Final Report

Lucien Eckert
Cornell University
Ithaca, U.S.
lee48@cornell.edu

Hedy Yang
Cornell University
Ithaca, U.S.
hy378@cornell.edu

Willem Fargo
Cornell University
Ithaca, U.S.
wf74@cornell.edu

INTRODUCTION

Our project investigated an implementation of a latte art robot that focuses on creating art and maintaining the created art – the idea being that a user could use the machine to draw an art pattern on the top of their coffee, and have the machine replenish the pattern between sips. As optional features, we also hope to design a device that can work with a variety of coffee cups – overcoming the single-mug shortcoming prevalent across many existing designs – and allows the user to select a design through a simple LCD-screen interface.

RELATED WORK

Latte Art Printing Machine [1]

This machine has a frame typical of a 3D printer; an arm is controlled on the 2D plane, and raised/lowered as needed. The art is drawn using colored syrups, rather than foamed milk like most latte art. This project doesn't support multiple cup sizes. The small mug is propped up with cardboard in the video, likely because there's a single optimal height. There's also no way to measure the liquid level in the cup, so our design could improve upon this aspect.

Text-Enabled Espresso Machine [2]

This product automatically prepares coffee for users based on the text message that users send to the machine. It uses foam to print the last three digits of users' phone numbers on the top of coffee to distinguish different orders.

Turn Inkjet Printer to Print on Coffee [3]

This project replaces the normal ink in an inkjet printer with edible ink to print patterns on coffee. It inspires us to take advantage of the current technology and modify them so they can work for our goals. One con of this product is that the pattern is created with edible ink instead of foam, resulting in less delicious flavor. Also, it limits the options of pattern and the size of cups.

WiibooxSweetin Coffee Printer [4]

Rather than using the construction of a 3D printer, this product looks more like just a coffee or espresso machine. It moves the cup around rather than using an arm to draw the pattern. Users direct what pattern to draw either with their phone (through WiFi) or choosing a preset pattern on the device's touch screen. It also uses some sort of food dye to print the pattern, rather than foamed milk used in traditional latte art. This means it may not be able to print well on cups of black coffee, or nonstandard liquids.

Costa Coffee BaristaBot [5]

This product produces drinks with aesthetically-pleasing designs formed with frothed milk. The machine functions by moving the cup around, rather than moving an arm around the cup to operate. It also has a huge form factor, taking up the same space as about three standard vending machines. The user interfaces with the machine using a touchscreen, which seems to be a common mode of input for this type of device that aims to inspire a high-end feeling.

Ripple Maker II [6]

This product is designed to embed patterns on the top of various drinks (from coffee to cocktails) by depositing precise drops of "ripple pods," edible dyes developed by the company, onto foamed drinks. These drops can be flavored, or simply aesthetic. The movement of this device is a midpoint between the arm and cup-based systems some of the above devices use – the drink is raised into a box, where it is then operated on with a combination of cup movement and a printer arm scanning over it.

Colorato Coffee Printer [7]

This product is a good comparison to the Ripple Maker mentioned above. The main difference is being able to upload selfies on the Coffee Colorato, whereas one can only select patterns from a library on The Ripple. Additionally, the Coffee Colorato is designed for businesses to implement, whereas the Ripple and other similar products seem to be geared towards consumer use in their home kitchen.

EVEBOT [8]

This product has all the features of the commercial coffee printers mentioned above. What's more, it supports various sizes of cups. Users can select the size of the image by using a slide bar or pressing the small, medium, large button. There is a laser sensor inside the machine to detect the height of the cup, but it cannot read the height of clear glasses. So when using clear glasses, the drink has to be filled to the rim of the glass for the laser to pick it up.

How Water Level Sensor Works and Interface it with Arduino [9]

The article was included because one of the design aspects that we are considering is being able to redo the pattern at any point in your beverage experience (e.x. when the coffee is halfway gone). We need a way to accurately tell the machine where the level of the liquid in the cup is. This sensor is placed within the liquid and basically acts as a variable resistor, with the amount of resistance increasing as the water level drops. The water connects sense and power traces on a board, which leads to the increase in resistance.

Peristaltic Liquid Pump with Silicone Tubing [10]
The Peristaltic Liquid Pump with Silicone Tubing is a product that we found on Adafruit. Its more recent product that they currently sell runs on 12V DC power and I would assume it works with the microcontroller we use in class. The DC motor inside the pump uses a 'clover' pattern of rollers to press the fluid through the silicone tubing. Similar to the motor we just learned how to control in class, one can utilize PWM to speed up or slow down the motor inside the pump, and consequently the flow rate. we would imagine this product could be used inside the printing head on the arm of our 3D printer-style design to dispense the foamed milk or other liquids.

