Pancake Printer

A Technical Report submitted to the Department of Electrical and Computer Engineering

Presented to the Faculty of the School of Engineering and Applied Science

University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree

Bachelor of Science, School of Engineering

Kendall Reyna Livesay

Spring, 2022

Technical Project Team Members

Enkhbilig Batsukh

Devin Gardner

Yudel Martinez

Maria Parnell

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

Harry Powell, Department of Electrical and Computer Engineering

Todd Delong, Department of Electrical and Computer Engineering

Don't Throw in DePowell - Pancake Printer

Bilge Batsukh, Devin Gardner, Kendall Livesay, Yudel Martinez, Maria Parnell

January 19th, 2021

Capstone Design ECE 4440 / ECE 4991

Signatures

Enkhbilig Batsukh

Devin Edward Gardner

Kendall Reyna Livesay

Yudel Alejandro Martinez

Maria Alaina Parnell

Statement of Work:

Bilge Batsukh

Throughout this project, my primary responsibility was implementation of embedded software components. The microcontroller directly operates the gantry's bump switches, pump and stepper motors, taking instructions over UART from a Raspberry Pi. Specifically, my work covered implementation and testing of interrupts and timers to regulate the pump speed, UART receive and transmit process, and drive speed. I also implemented a set of preprocessor directives for port mappings to allow simple portability of the system between various MSP430 launchpads as necessary.

When each of these subsystems were completed and tested individually, they were also sequenced and tested together, with several test configurations for verifying system integration. Selection of the launchpads we used in the design of this project, namely the FR2355 and FR2476, was an important part of the project scope on my end. Finally, I also contributed to reviewing the design and layout of our PCB system. I worked closely with Yudel throughout these steps, particularly when testing the stepper motors.

Devin Gardner

My primary task for this project was implementing the software that acted as the image processing module for the pancake printer. The purpose of the image processing module was to detect the edges and colors in the image in order to identify appropriate pixel coordinates that the printer could use to effectively recreate the two-dimensional image in pancake art. In order to do this, I wrote a function in Python using the popular OpenCV computer vision library that would accept a given image as input and return the edges and colors detected in the image as a list of binary masks.

The implementation of the image processing module required knowledge of Python programming and image processing techniques, as well as experience with setting up an executable environment on a Raspberry Pi. Previous experience with software engineering gave me the skills I needed to program in Python and manage my code with Git version control, and as such most of my time working on the project was spent researching the image processing algorithms and techniques I needed to employ in order to extract the edges and colors from a given image. After ample research, I was able to narrow down the potential methods I could use for edge and color detection, and I incrementally tested each one on a training image to compare their results and see which method produced the most clearly defined edge and color masks. From there, I tested my script on the Raspberry Pi 4 to ensure OpenCV would run without needing to use more memory than the Raspberry Pi had to offer.

Moreover, I worked closely with Maria to ensure that the edge masks I returned from the image processing module would be acceptable for the path decision script she was writing. We needed to coordinate the expectations for the data type and structure of the edge masks in order to handle them appropriately, and if the quality of the edge masks were acceptable enough to deduce a closed-loop path from them.

Kendall Livesay

Throughout the duration of the project, I was responsible for designing the dispensing mechanism and centralizing the user's experience in the development of other subsystems.

Initially, I manually reproduced the task we were attempting to automate to determine the expected output and areas to be optimized. The design of the dispensing system required selecting a pump that would allow for volume control and batter isolation from other components to comply with best practices and industry standards for kitchen appliances. This led to the selection of a peristaltic pump with a 500 ml/min flow rate. The nozzle was designed with the help of the rest of the team to include high resolution desired by path creation and rigidity to withstand the movement of the carriage system, while also being constructed of food safe materials. A tubing extension was created from brass pipe connectors and a 1.0 mm stainless steel 3D printer tip was attached to satisfy these specifications.

Centering the user's experience included considering common use cases, designing elements of the housing, and fitting attachments. Use cases that were tangential to the actual printing, like cleaning and reuse were incorporated into other subsystems PCB design and web application development. The housing took into consideration that the product would be targeted at families that might have small children and the LEXAN provided a barrier between the heating element while not limiting visibility from the outside. Attachments for replaceable batter holders and removable housing shields allow for easier reuse and cleaning.

Maria Parnell

My primary responsibility for this project was conceptualizing and implementing algorithms to transform images into motor instructions for the microcontroller, which I did in Python. There were elements of image processing involved since these algorithms work best with one-pixel-wide lines; I skeletonized input images to pare down edge-detected results to the minimum essentials. I then manually processed the images to create a path describing how to draw each pixel grouping and in which order. I put systems in place which aimed to draw the image intuitively by maximizing the time spent dispensing batter in smooth, continuous lines. Since batter leakage was found to be inevitable, I also focused on minimizing the number of times the dispenser would need to readjust its placement between continuous groupings. Within this process, there remained a focus on optimizing my code so it could run on a device with low processing power such as the Raspberry Pi without creating excessive waits from inefficient runtime. I also optimized the output paths themselves by removing unnecessary internal commands, allowing the hardware to achieve the same paths with fewer instructions. For debugging and demoing purposes, I wrote a script which interpreted my output files similarly to the hardware driver and displayed visual representations of the paths created by my code.

This task required communication on both ends of my system. I worked with Devin to coordinate the transfer of images between his system and mine, and we also discussed the overall goals for the software system's output to make sure our understanding of how the images should be handled was in alignment. I also communicated with Yudel and Bilge to ensure the output commands were formatted in such a way that they could be interpreted properly into motor movement.

Yudel Martinez

Based on previous research experience, I had responsibilities along the entire range of the project. In the beginning, I worked with Bilge designing the hardware. Then, I designed the main PCB and a breakout PCB for the driver board. I decided to separate these to accommodate excess heat dissipation from the motor driver chip given the high load being supplied. This proved to be useful to keep overheating issues isolated to the breakout board keeping the main PCB thermally safe.

Later I wrote the UART driver along with Bilge and created a command queue to help conserve the RAM resources on the MSP to prevent significant data loss when the path commands are being transferred.

Finally, I created the server and web application for the user interface to the printer. I developed the server using Express and built the front end using Fluent UI. I collaborated with Kendall and Maria to design the user experience based on common use cases and desirable functionality that would increase the ease of using the printer including options for filling in the pancake and the ability to select the level of detail in the final design.

Table of Contents

Table of Figures

Abstract

The pancake printer translates digital art into edible pancake designs for household and professional use. For ease of operation, users can upload a desired image to a connected mobile application. The image is then processed with computer vision in order to identify crucial edges and translate the image into line art. Using path decision algorithms, the line art is then converted into a set of instructions for tracing the outlines present in the image. A pancake batter dispenser is mounted on a two-dimensional carriage system and follows the instructions to dispense batter tracing the line art onto a hot griddle. Once the designs have darkened on the griddle, the dispenser fills in the remaining parts of the image to create comprehensive pancake art. The completed pancake printer serves as a working proof of concept for non-traditional additive manufacturing machines with applications in the medical field for producing 3D-printed organic materials based on imaging of patients.

Background

Rationale and Prior Work

The pancake printer was chosen due to its level of difficulty as well as the wide range of engineering topics touched upon by the design. Since the team of students contained both computer engineering and electrical engineering majors, it was ideal to select a project that integrated elements of computer vision, analog control circuitry, and embedded systems. Subject matter aside, the pancake printer was an enjoyable project to work on for several months since it combined complex technical and aesthetic aspects. Despite its lighthearted intent, the pancake printer serves as proof of concept for more socially impactful works of engineering. Specifically, as a device which can process images and automatically create corresponding mediums, the pancake printer is relevant to the numerous applications of 3-D printed materials in the medical field. In 2019, researchers at the Wake Forest Institute for Regenerative Medicine [1] discussed realistic expectations with 3-D bioprinting. The process requires not only replicative shape design, but temperature and environmental control for the printed bioproducts. With the integration of more advanced computer vision techniques and a more finely tuned 3-D printer using organic materials, the creation of bio-printed tissue replacements is likely feasible for a device such as the pancake printer.

Distinctions

Similar work has been completed in the vein of printers which can create pancake art, but what separates this project from its predecessors is the comprehensive and intuitive user experience it provides. Particularly, the inclusion of mobile image upload capabilities and computer vision differentiates this device from its competitors. Previous iterations of pancake printers have required the use of specific software to create images the printer could process, as well as the physical insertion of an SD card to upload the created images [2]. In contrast, the design of our pancake printer allows users to upload images of any format from their own mobile device, which are wirelessly communicated to the printer and processed into a printable format.

As such, the nature of our pancake printer leads future work away from "edible art" to essential applications in other fields. A 3-D materials printer may be useful in the replication of images taken by non-technical individuals without sufficient time or wherewithal to create CAD instruction for rapid prototyping or fabrication. Another advantage of our design is that it provides an affordable and viable alternative to most pancake printers on the market, which are significantly above the price range of the average consumer, and in many cases discontinued [2].

Coursework

The design of the pancake printer drew upon a wide range of engineering topics in both computer science and electrical engineering. The development of the user-facing application was aided by the students' experience in CS 3240 (Advanced Software Development), CS 4640 (Programming Languages for Web Applications), and individual research experience. The image processing module using OpenCV 1 [3] closely resembled work completed in CS 4774 (Machine Learning), while the process for translating line art into a set of movement instructions was supported by experience from CS 4102 (Algorithms). Printed circuit board components were designed and integrated similarly to projects in ECE 2630, 2660, and 3750 (ECE Fundamentals Series). Lastly, communication between boards as well as the control of the stepper motors adjusting the dispenser position closely resembled coursework from ECE 3501 and 3502 (Embedded Computing and Robotics).

Constraints

Turning images into pancake art is no simple task, and as such the accomplishment of this culinary feat requires multiple coordinating components. With a system as complex as this one, there are numerous functional, financial, and social constraints to consider, all of which must be addressed during the design process.

Design Constraints

Several key design considerations guided the selection of hardware to match the scope of the project, while constraints were respected to maintain project feasibility. Sufficient I/O and memory on the central microcontroller were prerequisites for allowing baseline functionality of the system. At least 16 general purpose I/O pins were necessary to operate the motors, limit switches, pump, and UART modules. As a design consideration, one easily accessible UART and a hardware timer were necessary to allow UART to receive and transmit. Pulse-width modulation was necessary to vary pump speed, and interrupts were necessary for proper bump switch functionality. It was found that 32 KB of memory would be sufficient to buffer UART inputs from images processed by the printer.

Among other hardware constraints, all parts used in the system needed to be both food-safe and heat tolerant. The pump itself required the capability to supply highly viscous pancake batter. Part availability was a constant concern, with shipping and lead times for many potential microcontrollers exceeding several weeks and immediately removing certain models from consideration. The selected processor was switched based on availability on more than one occasion.

No major software constraints were placed on the system, allowing flexibility in the implementation of the algorithms for image recognition and processing. The choice of a Raspberry Pi made for this convenience but was a constraint in and of itself. By aiming to utilize relatively complex image processing software packages such as OpenCV, a higher-power computing system as compared to the primary microcontroller was necessitated to offload computational activity. Due to supply chain issues, purchasing a Raspberry Pi Model 4 during the Fall of 2021 was almost impossible. However, the team was fortunate to have access to Raspberry Pi devices already owned by team members, so this supplier limitation was overcome with good fortune. This project may not have been possible in its current form if not for the prior possession of a Raspberry Pi.

Economic and Cost Constraints

While many of the mechanical components for the pancake printer are inexpensive, some of the electronic components are far more costly. The peristaltic pump for pumping batter is the most expensive item by far, closely followed by the Raspberry Pi 4B microcomputer, NEMA stepper motors, griddle, and AC power source. This means extra caution needed to be taken when regulating power to these electronics during prototyping, otherwise ordering replacement parts would quickly run up the budget and halt progress throughout the design process. Given these expensive parts, as well as others such as the wire sets and OPTIX acrylic sheets, it should be of no surprise that the total cost of the pancake printer is only slightly under the allotted budget. However, since this product was intentionally designed as a luxury kitchen appliance in the first place, there was never any major concern about limiting the cost of production or maintenance, which was ideal since the functionality of the pancake printer primarily relies on its most expensive parts.

Health and Safety Constraints

With this product housing computing devices, motors, and a heated surface all in close proximity of each other, it should be no surprise that the primary safety concerns for the pancake printer during the design process were its potential to start fires and cause burns. For example, one likely possibility is that a user may be burned by the griddle of the device while it is operating. In order to prevent this situation, the team exercised caution around the hot surface while it was heating up, while it was at full heat, and for a period after it was turned off. The griddle was not heated without supervision, and in the case that the team members had to leave the area while the griddle was still cooling, a sign was posted to warn others of the hot surface. To minimize the risks, the griddle was always turned off, unless it was being actively used to test pancake designs.

As an additive safety measure, transparent acrylic sheets were attached to two sides of the carriage system in order to shield those not actively operating the device from the heat of the griddle, which is an especially useful feature for protecting small children and other onlookers from accidentally burning themselves on the griddle. Moreover, the plexiglass sheets essentially acted as a heat shield between the active griddle and the electronics, effectively ensuring the electronics did not overheat, or possibly even catch fire, during device operation.

External Standards

Several project standards will be met in the delivery of this project. Firstly, safety standards regarding kitchen appliances will be followed – IEC 60335 [4]. Code development standards were required for ease of operation, and the set to achieve in this project were those provided by the Barr Group [5] for C code. In wireless communication, IEEE 802.11 [6, p. 11] needed to be observed for the network set up by our Raspberry Pi, and were as such satisfied with that computer's built-in tools. To enclose the robot's electronic systems, the NEMA 1 standard for general purpose indoor enclosures [7, p. 1] was met to prevent dust and light splashing during cleaning of the robot from damaging the device, using plastic covers and acrylic barriers where necessary. IPC class 2 standards for board assembly were met by the manufacturers hired on the custom PCB utilized on the device [8]. NEMA 17 compliant stepper motors were used in the carriage system [9, p. 17]. For communication, HTTP/2 [10] was used for the server and UART [11] was used for the path transmission to the MSP.

Tools Employed

Throughout the course of this project, a wide variety of tools for designing, developing, and testing both hardware and software were utilized. While some of these tools were familiar and easy to use, others required extensive research for effective employment. As such, the tools for each job were picked meticulously based on their ease of use, overall efficiency, and compatibility with corresponding team members.

Hardware Tools

On the hardware side, the schematic designing software KiCAD [12] was used to design the custom PCB, which proved to be a learning curve for the team. On the one hand, KiCAD is an incredibly helpful tool to use due to its intuitive interface and support from community libraries. On the other hand, the ample amount of time it took team members to learn the tool reduced its overall utility. In short, while learning PCB design in KiCAD was a good educational experience for team members, the choice to use it proved to be more detrimental than helpful. Next, the circuit simulation software Multisim [13] was used to test basic circuit designs ahead of time and verify functionality before fabrication. Multisim was the ideal tool for the job as our team already had extensive experience with Multisim's intuitive interface and helpful features from previous coursework.

Firmware Tools

On the firmware side, the programming languages of choice for the MSP430 were C and C++, as they provide the kind of features necessary for developing applications for embedded systems, such as quick execution time and memory pointers. Code Composer Studio [14] was the IDE used for C and C++ development, as it is the development environment that was specifically created by Texas Instruments for embedded microcontrollers like the MSP430. Thankfully, the programming languages and development environment of choice for coding on the MSP430 required hardly any research at all due to previous coursework experience our team has had with these tools.

Software Tools

On the software side, the programming language of choice for the Raspberry Pi was Python, as the Raspberry Pi Foundation itself states that Python is the ideal language for developing on their device. Visual Studio Code [15] and PyCharm [16] were the primary IDEs used for Python development, as they provided valuable debugging tools for the language. Fortunately, both the Python programming language and chosen development environments required little research to use, as previous coursework and industry experience provided our team with the skills and knowledge to utilize these tools effectively.

For the image processing module, the widely popular computer vision library OpenCV [3] was employed to apply simple off-the-shelf solutions for edge and color detection in images. In addition, the mathematical functions library NumPy [17] was utilized to effortlessly operate on high-dimensional image matrices, which assisted the image processing algorithms provided by OpenCV. These tools required extensive research in order to use efficiently, as these libraries and the field of computer vision itself were entirely new to our team. As such, work on the image processing module gave team members a good deal of experience with using OpenCV and NumPy, as well as a better understanding of classic image processing techniques and computer vision concepts as a whole. For the path decision algorithm, the image processing library SciKit-Image [18] was used to operate on the image before extracting a path from it. Using this library was a learning curve for our team, but fortunately not much research had to be done as it was used sparingly in application.

Lastly, the web application used to upload images to the printer was developed using the programming languages HTML and JavaScript, as well as the web application framework Express.js [19]. These tools were optimal for the job at hand, as they not only made application development quick and easy, but also were easy to use as previous industry experience gave our team the skills necessary to expertly make use of these tools.

Ethical, Social, and Economic Concerns

While at first glance the pancake printer may seem like nothing more than a harmless kitchen appliance, in reality this device has more than its fair share of social and ethical concerns. In particular, the ethical dilemmas, economic issues, and health hazards this product poses are quite alarming if not taken seriously. That being said, the pancake printer is still considered a low-risk product in general, as many of the significant concerns with the device were kept in mind and accounted for during the design process. Moreover, the most distressing issues associated with the pancake printer can easily be mitigated with responsible actions on the consumers' part. Examining the major impacts and concerns of an innovation is an essential part of the engineering design process, and the foresight to do so is what often separates a good engineering team from a great one.

