ARTAIC ENTERPRISE SOLUTIONS

A Major Qualifying Project Report
submitted to the Faculty
of the
WORCESTER POLYTECHNIC INSTITUTE
in partial fulfillment of the requirements for the
Degree of Bachelor of Science
by

John Depot
Daniel Karol
Demetrios Kechris
Zachary Miller

Date: May 1, 2011

Approved:

Professor Gary F. Pollice, Major Advisor

Professor Stephen S. Nestinger, Advisor

Professor Kenneth A. Stafford, Advisor

Table of Contents

List of Figures

List of Equations

List of Tables

Abstract

Artaic Innovative Mosaic used a slow, expensive and error-prone system to assemble custom tile mosaics. The flaws in the original mosaic assembly system were inherent to the design and no reasonable upgrade would have substantial improvement in performance. This project aimed at developing a fast and accurate robotic mosaic assembly system solution, the TileJet, for the assembly of tile mosaics used in the residential and commercial sectors. The novel design feature of the TileJet system is the use of an array of tile placing mechanisms to assemble mosaics. Having an array of tile placing modules provides for parallel tile placement which significantly enhances mosaic assembly time. The modularity of the TileJet allows Artaic to expand and complete the system as per their requirements.

Acknowledgements

The project team acknowledges Artaic Innovative Mosaic for their sponsorship of the project, and Professors Nestenger, Pollice and Stafford for their continual support throughout the project. The team also wishes to acknowledge Haas Technical Education Center for providing CNC facilities, Solidworks for providing CAD software, Adam Sears and Neil Whitehouse for machining guidance, Joe St. Germain for technology support

Executive Summary

This project was sponsored by Artaic Innovative Mosaic, a startup mosaic design and assembly company in Boston. The goal of this project was to increase the company's production rate by designing and building a new assembly robot. The team analyzed several assembly systems of Artaic's top competitors to gain an insight into existing assembly methods.

Several unique assembly concepts were developed. The majority of the systems were not based on preexisting concepts. The team then compared the concepts to Artaic's requirements using a weight chart. The strongest design candidate was the TileJet parallel printer. The TileJet uses a parallel processing method to place a large number of tiles simultaneously. Individual basic components called print modules move print heads back and forth over a conveyor, placing tiles on an adhesive substrate backing. Each print module is responsible for two tile lines or colors and can also reload itself without waiting for operator intervention. Once all print modules are done placing tiles on their given rows, the belt increments forward to the next set of rows and the process is repeated. Unlike many assembly methods the TileJet can assemble several square feet at a time. By using a modular system, print modules can be added to expand the TileJet up to 510 colors. The time it takes the TileJet to complete a mosaic is dependent on its complexity number of print modules dedicated to the project. The modularity has the added benefit of redundancy. In the event that a print module stops operating correctly it can be removed and taken off the TileJet without stopping production.

The TileJet prototype has a theoretical placement accuracy that should allow the system to place tiles directly on a substrate while appearing to be straight. When complete, it will be able to produce a square foot section in less than nine minutes with lower average run times.

2 Introduction

This project is a Major Qualifying Project (MQP) spanning four terms and three majors: Mechanical Engineering, Computer Science and Robotics Engineering. It is sponsored by a small start-up company in Boston, Artaic Innovative Mosaic (Artaic). Artaic creates custom mosaics in sizes ranging from kitchen back-splashes to the walls of a three story tall building. What makes Artaic unique in the custom mosaic industry is their assembly method.

Artaic uses a robotic system to pick and place tiles at high speeds into square foot sections, allowing high output at a relatively low cost. For comparison it takes a person about one and a half hours of monotonous labor to create a square foot of mosaic with a wide variety of color. Artaic's robot can produce the same square foot in less than fifteen minutes.

The original goal of the project was to improve upon the assembly process by reducing error within the system, increasing the range of usable tile, and creating a suite of computer software to increase production stability. After spending some time testing the current system, a conclusion was reached that any improvements would only result in a minimal improvement in production. As a result, the focus was shifted to creating a new robotic assembly system. This new system would be faster and more accurate than the original assembly system could have been, even with improvements. Additionally, the development of a new system allows Artaic to continue production with both systems once completed, more than doubling their production speed. The new system will fulfill the goals and requirements proposed during the creation of the original system, which were not met in the design.

The system designed by this project used multiple print heads running in parallel to greatly increase efficiency. The speed of the system comes from the fact that each print module is able to simultaneously print a different color on a separate row. Adding more print heads increases the throughput of the system. The extra printer can be used to print a new color or can be used to split the work of the most frequently used color. The control system can manage at most 255 print heads, which can make use of 510 separate tile lines simultaneously. Most mosaics can be completed with only twenty to forty colors, requiring ten or twenty print heads, though more can be added later on. This scalability allows for plenty of system growth. The time necessary to print is only dependent on the complexity of the mosaic and the number of place operations necessary to complete it.   Assuming the system has enough heads to place all of the colors used, it can print a square foot of ten millimeter tiles in under nine minutes.

3 Review and Background

3.1 Artaic Production System

Artaic Innovative Mosaic is a custom tile mosaic design and manufacturing firm that utilizes an online program to allow customers to design personalized mosaics. These mosaic designs are based on uploaded pictures and available stock. The company also uses in-house designers for more complex projects, the largest of which was 460 square feet (). The variety of tiles the company offers when creating the mosaics is numerous, from ten millimeter glass tile, to one inch painted ceramic, all in a variety of colors. The different materials allow for beautiful mosaics that fulfill customer's needs, but require the production system to be more complex.

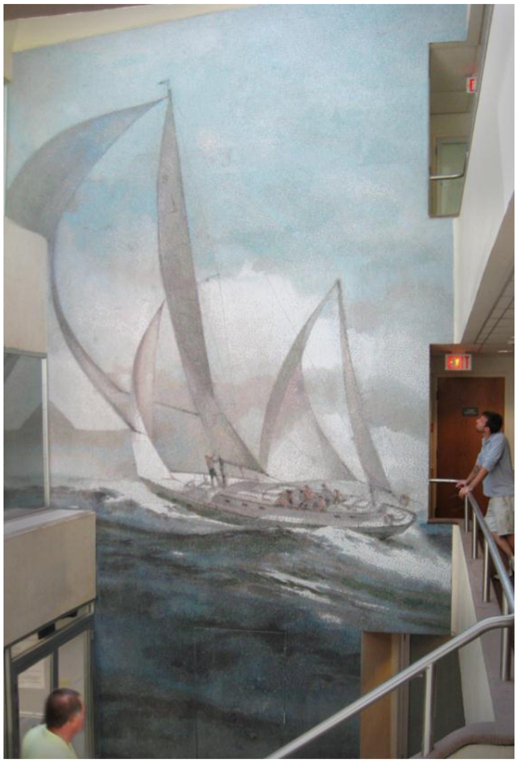
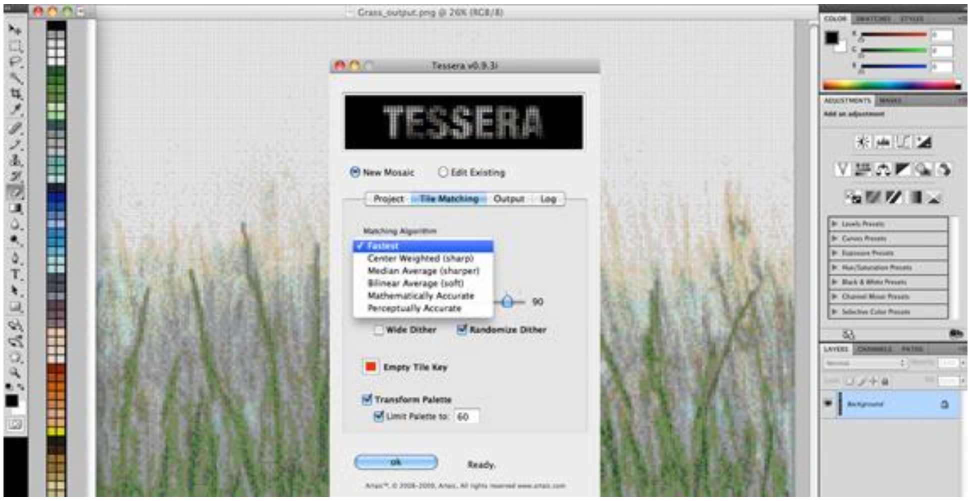


Figure 1: A Completed Mosaic

he online mosaic generator, Tessera (Figure 2), accepts an image file and multiple artistic settings, which it then uses to generate a representation by compiling digital images of actual mosaic tiles to render a preview of the final project. The image is rendered using digital photos of individual tiles. This image is touched up by artists at Artaic, or regenerated with different settings, until the final product is satisfactory. Once a design has been approved, the operator uses Tessera to generate a file which is representative of the design. This file is used to interpret pick and place commands for the robot.



Grass_output.png @ 26% (RGB/8)

Tessera v0.9.3i

TESSERA

● New Mosaic ○ Edit Existing

Project | Tile Matching | Output | Log

Matching Algorithm

✓ Fastest
Center Weighted (sharp)
Median Average (sharper)
Bilinear Average (soft)
Mathematically Accurate
Perceptually Accurate

90

☐ Wide Dither ☑ Randomize Dither

🟥 Empty Tile Key

☑ Transform Palette
☑ Limit Palette to: 60

ok

Ready.

Artaic™, © 2008-2009, Artaic. All rights reserved www.artaic.com

Figure 2:Tessera

The robot arm (Figure 3) is given a text file containing data that instructs the robot which tile lane to take from and where the tile should be placed on a plastic grid. This procedure is very user intensive, slow and complex. It limits the robot's capabilities and does not allow for future production expansion. The file format is restricted to Cartesian grid placement only, and does not allow for arbitrary coordinates or rotation.