DESIGN

Functional block 1: Movement system for drawing arm

The system that controls the drawing arm, from which the foamed milk is extruded onto the coffee, is designed to allow movement on two dimensions. This will allow the device enough movement to trace any 2D pattern, using two Nema 8 stepper motors to drive timing belts affixed to plastic mounts. These motors were chosen due to the light load of the drawing arm, and their small form factor allowing for as much space as possible within the frame to be dedicated to movement. Our initial design also envisioned using a third motor to raise and lower the drawing tip, allowing for material to be extruded directly at the top of the liquid, allowing for greater precision when drawing patterns.

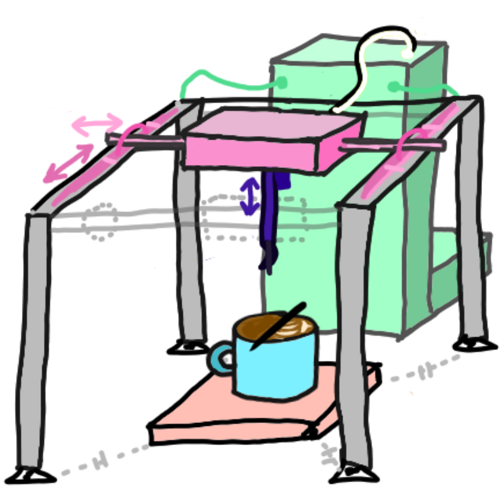

Figure 1. Revised sketch of our first feature. Most important to this feature, the drawing arm (purple) and arm mount (pink) are depicted. The mug is split into two halves, with the left and right showing the before and after states respectively.

Functional block 2: Pump and valve for controlling drawing material

In our initial design, we were far too ambitious and wanted one our design goals was to allow the system to detect that the liquid level has decreased and adjust the height of the dispenser head accordingly. Figures below depict our initial sketches for that feature. The feature was ultimately dropped as we decided to utilize only two axes of movement, X and Y, for simplicity's sake. Regardless, the sketches show the implementation of silicone tubing, a pump motor, and a nozzle (which in later iterations includes the solenoid valve component).

The revised sketch shows the final implementation of the pump and valve system. Liquid is drawn into the pump from a reservoir, pushed through the silicone tubing to the top-side of the solenoid valve, the solenoid valve opens and the liquid flows through, and then the liquid is accurately directed via the nozzle. Simplifying the liquid extrusion function to not account for cup or water height allowed us to focus on the precision and overall performance, measured by the clarity of the pattern.

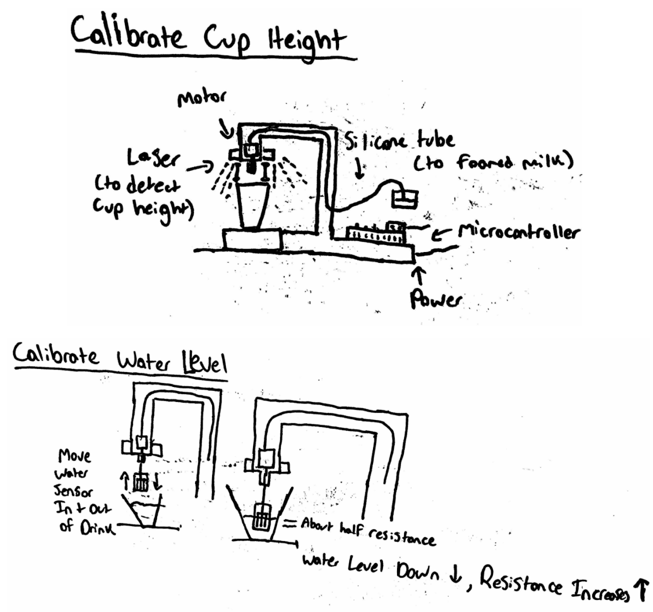

Figure 2. Iterations of planned liquid-level calibration methods. This component of the functional block was eventually scrapped.

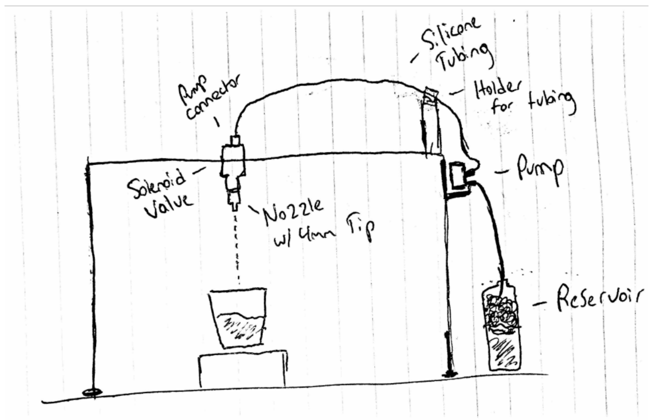

Figure 3. Revised sketch showing function between pump and valve, liquid extrusion.