Ethical Concerns

In using the pancake printer, the largest potential ethical risks at play are involved with privacy and security. These concerns primarily arise during consideration of design decisions regarding how images are uploaded and saved to the Raspberry Pi. In order to upload an image to the pancake printer, one must first connect their personal electronic device to the Raspberry Pi network hosted by the printer, and then use their device to upload an image to the pancake printer through a provided web app. Given that the pancake printer is only communicating with the consumer's device through its own private network, it usually can be assumed that there's no risk involved with transmission of sensitive data. However, if the pancake printer was connected to a public network, such as in some restaurants or cafes, then a malicious actor could potentially intercept sensitive images during transmission. This could compromise the privacy of the user, not to mention the integrity of the pancake printer itself. In order to best prevent this situation, a warning should be provided with the pancake printer that informs users to connect the device to public networks at their own risk due to potential malicious actors.

Similarly, when images are uploaded to the pancake printer, the privacy of the user is at risk due to the process by which images are stored. Upon receiving an uploaded image, the Raspberry Pi in the device first checks if the image already exists in storage: if it does not, then the image is stored in main memory, otherwise the transmitted image is discarded. Given that images are stored locally on the device itself, it is safe to assume that only the consumer has access to this sensitive data, and as such the privacy of the consumer is safe. However, if images were uploaded to an external server instead, indefinitely storing the images on the server would have major ethical implications. Storing potentially sensitive images uploaded by consumers on private servers would not only compromise their privacy, but also set up their sensitive data as an easier target for malicious actors. In order to prevent such a scenario, images should be deleted from storage after a certain period of time. This should ensure that no image stays in storage indefinitely, thus protecting the privacy of the user by not having their sensitive data stored for the long-term in the first place.

Finally, there is one last ethical issue to consider: with the production of this device commercially, there arises a potential situation where the pancake printer is used in restaurants to replace employees. Given that the pancake printer is calibrated to levels where it can create impressive works of edible art, this product could feasibly replace artisan cooks whose employment had previously depended on their specialization in food art. The ethics of replacing artists with automation are surely debatable, especially in the initial stages of technological change where workers are often left jobless. However, designed to serve primarily as a home good appliance, this is unlikely to provide a worthwhile economic proposition for restaurants that rely on scale to operate. Additionally, given the size and speed of the device it would likely not outperform a skilled trained worker when presented with a larger output. Further iterations of this device by either this team or others, unfortunately, still leave this risk open.

Economic Concerns

With the impressive array of technologies that the pancake printer has to boast, it should be no surprise that the cost of this product is relatively high. One economic concern regarding this product is that the electronic and mechanical components add up in price to create a kitchen appliance that is a luxury rather than a necessity, which means that it is not affordable to the common household. As such, this product would not be widely available if it were to be released on the market, and many markets would most likely not carry the product given the high purchase price to obtain it from suppliers. Another economic concern is that the pancake printer will be more expensive to mass produce in comparison to other kitchen appliances, which may present an obstacle to suppliers looking to produce it.

Social Concerns

What is important to remember about luxury devices such as the pancake printer is that they not only serve as functional appliances, but also can serve as symbols of social status. Given the higher cost of the printer compared to the traditional mode of making pancakes and the luxury it provides, it is of no surprise that the purchase of this product serves as a clear indicator of the economic status of the buyer. More importantly, the mere possession of this product may serve to further the divide between economic classes, which is a significant social concern to consider. The creation of luxury devices such as this one contributes to the social divide between classes by acting as a separator, becoming items to own that differentiate the poor from the rich. If possible, the social divide this product imposes could be reduced by decreasing its cost of production, making it more equally available to all regardless of economic or social status.

In terms of the function of the product, given that this device is designed to make breakfast easier and faster, it is interesting to contemplate how the device may impact consumer expectations and ideals. For instance, consider the following: if one becomes used to having all their everyday tasks automated for them, how will this affect their work ethic? How will this new lifestyle affect their patience and attention span? It could be assumed that one would become less patient and less hardworking over time in such a situation. Now, in a similar fashion, if one becomes used to a more effortless and quick breakfast experience with the pancake printer, would they not find their patience and diligence thinning over time? It may not be a drastic change in behavior, or one that may happen right away, but it would be a change in social traits, nonetheless. When creating a luxury device such as this one, it is important to consider how the product will change the social behavior and lifestyle of consumers.

Environmental Concerns

As a consumer electronic, possibilities for environmental and ecological impacts exist in the supply chain for all mechanical and electronic parts utilized in the pancake printer. These concerns are difficult to quantify on the scale of an individual unit of the pancake printer, but nonetheless do exist and thus must at least be qualitatively considered.

The primary point of environmental concern for the pancake printer is the possibility of food waste. Agricultural transportation and waste are major sources of environmental and ecological harm both globally and in the United States [20] Usage of consumer electronics to encourage otherwise unlikely consumption of novelty food products could create demand for the food products' ingredients, namely flour, grains, and eggs in the case of the pancake printer. These are carbon-emission intensive food products [21], and so the change in consumer consumption habits after the introduction of this product may result in net increases in agricultural waste and emissions. While agricultural waste is a difficult environmental issue to mitigate, engineers must keep in mind the impact on the supply chain and demand their innovation will create upon being introduced to the market.

Sustainability Concerns

The pancake printer should offer a sustainable design since it was engineered to be a reusable kitchen appliance, but one sustainability concern for this product is that it could potentially end up as consumer appliance e-waste [22]. Whether the device offers actual longevity or not depends on two things: how long the individual parts of the system will last through prolonged use, and the demand for automated pancake art over time. Surprisingly, despite producing a large amount of heat and correspondingly consuming high amounts of power, the griddle will likely last the longest out of any of the parts, over a span of ten or more years [23]. Many of the other electrical components will last this long as well, given that there are no irregularities in the power supply. The two parts that have the shortest lifespan are the belt drive and the pump. Belt drives are prone to warping and tearing over time [24], and pumps may begin to leak after periods of frequent use [25], so these parts may need maintenance every few years or so. To mitigate sustainability concerns over replacing these parts, a pamphlet should be provided with the pancake printer that instructs the consumer on how to properly take care of the printer and effectively reduce the wear and tear on its individual parts. By lengthening the lifespan of each part, the overall electronic waste produced by the device over time should decrease, thereby effectively creating a more sustainable product.

Regarding the need for pancake art as time goes by, it should be reasonable to claim that the desire for automatic pancake art should be sustained for as long as people enjoy pancakes, which is practically a given. Therefore, with a reusable appliance that requires infrequent maintenance and a long-term need for the service, this pancake printer design will be sustainable for years to come.

Health & Safety Concerns

Given this device possesses considerable potential to start fires and cause burns, health and safety hazards were always the forefront concerns during the design of the pancake printer.

In order to mitigate these issues, two safety measures should have been implemented: a physical case that both houses the electronics and provides them a heat shield to prevent them from overheating, and an internal switch to cut off the power of the griddle in case it continues to stay turned on after a certain amount of time since the last print. While a version of the heat-resistant shield for the electronics was implemented, due to excessive technical issues during the design process, the internal power switch for the griddle was not; as such, the device currently has no built-in protection against accidental fires and burns caused by prolonged heating. With no such protection, this means that if the griddle is not manually turned off after printer use, which could occur due to human error or other similar mishaps, it could potentially start a fire or inflict a first or second-degree burn on an unsuspecting user.

Hence, in place of this feature, an operation manual should be provided with the pancake printer that instructs users on how to safely operate the device and how to handle potential burns or fires, as well as clear warnings and demarcations as to where users should and should not touch. While this is not a perfect solution, it is an ethical alternative that nonetheless promotes fire safety and prevents possible fires and burns through educational means.

Intellectual Property Issues

As a form of additive manufacturing, the intellectual property documentation, literature, and exclusive rights pertaining to the pancake printer has expanded greatly in the past twenty years. With the first US patent for fused deposition modeling ("FDM") in 1989, increasingly discrete intellectual property constraints have been demonstrated regarding the commercial viability of the Don't Throw In DePowell Pancake Printer [26].

To begin with, [27] is a US patent for "rapid prototyping and fabrication method for 3-D food objects" granted in 2001. The single independent claim, claim 1, of this patent consist of a "freeform fabrication method for making a three-dimensional food object from a design created on a computer, comprising" several sub-characteristics reciting supports, material dispensing methods and control signal integration. This patent, being primarily directed at foodstuffs which may need construction in several layers and without any necessarily decorative characteristics, does not pose a direct intellectual property constraint to the pancake printer. However, it is to be noted that several steps of the disclosed method are integral to the design of the pancake printer, and several dependent claims are also applicable to the printer's methodology. Ultimately, since the pancakes produced by the printer do not "rapidly reach a physical state of rigidity and strength sufficient for permitting said food composition to be self-supporting while being built up layer by layer into a three-dimensional shape in a non-solid state" the independent claim is not applicable to the pancake printer.

Secondly, [28] is a US patent protecting a "system and method for solid freeform fabrication of edible food" granted in 2017. The first independent claim, claim 1 of this patent, is a "system for solid freeform fabrication of edible food comprising" several mechanical elements, a software system controller, and a distinguishing "sealed, soft-shelled packet configured to contain edible food…capable of being pressurized to deform…and dispense the edible food" which notably separates the pancake printer from the claimed subject matter. The second independent claim, claim 6, is much the same, with the substitution of pressure deformation to instead rely on a system of mechanical rollers to dispense from the "soft-shelled packet" instead. Since the pancake printer does not utilize either of these systems, the dependent claims are rendered moot in application. The mechanical construction of the printer is largely disclosed, being a typical machine-driven gantry system, and a typical software system is also claimed within the bounds of the patent. This patent could easily be used to demonstrate at least a few constraints as prior art to the pancake printer, if such a rejection were to be made.

Next, [29] is a US patent for a "method, system, and apparatus for creating 3-D-printed edible objects" granted in 2018. This patent demonstrates several in-software steps for the creation of the foodstuffs disclosed within, and particularly covers the claimed method with the additional dependent claim of "instructing the one or more extruders to print 3-D text onto the printed 3-D object." This patent's independent claims, 1 and 11, are however explicitly directed to mixing cannabinoids or dosed medications into the printed 3-D edible objects, rendering the patent inapplicable in exercise against the actual pancake printer itself.

Second to last, [26] is a US patent for a "method and apparatus for drawing cakes" granted in 2017. This patent's subject matter is explicitly a pancake printer for "capturing a design to dispense the batter onto a cooking surface" and several controllers to "create a contrasting color between foodstuff dispensed at various times." What is independently claimed in claim 1 is a "foodstuff making machine comprising" what is essentially the Don't Throw In DePowell Pancake Printer, excepting the addition of several discrete, three separate controllers for flow rate control, timed delays, and a requirement for "no contact of food stuff and mechanical or moving parts." The independent claim may or may not immediately cover the subject matter of the Don't Throw In DePowell Pancake Printer, though it certainly uses largely the same processes and could be considered "equivalent" under the doctrine of equivalents [uspto.gov mpep 2186]. Seeing the in-purpose and technical similarities of the patent to the Pancake Printer, and the patent's commercial implementation as "PancakeBot" this is likely a direct and complete intellectual property challenge to the commercialization of this project.

Lastly, [30] is a Patent Cooperative Treaty patent, applicable in the United States, for "an autonomous convertible cooking device" granted in 2020. There are two independent claims, claims 1 and 18. Claim 18 pertains to remote operation of the device based on caloric/quantity inputs over an internet connection. The patent recites several elements of the pancake printer in claim 1, namely a "platform with grooves for accommodating the gantry and pan structure;" a "cylindrical container to store the batter;" a "pushing means to discharge the batter;" a "heating pan structure;" a "input/output device for a user to access the autonomous cooking device;" and "at least one communication device[.]" This patent's subject matter is directed towards a purpose substantially different from the pancake printer, and does not produce customized 'art' in foodstuff form. However, the mechanical, electrical, and computer systems disclosed do cover substantially the pancake printer's description. The potentially applicable independent claim is likely inapplicable due to its narrowness in requiring use-from-a-distance capability in the claim, as well as its inability to create customized shapes.

Ultimately, the pancake printer is likely not patentable in the US. Not only has there already been issued a patent [26] a device that does the same aesthetic function, but several other patents and pieces of prior art as described in this section could be combined by persons with ordinary skill in the applicable arts to produce this device.

Technical Description of Project

High Level Overview

The fundamental project is a viscous fluid dispenser (VFD) mounted to a two-dimensional carriage system positioned over a griddle. It uses a central microcontroller communicating with stepper motors, and a batter pump system to recreate shapes and images uploaded via computer or mobile application to a microprocessor in the form of pancakes.

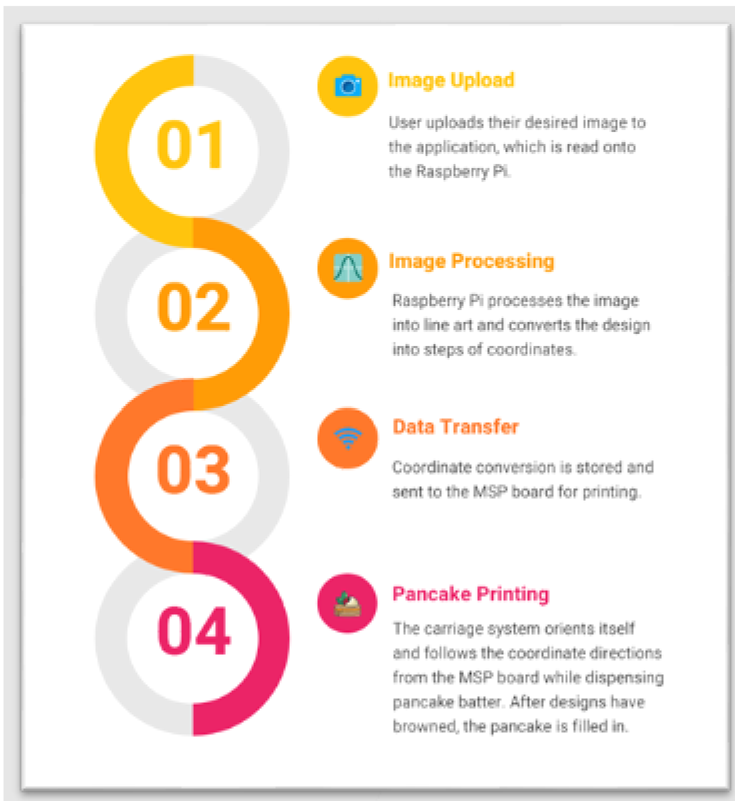


Figure 1: Process Flow Diagram

The pancake printer operates in the procedure described by Figure 1, with a sequential series of simple automated tasks run on three distinct computing units. The first task is user initiation by uploading an image via wireless communication with a Raspberry Pi unit for image processing purposes. The image is processed with computer vision into line art, and then converted into a custom coordinate system instruction set tailored for usage on the pancake griddle contained within the robot. This instruction set lays out the plotted course of the VFD, enabling the stepper motors on the device to precisely and rapidly recreate two-dimensional shapes and detailing. By first drawing out the line art with pancake batter onto the griddle, the highly contrasted edges will darken onto the griddle before the background of the art is filled in, creating a full pancake with shaded detailing to provide image contrast for a pleasant and novel viewing and eating experience.

Technical Overview

Figure 2 provides a comprehensive visualization of the discrete physical sections of the pancake printer as well as their relationships to one another. The following sections will describe in more detail the functionality of each section of the project, as well as explain certain design decisions.

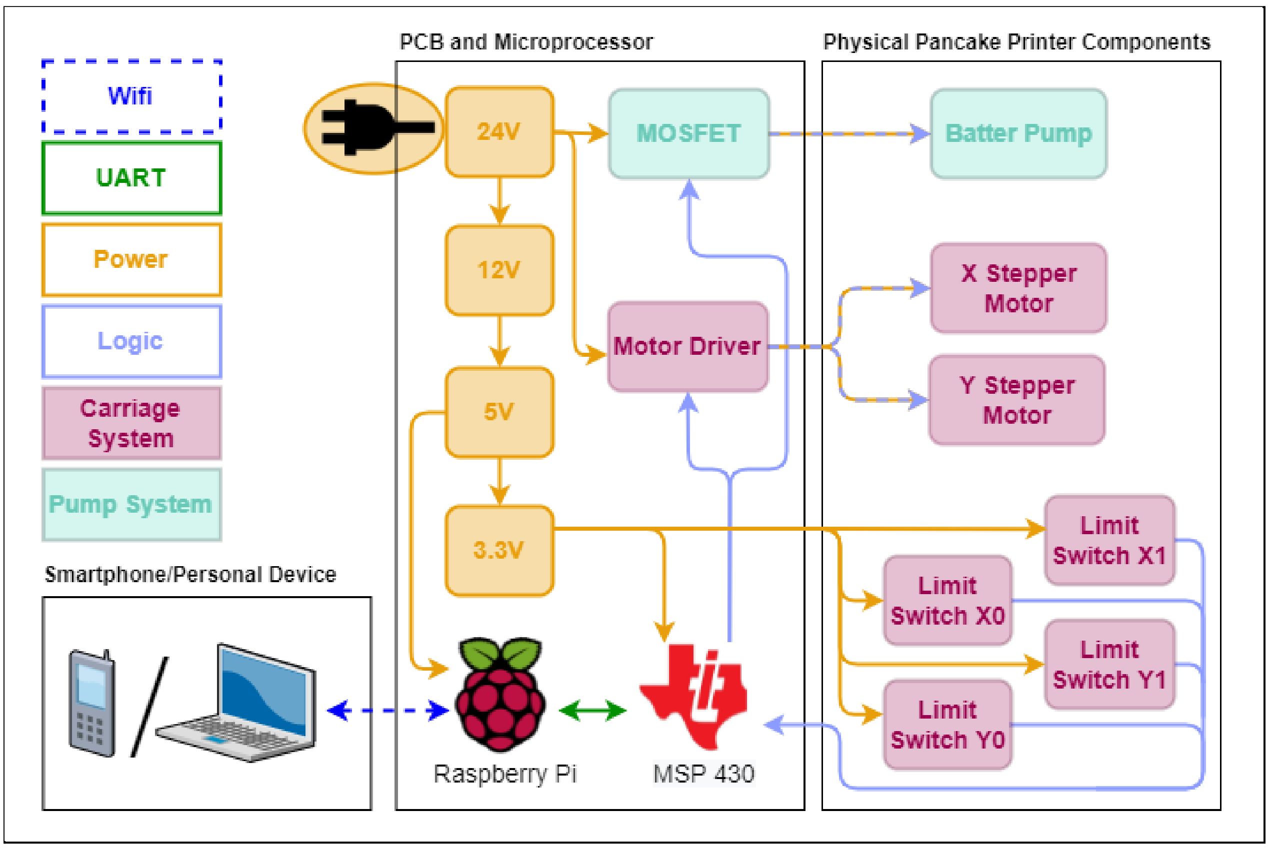


Figure 2: Technical System Flowchart

Image Upload

To allow users to easily upload images from their personal devices, the Raspberry Pi's Wi-Fi chip was configured to act as a wireless access point from which users can access using their personal mobile device or computer. Once connected, the user can access the web address pancakeprinter.local:8080 to interface with the application. The front-end of the web application was built using a simple HTML form to allow users to submit their images to the printer. The backend used Nodejs to run JavaScript which reads in the user's image and saves it to a temporary location in the Raspberry Pi so that the image processing script can access it. Once saved, the script returns five different image options with varying levels of detail to the application for the user to select between.