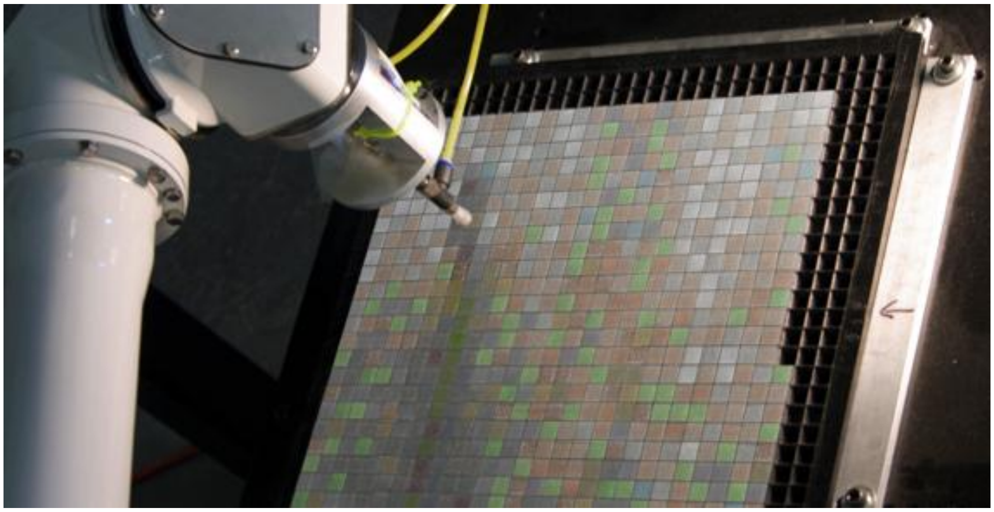


Figure 3: Artaic's Assembly Station

At best the arm is capable of completing a square foot in just over 13 minutes. Due to precision requirements, the arm runs at a slower speed and will complete a square foot in roughly 15 minutes. The production station is the bottleneck in the manufacturing process and cannot be improved upon significantly. The speed and accuracy limitations are imposed by the use of the FANUC H-430iA robotic arm.

At Artaic, the process of preparing tiles for assembly begins by receiving the loose tiles from manufacturers in twenty-pound bags. The tiles are cleaned to remove debris. They are stored in labeled plastic bins and organized by color code on shelves. When a quantity of tile is needed for a project, the tile is checked for defects and stored in extruded plastic tubes. The filled tubes are stored until needed by the assembly system.

The process of operating the assembly robot is not a simple one; there are eight steps before the robot can place its first tile. The first step in running the assembly robot is to initiate the assembly program and start the control computer. The program informs the operator what lanes the colors belong in and waits for the operator to confirm they are in the correct positions. The assembly operator then manually controls the Fanuc robotic arm to locate and record the first lane location. Other lane locations are based on calculations done by the robot control software. The operator places a plastic grid into a cradle in-front of the robot. The grids are different sizes for each type of tile and the grout size. The operator manually controls the robotic arm to locate and record the location of the grid.

During assembly, the robot requires constant maintenance. The lane tubes must be refilled as the robot places tile; the operator corrects misplacements and failures while the machine is running. The pick-up lanes may also become jammed, and require clearing by the operator. These interactions pose a safety hazard to the operator.

Once a grid is filled, the operator removes the full grid and replaces it with an empty one to continue manufacturing. The completed section of mosaic is adhered to an adhesive plastic substrate and the tiles are removed from the grid. The clean grid is now ready to replace the current grid when assembly is complete. During this process the constant manual manipulation and babysitting needed by the robot, slows down the production and creates an error prone environment.

4 Methodology

This project implemented phase driven development. Each phase has a set of goals that must be met before the next phase of development could begin. If one phase failed to be completed then the project would be stalled at that point. There were four phases stretched out over the project period. The first phase was for background research; in this phase the team measured and calculated metrics for existing methods of assembling mosaics. The second phase of this project was the planning phase in which designs and mock-ups were created. The third phase was devoted to developing a prototype of the system. The fourth phase was used for refinement and polishing, in which the robot was to be used by the student team to identify defects in the robot and to document the system.

4.1 Background Research Phase

The background research phase allowed for the study of existing mosaic assembly techniques to determine the best method to improve Artaic's capabilities. Furthermore it allowed the team to become aware of the problems and caveats associated with working with tile. The research phase started with observation and analysis of the current implementation of the assembly line currently in use by Artaic. After gaining a full understanding of the current Artaic system, the team looked into how Artaic's competitors compared and how they accomplished similar tasks. At the end of this phase, the team was able to make an educated decision as to how to move forward with the project.

4.1.1 Current System Analysis

To gain understanding of how the Artaic system could be improved, the team performed an in-depth study of the capabilities and characteristics of their current assembly station. The information that needed to be garnered was related to the accuracy, speed and sources of error. To get an idea of the accuracy of the assembly station several tests were conducted. The first test was intended to reveal whether the arm was operating at peak precision, or if it had some room for improvement. The second test was designed to determine possible sources of error such as vibration, low title tolerances, or jamming. Finally, the last test performed was designed to reveal information about specific sources of error and whether they were causing the problems that were revealed in the previous two stages.

To ascertain if the robotic arm is performing at the advertised tolerance of ±0.5 millimeters, a dot placement test was used. In this test a fine point felt tip marker was attached securely to the robots end effector. The robot was then programmed to follow the same motion path that it would during regular operation to move the pen instead of dropping tiles. Next, a piece of paper was placed on top of the grid that is normally used to receive the tile. The robot was then calibrated so that when in the placing position the felt tip marker would be touching the paper. This test will produce a series of dots on a piece of paper each showing where the center of a tile would be placed by the machine given no external sources of error such as the tile itself or the tolerance of the pick locations. The piece of paper with the dots on it would be scanned and loaded into SolidWorks, a three-dimensional CAD tool where the placement of each point would be measured. The concentric circles are superimposed over the target positions; each circle is one millimeter larger in diameter then the previous ().

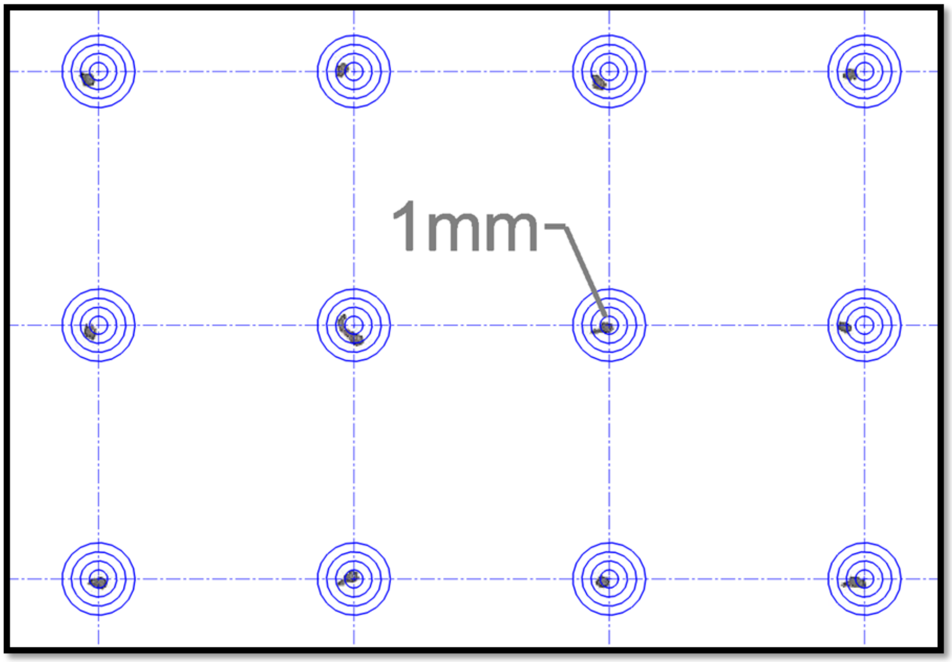


Figure 4: Results of the Dot Test

The second test that was conducted was a full system tolerance check. This test was performed for two reasons. The first reason was to measure the overall tolerance of the system. The second was to give a target tolerance to verify that all sources of error had been found. In this test the robot was loaded with random lines of tile and a random mosaic tile map was loaded into the system. A square foot of adhesive backed substrate was then placed over the grid with the sticky side up, so that instead of allowing the tile to fall into a grid which justifies it, the tile would be stuck in place where it landed. Once all the tiles had been placed onto the substrate another piece was cut out in the same size as the first and placed over the top of the tiles to bind them between the two layers so that there would be no chance of them accidentally wandering, sliding or falling off. Then, the grid of substrate would be scanned in to a computer and loaded into SolidWorks in a similar method as described in the dot test. Then the bottom left-hand corner of every single tile was measured against the others and the discrepancies became the tolerance of the tile placement.

The last operational test was performed to see if there was a link between movement angle and vibration. For this test, a laser was affixed to the end effector in a similar manner as to the felt tip pen mentioned in the dot test. The lights in the room were then turned off and several long exposure photographs were taken while the machine ran the same program used in the dot test. Areas where the long exposure photo of the laser lines that streaked across the placement grid were blurry showed that there is significant vibration along that path. Conversely, when the laser line was clean and crisp there was relatively little vibration in the arm the system.

The tiles were also measured for size tolerance. A handful of randomly assorted 10 mm tiles were measured using high accuracy caliper to determine the distribution of error.

4.2 Design Phases

During this phase of development two key milestones were met. First, several broad system concepts were developed. Then each system was analyzed for run time, functionality, and operational time to determine and select the strongest concept.