Functional block 3: User interface for selecting pattern to draw

Our original goal of this feature is to allow users to customize the patterns they wish to print on the top of their drinks. A control screen will be connected to the microcontroller of the main body of the machine. Users will pick the pattern and its size on this screen, and once the selection is finished, the product will start drawing. A few simple patterns will be stored locally for users to print at a fixed size. If time allows, our product will also allow users to upload images and select a size to match the size of their cup before they print.

After further discussion, we decided to make some changes on this feature in order to focus on the hardware. Instead of providing many pattern options, our product will only support printing one specific pattern in the early phase. In the later phase of development, we can add a display screen on the frame to display the pattern that is being drawn. We also update the outlook of the machine to make it consistent across the three features.

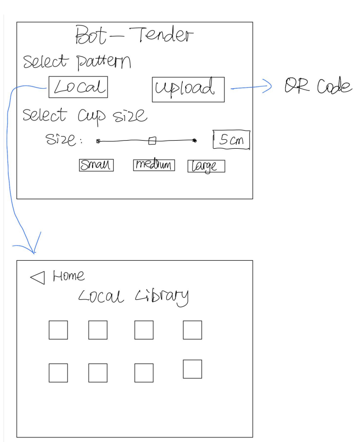

Figure 4. Initial sketch of feature 3. UX design for the touchscreen interface. Touchscreen was too expensive for this prototype so we ended up using a 8x8 LED screen.

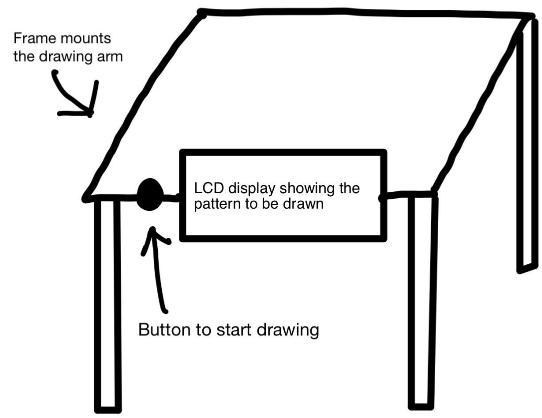

Figure 5. Revised sketch of feature 3. The outlook of this latte art machine is consistent with the other two features.

IMPLEMENTATION

Functional block 1: Movement system for drawing arm

We primarily structured our movement system with 3D printed components, either mounted to the device's frame or resting along cylindrical rods. Since these rods couldn't be cut down, we used rubber bands to apply tension and prevent them from sliding around. Ball-bearing platforms were used in some cases where movement would otherwise not be smooth enough for precise control.

The primary difference between our design and final prototype was the removal of the mechanism raising and lowering the drawing arm. This was done to reduce cost and complexity, and allow greater focus on precise 2D movement.
Also differing from our initial design, the movement system in our final prototype is driven by Nema 17 stepper motors. This was necessary due to the DRV8825 stepper module we used to control the motors, as the minimum voltage requirement for the module was much higher than the voltage rating for the Nema 8 motors. This caused a redesign of the plastic mounts, and reduced the size of our drawing area. Each motor was then driven with its own stepper module, allowing for totally independent control.

Finally, the position and orientation of the timing belts was also redesigned after the belts affixed to our initial prototypes were unstable. Learning from other groups' prototypes, we made one of the belts rotate along the same plane that the associated carriage would move.

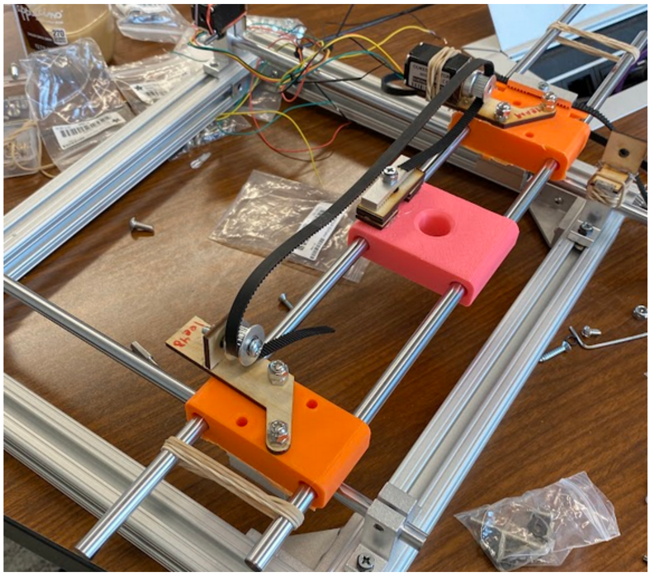

Figure 6. The initial prototype's movement system is affixed to the top of the frame. Showcases the smaller Nema 8 motors, and unstable mounts that would be revised before the final prototype.