Several factors were considered when designing the upload system. Most users of the device would not want to waste time or space downloading an entire application to use the pancake printer, which makes a hotspot a much more appealing option. It was also determined that creating a full mobile application would be time-consuming for the team to develop without providing much extra value to the project. Between online resources and prior knowledge, configuring the Pi to host the wireless access point was a convenient design decision. In the end, the image upload process using wireless access turned out to be a consistent, useful, and low-maintenance aspect of the software system. Furthermore, it was found experimentally that different levels of image dilation were required to achieve satisfactory results for different images, so the decision was made to provide the user with more than one option in the application. By utilizing the application to allow the user to select between options, the pancake printer is better able to provide aesthetically pleasing pancakes for a greater variety of images.

Image Processing

In order to transform the uploaded image into pancake art, image processing techniques must be applied to extract the pixel coordinates of the edges and colors in the image. Such techniques could be implemented manually from scratch, or they could be provided from an off-the-shelf software package. Given the time constraints of the project and technical background of the team, the second method was chosen to save time and effort. The widely popular computer vision library OpenCV was chosen for its widespread documentation and ease of use. In addition, OpenCV is compatible with Python, which is the preferred language for the Raspberry Pi.

To distinguish the edges in an image, the Canny Edge Detection method [31] was chosen due to its popularity and reputable performance. This algorithm goes through a four-step process to identify edges: noise reduction, intensity gradient calculation, false edge suppression, and finally hysteresis thresholding. In the first step, a Gaussian blur filter is applied to the image to reduce variation in pixel intensity around the edges, which helps to eliminate any insignificant edges. In the next step, a Sobel kernel is applied to calculate the intensity gradient magnitude and direction for each pixel. In the third step, a technique called non-maximum suppression is used to filter out insignificant pixels that are not needed for the edges. Lastly, the calculated gradient magnitudes are compared to two thresholds to determine the edges: if the gradient magnitude is greater than the larger threshold value, then the pixel is marked as an edge, otherwise if it is less than the smaller threshold value, it is thrown away. This entire process is carried out by the OpenCV Canny(image, threshold1, threshold2) method, which accepts an input image and two threshold values for the hysteresis thresholding. Upon testing this algorithm with a simple image, the result in Figure 3 was produced.

Figure 3: Simple Edge Detection Test

While this test proved to be quite successful, the algorithm still struggled with detecting the edges for more intricate photos. For the next steps, the contrast and resolution of the images were altered to see if it would improve the edge detection capabilities of the algorithm. While changing the contrast of the image proved to be negligible in terms of improving the edge detection performance, changing the resolution of the image to be 150 by 150 pixels greatly increased the number of edges we could extract from the image while decreasing the amount of memory needed to store it on the Raspberry Pi. Another valuable technique that enhanced edge detection performance was normalizing the color space of the image. In normalization, the pixel intensity of each pixel in the image is fitted to be in the grayscale color space range of 0 to 255, which effectively creates a greater difference in the intensity gradient between each pixel of the image, and as such allows the Canny Edge Detection algorithm to more successfully detect edges in the image. Lastly, the morphological operation of dilation was applied to the image, which used a kernel of determined size to make the existing edge pixels thicker. In effect, dilation is incredibly helpful towards edge detection as it connects loose edges and better approximates what the image will realistically look like when drawn with pancake batter. The final function developed returns the edges as 150 by 150 black-and-white binary masks that are passed from the Raspberry Pi to the MSP430 via UART.

In terms of detecting colors in the image, the initial plan was to grayscale the image and then apply a gradient mask that will identify five color layers: white, light gray, gray, dark gray, and black. However, it was determined that having five color layers would be unnecessarily elaborate for this application, so the five layers were compressed down to three: white, gray, and black. The OpenCV inRange(image, lower, upper) method was used to iteratively find the pixels in the image within the color space range of each layer, and each set of pixels was returned as a black-and-white binary mask. As done in the preprocessing for the edge detection, the color space of the image was normalized beforehand in order to create a greater difference in color between each pixel in the image, thus more effectively separating the colors of the image into three separate layers. Upon testing this method with a simple image, the results in Figure 4 were produced.

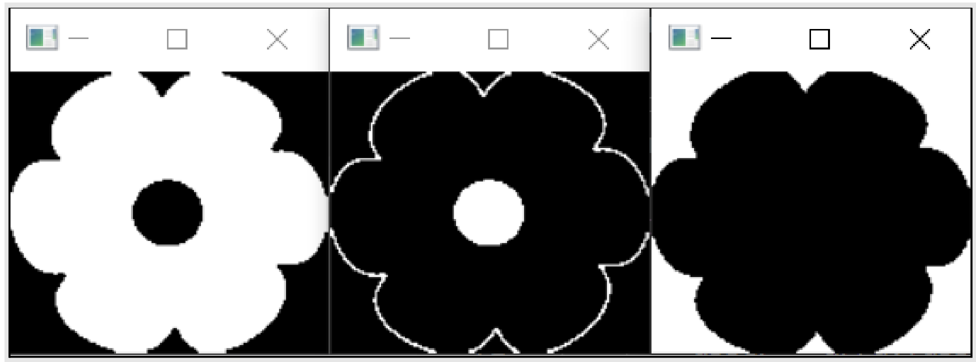


Figure 4: Simple Color Detection Test

While the color detection algorithm was able to isolate the colors of the image adequately for the application, this feature did not end up being included in the final product. The original plan was for color detection to be used to determine which areas of the pancake art should darken the longest in order to create the effect of different shades of color in the image, but this feature was scrapped upon running into other technical issues that delayed progress. In future versions of this product, it would be beneficial to use the color detection script developed for this project in order to create shades of color in the pancake art itself, which would result in more artistic and delightful culinary creations.

Path Decision

At the start of the path decision process, while the edge detection processing did transform images into line art, it quickly became apparent that there were still several unnecessary pixels adding thickness to each detected edge. These extra pixels make traversal difficult and slow down the processing time, so it was optimal to reduce each line to a pixel thickness of one. A scikit-image module named ‘skeletonize’ was found to be able to achieve the desired results. Though the module is somewhat bulky and took some time to download onto the Raspberry Pi, it was overall the best design decision because the module delivered consistent results and saved a lot of development time.

First, the skeletonized images were processed to identify discrete groupings of connected pixels with the intent to draw the most continuous path possible within each grouping. To draw each grouping in an intuitive way, the pixel which occurs farthest from the grouping’s center was identified as the starting point. Then, a depth-first searching algorithm was applied from the starting point. By nature, this algorithm creates paths which extend as far as possible in one direction before jumping back to any intersectional decision point, so applying DFS achieves connected paths which are easier for the motors to follow smoothly.

Certain pre-existing path-finding modules were considered to achieve similar results, but in the end, it was deemed less complex to write an in-house version. The related functionalities were not tailored for use with images or pancake batter and did not allow much flexibility for alteration. It became extremely useful to have the ability to directly tweak the input and output formats as well as algorithm behavior based on progress of the other parts of the project.

Closer to the project deadline, it became apparent that continuous batter drippage was a recurring issue within the design. While some tested systems had shown that running the pump backwards was effective in stopping the flow of batter quickly, the short timeline for this project did not allow enough time for this extra feature to be added. Therefore, several variations on the initial path-finding algorithms were developed to mitigate this issue.

The initial algorithm focused heavily on optimizing the path within discrete groupings of pixels but did not focus on the length of a jump between groupings due to the assumption that the batter output could be stopped. Thus, the first proposed variation was to connect each of the discrete groupings of pixels into one fully connected grouping in the least noticeable way possible. This was achieved by iteratively finding the minimum distance between each grouping and any of its neighbors until the entire network was fully connected. From the traversal standpoint, to entirely avoid drawing outside the connected group, the depth-first search algorithm from earlier was edited to include "backtracking," or retracing back through paths that have already been explored to reach new paths. In software, this method works very cleanly. The turtle visualizations showed that this method does not produce visually distracting excess jump lines. In Figure 5, the left-hand image shows the pathing assuming the ability to stop batter. The right-hand image shows the pathing with the connected model. The red lines represent either or a line which was retraced or an added jump between groupings. The turtle visualizations make it clear that almost the entire path would be drawn over more than once. Instead of drippage between groupings, this raised a new concern. With a standard batter flow, retracing every line led to excessive batter pooling, obfuscating the picture.

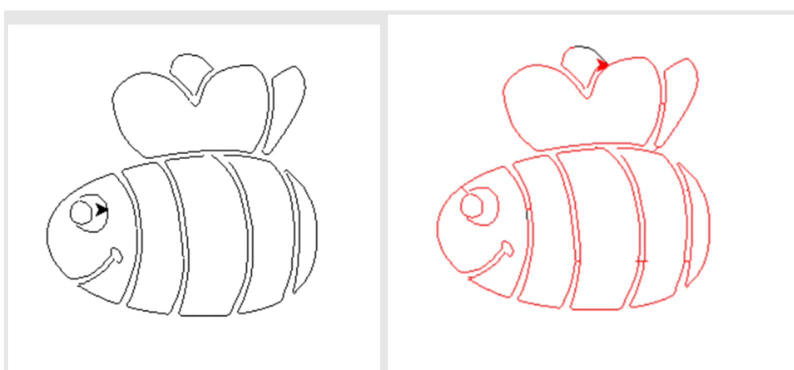


Figure 5: Fully Connected Drawing with Retrace and Minimal Jump

Therefore, an algorithm which combined reasonable retracing and the shortest jump distances possible was also developed. Once reaching a dead-end, the algorithm would search for the closest undrawn pixel ('target pixel'). If the target pixel existed in the same connected grouping as the dead-end pixel, the algorithm would find the shortest path using breadth-first search to retrace through the current group and approach the target pixel. If the target pixel existed in a different connected grouping as the dead-end pixel, then the closest pixel in the dead-end pixel's connected group to the target pixel would be found. The algorithm would traverse to this pixel before performing a jump to the disconnected target grouping.

This solution provided the best results visually and had reasonably low levels of retracing and jumping. Figure 6 once again shows the original solution assuming zero drippage on the left and the hybrid solution on the right. The right-hand solution displays a combination of retraced lines and jump lines with a red color indicator. While some lines are redrawn and there are a couple noticeable jumps, this solution avoids greatly obfuscating the image in batter and thus was deployed in the final version of the device. An extra feature was added such that the path decision would begin and end in the top left corner of the pancake so that no unnecessary lines were layered on the image as the nozzle returns to its 'done' position.

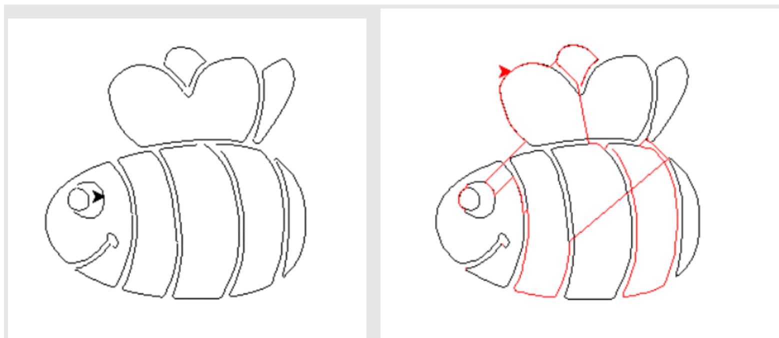


Figure 6: Fully Connected Drawing with Retrace and Jump Balance

Rail System

The structure of the rail system was built out of 80/20 and 40/20 T-slot extruded aluminum due to prior availability, and the aluminum pieces were cut to frame the griddle. L joints were added to most corners to cleanly attach and stabilize the system. The T-slots allowed for the routing of wires through the gaps as well as provided a secure track on which the motors could travel.

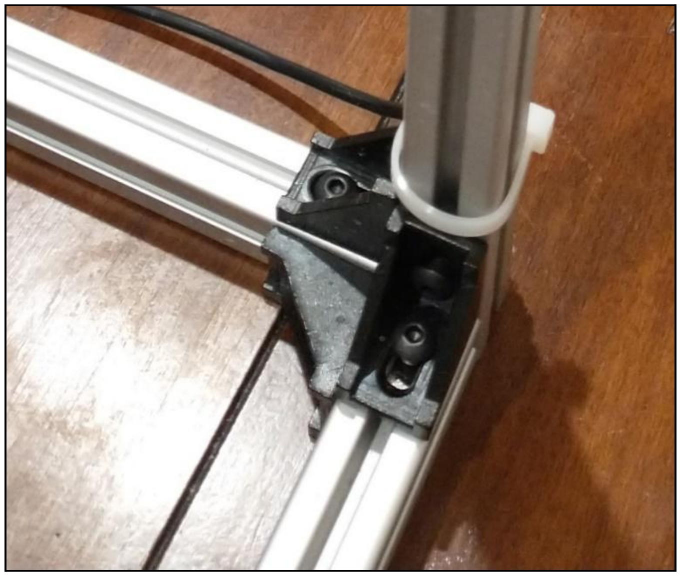


Figure 7: Rail System Joints

Two DC stepper motors were utilized to control the movement of the printer's nozzle on the x and y planes. Stepper motors were chosen due to their ability to execute the precise movements necessary for creating art. Two motors per axis were considered but using a single motor per axis proved to be a viable solution and saved on complexity in the form of utilized ports, motor costs, and debugging requirements. Each OpenBuilds NEMA 17 stepper motor controlled a belt and pinion system which were easily integrated onto the T-slot aluminum due to OpenBuilds standards.

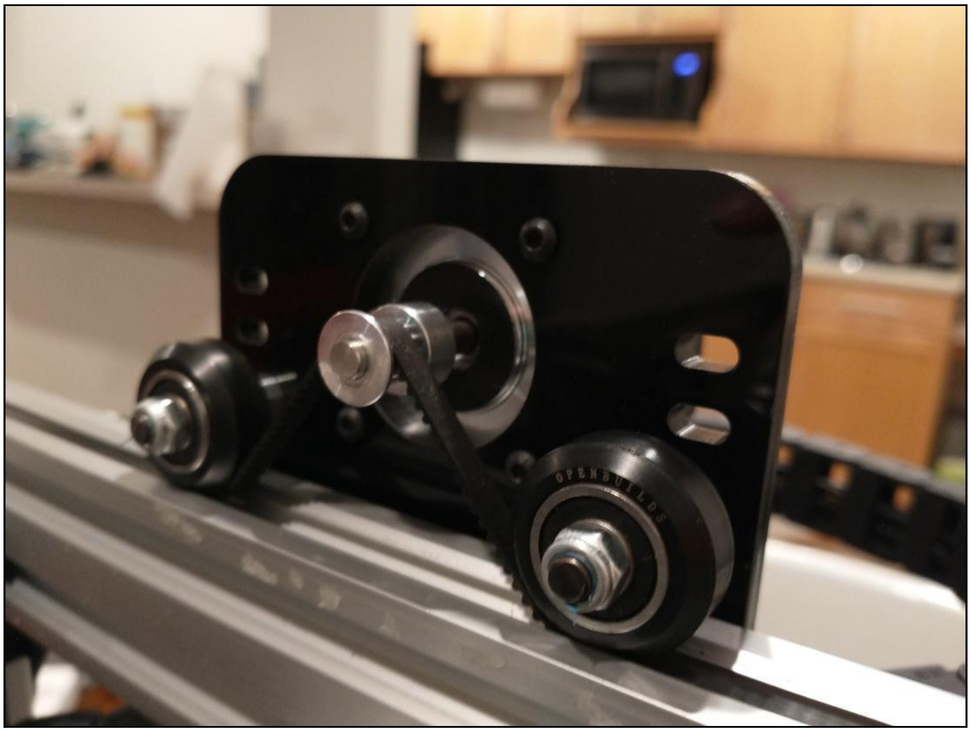


Figure 8: Belt and Pinion System

A DRV8821 [32] , a 2-Stepper driver and controller with micro-stepping support, was selected as the primary stepper motor controller for this project. A breakout board, and headers on the main PCB were added to ensure testability. The DRV8821 utilizes between 8 and 32-volt power connections, accepts logic levels at 3.3V for input pins, and could output up to 1.5A per winding, enough to satisfy both stepper motors utilized.