4.2.1 Brainstorming

The objective of this phase of the project is to synthesize several independent unique concepts. This phase leverages creativity and independence to form a more successful end product. To ensure each concept is unique and independent each team member created three unique concepts with at least three design generations apiece. Only one of these concepts per person can be based on a pre-existing design. A proper design report includes a detailed drawing of the system that outlines all major components and clarifies the functionality of the system, a description of the design and a list of the pros and cons of the system.

4.2.2 Design Analysis and Selection

Once concepts have been created and elaborated, it is necessary to compare them in order to determine the strongest design candidate. Analysis included four milestones. The first was determining all the important variables that define how one system compares to another. The second milestone was requesting Artaic to create two sets of ratings that describe how these factors interact in terms of importance, quantitatively and qualitatively. The third milestone was assigning a performance score to each variable for each concept. Finally once the first three milestone were completed, a weight chart could be generated using the information from the two previous milestones. The concept that scored the highest on the weight chart would then be chosen for development.

4.3 Development Phase

In order to ensure uniform system development, the hardware and software components of the robot were separated. Two students primarily focused on the development of the hardware aspects of the system while the other two students worked on the software. Because it is much more expensive to develop hardware then software these two separate subsections of the project were handled differently. The hardware of the system was produced using a review based system where the team would only be able to work on a very small and specific portion of the hardware and got approved as a final design. The software on the other hand, was produced using iteration driven development where Artaic specified several aspects of the program to be completed. Funding for materials would be allotted by Artaic at the end of each week based on progress completed.

4.3.1 Progressive Hardware Development

The development of the mechanical portion of the system was driven via an objective and review system. For each objective the team had to demonstrate a completely functional subsystem of the robot, working from simplest and most crucial subsystems to larger more general systems. Funding for prototypes of each objective was provided at the end of the previous review section. This funding scheme prevented the cost of the system from getting exorbitant before a fully working prototype was created. The objective order and descriptions are below:

1. Print head: The most basic part of this entire system, the device that holds the tiles and places them onto the substrate, Does not include the positioning system in any way. The placement tolerance goal was ±0.1 mm.
2. Print module: The subsystem is capable of positioning a print head anywhere along the X-axis of the robot. When combined with the print head a tile must be able to be placed within tolerance.
3. Tandem print modules: The concept of this robot mandates that several print modules will need to operate in tandem. The objective of this development period is to prove that two of them can work together given manual input of a virtual belt.
4. Conveyor: The component that holds all of the individual print modules together and allows them to work in concert. The belt includes the table in the method of moving the

substrate through the robot in the Y dimension. The belt must be able to position itself within tolerance.

5. Conveyor and print module interaction: The last objective that must be met before the full system can be produced is demonstrating that the belt and the tandem print modules can work together and still maintain the tolerances specified in each subsystem.
6. Full System: Producing the deliverable robot with all of its print modules capable of working together.

The first and most important subsystem to perfect before moving on to the rest of the robot was the print head, specifically the module for refilling and placing tiles. Once it was proven that a print head could be reloaded and place tiles within a tolerance of ±0.1 mm, the team would be allowed to proceed to the second stage of development, creating the print-module. After the physical printing mechanism, the most important subsystem of the robot was a single working positioning module for the printer head also known as a rail. The positioning system must first be proven to be capable of placing tiles within a tolerance of close to a 10th of a millimeter. To prove the scalability of the system it must demonstrate that the two print heads are capable of holding tolerance when working in tandem. Once all of these subsystems have been proven the team would then begin work on the conveyor subsystem.

4.3.2 Software Design

Software development on the project used a top-down approach to keep up with the changing specifications of the mechanical systems by encapsulating variability. As more details about the design emerged, the software design became more elaborate, until software components were well-defined enough to be written and tested.

The control software of the project was split into two facets at this time. The system control software manages high-level items, such as delegating tasks to print heads and parsing mosaic information. The robot control software controls the actuators and sensors of the hardware. The separation was done to ensure that the time difference between being able to develop and test hardware would not interfere with software development. The system control software was developed using physical simulations of interface specifications, and the robot control software would be developed based on what mechanical systems were available for testing.

After the mechanical design of the system was completed, the development of software components was timed to coincide with what hardware they would be used with, so that items could be tested immediately as they were completed. For example, if a gearbox with a motor, H-bridge, and encoder were to be finished, the embedded software development focused on those component libraries in the system. As other hardware components were developed, the existing hardware would be tested with the software to ensure functionality.

The system control software was developed with object-oriented, high-level aspects in mind. The hardware was simulated with mock objects, using well-defined and documented interfaces that would be followed when the hardware was ready to integrate with the system. The system design used a master-slave approach, with clearly defined roles and system-knowledge between master and slave. The master would retain knowledge about the entire mosaic, but not about the actions of individual print heads. It would use an aggregation scheme to gather and reduce information about the print heads to a single unit, rather than control print heads individually. Likewise, the slaves would receive self-contained instructions, and operate in their own control loops, rather than simply reporting sensor data back to the master and waiting for actuator control data. The slaves cannot physically interact with each other, and thus would not communicate with each other, relying instead on the master to ensure that a print head does not drop a tile on top of another one.

4.4 Polishing Phase

The polishing phase of the robot occurs when the development phase ends. At the beginning of this phase, there would have been a full working robot with all the necessary hardware and software components. In an attempt to ensure that the system was bug free and the testing phase was implemented, the robot would have been used for an extended period of time, identifying areas of the system that were weaker in their function than others and addressing those problems. Once the bugs were worked out of the system and the final hardware assembly was complete, metrics and tests would have been performed on the system to compare how it performed with the ideal case.

4.4.1 Positioning Tests

The most fundamental source of failure in any interdisciplinary system comes from the failure to accomplish the most essential tasks. For the TileJet the most essential task is positioning the print head. This test accomplishes two goals. It ensures the print head can position itself correctly. With the tolerance information gathered from this stage, a tolerance stack estimate can be built when combined with information from the other tests. The procedure for this test can be described as follows:

1. All tiles were removed from the robot
2. The system was calibrated
3. The system moved between the three following positions without recalibrating
    i. Five inches from the idler bracket docked position
    ii. Five inches from the gearbox docked position
    iii. Third: eight inches from the far dock position
4. For each position, a caliper was used to measure the distance from the print head to the calibration points (found of the idler and gearbox docks)

4.4.2 Placement Tests

The largest source of error on the system likely arises is from the non-uniform shape of the tiles that pass through the diaphragm when the print head is actually printing. To verify that the diaphragm is up to par with the design specification the project, an overall placement test must be conducted. This test also shows the overall system tolerance of tile placement. The procedure for this test was as follows:

1. A thin strip of substrate was placed on a flat and level support directly beneath the printer rail. The substrate must have been at least 13 inches long and two inches wide. The substrate strip was ensured to be parallel to the axis of motion.
2. The printer was loaded with 50, 10±0.5 mm trapezoidal opaque tiles.
3. The system was calibrated.
4. The printer was commanded to place tiles so that a gap of one inch existed between each tile along the length of the substrate
5. The position of the tiles was scanned into a computer using a scanner with a resolution of 600 dots per inch (dpi) or more.
6. The scanned image was loaded into SolidWorks. The size of the picture was to be calibrated by matching a square with one of the tiles so they are both of the same size. Then, two lines were drawn parallel with the line of tiles such that they were tangent with the tile farthest to each side.

7. The distance between the lines was the tolerance of the diaphragm

4.4.3 Control Tests

The final section of the project that must be verified was the scalability of the controller. The overall design of the system called for a nearly unlimited number of print heads to work simultaneously. There were a considerable number of errors and nuances that could arise from attempting to run many print heads at once. To prove the control system was capable of handling the full extent of its capabilities the following procedures were followed.

1. The prototype was connected to the PC as it would be in the final product. Additionally, a simulated print head would be added to the system.
2. A mosaic that uses colors from each logical print head will be printed with the system.
3. The physical and simulated print heads must correctly print the tiles they are assigned.
4. If the control system can handle two logical print heads, the test will be performed with more simulated print heads to check for any scalability issues

5 Development

When building a large system, adaptations and changes to the original methodology occur often. These changes occur for a variety of reasons, ranging from delays to concept reorientation. Furthermore, many design decisions implemented in this project are less than obvious and must be described in detail.

5.1 Concept Creation

The first stage of the development process was to create and refine designs for a new tile-placement system. Concepts would be analyzed for feasibility, reliability, scalability, and ease of use.

5.1.1 Delta Cartridge Placement

This design incorporates a delta-arm robot with a tile-cartridge end effector. It makes use of the high speed and accuracy of a delta robot (ABB) to quickly position the printer cartridge at the target position. It also uses a tile cartridge to reduce the time spent grabbing new tiles. It would only be able to place one tile color at a time, and different cartridges would be required for different tile sizes. It also can only build one section at a time thus is only an incremental improvement over the serial manipulator robot arm.

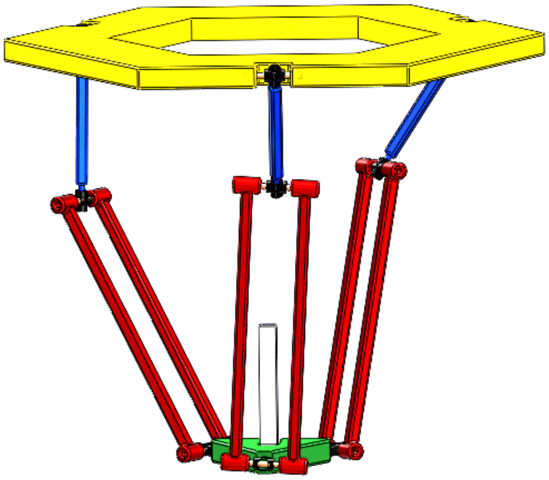


Figure 5: Delta Cartridge Robot

5.1.2 Linear-feed Row Placement

This design places rows of tile at a time, by queuing them in the proper order before picking them up with a suction arm. It works very quickly, assuming the feed mechanism can feed tiles quickly enough, because it places entire rows at once. Its production speed is consistent as well, and does not vary by Mosaic complexity. The complexity of the system is a result of the tile-ordering mechanism, which must deposit tiles in a precise order and line them up well. It is not well suited for changes in tile geometry or grout spacing, as the suction points on the arm would need to be repositioned. It also must be careful to avoid systematic error. If designed wrong, if one tile is missed in the queue the order of all following tiles will be shifted by one and will be difficult or impossible to fix after production is complete. The system would be generally large, and is incapable of placing tiles at an arbitrary position or with a rotational component.