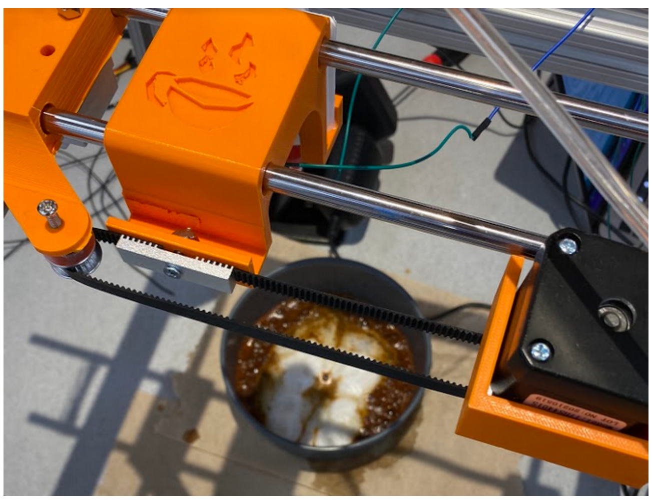

Figure 7. Depiction of revised timing belt position for one axis, directly attached to the drawing arm. The larger Nema 17 motor mounts are visible.

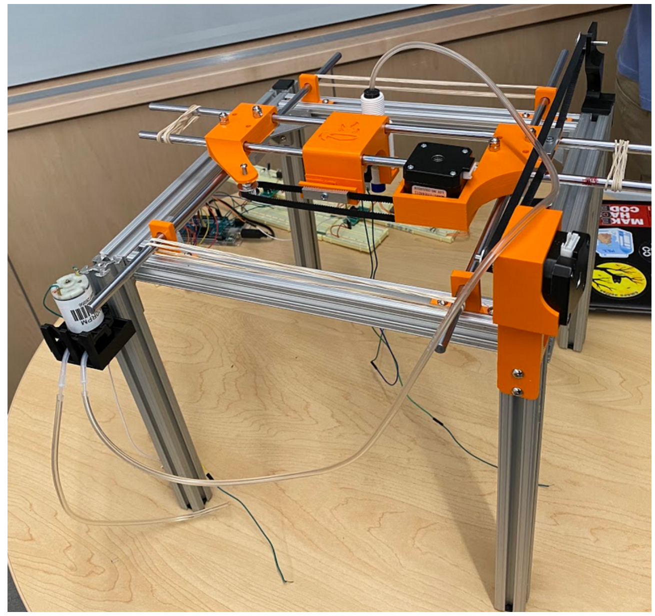

Figure 8. The final integration of the movement system and the pumping/valve system, our first two functional blocks.

Functional block 2: Pump and valve for controlling drawing material

As previously mentioned, we dropped the need to account for the water level and cup height. We utilized the same cup throughout the iteration to keep the height the same. The water level was not as much of a factor in the clarity of the pattern as we had initially expected. The nozzle did not actually need to move closer to account for a few inches less liquid. One could certainly run the pattern multiple times into the same cup and liquid.

Multiple variations of the nozzle component were tested. We found that a longer nozzle did not provide enough benefit to the accuracy to counteract the stability loss. We found a middle-ground in terms of nozzle length, as can be seen in the figures below. Additionally, we had issues with liquid overflowing from the top of the solenoid valve. To counteract this, we both sealed tubing into the 3D printed connector with hot glue and made more room on the inside of the connector and nozzle to give the valve more time to catch up to the pump.

From a code perspective, the biggest challenge for this functional block was to find the correct amount of power to give the pump and the timing between the pump and the valve opening and closing.

Initially we thought that the pump could run continuously and the valve would simply open when liquid needed to be dispensed. We found that this caused far too much pressure inside the valve and that the pump could be controlled accurately enough to stop the flow of liquid when needed. Theoretically, we could have just used the pump, as there was minimal extra liquid being dispensed after the pump stopped. The valve is truly only there to stop the small amount that may leak out.

Therefore, we landed on opening the valve for the entire pattern extraction process and only closing it when the pattern had been fully realized. We used full power on the pump as too little would allow for bubbles to form on the top of the nozzle.

