dispensing Mechanism

Next semester, the first step for the dispensing mechanism will be 3D printing the plastic parts assembling everything for the first time. Once this is complete, it will be tested to see if there are any issues with the design or if there are any improvements that can be made. Once the mechanism is working properly and has been optimized for the best performance possible, it will be attached to the x, y, z coordinate movement system and be further tested. The 3D printed parts can be adjusted and remade quickly which will help speed up the design process.

Supporting Structure and Safety Enclosure

At the start of next semester, the first step is to examine the parts of the 3D printer structure to get a better understanding for the amount of material needed. If it is possible to efficiently use the structure from the 3D printer, this can negate the need for more 80/20 and brackets. Having the printer in hand will give the accurate dimensions needed for the supporting structure. Once this is finalized in SolidWorks, purchasing of the necessary amounts of acrylic, 80/20 T-slotted single rails, and screws and nuts are needed. The brackets and door handles will need to be 3D printed. At the beginning of next semester, the brackets will continue to be tested to ensure the durability of the structure. Finally, everything will need to be put together and the enclosure will be built around the instrument.

Conclusion

This design will make microarray systems more affordable and accessible. The path taken to achieve the goal was done so by careful consideration and diligent testing. Multiple solutions for each part were developed and analyzed in order to produce the best design possible. The testing was used to optimize droplet size and lower the budget through 3D printing bracket parts rather than purchasing them. The designs and planning done in the fall ensure that it can be successfully built in the spring on time and within budget.

References

Badami, V. (2019, May 19). A tour of the Arduino UNO board. Retrieved December 07, 2020, from https://www.hackerearth.com/blog/developers/a-tour-of-the-arduino-uno-board/

Biocompare. (2017). Microarray Equipment. Retrieved December 09, 2020, from https://www.biocompare.com/Lab-Equipment/6580-Microarray-Equipment/

CDC. (2020, November 03). Using Antibody Tests for COVID-19. Retrieved December 08, 2020, from https://www.cdc.gov/coronavirus/2019-ncov/lab/resources/antibody-tests.html

Last Minute Engineers. (2019, December 22). In-Depth: Control 28BYJ-48 Stepper Motor with ULN2003 Driver & Arduino. Retrieved December 07, 2020, from https://lastminuteengineers.com/28byj48-stepper-motor-arduino-tutorial/

M2-Automation. (2020). InstrumentZERO. Retrieved December 07, 2020, from https://www.m2-automation.com/izero.html

Raj, A. (2017, August 15). Arduino Stepper Motor Tutorial - Interfacing 28-BYJ48 Stepper Motor with Arduino Uno. Retrieved December 07, 2020, from https://circuitdigest.com/microcontroller-projects/arduino-stepper-motor-control-tutorial

Raspberry Pi. (2020). Buy a Raspberry Pi 4 Model B. Retrieved December 07, 2020, from https://www.raspberrypi.org/products/raspberry-pi-4-model-b/?resellerType=home

Raspberry Pi. (2020). Buy a Raspberry Pi Touch Display. Retrieved December 07, 2020, from https://www.raspberrypi.org/products/raspberry-pi-touch-display/?resellerType=home

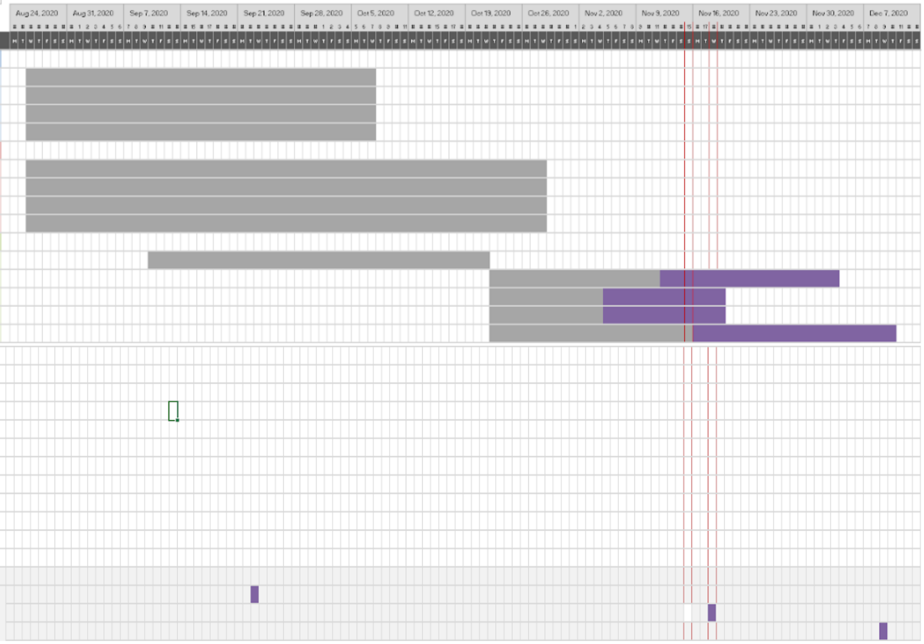
Appendix A: Schedule

Automated MicroArray System

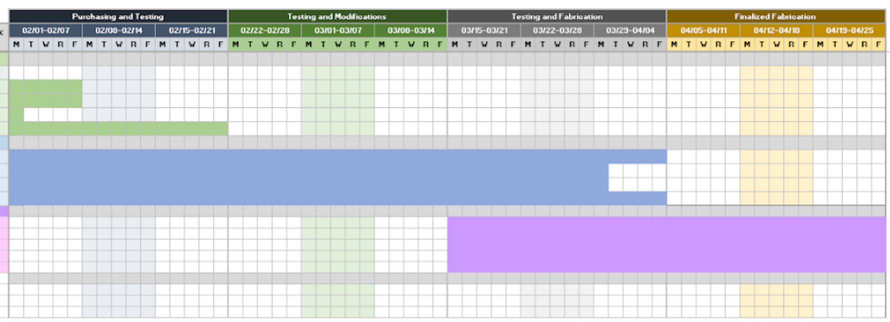
Students: Miranda Griffith, Michael D'Arpino, Michael Sherman, and John Handy
Professor: Marn