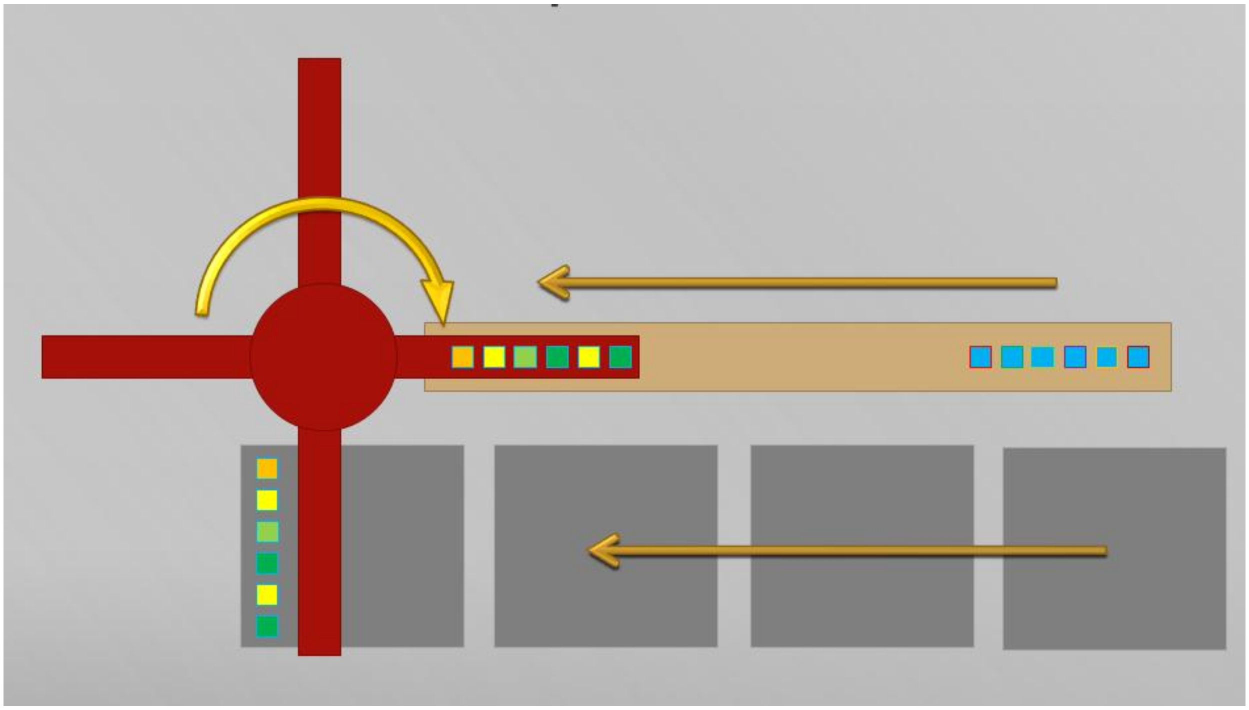


Figure 6: Linear-Feed Row System

5.1.3 Carousel Cartesian Robot

A central wheel of cartridges holds multiple tile colors and rotates one into the placement position as needed. This carousel of colors would be mounted on a Cartesian plotter platform, allowing for accurate placement and adequate payload. The robot's footprint is very small, and can change colors very quickly. Depending on the size of the cartridge, it may not require frequent reloading, and assembly speed can be improved by optimizing placement order. Its color selection is limited by the size of the cartridge, and could only realistically be able to hold sixteen or twenty different colors at a time. It would also need a mechanism to reload or replace carousels as they are exhausted.

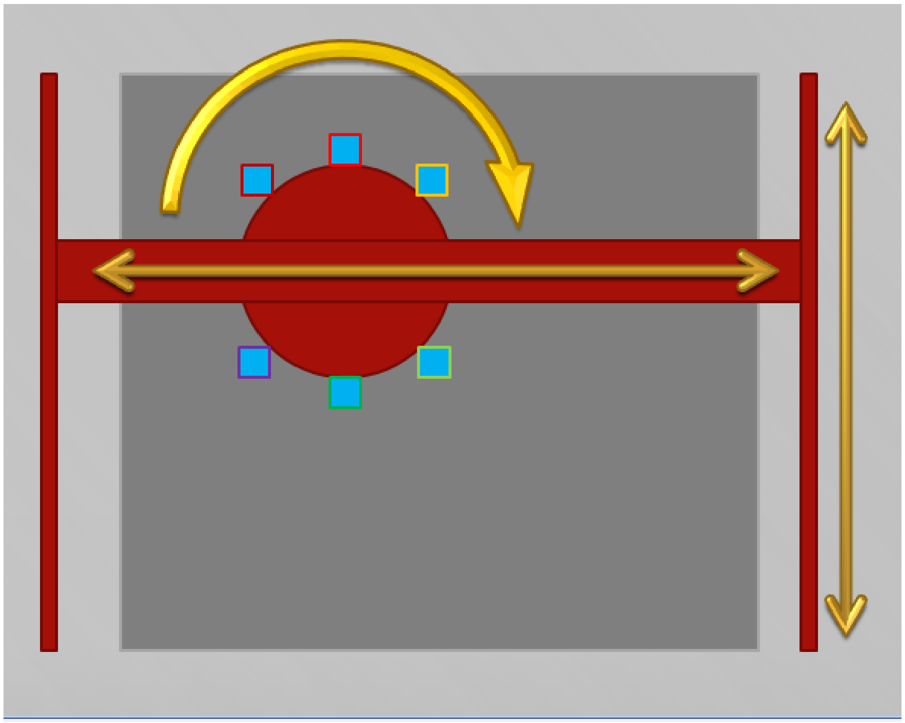


Figure 7: Carousel Cartesian System

5.1.4 Direct Stamper

Another row-placement design, this machine lays an ordered sequence of tiles on a conveyor running underneath an adhesive backed substrate. To affix the tiles, the substrate is pressed onto the tiles on the belt below it. By eliminating the moving arm of the row placing design above, this design is slightly faster, but suffers from the most of the same pitfalls. It is still possible to change the grout size by dropping tiles at larger gaps on the conveyor, but would have difficulty placing tiles at arbitrary positions. An ordered line of tiles is placed directly onto substrate. The placement method can either be moving the substrate onto the line, or moving the line onto the substrate.

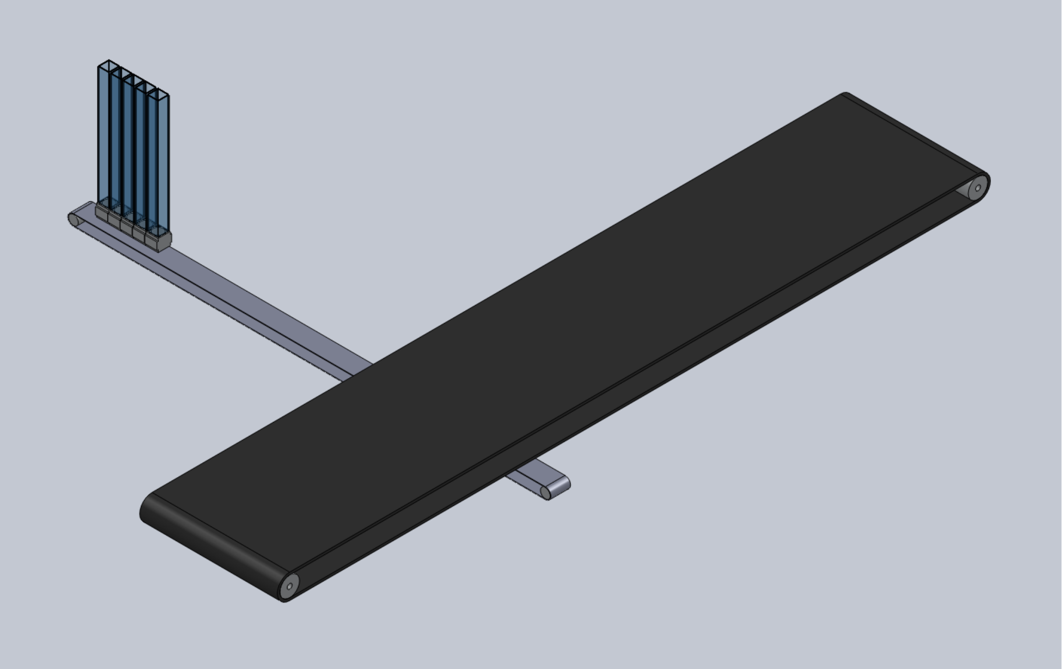


Figure 8: Tile Stamper System

5.1.5 Tile Wheel

The tile wheel concept is loosely based on circuit board assembly robot. This system uses two “wheels” (Figure 9) each with multiple suction cups used for picking up tile. The tile is feed to the wheel presorted in the order it will be placed. As one wheel rotates to pick up tiles the other is placing. When the refilling wheel is full and the placing wheel is empty they switch positions. This system requires a feeding system to feed a continuous stream of sorted tiles to the refilling wheel. Any error within this secondary system such as missing a tile will cause all following tile placements to be off by one space effectively disrupting the following sections. The tile wheel is not capable of arbitrary positioning or placing with a rotational component.