These bubbles on the nozzle tip were another one of our significant challenges. With too wide of a nozzle tip diameter, bubbles would form easily and interrupt the flow of liquid. Therefore, we settled on a nozzle tip with a 4mm diameter to ensure accurate liquid dispersion and reduce the chance of bubbles forming.

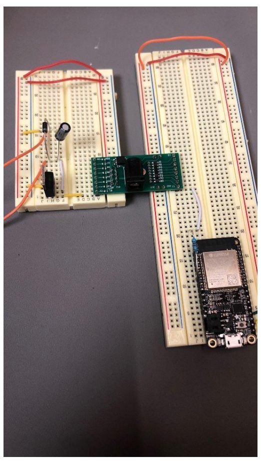

Figure 9. The circuit used for the peristaltic pump and solenoid valve. PWM control (one direction) is needed for the pump and analog control for the valve, which can both be provided by the pins utilized.

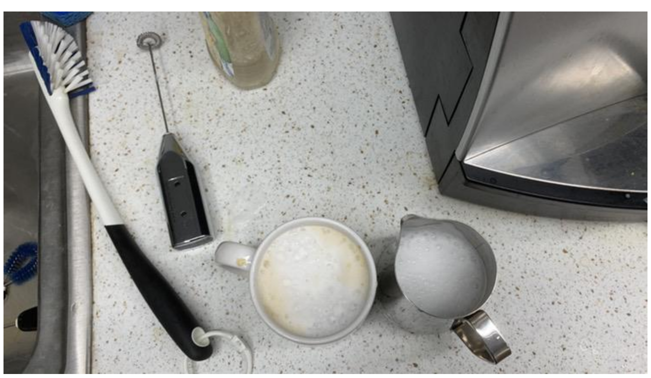

Figure 10. Testing liquid interaction by hand. We found soy sauce and soapy water foam was an acceptable alternative to foamed milk and coffee. We also found that pouring liquid from a higher position would cause bigger bubbles in the mug.

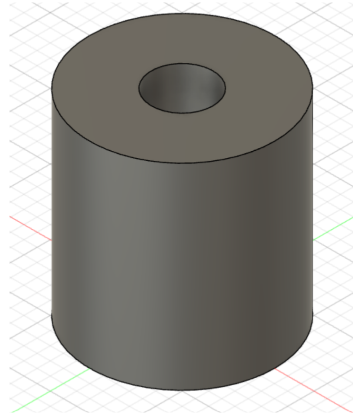

Figure 11. 3D model for pump side connector on valve. (Initial)

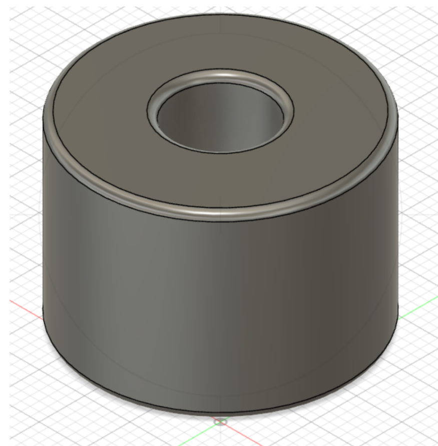

Figure 12. 3D model for pump side connector on valve. (Final)

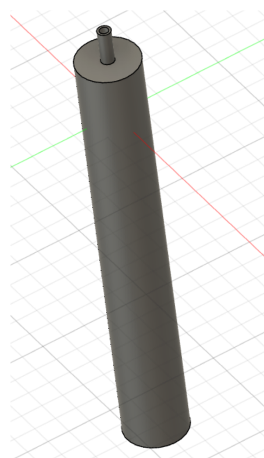

Figure 13. Various iterations of the nozzle were printed and tested. Length and nozzle tip diameters were adjusted. Four different nozzles included: Short Length w/ 2mm Tip, Long Length w/ 2mm Tip, Short Length w/ 4mm Tip, Long Length w/ 4mm Tip.