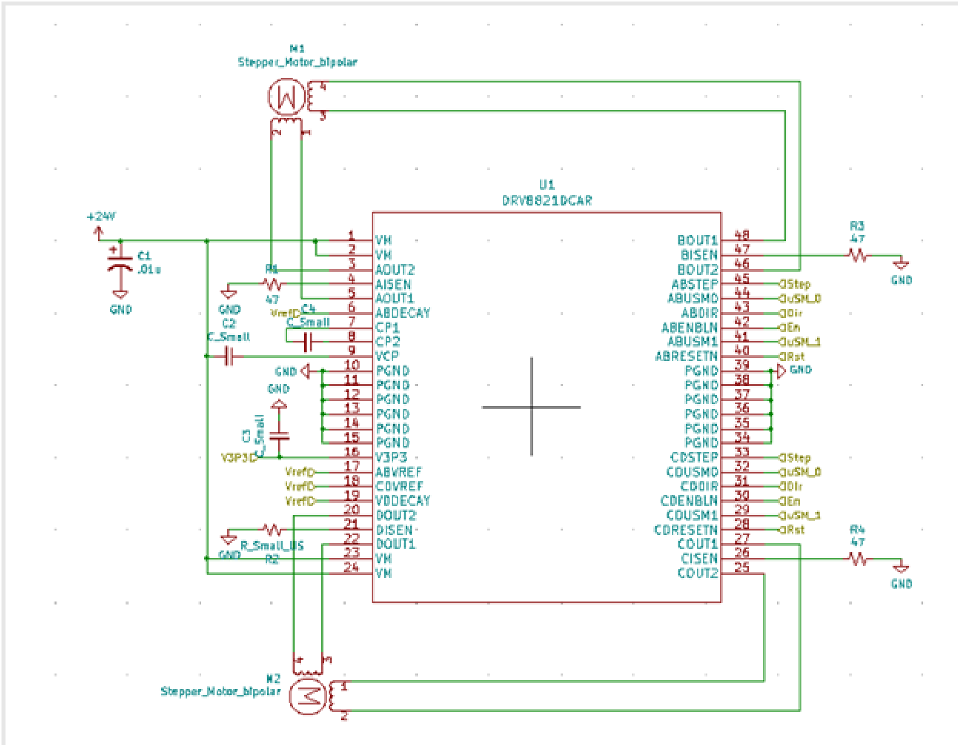


Figure 9: DRV8821 Pinout

The DRV8821 was surface mounted onto the breakout board to allow for easy integration with the rest of the PCB (Figure 10, Figure 11).

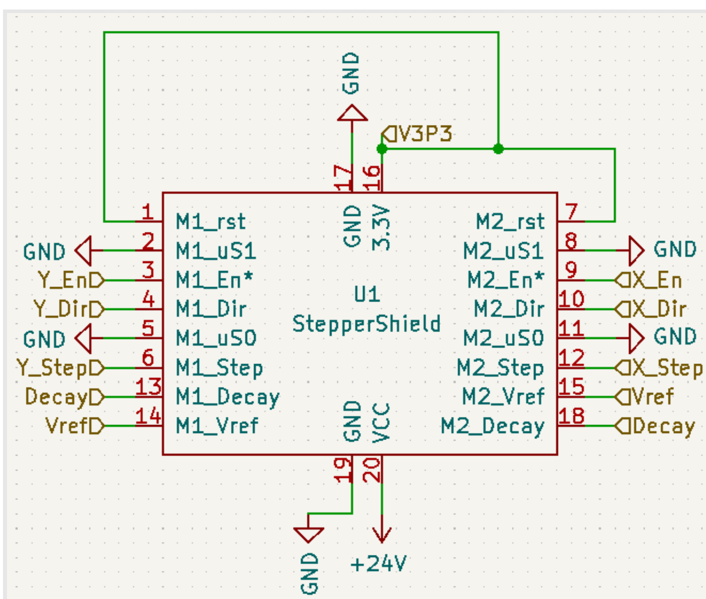


Figure 10: DRV8821 Breakout Board Pinout

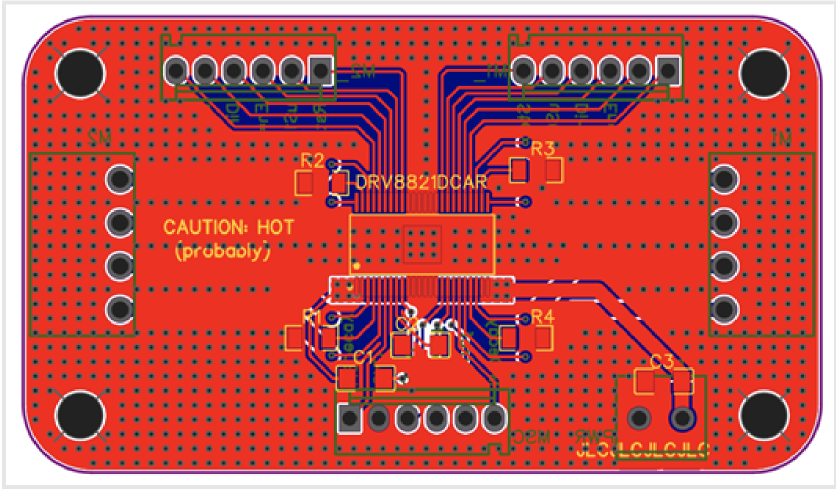


Figure 11: DRV8821 Breakout Board PCB

To prevent the motors from overrunning the gantry assembly's bounds and causing damage to the motors or structure, limit bump switches were integrated and connected to the microcontroller (Figure 12). The limit switches were configured to create a software interrupt when tripped, issuing an error code and to stop motors based on which switch was triggered. They were also used to calibrate the motion range of the printer at boot-up, which allowed a precise count of the usable step range for each stepper motor. Each bump switch was wired into an interrupt-driven pin on the MSP microcontroller and was powered by the 3.3V rail on the PCB (Figure 13).

Figure 12: Limit Switch

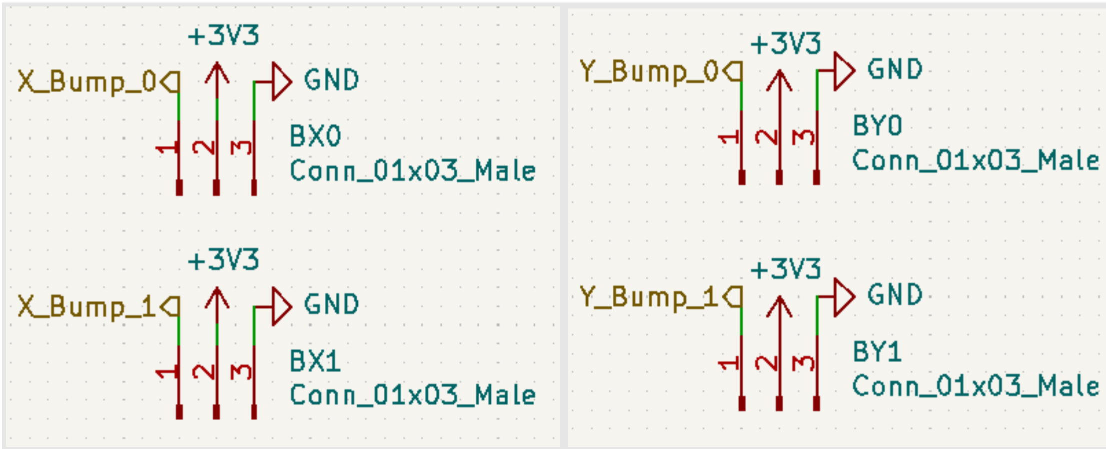


Figure 13: Limit Switch Schematic

Pump System

The physical pump system consists of a peristaltic pump [33], a precision nozzle, and tubing which routes from the batter storage basin through the pump to the nozzle. The specific variety of pump was selected for several reasons. Peristaltic pumps consist of a U-shaped tube around an irregularly shaped roller. As the roller rotates, it compresses parts of the tubing in a circular motion. This pulls fluid through the tubing by creating a vacuum which moves through the tubing. By nature, this system works well with viscous fluids such as pancake batter because the fluid remains within thick tubing. Furthermore, positive displacement pumps such as this one allows for the direct control of dispensation by volume. Running the roller motor faster is linearly correlated with an increased volume of fluid pumped, which is ideal for controlling the flow of the pancake printer to create consistently sized lines. Another crucial feature of a peristaltic pump which benefits this project is that the fluid being pumped only comes into contact with the tubing. This allows for an easier clean-up and ensures that the device will remain food-safe, assuming proper cleaning fluids are passed through the pump after use.

Figure 14: Pump Configuration

The tubing was routed from the pump into various metal connectors which were selected for their availability in nearby stores. This hardware funnels the batter into a thinner opening. Connected to this opening were the metal nozzle tips which were attached via a small segment of thinner rubber tube. Several sizes of nozzle were tested and eventually the 0.8mm nozzle was selected; this size was small enough to allow for precision without causing an excessive buildup of pressure.

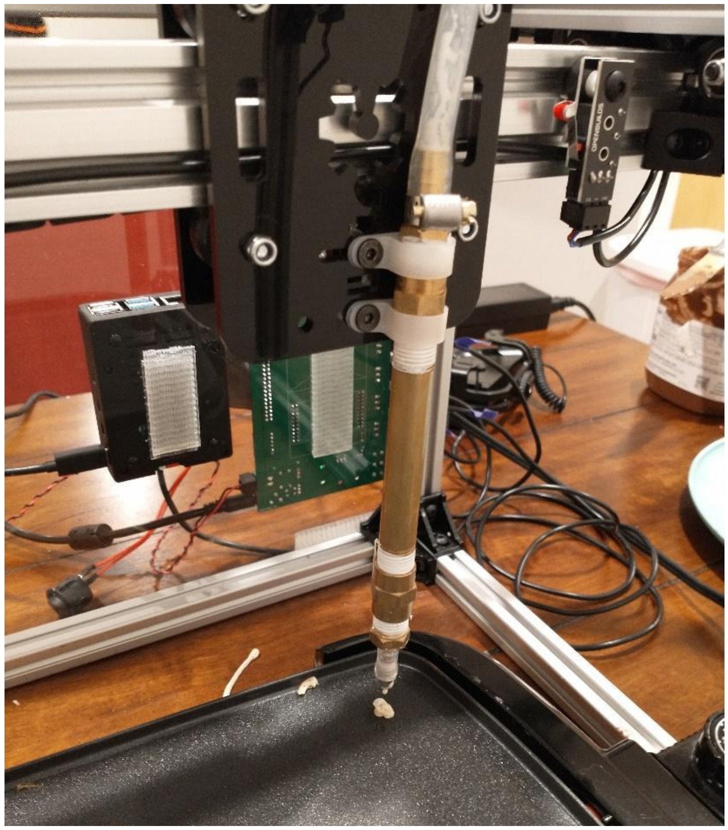


Figure 15: Nozzle Setup

To drive the peristaltic pump, the following schematic was incorporated into the PCB, where M3 represents the pump (Figure 16). The enable signal is a PWM output at 3.3V to a power MOSFET [34] from the MSP board, allowing variable control of up to about 24V through the motor. To maintain proper pump speed and subsequently a reasonable batter displacement, the motor was generally operated at about 50% capacity. Flywheel diodes [35, p. 4] were included to prevent backcurrent, and the pump was run in one direction as necessary. Considerations of implementing bidirectional motor control were made but were ultimately deemed more complex than necessary for the needs of the project. Reducing slightly the extraneous flow of the pancake did not contribute to solving the primary fidelity issues regarding pancake art resolution.

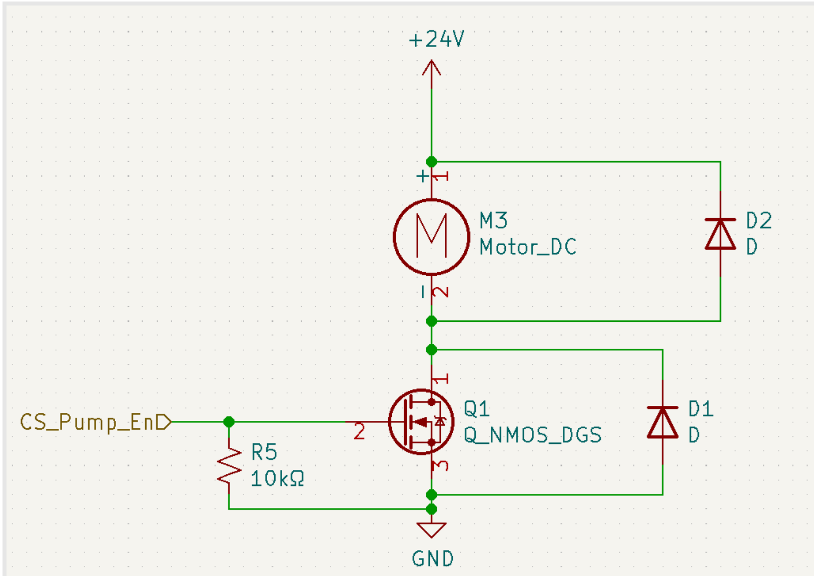


Figure 16: Pump Driver Schematic

Power System

The PCB takes a 24V input from a wall-connected barrel jack. Besides directly powering the motor shield and the pump with this input, a power regulation system was required to safely supply power to various parts of the project's logic and electronic systems (Figure 17). In conjunction with a filtering capacitor and bypass capacitors, an L7812 regulator [36] was used to reduce the 24V input down to 12V. While 12V is not directly used in the system, this regulator was included to reduce the power dissipation through any one component and prevent overheating. The signal was then further reduced from 12V to 5V with an L7805 regulator [37], which was used to power the Raspberry Pi. Using an LM1086-3.3 regulator [38], the power was decreased to 3.3V to power the MSP microcontroller and the bump switches. While 3.3V could have been sourced from other locations on the PCB, the decision was made to directly attenuate the input voltage for isolated testability purposes.

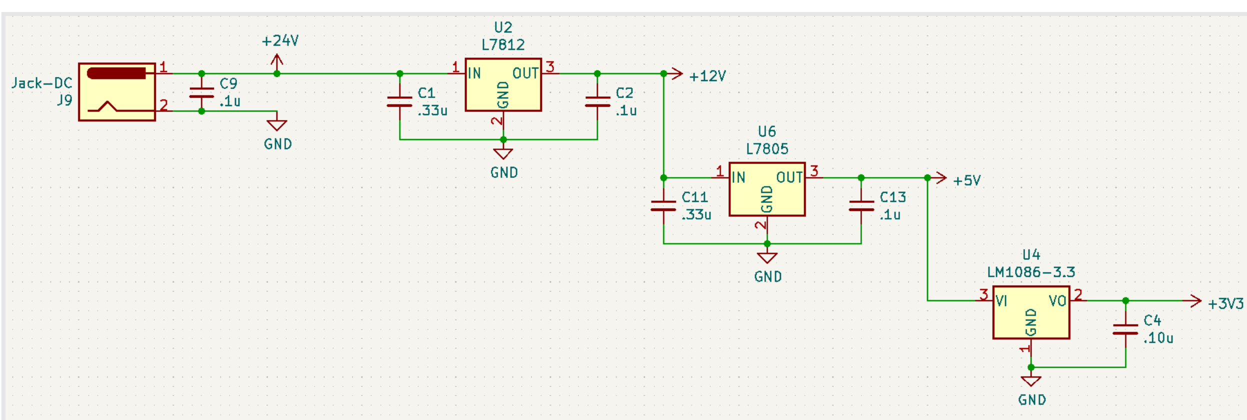


Figure 17: Power Regulation Schematic

Integrated Hardware System

The primary embedded microcontroller used in this project was the MSP430FR2476 [39]. However, the design was initially created for the MSP430FR2355 [40, p. 23]board, so the pinout labeling in Figure 18 corresponds to the MSP430FR2355 architecture. For reference, Appendix A shows the relative placements of pinout between the MSP430FR2355 and MSP430FR2476 boards physically. The destruction of the initially ordered MSP430FR2355 boards in combination with the subsequent shortage of this board led the team to find the MSPFR2476 as a comparable board. Minimum functionality for the microcontroller was identified for all embedded computing requirements, with four areas of special interest. The first was UART communication from the Raspberry Pi, second sufficient I/Os for stepper motor controls, the third was a timer for pump control, and the fourth was I/Os for bump switches. The MSP boards were selected on the basis of sufficient RAM and general-purpose input-output pins to satisfy the project requirements as well as enough external communications support to simple UART communications with the Raspberry Pi. Furthermore, interrupt-based designs were necessary for the bump switches and PWM capabilities were necessary for the pump output, further specifying the microcontroller demands. The MSP430FR2355 and the MSP430FR2476 both accommodated these requirements.

Due to market considerations throughout the course of the project, the initial designs were produced and tested on an MSP430FR2355 system, while the final design is based on the MSP430FR2476 launchpad.