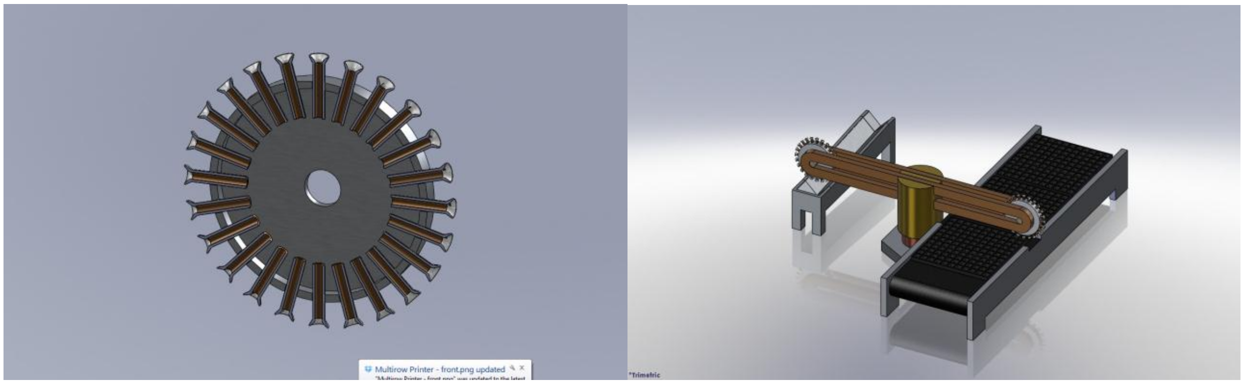


Figure 9: Tile Wheel and Perpendicular Wheel System

5.1.6 Tile Saw

The “tile saw” uses a belt with multiple, self-contained tile placement modules, each containing a suction head capable of axial translation and rotation. It runs perpendicular over a conveyor belt of adhesive substrate. Above the “saw” are tile dispensers for different color tiles. As the belt rotates, the modules grab tiles from the dispensers on top, and drop them when they follow around to the bottom. The substrate conveyor controls the vertical position of the tile, and the timing of the belt determines the horizontal position. As the suction heads can twist, the tile can also rotate, and the system is capable of arbitrary placement at a much slower pace. The belt containing placement modules does not stop moving during production.

By far the most complex design, there are many potential sources of error. The placement modules must pick and place tiles in motion, and any inconsistency in suction or release will reduce accuracy. If a module fails to pick, the robot will have to wait until a free head comes to make another pass before continuing on. The placement modules themselves are also very complex, and may be prone to unreliability.

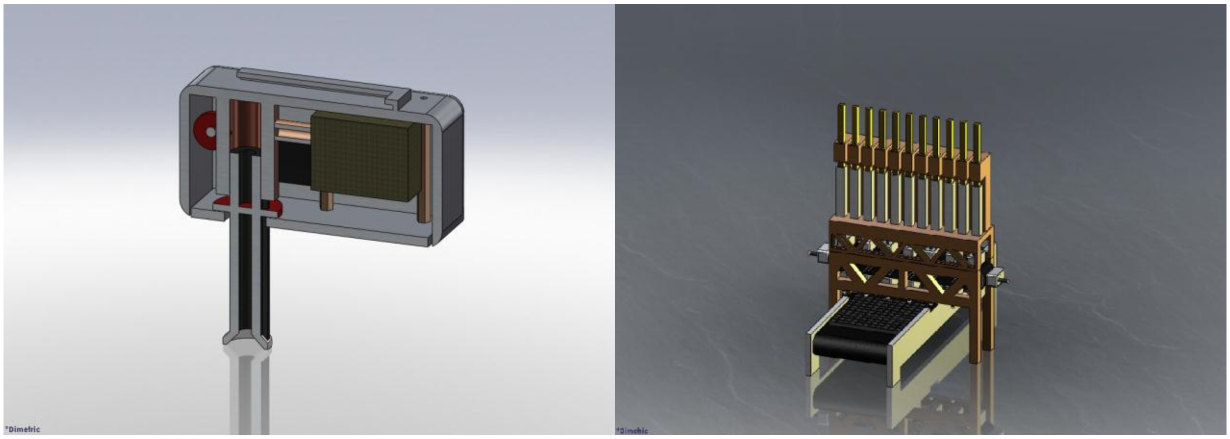


Figure 10: Tile Saw Module and System Design

5.1.7 Multi-row Printer

This design uses multiple independent print heads, each capable of placing a different color tile at a different place in the mosaic. A conveyor of adhesive substrate travels below a line of print heads as they move back and forth, dropping tiles. Its speed arises from its ability to accurately place more than one tile at once. The shared conveyor determines the vertical position, and the print head controls the horizontal. This design has a high throughput; at worst the speed of a Cartesian plotter, at best, a Cartesian plotter only placing one color. If the print heads can rotate the tiles, it is capable of arbitrary tile placement. Production speed can be augmented by adding additional print heads, and further improved with path optimization algorithms. As the print head uses tile cartridges, different cartridges must be used for each tile size, and a reloading system must be developed to load the cartridges.

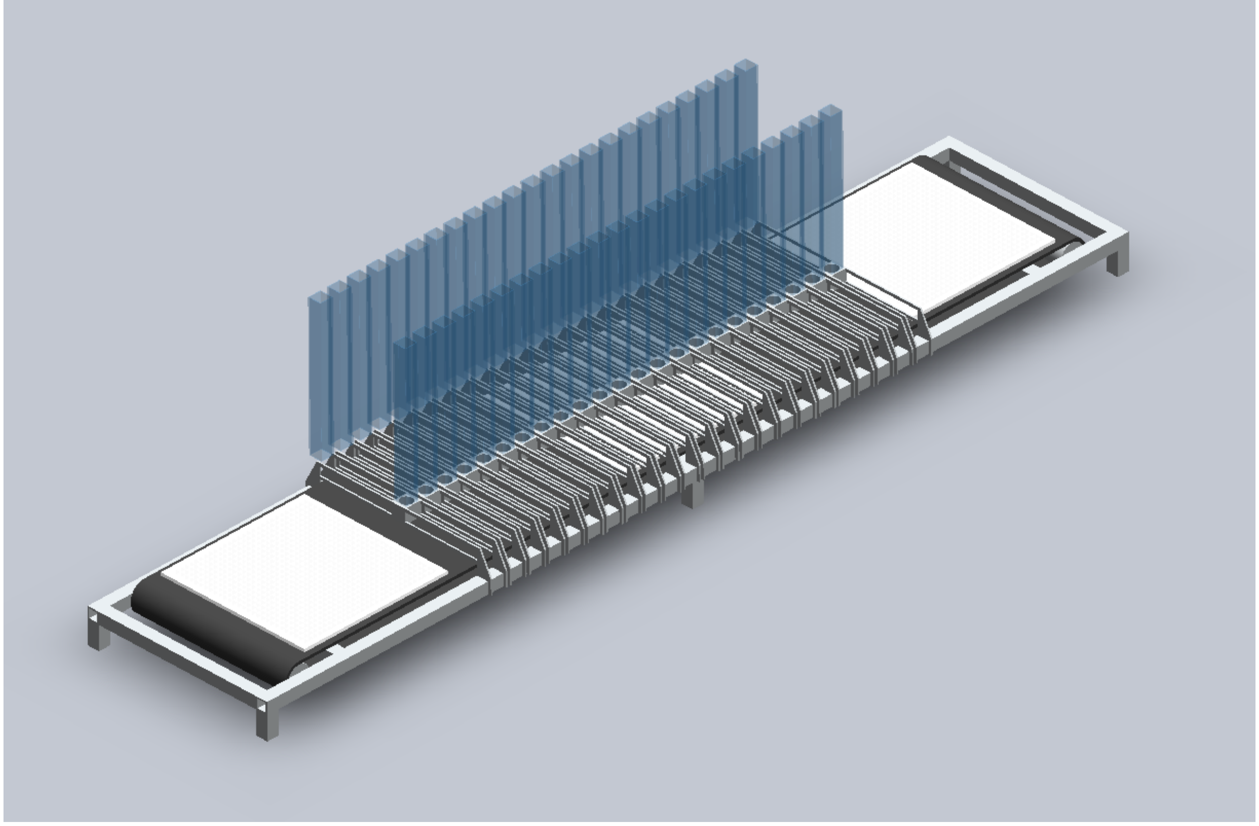


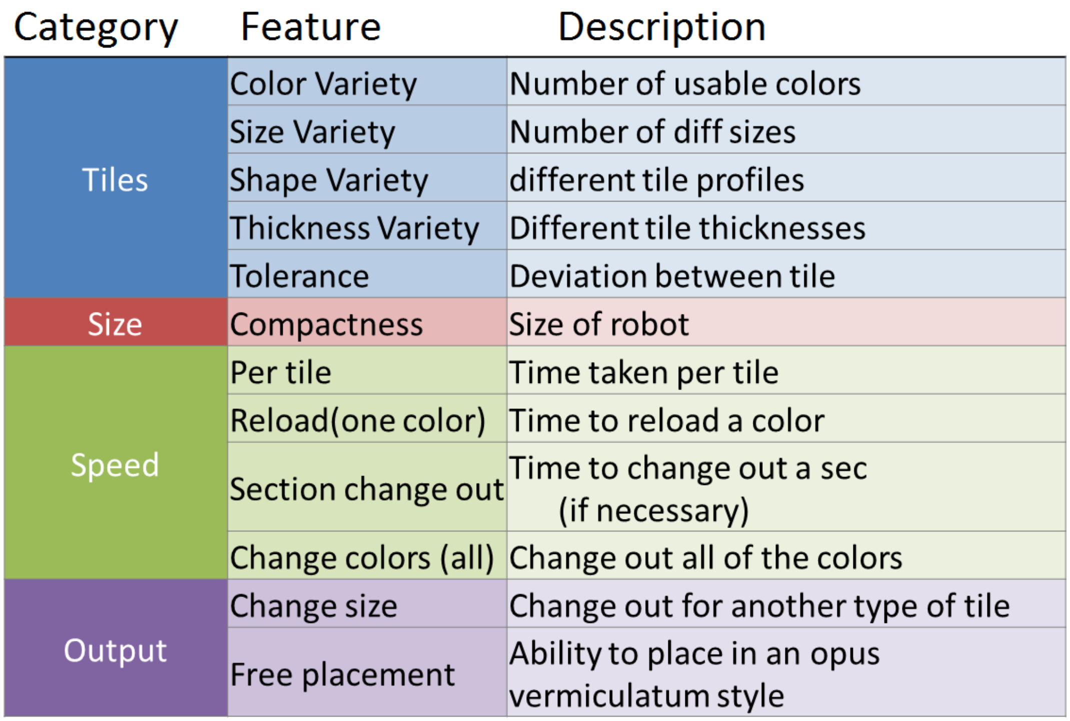
Figure 11: Multi-Row Printer System

5.2 Design Selection

A few tile placement concepts were selected for in-depth analysis and selection through discussion of their general qualities. The selected designs were analyzed for theoretical quantitative features, and compared to each other using feedback from the customer regarding how important each feature was. The most important factors as indicated by the customer were the speed of the system and the number of different usable tile lines. As such, designs that maximize these criteria perform well in the analysis.