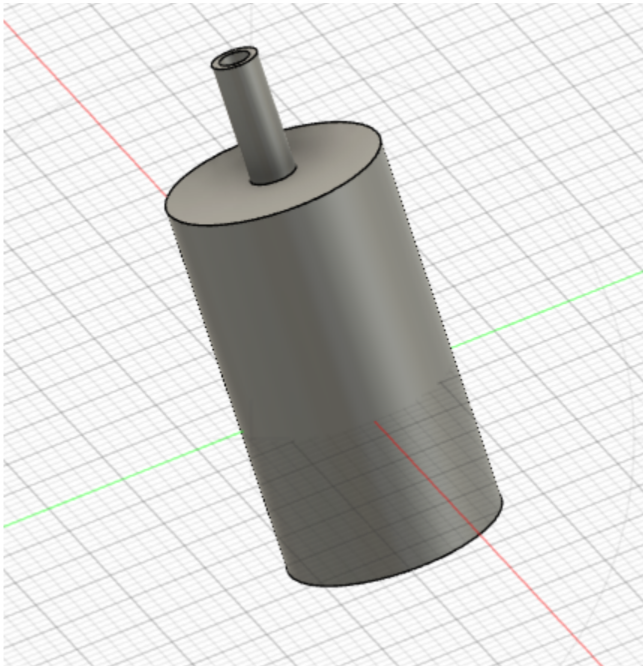

Figure 14. 3D model for nozzle on solenoid valve. (Final)

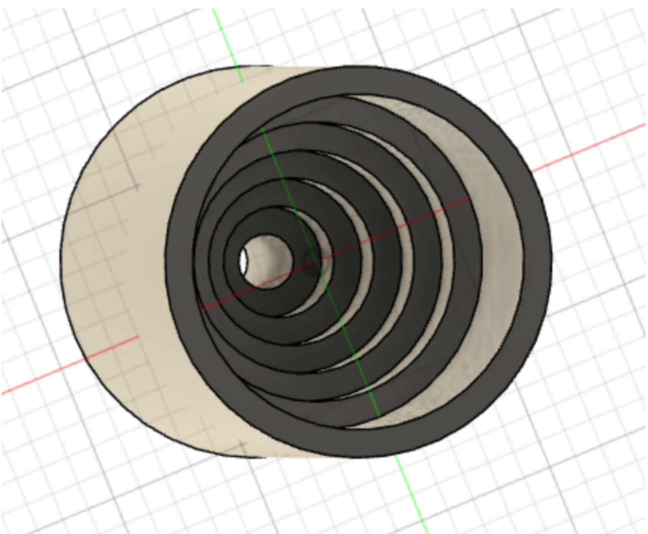

Figure 15. The interior of the nozzle was modified after testing to allow more space for liquid in order to reduce backfill and overflow from the top of the valve.

Figure 16. Connectors for both sides of the solenoid valve. Pump side to interface with tubing, bottom side is initial version of nozzle for liquid extrusion

Figure 17. 3D printed piece to hold the solenoid valve in place at the center of the drawing arms.

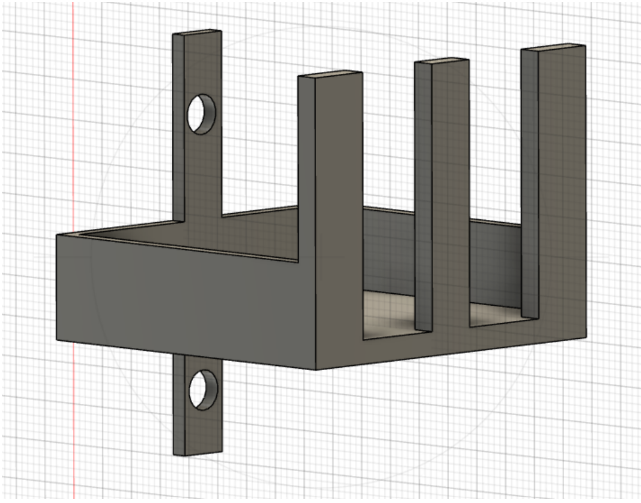

Figure 18. Holder for peristaltic pump. Connects to extrusion bars of frame.

Functional block 3: User interface for selecting pattern to draw

The final implementation of this feature is almost the same as our design. Five patterns were provided in the LED display, and users could use a joystick to switch between different patterns by moving it left or right. Once the joystick button is clicked, the current pattern displayed on the screen is selected, and the other functional blocks will be triggered to start the drawing process. The only minor difference between the final implementation and the initial design is the location of the joystick. In the initial design, we put it on the frame next to the display screen. In the final implementation, we discovered that it would be easier for users to control the joystick if they could hold it in their hands, so we didn't fix the joystick to the frame.

In our initial prototype, we only provided two patterns: a smiley face and crying face. the LED display was very vague due to some coding issue, as seen in the picture below. The patterns were stored in 2-dimensional arrays, and we hardcoded how to display two patterns in LED on Arduino, making the code redundant.

In the final presentation, we successfully debugged the LED display issue to make it much brighter. We also improved the code reusability. Instead of storing patterns in multiple 2-dimensional arrays, we store patterns in a single 3-dimensional array. The index of each pattern in this 3d array is used as its identification number. We only have one block of LED display code, and this block of code is reused for every available pattern. Because of this improvement, adding new patterns becomes much easier, and we were able to increase the number of available patterns to five.