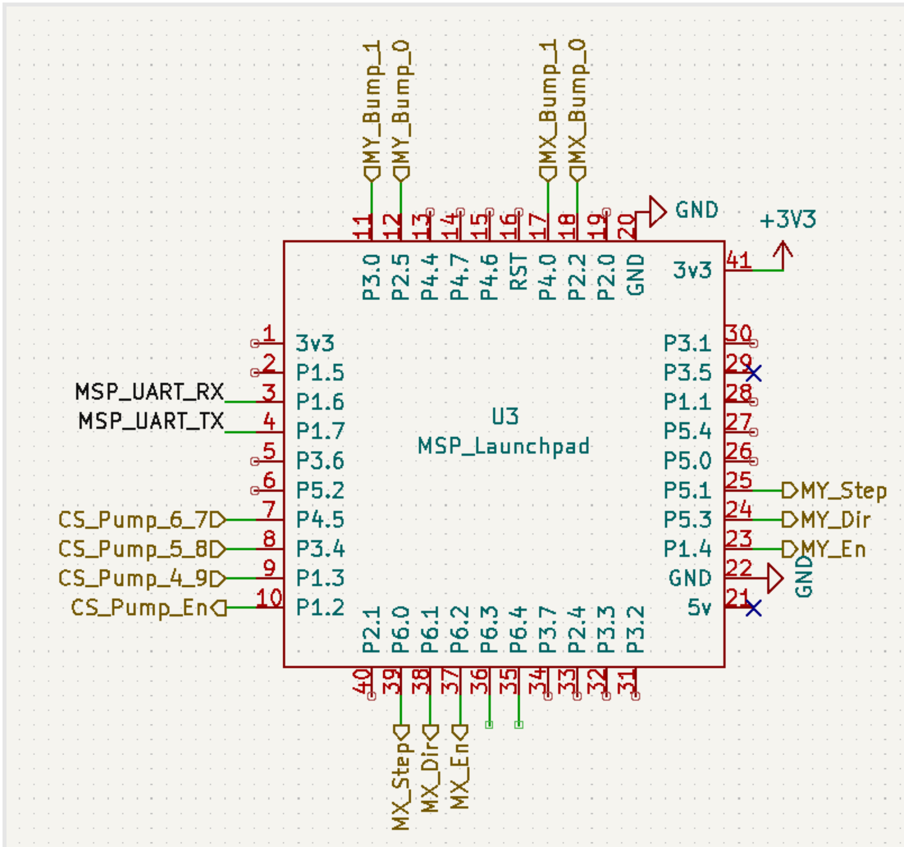


Figure 18: Microcontroller Pinout

The PCB power system presents a 5V power line, which can be used to power the Raspberry Pi. Connectors from the Raspberry Pi pull this power supply and the UART TX/RX lines as necessary to transfer motor instructions.

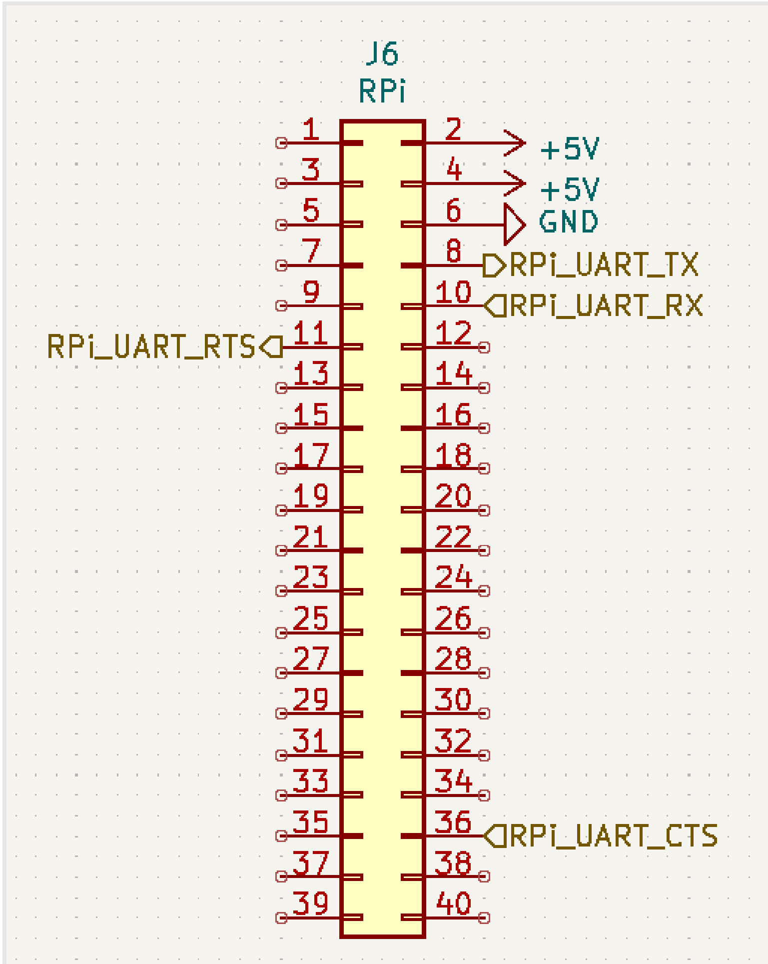


Figure 19: Raspberry Pi Connection Layout

The following figures include the schematics summarized into a hierarchal block schematic (Figure 20), the full PCB which incorporates each schematic (Figure 21), and an image of the fully integrated PCB and Raspberry Pi system (Figure 22).

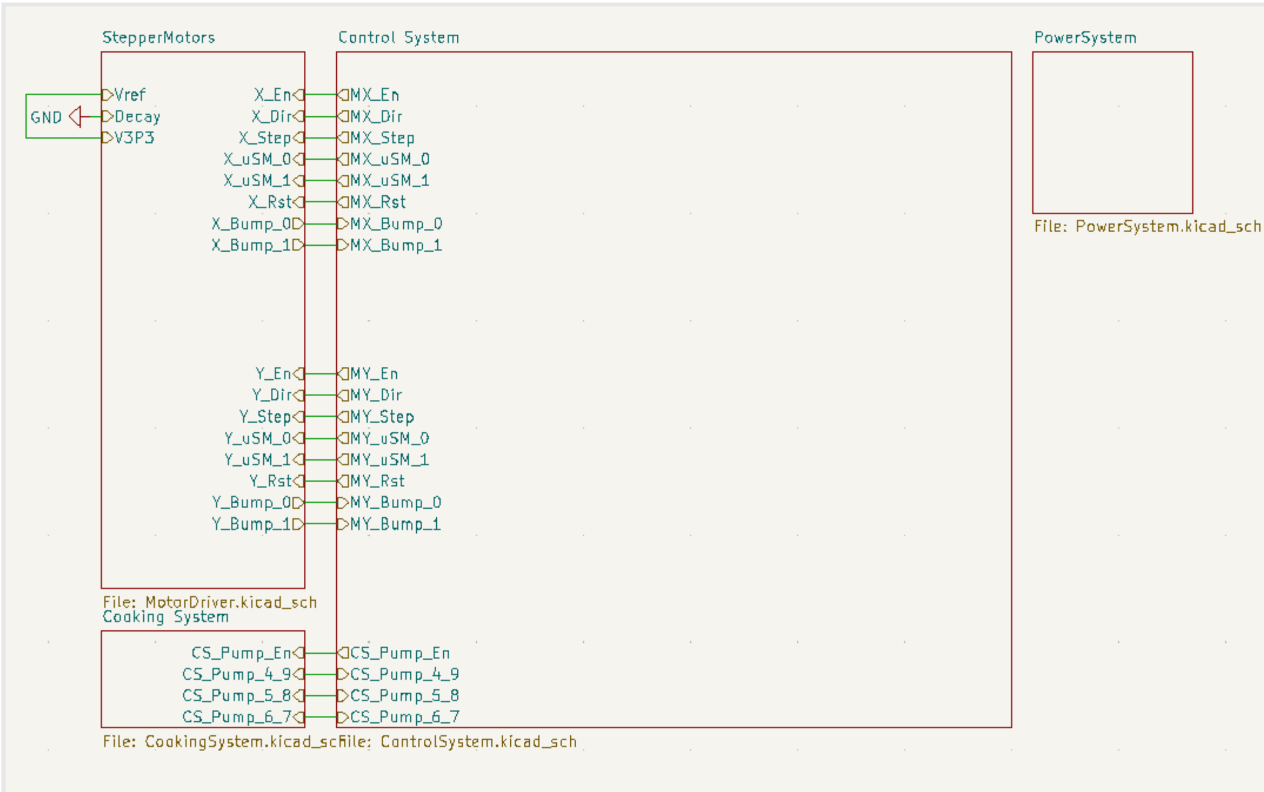


Figure 20: Hierarchal Block Schematic

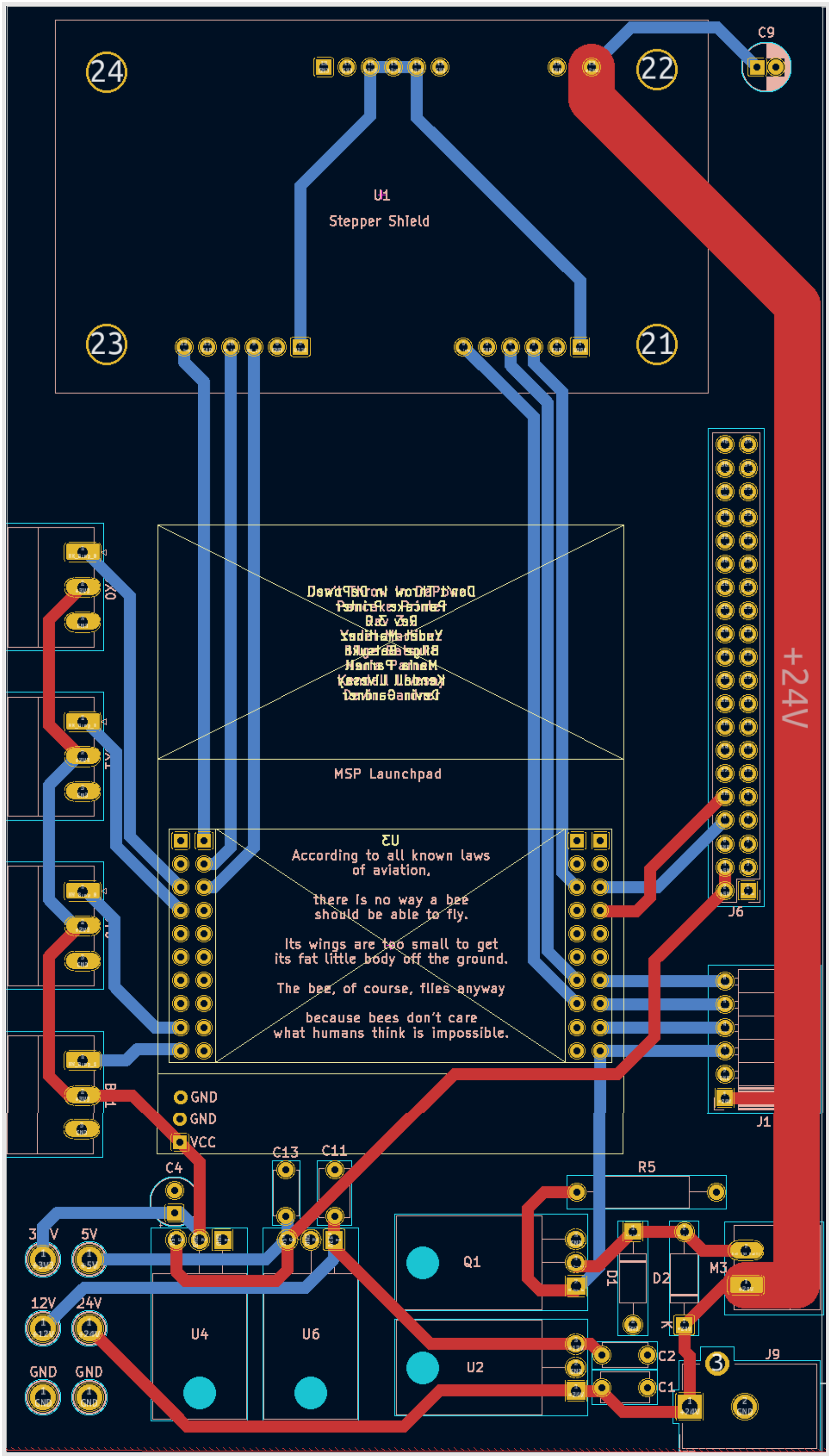


Figure 21: PCB Layout

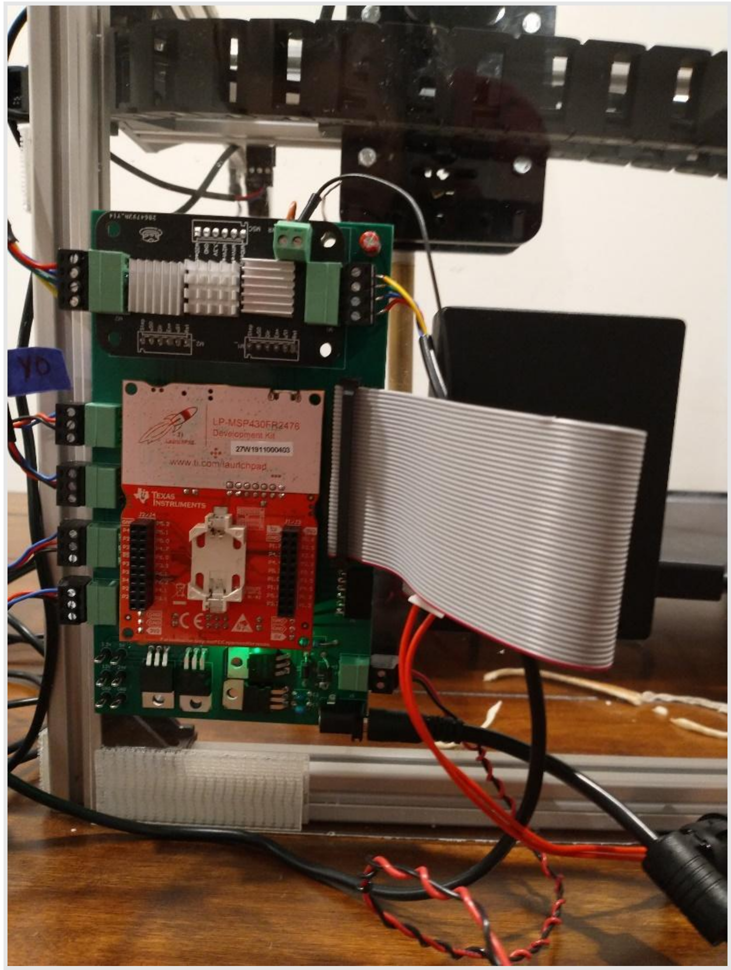


Figure 22: Populated PCB and Raspberry Pi

Data Flow

The flow of data in general is only required to move in one direction, simplifying communication between discrete portions of the project. Images were able to be transferred from the user's personal device to the Raspberry Pi using a wireless hotspot hosted by the Pi as well as a simple HTML form to upload the image. Once the image is uploaded to the Pi, it is processed into coordinate instructions locally on the microcomputer. The processed instructions and their associated coordinates are fed into the MSP microcontroller through a physically connected UART feed. Data is transmitted and buffered into microcontroller storage until capacity is reached, at which point the MSP transmits a hold signal, instructing the Raspberry Pi to cease further transmissions until the hold signal is released.

However, this design, while simple, has some limitations. The printing process is entirely triggered by the upload of a new image to the device and does not contain safeguards to prevent interference or queueing instructions from a new upload in the middle of the process. In the case that the pancake printer is developed into a commercially available product, these drawbacks would need to be addressed.

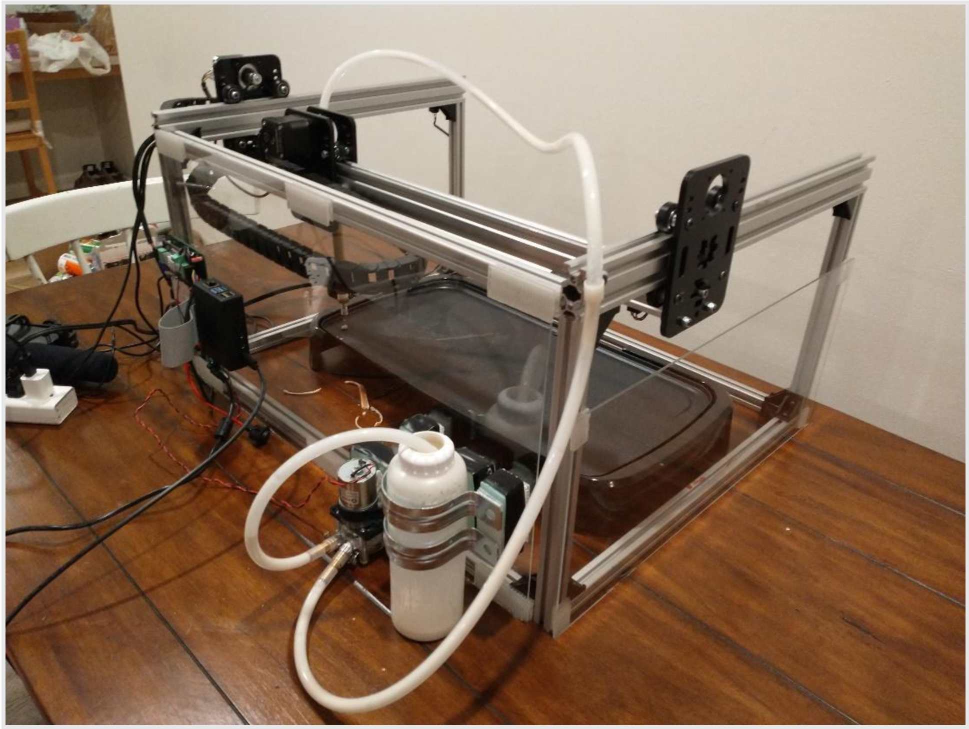


Figure 23: Full Pancake Printer

Project Timeline

Division of Tasks

The technical aspects of the project were split amongst the team members based on interest and comparative expertise. Due to Bilge's experience as an embedded teaching assistant and interest in electrical engineering, his focus was designing the printed circuit board and MSP board's software system, with a secondary responsibility of working on the 2D carriage system. Devin's enthusiasm for machine learning as well as related work in personal projects makes the computer vision aspect of the project a natural fit for him. Kendall's primary task was designing the dispenser system. She primarily researched parts for dispensing pancake batter effectively. Yudel has extensive experience with robotics and embedded systems, so he focused on the 2D carriage system design and managed the networking and application development as well, since he has completed research focusing on those topics. Lastly, Maria's interest and experience working with algorithms made her a natural choice to develop the path decision algorithm, while she secondarily worked with the related software systems (computer vision and application development). It is important to note that in the proposed project timeline (Figure 24), tasks are assigned to the person primarily responsible for each assignment, but this did not mean they will not be aided by team members, especially those who are not assigned a parallel primary task.