A weight chart was used to compare the analyzed designs against the customer's specifications. The design that performed with the highest performance score was selected for the project.

Table 1: Weight Chart Categories



Category	Feature	Description
Tiles	Color Variety	Number of usable colors
	Size Variety	Number of diff sizes
	Shape Variety	different tile profiles
	Thickness Variety	Different tile thicknesses
	Tolerance	Deviation between tile
Size	Compactness	Size of robot
Speed	Per tile	Time taken per tile
	Reload(one color)	Time to reload a color
	Section change out	Time to change out a sec (if necessary)
	Change colors (all)	Change out all of the colors
Output	Change size	Change out for another type of tile
	Free placement	Ability to place in an opus vermiculatum style

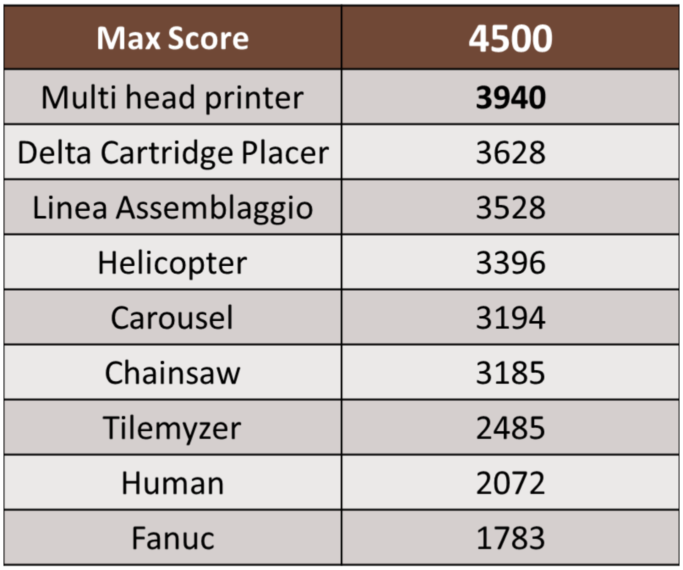
The customer was given a one hundred ‘points’ of importance to assign between features listed in the chart. Each feature was rated from one to ten for each design. These ratings were scaled by the number of points the customer assigned to determine their final score. These scores were added up to find the design with the highest system score. A system’s score on any given feature could be described as:

Equation 1: Weight Chart System Scores

$$Concept\ Feature\ Score = Quantitative\ Score * Qualitative\ Score * Feature\ Rating$$

$$Full\ System\ Score = \sum Concept\ Feature\ Scores$$

Table 2: Weight Chart Scores



Max Score	4500
Multi head printer	**3940**
Delta Cartridge Placer	3628
Linea Assemblaggio	3528
Helicopter	3396
Carousel	3194
Chainsaw	3185
Tilemyzer	2485
Human	2072
Fanuc	1783

With a score of 3940, the multi-head printer is the strongest design candidate and was moved to further stages of development. As an interesting side note, the Fanuc arm in use by Artaic scored lower than a human, due to the high importance placed on versatility and accuracy.

5.3 Proof of Concept

After selecting the multi-head printer method as the strongest design candidate, it was necessary to prove that the each of the major components of the design was feasible. This phase also served as a method of testing one way of doing something against another. The two most difficult tasks that needed to be proved possible before the team could continue were placing tiles and reloading the robot.

5.3.1 Tile Placement Methods

Controlling non-uniform, lightweight tiles is no easy task. When forces are applied to tiles with a trapezoidal profile, they have a tendency to shingle, or stack on top of each other inside their storage tube, making them unusable in most cases. To prove it is possible to manipulate more than one tile without using a vacuum system, Artaic felt it was necessary to have a working proof of concept showing the ability to convert their tile storage tubes to stacked tubes successfully.

5.3.1.1 Indexing Wheel Method

The indexing wheel method was created in an attempt to use the preexisting tubes instead of using custom cartridges. The tube is inserted into the print head vertically so that each tile enters the system stacked on the thin edge of the one below it. This method is resilient to tiles stacking up on each other in strange ways because the tube of tiles feeds into an indexing wheel that allows only one tile to progress through the system. That tile falls into a diaphragm waiting area where it is pushed onto the substrate or grid with the use of a linear actuator. Three beam detectors used in this design: one to detect a tile in the diaphragm and the other two to detect when a tile passes by an indexing wheel.

A modification of this method exists where instead of using an indexing wheel, a small linear actuator would be used to push a single tile into the diaphragm.

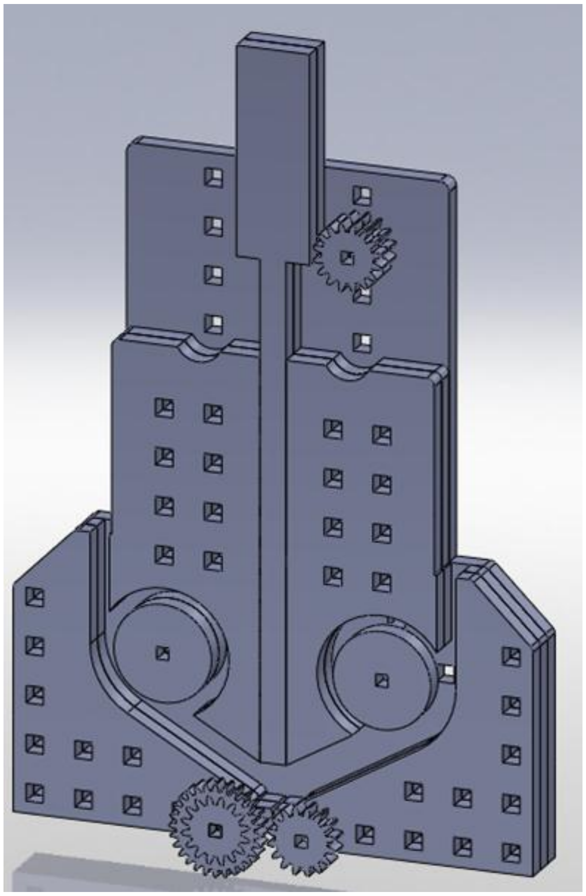


Figure 12: Indexing Wheel Print Head

5.3.1.2 Linear Actuator Method

The “push-down” method uses two independent linear actuators to push a stacked column of tile through a diaphragm. This method requires the tiles to be correctly stacked and assumes that every tile is sitting in the correct orientation. Each cartridge would contain a specific tile type. To place a tile of that color into the mosaic, the system would move the print head to the intended location, and the linear actuator would push a tile through the diaphragm and onto the substrate. To detect when a tile had successfully dropped this method would either use a pressure sensor located at the tip of the linear actuator or a broken beam detector underneath the diaphragm. The original design used a modified rack and pinion as the linear actuator, reducing the weight and cost of this print head. The rack and pinion was later replaced by a threaded rod and nut.

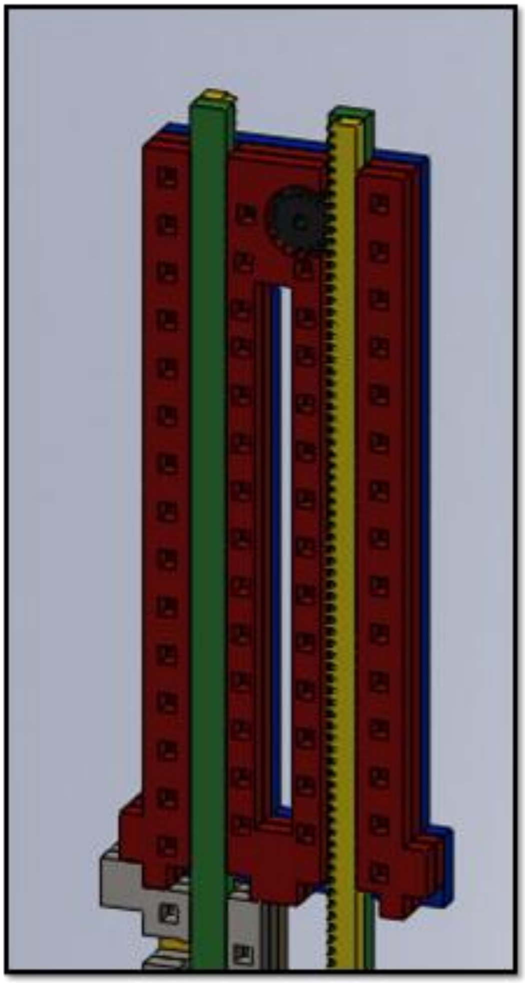


Figure 13: Linear Actuator Print Head

5.3.2 Cartridge Filling Methods

Most of the tile placement concepts required tiles to be loaded into the robot in a different configuration than what Artaic uses for their Fanuc assembly station. In order to use these designs a method for filling the cartridges was to be developed.