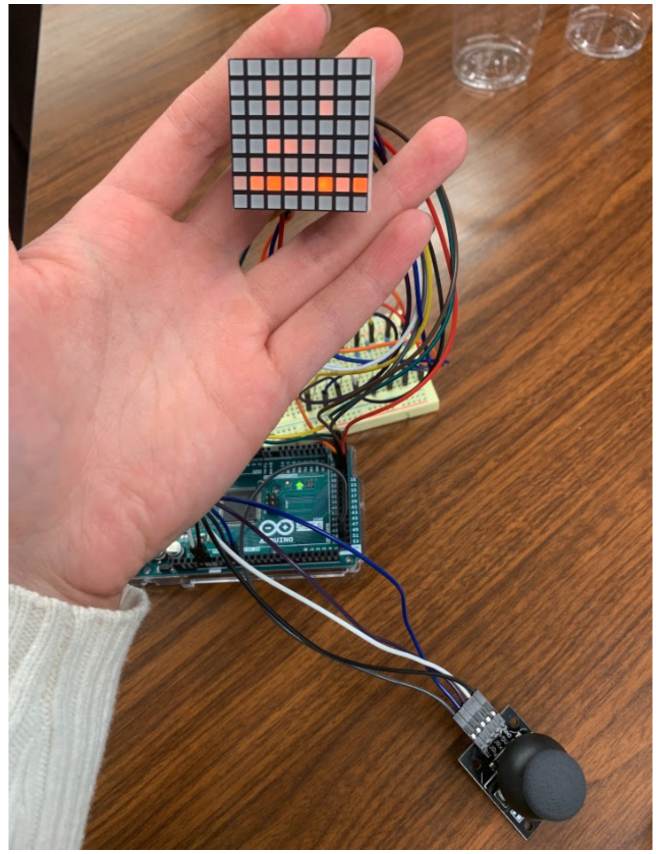

Figure 19. Initial prototype for functional block 3. A crying pattern is displayed on the initial prototype. The LED lights are very dim. Users can move the joystick left or right to select different patterns.

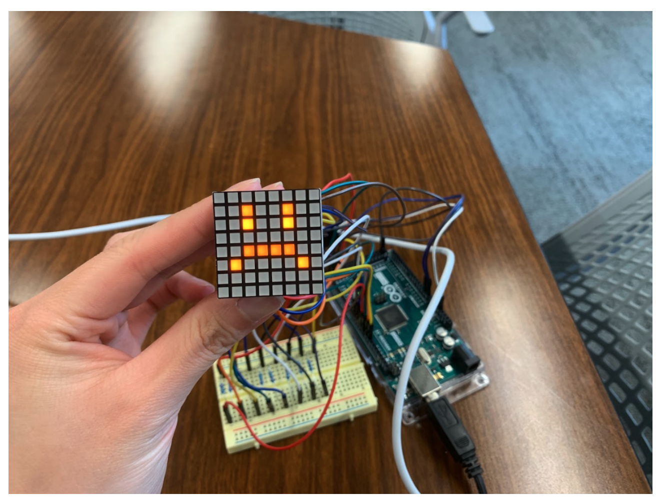

Figure 20. Revised LED display. The LED brightness has been improved, and the crying pattern can be easily distinguished.

LIMITATIONS AND FUTURE WORK

One major limitation of our prototype was the inability to raise or lower the drawing arm. This meant our drawings were less precise, as droplets of the drawing material would gain speed as they fell onto the liquid's surface, causing a greater spreading of the material than we expected. This reduced the precision of our pattern drawing. Finding a way to mount a servo to the drawing arm to allow for that would be a good step for future work in order to produce more precise pattern drawing.

Another limitation we ran into had to do with materials. To avoid the cost and mess of using actual milk and espresso, we used soap and soy sauce respectively as replacements for our demo. This worked well visually, but had the unexpected downside of causing the soap to form bubbles at the tip of our drawing arm. This caused it to be extruded inconsistently. While this problem was present during our demo, we did not run into it when using actual foamed milk with our prototype. For both soap water and milk, we were facing the need to constantly replenish the material. We had to manually froth the milk or soap water to make it good to draw. It would be necessary to include another functional block for future work to automatically froth the milk and get the material ready to draw.