Timeline

In the proposed timeline (Figure 24), tasks were organized based on processes that were either distinct or tightly coupled. The carriage assembly, software system design, and PCB design were distinct processes that could occur in parallel. Hardcoded carriage movement and path decision algorithm design are tightly coupled processes that will highly inform one another, so they will also be conducted in parallel. On the other hand, software system design and the path decision algorithm as well as the PCB design and the hardware integration are two continuous parts of the same process so they must be done in series.

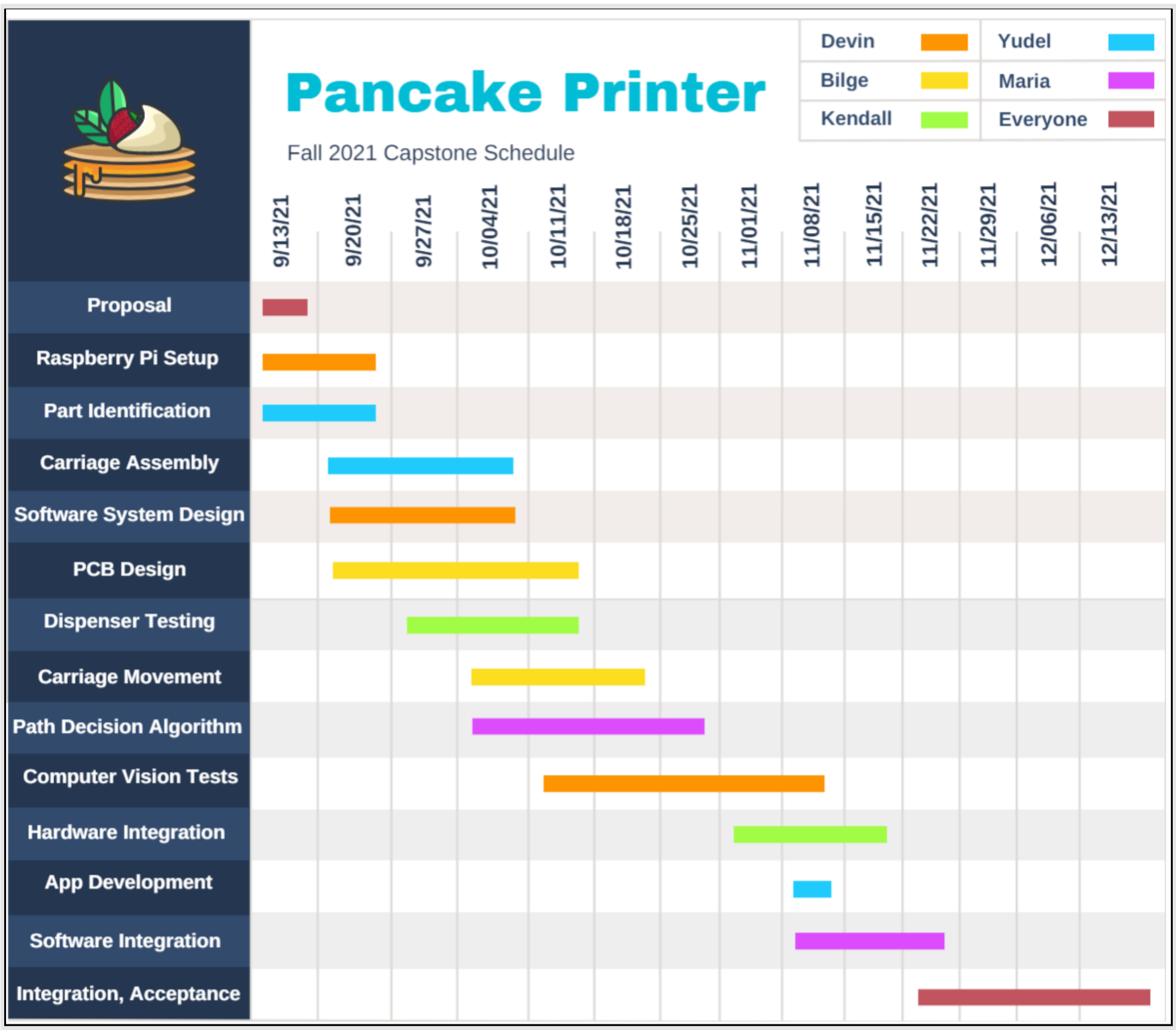


Figure 24: Proposed Gantt Chart

Looking in retrospect, the proposed schedule did not sufficiently capture workflows and parallelization on the project, because it failed to deploy at demonstration time. Several failures in timeline on fabricating and populating a new master PCB caused several complications in integration and testing. Damage to critical components also caused setbacks, such as damaged microcontrollers. With these extensive technical issues, many of the significant sections of the project such as the carriage assembly, software system design, carriage movement, and path decision algorithm had massive delays in their progress, as reflected in the timeline in Figure 25.

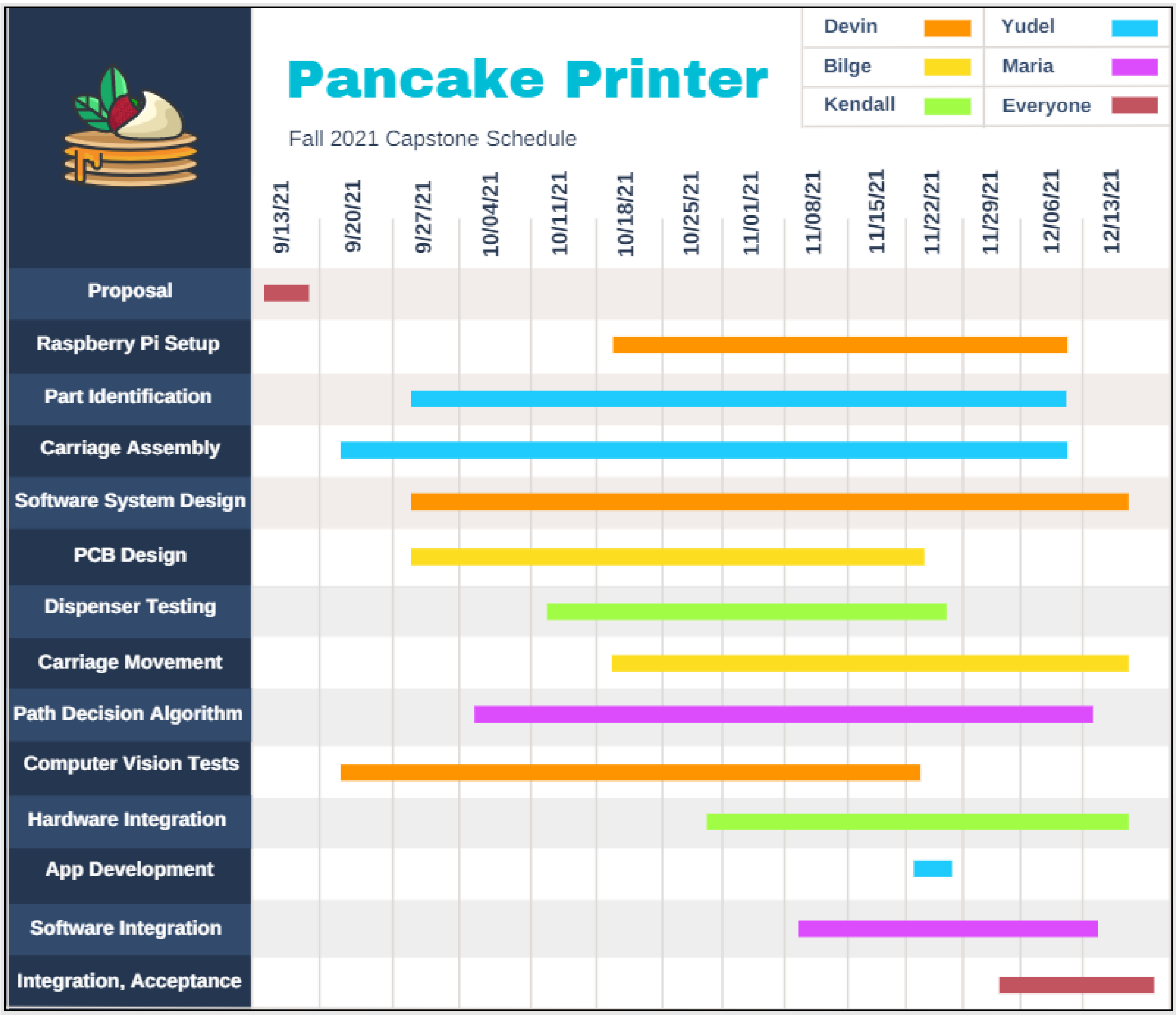


Figure 25: Semester-End Gantt Chart

The further extended Gantt (Figure 26) chart reflects additional tasks completed after the scheduled demonstration day. The primary tasks for the finalized timeline were development of a new user interface for the image upload web application, populating a new printed circuit board, replacing the bump switches with cleaner and mechanically sturdier alternatives, and experimentation with cooking times and heat application during the printing process. Refinements to the path decision algorithm were also made to account for the mechanical properties of the pancake batter used by the device. In this period, Yudel, Kendall, and Bilge focused on the mechanical system and the embedded processing, while Devin and Maria added improvements to the image processing system, based on tests in our finalized dispenser system. The new printed circuit board, which was obtained and fabricated in January, was quickly verified, allowing final testing to proceed. Heat management issues in the motor drive board were overcome, and the Raspberry Pi-MSP430 link was made reliable once more. As such, many of the sections that suffered from massive delays, such as the carriage assembly, software system design, and carriage movement, were finally implemented towards the end of the timeline in January, as shown in the timeline in Figure 26.

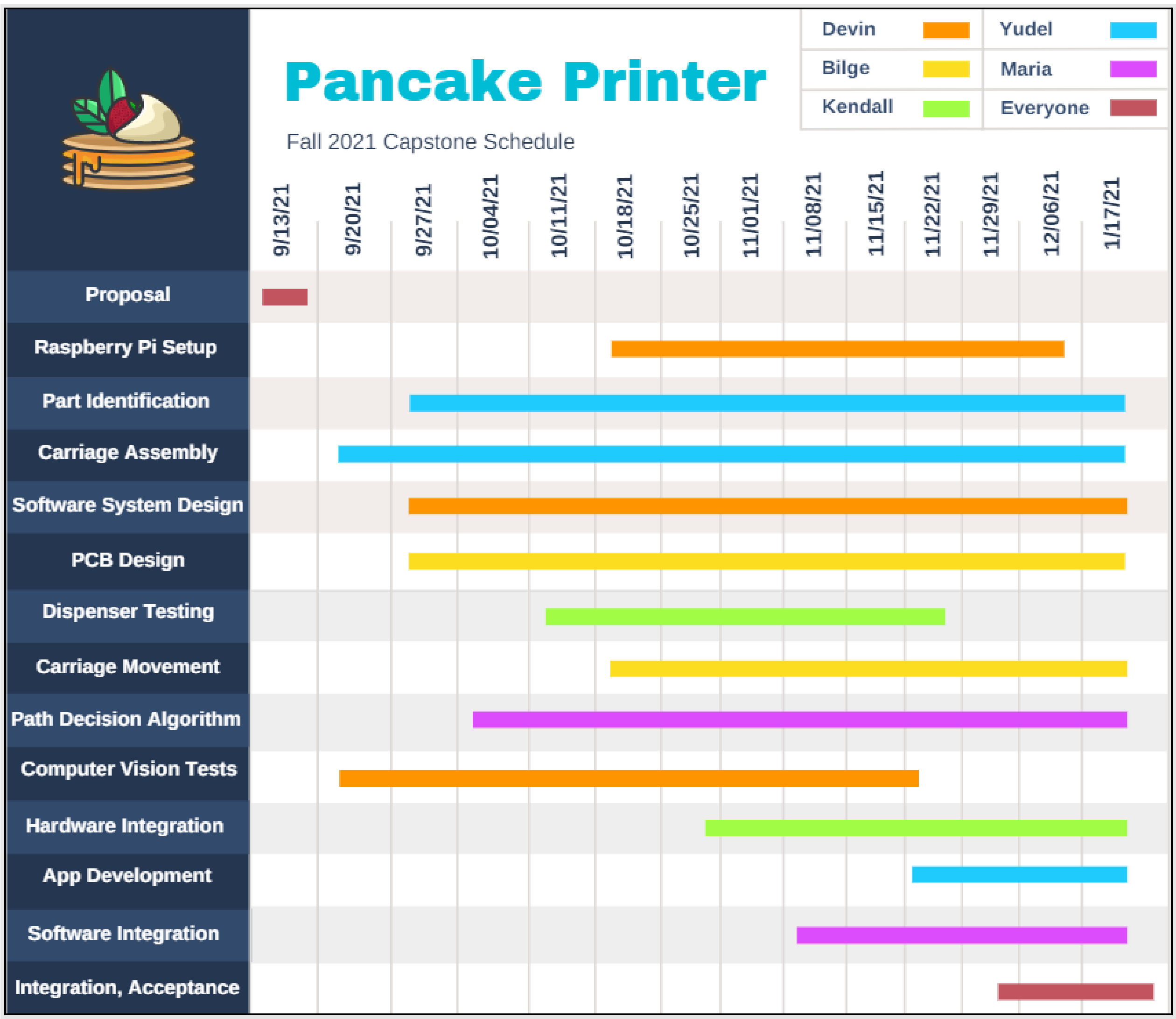


Figure 26: Extended Gantt Chart

Test Plan

The test plan for this project was guided by the chronological segmentation inherently present in this project. To minimize confusion when encountering an error, sections of the project were designed and troubleshot separately before linking up each piece. Discrete development also became necessary due to the requirement of parallel work between team members to conform to the project timeline. The separately testable systems of the project can be described broadly as image upload, image processing, path decision, rail movement, pump system, griddle system, and integrated hardware system.

Image Upload

The image upload was made possible by creating a wireless access point with the Raspberry Pi's Wi-Fi chip, which hosted a simple HTML/JavaScript application. Due to the simplicity of this process, an in-depth test plan was not required beyond standard debugging and verification techniques. The process was tested by using various devices to connect to the hotspot and upload images. It was verified that the application was accessible from the access point as well as functional in transferring images from mobile devices to a temporary location within the Raspberry Pi. After successful completion of an image upload, the image could be found directly in the necessary environment, fulfilling the proper prerequisites for the image processing stage and removing the need for excessive inter-stage testing.

Image Processing

For the image processing stage, two test plans were required: one for measuring the performance of the edge detection and color detection algorithms themselves, and another for measuring the functionality of the algorithms on the Raspberry Pi. For measuring the performance of the algorithms themselves, the test plan was simple: use a select set of images as the test set, generate edge and color masks from them, and display the masks to see if they produced the desired results. For example, for an image with greater detail, the displayed edge mask was used to validate if all the desired details in the image were captured by the generated edges; for an image with a wide array of color variety, the displayed color masks were used to validate if all the desired colors were captured by the generated colors. If the desired results were not present in the displayed masks, then the algorithms would be altered to produce more desirable results.

For measuring the functionality of the algorithms on the Raspberry Pi, the image processing script was uploaded to the device, and then the runtime of each algorithm was measured to see if each one ran within a desirable amount of time. For example, an image from the test set was ran with the edge detection algorithm on the Raspberry Pi, and if the program took longer than approximately five seconds to run with the RAM and processing power of the Raspberry Pi, then the algorithm would be altered to run within a shorter, more desirable runtime. This could be accomplished by removing unnecessary loops or data structures in the algorithm, which would drastically reduce its runtime.

Path Decision Test Plan

Other than simple code debugging, the testing for the path decision section [also] depended primarily on visual feedback. At every stage, visuals were generated to validate results. When edge groupings were identified within images, the pixel colors were changed to match all other pixels within the same group and displayed to the user in order to verify correct results. The Python turtle library, which has drawing capabilities, was used to visualize the results of running the path-finding algorithms on images. A driver was written to read in the generated coordinate instructions and draw the corresponding image using turtle. With these methods, any unexpected behavior was quickly identified.

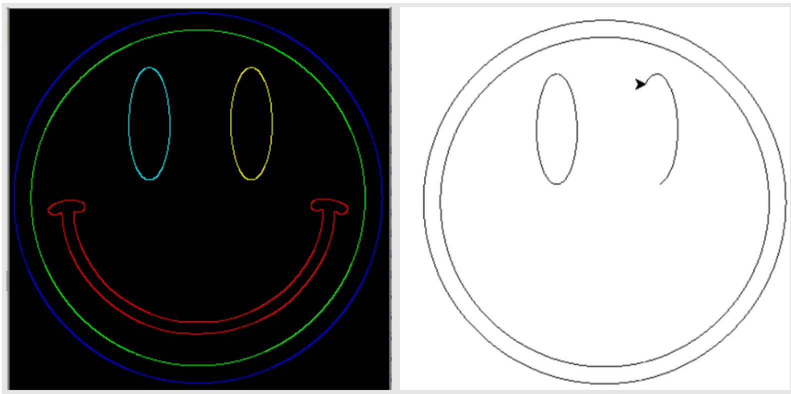


Figure 27: Colored Groupings and Turtle-Drawn Pathing

Furthermore, while the path decision algorithms were developed outside the final deployment environment, steps were taken to ensure compatibility. The code did not utilize extensive external resources other than a couple image processing libraries which are compatible with Raspberry Pi boards. The processes were also developed with the constraints of the microcomputer in mind. For example, certain recursive solutions may not have executed properly within the constraints created by Raspberry Pi's limited memory and thus were avoided. This ensured a smooth transition when the code was eventually deployed on the microcomputer.