5.3.2.1 Push-up Cam

The sawtooth shaped CAM pushed tiles up into the cartridge. In order to work properly each CAM tooth must be shorter than the length of the tile and the height must be taller than it. Short stocky tiles require a rather steep and jagged sawtooth pattern and more force to load them, along with a higher chance they could get twisted and jammed. This method could operate with a continuous stream of tiles entering the system and doesn't matter if it gets backed up.

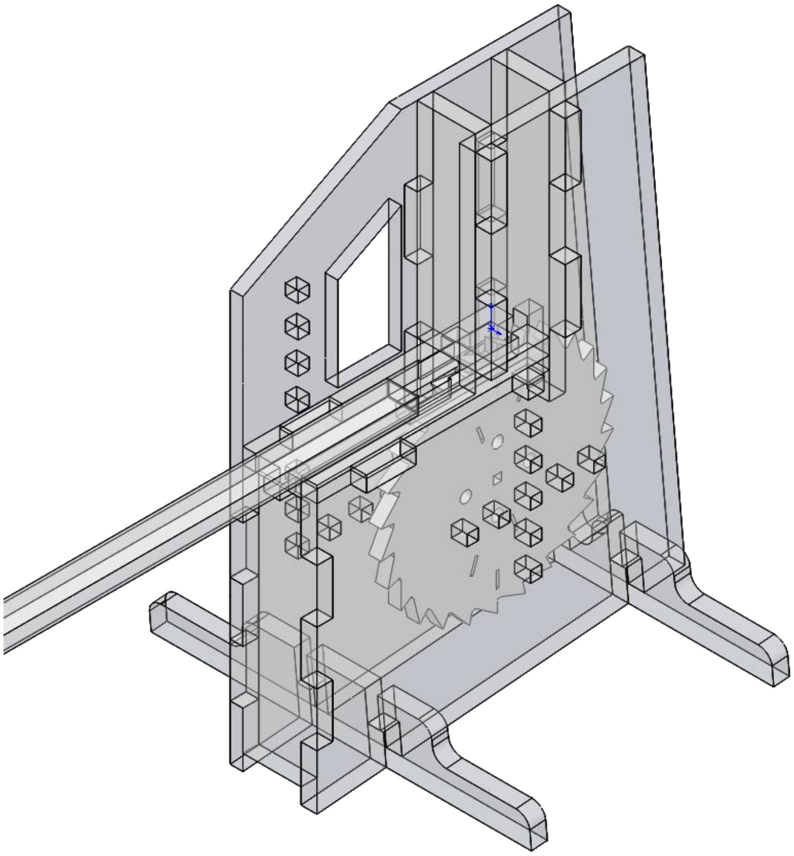


Figure 14: Push-up Cam Filler

5.3.2.2 Spiral Cam

The spiral cam gets its name from a spiral shaped slit and a large drum. The tile enters near the center of the drum and as it spins the tile is pushed outwards. The tile is prevented from translating sideways at its edges by the support plates. The center of the tile get pushed up by the spiral cam and the outer edges push against a vertical wall. One of the benefits of this system is that it can continuously run without the use of any sensors and will not get backed up.

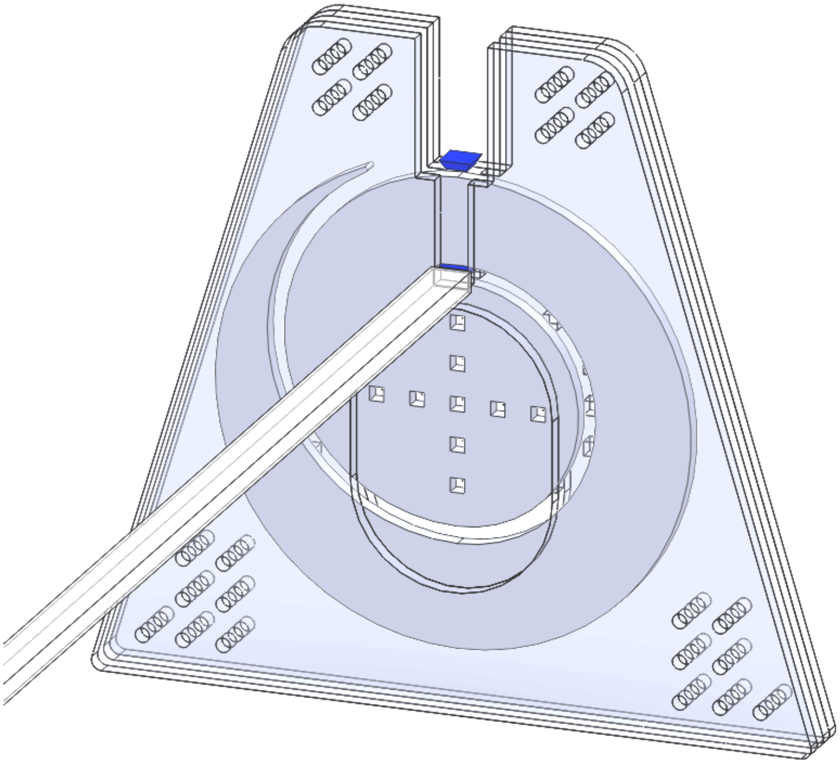


Figure 15: Spiral Cam Filler

5.3.2.3 Push-up Linear Actuator

Using the same linear actuator method used to dispense tiles; this system consists of a linear actuator rod inside a square storage tube. This system allows tile to slide from the inspected tube storage into a stacked position. When a tile slides into the tube and is detected on top of the stack, using a broken beam detector, the linear actuator moves down to accommodate another tile.

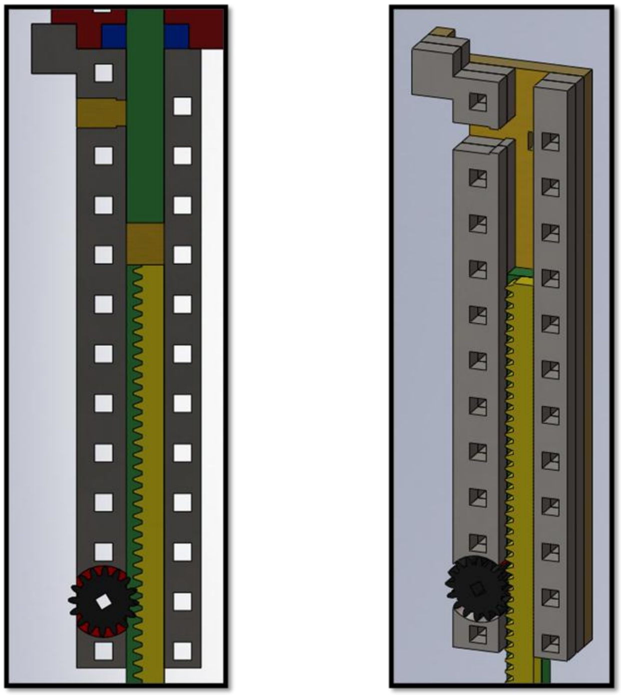


Figure 16: Push-up Filler

5.3.3 Print Head Reloading Methods

When the print head runs empty of tiles it must be refilled in order to continue operation. To reduce the time a print head must wait for its refill before returning to printing tasks the refilling process will be automated. These reloading methods all assume that the printer uses a cartridge based placement method.

5.3.3.1 Pass Through

This concept uses a linear actuator in combination with a broken beam detector to allow tiles to slide into a temporary holding cartridge from a preexisting storage tube. To reload, the print head would position itself above the tile stacking mechanism where the print head's linear actuator rod would push down to meet the top of the stack of waiting tile. Once the tile has been secured by the rod, the bottom linear actuator would push the stack of tile up into the print head, and the constant pressure provided by the rod above would minimize movement of the tile. Once the print head rod has reached the filled point the reloader would descend with the excess tiles. This reloading system would have the capacity to fill a print head multiple times so that an operator does not have to constantly refill each lane. The operator would instead place an inspection storage tube into the refilling system and be able to continue working while the tube was emptied into the stack. This method allows continuous production with minimal delays caused by operator intervention.

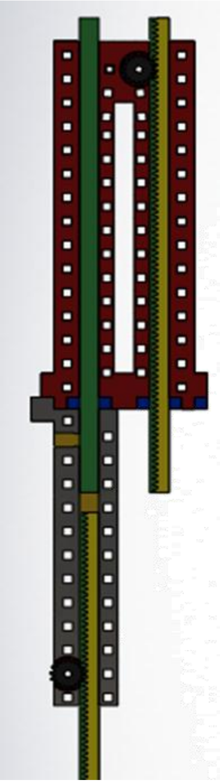


Figure 17: Pass Through Reloader

5.3.3.2 Magnetic Cartridge Swap

The magnetic swapper is a simple low profile system to grab and release cartridges without the use of a motor. A pair of circular magnets would have been affixed to each cartridge, one at the top and one at the bottom. These magnets would have had a groove around their circumference at the midpoint of the cylinder (Figure 18). A push-pull electromagnet would be affixed within a spring loaded latching mechanism. The electromagnet is used to engage and disengage the magnet when a cartridge is empty.