For the function of the liquid extrusion components (pump and valve), 12V power was needed for both motors. Both motors could not be driven off the same 12V power source and consequently we needed to use two isolators and three breadboards, two with the circuits pictured above and the last as a medium between the Arduino MEGA and the isolators. In addition to the stepper motors needing to utilize 9V power, we ended up with too many breadboards and a mess of wires in the end. In future iterations, we would seek to consolidate the breadboards and employ better cable management. The bundle of cables sometimes caused things to disconnect and slowed down our progress at times. Cable sleeves may be an effective solution.

One final future step to take could be adding a spinning platform below the cup being drawn on. This would allow investigation of more common "swirly" latte art patterns, which our grid-based pattern encoding system did not support.

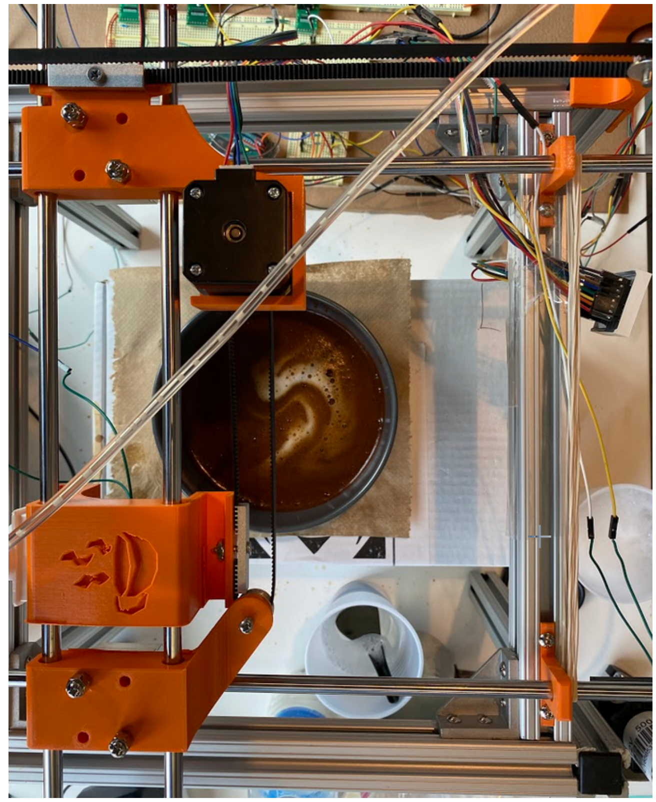

Figure 21. A "swirled" latte art, produced accidentally after the machine drew a pattern and the cup was rotated.

VIDEOS

Initial Pump Function:
https://youtube.com/shorts/kagEkd6z6JA?feature=share

Pump Function w/ Water:
https://youtube.com/shorts/V_YLSIhLaww

Pump and Valve on Arduino MEGA:
https://youtube.com/shorts/xG8DeWtePMg?feature=share

Final Presentation
https://drive.google.com/file/d/163IOMc9wnN7eKQdaCrfHgEu2UStIq4OC/view?usp=sharing

REFERENCES

1. Jamogo_999. Latte Art Printing Machine. Retrieved February 20, 2022 from https://www.instructables.com/Latte-Art-Printing-Machine/
2. Mark Frauenfelder. Text-Enabled Espresso Machine Prints Your Phone Number on the Foam. 2012. Retrieved February 20, 2022 from https://makezine.com/2012/05/06/text-enabled-espresso-machine-prints-your-phone-number-on-the-foam/

3. Baffouri. How to Turn Inkjet Printer to Print on Coffee. Retrieved February 20, 2022 from https://www.instructables.com/How-to-turn-inkjet-printer-to-print-on-Coffee/
4. WiibooxSweetin. WiibooxSweetin Latte Art Coffee Printer Barista Machine Digital Inkjet Cake Desserts Decoration Maker. Retrieved February 20, 2022 from https://www.wiibooxsweetin.com/coffee-printer.php
5. CostaCoffee. The Costa Coffee BaristaBot. Retrieved February 20, 2022 from https://us.costacoffee.com/baristabot
6. Ripples. Ripple Maker II. Retrieved February 20, 2022 from https://www.drinkripples.com/product
7. Colorato. Our World Famous Coffee Printer. Retrieved February 20, 2022 from https://en.coffeecolorato.com/
8. Evebot. How it works. Retrieved February 20, 2022 from http://www.evebot.co.za/how-it-works/
9. Last Minute Engineers. How Water Level Sensor Works and Interface it with Arduino. Retrieved February 20, 2022 from https://lastminuteengineers.com/water-level-sensor-arduino-tutorial/
10. Adafruit. NEW PRODUCT – Peristaltic Liquid Pump with Silicone Tubing. 2019. Retrieved February 20, 2022 from https://blog.adafruit.com/2012/12/19/new-product-peristaltic-liquid-pump-with-silicone-tubing/