Heating System

The heating system was simplified to provide maximum reproducibility in pancake results. As a result, the only steps necessary for the heating system was to set the heat to 450 degrees Fahrenheit and allow the griddle to heat up before printing, much like any traditional pancake cooking setup. Testing consists of ensuring the heating element is correctly plugged into the main body of the griddle. Ensure the acrylic safety barriers for heat protection are fastened to the chassis, and that nothing flammable or temperature sensitive is in the bounds of the gantry, to prevent accident or injury.

Rail System

In testing, the initial mechanical assembly of the gantry crane proved to be largely unstable at higher stepper motor speeds. This was approached by reassembling the crane with more angle brackets to hold tension against vibrations and lowering the center of gravity. When taken together, these countermeasures reduced shifting of the device while on a table, reducing the need for repeated calibrations during testing.

Integrated Hardware

While the software systems could be integrated with relative ease, there was a higher level of interaction between each physical component, necessitating a more in-depth test plan for the integrated hardware system.

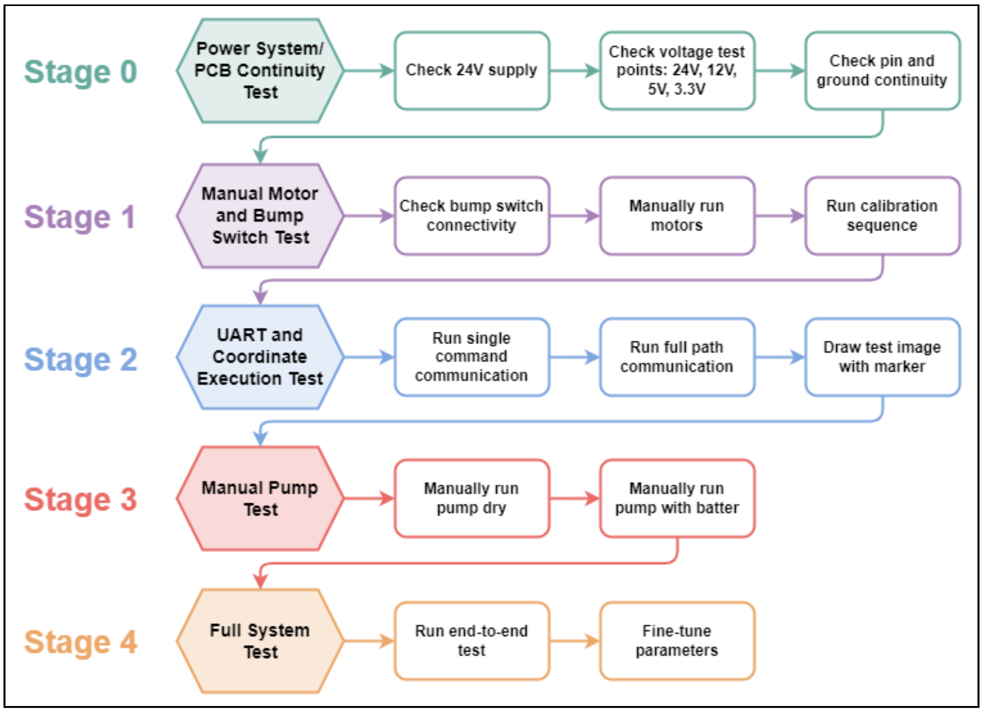


Figure 28: Hardware Test Plan

In hardware testing, the first necessary step is checking the function of the power delivery system. To this end, once the power jack is plugged in, test the supplied voltages at the 24V, 12V, 5V and 3.3V test points on the PCB. Once these are verified, check ground continuity between the breakout motor PCB and the MSP430 Launchpad to ensure mechanical connections have not failed between the components of the device. At this point, it should be simple to see the power-on lights on the Raspberry Pi and the MSP430.

Once proper functioning of the power delivery system has been verified, the bump switches can be checked. Simply pressing each bump switch should evince the switching function, with a red LED enabled when the switch is closed. Finally, the motor calibration sequence can be run to verify the motors are properly connected and aligned with the appropriate bump switches.

The pump system can be tested by connecting the pump control pin, P1.2 to a 3.3V line, to simulate a 100% duty cycle in controls. If this succeeds in driving the pump, then its electronic systems are functional.

Though not relevant to testing for system function, there were design considerations discovered in testing that influenced the final product. The nozzle selection is a 0.8 millimeter 3D printer filament nozzle, which is both food-safe and provides a thin enough stream of batter to provide higher resolution when printing designs. This was tested against higher and lower nozzle sizes, which demonstrated a tradeoff in nozzle leakage and flow rate, which could distort the image or even spill batter depending on pump speed. Thus, the current 0.8mm nozzle was selected to balance these competing interests.

In the final iteration of the printed circuit board, the motor controller board configuration was modified to simply tie high or low certain command inputs, so that they were held constant regardless of microcontroller function. The micro-stepping functions on the motor controller were disconnected from central control, as a source of potential error in the event of microcontroller failures.

Final Results

In this project's proposal, success was defined based on five categories displayed in Figure 29. Based on the descriptions for each category, the team self-assesses the final project as having achieved 10/10 of the described points. The application interface is seamlessly accessible from the Raspberry Pi hotspot and allows users to easily upload their local images without data loss. These images are then edge-detected into detailed line art which reflects the key details in the image. While several of the options displayed on the application capture minimal details only, this was an intentional design decision. Attempting to draw too many details is simply incompatible with pancake batter as a medium, so various image processing methods such as dilation were employed to minimize detail. The selected pattern is then converted into viable motor instructions without error.

When dispensing, the batter flows continuously and consistently from the nozzle. The occasional gap in batter flow may be caused by a low batter supply or, very rarely, by air bubbles inherently present in pancake batter which are impossible to fully remove. To prevent clogging, the batter is strained before use, which is a technique employed by manual pancake artists as well. It was found that room-temperature batter performs best in the system since it allows free movement without being overly runny. While using refrigerated or warm batter could cause slight problems, the system works consistently with room-temperature batter.

The stepper motors follow the instructions accurately to draw the designs while dispensing batter, which produces recognizable images on the griddle. The greatest limitations of the current system relate to the lack of ability to cleanly stop dispensing batter, however, this problem was mitigated in the path decision software, so recognizable pancake art is still produced. There is a noticeable coloration difference between the in-fill and the pancake designs, even in fully cooked pancakes. The inclusion of an in-fill button on the application even allows the user to control their preference in terms of coloration differences.

Points	Carriage System	Batter Dispenser	Heating System	Image Processing System	Application Interface
2	Sufficiently moves the batter dispenser to produce recognizable pancake art	Cleanly dispenses batter into patterns with little clogging issues	Adequately cooks pancake art for pancakes to be edible with visible coloration differences	Effectively detects edges in simple uploaded images, capturing key details	Easily uploads images to the printer with no corruption or data loss
1	Irregularly or erratically moves the batter dispenser to produce indefinite pancake art	Dispenses batter into patterns with noticeable, disruptive clogging issues	Slightly cooks pancake art for pancakes to be mostly edible with somewhat visible coloration differences	Detects some edges in Simple uploaded images, capturing minimal details	Uploads images to the printer with some corruption or data loss
0	Fails to move the batter dispenser consistently, producing unrecognizable pancake art or no art at all	Fails to dispense batter into patterns or dispense any batter at all with constant clogging issues	Fails to cook pancakes consistently or at all, producing inedible pancakes without art	Fails to detect most edges and colors in uploaded images, capturing little to no details	Fails to upload images to the printer most of the time or at all

Figure 29: Success Criteria

Points	Grade
8 - 10	A
5 - 7	B
2 – 5	C
0 – 2	D

Costs

As indicated by the final cost sheet in Appendix C, the total cost of the final product is $433.64. Whether considering the total project budget of $500 or the flat price itself, it's certain that the pancake printer can be considered an expensive luxury item. As the device was always intended to be a luxury kitchen appliance and not a common household tool, it seems reasonable that the price is relatively expensive without exceeding the project budget. According to the final cost sheet, the cost of many of the circuit components was negligible, as many of them cost less than $1 each. The most expensive component by far was the peristaltic pump, which came in at $47.80 for the single part. While it was surprising that the pump cost so much, it was less surprising that the next most expensive part was the Raspberry Pi 4B at $35, given that it is a complete microcomputer with impressive processing power. Besides the electronic components, which were by far the most expensive purchases for clear reasons, the most expensive non-technical parts were the OPTIX acrylic sheets for heat shielding, coming in at $28.96 for two sheets of the material. Overall, while the pancake printer ended up being more expensive than we planned due to the cost of the heat shields, wire sets, and pump nozzle, for it being a luxury item, the price is reasonable given the final product performance and the $500 budget.

If the pancake printer were to be mass produced, the overall cost would decrease significantly when bought in bulk. For example, if 100,000 units of this product were to be produced, the effective price of each unit would become $202.08. This would result in around a $230.00 price decrease from the current level of production.

Future Work

Design Suggestions

Several improvements could be incorporated into the pancake printer in the future. The pancake art itself could be enhanced with various new features. The most notable goal would be retraction, which is achieved by running the pump backwards in order to prevent batter drippage from the nozzle. With this ability, the printer would no longer draw unnecessary lines in between disconnected groupings, making a clearer image. Another common issue encountered in the current design is excessive expansion of batter on the griddle, which muddles designs. Different pump types and speeds, tubing sizes and shapes, and nozzle sizes could be experimented with to achieve thinner lines, increasing image precision. With better precision, even more features could be added to the art. For example, the device could discern the image's shading within the preexisting line art and fill darkened pancake sections in order to better represent a greater range of art with coloring. The pancake printer could achieve several shades of darkness by precisely monitoring cook times to mimic complex coloring present in realistic images. Various colors could even be added to different batches of batter with food dye, and the printer could have several nozzles, alternating between colors to create more vivid art.

Furthermore, several improvements could be implemented to enhance the user experience with the pancake printer. The pancake printer could be further tailored to provide a more end-to-end user experience. To prevent burning and ensure the optimal artistic contrast is apparent on the pancake without undercooking the pancakes, a buzzer or ringer could be implemented to indicate the pancake's readiness for flipping. Since the griddle is a regulated temperature, the timing should remain consistent, but if a user has varying preferences for the darkness of their pancakes, there could be a feature on the application to adjust the cook timing. A more mechanically focused project could even add a robotic arm to flip pancakes so that the user does not need to concern themselves with flipping it at the correct time to preserve the artwork. The robotic arm would deposit the completed pancakes onto a separate cooling surface.

These capabilities could be expanded to create a fully automatic pancake system. Once the batter is deposited in the machine, the user could upload several photos to the mobile application. The application would queue the images as necessary and handle the entire process of pancake creation one by one, resulting in a full stack of pancakes by the end. While this would allow the creation of an artistic breakfast for the entire family without supervision, it would require a significant increase in complexity. Some of the current design decisions were made with the intent that data would only need to transfer in the forward direction. Alternatively, by implementing a fully automated process, information would need to flow backwards from the MSP board to the Raspberry Pi and the application. This project could prove interesting because it would increase the complexity of communication between modules.

Regardless of the specific project, any team who chooses to build upon the current state of the pancake printer will have to contend with certain difficulties. Thermal issues were a consistent problem throughout the process, especially with the addition of a hot griddle, so it would most likely benefit them to include more robust cooling systems than were included in this iteration. It is also of the utmost importance to experiment with batters, because a batter which is too thick or chunky will clog thin nozzles while a batter that is too runny can drip excessively. The batter selection is closely related to the pump and nozzle selection, so these should be chosen in relation to each other. Furthermore, the stability of the rail system can greatly contribute to success or failure. This project benefitted from a well-constructed project frame, but testing showed that an unsteady structure will create poor artwork as the motors will shake the entire system.

Lessons Learned

The project deadline was pushed back from the original date to accommodate various challenges which prevented timely completion. Most notably, crucial microcontrollers and other components were blown shortly before the deadline, causing backwards progress and last-minute redesigns. While the problems were mostly accidental in nature and somewhat unpredictable, a number of actions could have been taken to prevent this delay.

Preventative practices can help to protect hardware as well as mitigate the effects of accidents. All connections should be tested thoroughly before running power through sensitive components, and thermal management methods such as heat sinks can be employed to prevent burnout over time. If possible, redundancies should be ordered ahead of time to account for worst-case scenarios.

Another way to prevent issues like this in the future is to simply shift the project timeline earlier to build in extra time for potential failure. Strict adherence to a conservative timeline would have allowed more time to fix any design problems or order new parts. Lastly, increased communication within the team would allow for earlier identification of potential issues as well as decrease bottlenecks in productivity caused by overly specialized task allocation to certain team members.

References

[1] A. Atala and G. Forgacs, "Three‐Dimensional Bioprinting in Regenerative Medicine: Reality, Hype, and Future," *Stem Cells Transl. Med.*, vol. 8, no. 8, pp. 744–745, May 2019, doi: 10.1002/sctm.19-0089.

[2] "https://www.pancakebot.com/." https://www.pancakebot.com/ (accessed Dec. 17, 2021).

[3] "Home - OpenCV." https://opencv.org/ (accessed Jan. 18, 2022).

[4] D. Roman, "Introduction to IEC 60335 — Household and similar electrical appliances — Safety," in *2015 IEEE Symposium on Product Compliance Engineering (ISPCE)*, May 2015, pp. 1–6. doi: 10.1109/ISPCE.2015.7138702.

[5] webmaster, "Embedded C Coding Standard," May 26, 2016. https://barrgroup.com/embedded-systems/books/embedded-c-coding-standard (accessed Jan. 18, 2022).

[6] "IEEE 802.11-2020 - IEEE Standard for Information Technology--Telecommunications and Information Exchange between Systems - Local and Metropolitan Area Networks--Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications." https://standards.ieee.org/standard/802_11-2020.html (accessed Sep. 10, 2021).

[7] "NEMA 1 Enclosures - NEMA 1 Electrical Enclosure | Nema Enclosures." https://www.nemaenclosures.com/enclosure-ratings/nema-rated-enclosures/nema-1-enclosures.html (accessed Jan. 18, 2022).

[8] "Recently Released IPC Standards and Documents," *IPC International, Inc.*, Aug. 12, 2020. https://www.ipc.org/recently-released-ipc-standards-and-documents (accessed Sep. 10, 2021).

[9] "What do NEMA sizes mean? | Applied Motion." https://www.applied-motion.com/news/2015/10/what-do-nema-sizes-mean (accessed Jan. 19, 2022).

[10] "Introduction to HTTP/2 | Web Fundamentals | Google Developers." https://developers.google.com/web/fundamentals/performance/http2 (accessed Jan. 19, 2022).

[11] U. Nanda and S. K. Pattnaik, "Universal Asynchronous Receiver and Transmitter (UART)," in *2016 3rd International Conference on Advanced Computing and Communication Systems (ICACCS)*, Jan. 2016, vol. 01, pp. 1–5. doi: 10.1109/ICACCS.2016.7586376.

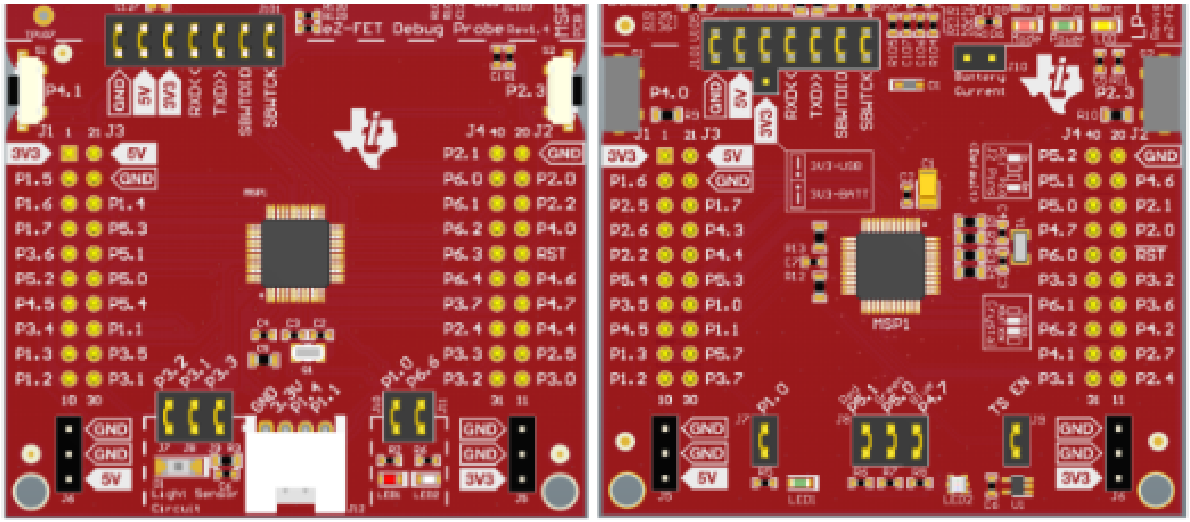
[12] "KiCad EDA - Schematic Capture & PCB Design Software." https://www.kicad.org/ (accessed Jan. 18, 2022).