Project Start: 8/26/2020
Display Week: 1

TASK	ASSIGNED TO	PROGRESS	START	END
Research				
GUI and Processing System	Michael D'Arpino	100%	8/26/20	10/7/20
Nozzle	Michael Sherman	100%	8/26/20	10/7/20
X/Y Coordinate Movement	John Handy	100%	8/26/20	10/7/20
Supporting Structure and Covering	Miranda Griffith	100%	8/26/20	10/7/20
Design Ideas				
GUI and Processing System	Michael D'Arpino	100%	8/26/20	10/28/20
Nozzle/Z coordinate Movement	Michael Sherman	100%	8/26/20	10/28/20
X/Y Coordinate Movement	John Handy	100%	8/26/20	10/28/20
Supporting Structure and Covering	Miranda Griffith	100%	8/26/20	10/28/20
Budget and Purchasing of Materials				
Completed Budget	Miranda Griffith	100%	9/10/20	10/21/20
GUI and Processing System	Michael D'Arpino	50%	10/22/20	12/3/20
Nozzle/Z coordinate Movement	Michael Sherman	50%	10/22/20	11/19/20
X/Y Coordinate Movement	John Handy	50%	10/22/20	11/19/20
Supporting Structure and Covering	Miranda Griffith	50%	10/22/20	12/10/20
Testing				
GUI and Processing System	Michael D'Arpino	0%	2/1/20	3/30/20
Nozzle/Z coordinate Movement	Michael Sherman	0%	2/1/20	3/30/20
X/Y Coordinate Movement	John Handy	0%	2/1/20	3/30/20
Supporting Structure and Covering	Miranda Griffith	0%	2/1/20	3/30/20
Fabrication of Design				
GUI and Processing System	Michael D'Arpino	0%	2/1/20	4/13/20
Nozzle/Z coordinate Movement	Michael Sherman	0%	2/1/20	4/13/20
X/Y Coordinate Movement	John Handy	0%	2/1/20	4/13/20
Supporting Structure and Covering	Miranda Griffith	0%	2/1/20	4/13/20
Finalized Design	Everyone	0%	2/1/20	4/13/20
Other Important Dates				
Presentation 1			9/23/20	9/23/20
Presentation 2			11/18/20	11/18/20
Milestone Report			12/9/20	12/9/20



TASK TITLE	TASK OWNER	START DATE	DUE DATE	PCT OF TASK COMPLETE
Budget and Purchasing of Materials				
Completed Budget	Miranda Griffith	09/10/20	11/16/20	100%
GUI and Processing System	Michael D'Arpino	10/22/20	02/07/21	50%
Nozzle	Michael Sherman	10/22/20	02/07/21	50%
XYZ Coordinate Movement	John Handy	10/22/20	02/01/20	50%
Supporting Structure and Covering	Miranda Griffith	10/22/20	02/21/21	50%
Testing				
GUI and Processing System	Michael D'Arpino	02/01/21	04/04/21	0%
Nozzle	Michael Sherman	02/01/21	03/28/21	40%
XYZ Coordinate Movement	John Handy	02/01/21	03/28/21	0%
Supporting Structure and Covering	Miranda Griffith	02/01/21	04/04/21	20%
Fabrication of Design				
GUI and Processing System	Michael D'Arpino	03/15/21	04/25/21	0%
Nozzle	Michael Sherman	03/15/21	04/25/21	0%
XYZ Coordinate Movement	John Handy	03/15/21	04/25/21	0%
Supporting Structure and Covering	Miranda Griffith	03/15/21	04/25/21	0%
Finalized Design	Everyone	03/15/21	04/25/21	0%
Other Important Dates				
Presentation 1		9/23/20	9/23/20	
Presentation 2		11/18/20	11/18/20	
Milestone Report		12/9/20	12/9/20	



Appendix B: Budget

I MATERIALS AND PARTS:				
	ITEM	QUANTITY	UNIT PRICE	TOTAL PRICE
1	Lead Screw	1	$4.15	$4.15
2	Guilding Rods	2	$12.08	$24.16
3	Small Stepper	1	$7.95	$7.95
4	Syringes	1	$6.89	$6.89
5	Coupler	1	$4.99	$4.99
6	Nut	1	$2.14	$2.14
7	3D Printer Filament	2	$20.99	$41.98
8	Syringe Holder (3D Print)	1	$0.00	$0.00
9	XYZ DIY 3D Printer	1	$135.00	$135.00
10	80/20 Single Rails (10ft)	1	$30.54	$30.54
11	Brackets (3D Print)	16	$0.00	$0.00
12	Door Handles (3D Print)	2	$0.00	$0.00
13	Screws and Nuts	1	$29.60	$29.60
14	Acyrlic Sheets (48"x48"x1/8")	1	$81.34	$81.34
15	Stepper Motor Driver Board	4	$1.99	$7.96
16	Breadboard	1	$0.00	$0.00
17	Pack of Wires	1	$0.00	$0.00
18	Arduino	1	$0.00	$0.00
19	Raspberry Pi	1	$29.99	$29.99
20	Raspberry Pi Touch Screen	1	$60.00	$60.00
21	Laptop Charger for power	1	$29.70	$29.70
			TOTAL	$496.39