[13] "What is Multisim™?" https://www.ni.com/en-us/shop/electronic-test-instrumentation/application-software-for-electronic-test-and-instrumentation-category/what-is-multisim.html (accessed Jan. 18, 2022).
[14] "CCSTUDIO IDE, configuration, compiler or debugger | TI.com." https://www.ti.com/tool/CCSTUDIO (accessed Jan. 18, 2022).
[15] "Visual Studio Code - Code Editing. Redefined." https://code.visualstudio.com/ (accessed Jan. 18, 2022).
[16] "PyCharm: the Python IDE for Professional Developers by JetBrains." https://www.jetbrains.com/pycharm/ (accessed Jan. 18, 2022).
[17] "NumPy." https://numpy.org/ (accessed Jan. 18, 2022).
[18] S. van der Walt *et al.*, "scikit-image: image processing in Python," *PeerJ*, vol. 2, p. e453, Jun. 2014, doi: 10.7717/peerj.453.
[19] "Express - Node.js web application framework." https://expressjs.com/ (accessed Jan. 18, 2022).
[20] O. US EPA, "Global Greenhouse Gas Emissions Data," Jan. 12, 2016. https://www.epa.gov/ghgemissions/global-greenhouse-gas-emissions-data (accessed Jan. 18, 2022).
[21] "Eat Low Carbon." https://www.eatlowcarbon.org/food-scores/ (accessed Sep. 10, 2021).
[22] D. Wroclawski, "Appliances, Not Gadgets, Make Up Majority of e-Waste," *Reviewed*, 2015. https://www.reviewed.com/laundry/news/appliances-not-gadgets-make-up-majority-of-e-waste (accessed Sep. 10, 2021).
[23] "How Long Will A Blackstone Griddle Last? This May Surprise You... - Griddle King." https://griddleking.com/how-long-will-a-griddle-last/ (accessed Jan. 19, 2022).
[24] "3D Printer Belt: What to Consider & Which to Buy," *All3DP*, Nov. 02, 2019. https://all3dp.com/2/3d-printer-belt-what-to-consider-and-which-to-buy/ (accessed Jan. 19, 2022).
[25] "Diagnosing and resolving water pump problems | Gates Europe." https://www.gatestechzone.com/en/problem-diagnosis/cooling-system/water-pump-failure-signs (accessed Jan. 19, 2022).
[26] M. Valenzuela, "Method and apparatus for drawing cakes," US9681774B2, Jun. 20, 2017 Accessed: Dec. 17, 2021. [Online]. Available: https://patents.google.com/patent/US9681774B2/en?inventor=Miguel+Valenzuela&country=US&status=GRANT&type=PATENT
[27] J. Yang, L. W. Wu, and J. Liu, "Rapid prototyping and fabrication method for 3-D food objects," US6280785B1, Aug. 28, 2001 Accessed: Dec. 17, 2021. [Online]. Available: https://patents.google.com/patent/US6280785B1/en?oq=US+6280785
[28] H. Lipson and J. I. Lipton, "System and method for solid freeform fabrication of edible food," US9723866B2, Aug. 08, 2017 Accessed: Dec. 17, 2021. [Online]. Available: https://patents.google.com/patent/US9723866B2/en?q=robotic&q=printing&q=B29C64%2f106&status=GRANT&type=PATENT
[29] W. Langeland, "Method, system and apparatus for creating 3D-printed edible objects," US9854828B2, Jan. 02, 2018 Accessed: Dec. 17, 2021. [Online]. Available: https://patents.google.com/patent/US9854828B2/en?q=B29C64%2f106&q=batter&status=GRANT&type=PATENT&page=1

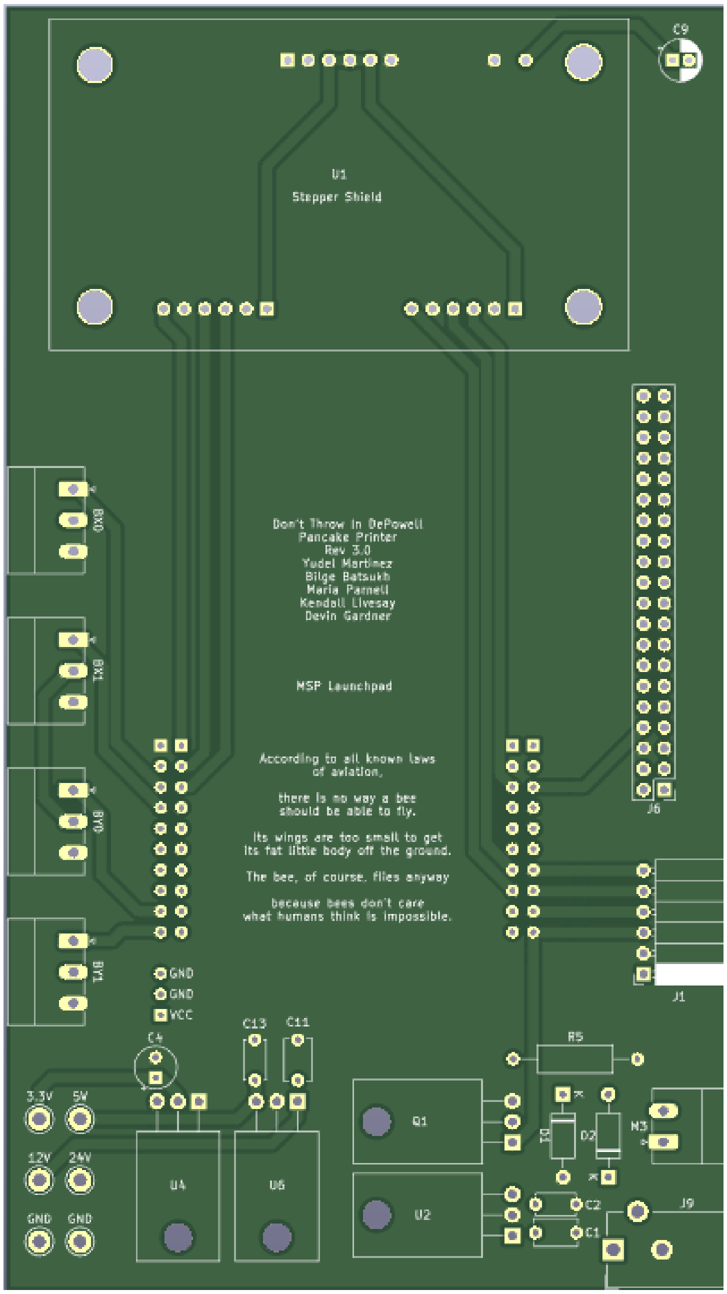
[30] S. B. Mohiddin, “Autonomous cooking device for making idly, dosas, omelets, crepes, pancakes, and the like,” WO2020003167A1, Jan. 02, 2020 Accessed: Dec. 17, 2021. [Online]. Available: https://patents.google.com/patent/WO2020003167A1/en?oq=WO+2020%2f003167
[31] “OpenCV: Canny Edge Detection.” https://docs.opencv.org/3.4/da/d22/tutorial_py_canny.html (accessed Jan. 18, 2022).
[32] “DRV8821 data sheet, product information and support | TI.com.” https://www.ti.com/product/DRV8821 (accessed Jan. 19, 2022).
[33] “How do peristaltic pumps work? | Watson-Marlow | WMFTG US.” https://www.wmftg.com/en-us/support/pump-principles/how-do-peristaltic-pumps-work/ (accessed Jan. 19, 2022).
[34] “IRLB8721PBF Infineon Technologies | Discrete Semiconductor Products | DigiKey Marketplace.” https://www.digikey.com/en/products/detail/infineon-technologies/IRLB8721PBF/2127670 (accessed Jan. 19, 2022).
[35] “1N47 Zener Diodes - Microchip Technology - Zener - Single | Online Catalog | DigiKey Electronics.” https://www.digikey.com/catalog/en/partgroup/1n47-zener-diodes/42060 (accessed Jan. 19, 2022).
[36] “L7812CV STMicroelectronics | Mouser,” *Mouser Electronics*. https://www.mouser.com/ProductDetail/511-L7812CV (accessed Jan. 19, 2022).
[37] “L7805CV STMicroelectronics | Mouser,” *Mouser Electronics*. https://www.mouser.com/ProductDetail/511-L7805CV (accessed Jan. 19, 2022).
[38] “LM1086CT-3.3/NOPB Texas Instruments | Mouser,” *Mouser Electronics*. https://www.mouser.com/ProductDetail/926-LM1086CT-3.3NOPB (accessed Jan. 19, 2022).
[39] “MSP430FR2476 data sheet, product information and support | TI.com.” https://www.ti.com/product/MSP430FR2476 (accessed Jan. 19, 2022).
[40] “MSP430FR2355 data sheet, product information and support | TI.com.” https://www.ti.com/product/MSP430FR2355 (accessed Jan. 19, 2022).

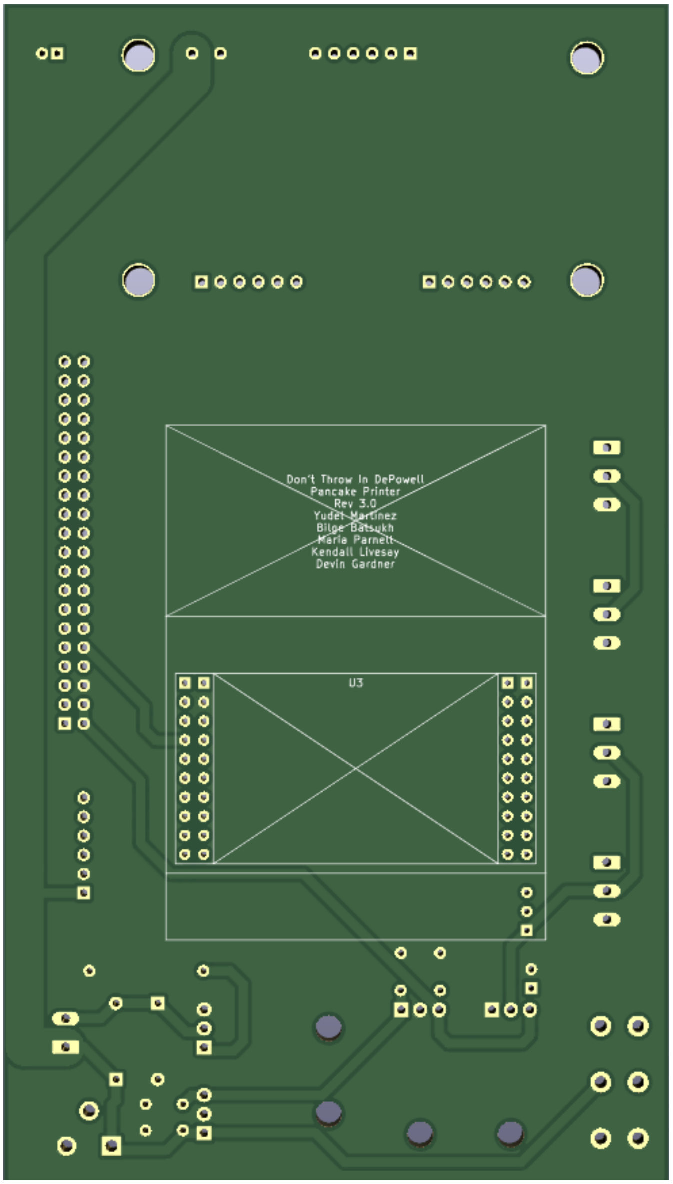
Appendix

Appendix A: Pinout Comparison of MSPFR2355 (left) and MSPFR2476 (right)



Appendix B: 3D Rendering of Unpopulated PCB




Don't Throw In DePowell
Pancake Printer
Rev 3.0
Yudel Martinez
Bilge Batsukh
Maria Parnell
Kendall Livesay
Devin Gardner
U3

Appendix C: Final Cost Sheet

Item (* Found in NI)	Supplier	Part Number	QT	Unit Price	Total Price	Tracking Budget
						$500.00
*V-Slot 20x20	OpenBuilds	320-LP	1	$21.39	$21.39	$478.61
*V-Slot 20x40	OpenBuilds	195-LP	1	$13.99	$13.99	$464.62
*NEMA 17 Steppers	OpenBuilds	623	2	$18.01	$36.02	$428.60
Griddle	Target	N/A	1	$25.99	$25.99	$402.61
Pump	Amazon	N/A	1	$47.80	$47.80	$354.81
Motor Stepper Shields	JLCPCB	N/A	1	$0.50	$0.50	$354.31
Main Custom PCB	JLCPCB	N/A	1	$0.50	$0.50	$353.81
144W 24V 6A Power Supply	Amazon	N/A	1	$23.99	$23.99	$330.82
Raspberry Pi 4B	Amazon	N/A	1	$35.00	$35.00	$295.82
MSP430FR2476	Digkey	296-50211-ND	1	$17.99	$17.99	$277.83
Pump Driver MOSFET	Adafruit	355	1	$1.75	$1.75	$276.08
2x20 Ribbon Connector	Amazon	N/A	1	$14.00	$14.00	$262.08
CONN HEADER VERT 2POS 2.54MM	Digikey	S1012EC-02-ND	1	$0.06	$0.06	$262.02
CONN HEADER VERT 3POS 2.54MM	Digikey	S1012EC-03-ND	5	$0.09	$0.45	$261.57
CONN HEADER VERT 40POS 2.54MM	Digikey	S2012EC-20-ND	1	$0.76	$0.76	$260.81
CONN HDR 2POS 0.1 TIN PCB	Digikey	S7000-ND	1	$0.32	$0.32	$260.49
TERM BLOCK PLUG 2POS STR 3.81MM	Digikey	277-8838-ND	1	$2.48	$2.48	$258.01
TERM BLOCK PLUG 4POS STR 3.81MM	Digikey	277-11432-ND	4	$4.89	$19.56	$238.45
CAP ALUM 1000UF 20% 16V RADIAL	Digikey	732-8804-1-ND	5	$0.42	$2.10	$236.35
CONN HDR 6POS 0.1 TIN PCB	Digikey	S7004-ND	2	$0.52	$1.04	$235.31
TERM BLOCK HDR 3POS 90DEG 3.81MM	Digikey	277-1207-ND	4	$1.32	$5.28	$230.03
Xtension Wire Set - 4 Conductor (7 foot)	OpenBuilds	2557	2	$9.19	$18.38	$211.65
Xtension Connector Sets (4 Pin Male & Female)	OpenBuilds	2535-Set	2	$0.00	$0.00	$211.65
Xtension Wire Set - 3 Conductor (3 foot)	OpenBuilds	2905	2	$4.29	$8.58	$203.07
Xtension Wire Set - 3 Conductor (7 foot)	OpenBuilds	2865	2	$6.79	$13.58	$189.49
Xtension Connector Sets (3 Pin Male & Female)	OpenBuilds	2530-Set	4	$0.00	$0.00	$189.49
Xtension Limit Switch Kit	OpenBuilds	2805-Kit	2	$6.29	$12.58	$176.91
BumpConn	Digikey	277-1207-ND	4			
C1 C11	Digikey	445-173379-1-N	2			
C2 C13	Digikey	445-FK28X7R1H	2			
C4	Digikey	478-1839-ND	1			

CAP ALUM 0.1UF 20% 50V RADIAL	Digikey	732-8847-1-ND	1			
J1	Digikey	S5481-ND	1			
J6	Digikey	609-6379-ND	1			
CONN PWR JACK 2X5.5MM SOLDER	Digikey	CP-102AH-ND	1			
TERM BLOCK HDR 2POS 90DEG 3.81MM	Digikey	277-1206-ND	1			
Test Points	Digikey	36-5011-ND	6			
Power Connector	Digikey	ED90389-ND	2			
U3	Digikey	296-53618-ND	1			
U4	Digikey	LM1086CT-3.3/N	1			
MSP_Header	Digikey	S6106-ND	2			
MSP_PWR	Digikey	S7036-ND	1			
5V Regulator	Digikey	497-1441-5-ND	1			
12V Regulator	Digikey	497-12406-ND	1		$38.79	$138.12
RES 0.47 OHM 5% 1/4W 1206	Digikey	73L4R47JCT-ND	4			
CAP CER 0.1UF 50V X7R 1206	Digikey	399-C1206C104	2			
CAP CER 10000PF 100V X7R 1206	Digikey	1276-1157-1-ND	1			
TERM BLOCK HDR 4POS 90DEG 3.81MM	Digikey	277-1208-ND	2			
IC MTR DRV BIPOLR 8-32V 48HTSSOP	Digikey	296-41245-1-ND	1			
TERM BLK 2P SIDE ENT 3.81MM PCB	Digikey	A98166-ND	1		$9.33	$128.79
1.0 mm Nozzel	Amazon	N/A	1		$0.45	
1cm 3mmx5mm silicon tubing	Amazon	N/A	1		$0.04	
5ft 1/4 in silicon tubing	Amazon	N/A	1		$4.50	
1/4-in x 1/4-in Barbed Splicer Adapter Fitting	Lowes	1/4-in x 1/4-in Ba	2		$7.76	
1/8-in x 1/4-in Barbed Barb x Mip Adapter Fitting	Lowes	1/8-in x 1/4-in Ba	1		$4.32	
1/4-in x 1/4-in Barbed Barb x Mip Adapter Fitting	Lowes	1/4-in x 1/4-in Ba	1		$3.88	
1/4-in x 1/4-in Threaded Male Adapter Nipple Fitting	Lowes	1/4-in x 1/4-in Th	1		$9.38	
0.6-in x 43-ft Plumber's Tape	Lowes	0.6-in x 43-ft Plu	1		$2.14	
OPTIX 0.08-in T x 18-in W x 24-in L Clear Sheet	Lowes	OPTIX 0.08-in T	2		$28.96	
Total Cost					**$433.64**	