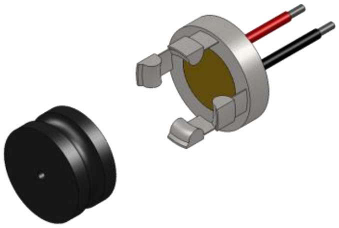


Figure 18: Magnetic Cartridge Locking Module

5.3.3.3 Lead-screw Cartridge Swap

The lead-screw method is similar to the magnetic swapper in concept, but instead of using an electromagnet to grab the cartridges, it uses a lead screw and nut system (Figure 19). The cartridge features a threaded hole set near its top and bottom. On the print head there is a bolt with the same thread pattern connected to a small motor. To pick up a new cartridge the print head slowly drives into the cartridge with the lead screw spinning. When the teeth engage the cartridge is pulled into the robot. To drop the cartridge, the lead screws are spun in the opposite direction until the cartridge falls free.

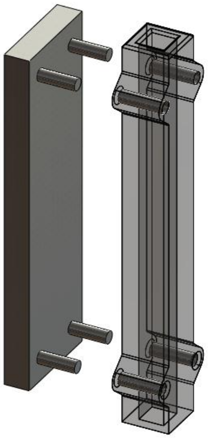


Figure 19: Lead Screw Cartridge Swapper

5.4 Initial Prototyping

Before a full print module could be started, a prototype print head must be built. This prototype must demonstrate all functional components and be capable of operating as mandated by the concept. The linear actuator method of placing tiles was chosen for further development due to its compact form factor and conceptual simplicity. The push up cartridge reloader was chosen because it was very similar to the tile placement method.

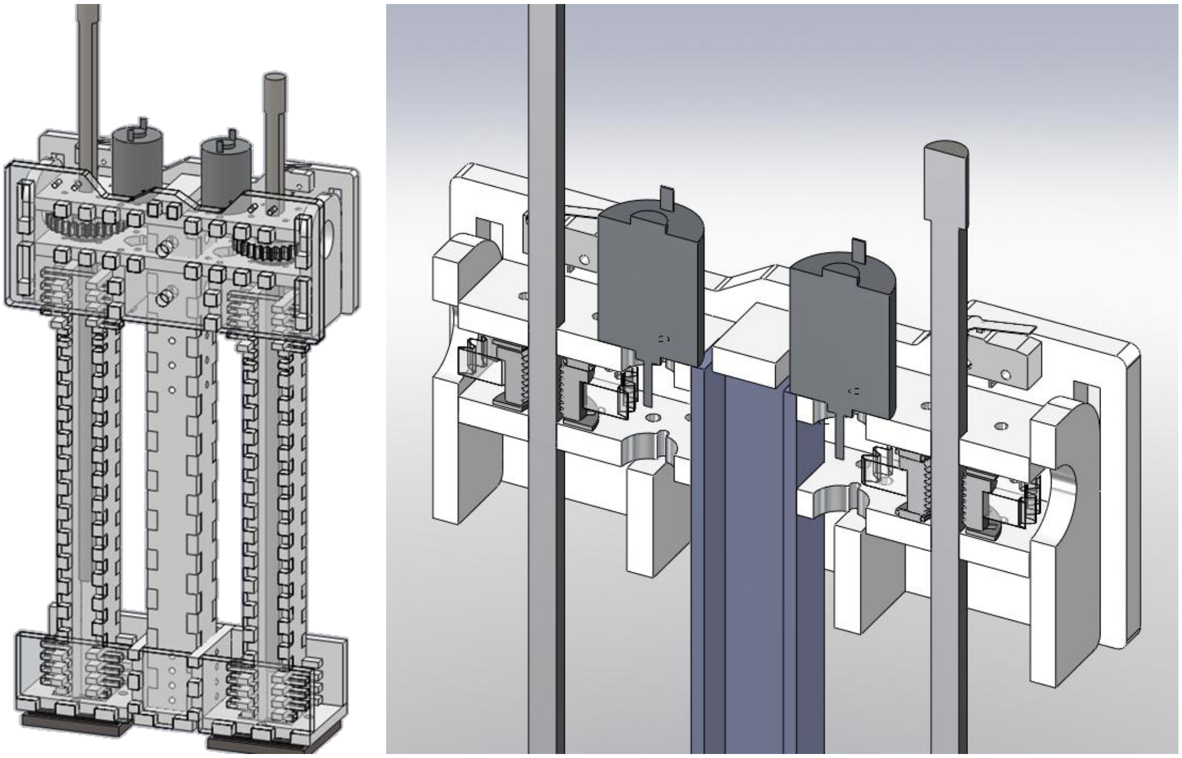


Figure 20: Print Head Prototype

The TileJet uses many print heads in a single system. With such a large number of print heads a small change in the cost of one module or component one could have a large effect on the cost of the total system. In order for this project to be affordable each print head must be made as inexpensively as possible. The linear actuator module forms the core of the print head. It pushes tiles onto the substrate and contains the print head's only two degrees of freedom. The proof of concept for this module used a rack and pinion drive to push tiles out. Unfortunately, a rack and pinion drive requires a gearbox of some sort in order to apply enough force to eject a tile or retract the rack. A possible solution to the problem was to switch to a threaded rod with a worm gear mounted directly onto the motor’s output shaft. This design had drawbacks as well;

a worm that is larger than the motor that drives it is likely to be very expensive and possibly too large for the motor to easily turn. This problem can be solved by using a nut instead of a worm as the device that transmits the power to the threaded rod and placing a gear around the nut (Figure 20). When the gear is turned, the nut turns with it and applies a force on the shaft. To prevent the shaft form spinning, so only the nut is capable of revolving two flat surfaces where milled into the threaded rod with a matching D slot on the housing. Finally, a pinion was press fit on the motor shaft to drive the nut-gear combination.

To center the variable sized cartridges, overhanging flanges where built into the cartridges that press up against the side walls of the print head. A gate was featured on the linear actuator housing to allow the removal of the cartridges easily by hand to switch the system over to another tile line. To remove a cartridge, simply pull the gate up, out of the way and retract the linear actuator and the cartridge comes free.

6 The Final Design

The final design of the Tilejet (Figure 21) incorporates many innovations over its predecessors. It includes refinements to the structure and assemblies of the print head, as well as provisions for the remaining conceptual mechanisms: the rail and reloader. The refined print head and rail system were fabricated and assembled, and designs for the reloading system were formalized.

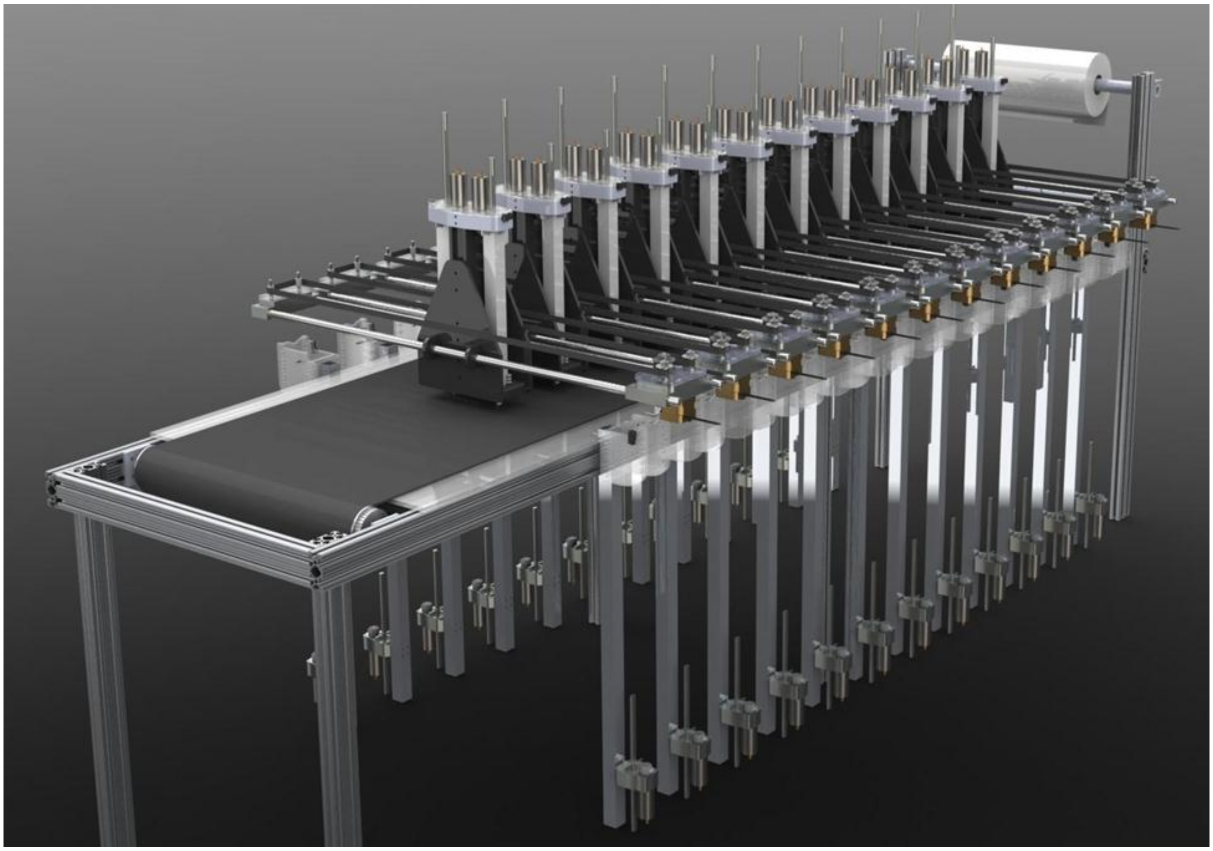


Figure 21: The TileJet Mosaic Printer

6.1 Mechanical System

The print-module's mechanical system was created with three subsystems, each handling a different task. The print head is the central system that the entire device is built around. It is responsible for carrying and depositing tiles. The second subsystem is referred to as the rail, which controls motion in the X-axis. The job of the X-axis is to position the print head where it needs to be to can take whatever action it needs to perform. The final subsystem is the re-loader, which accepts additional tiles into the system and passes them into the print head when it is nearly empty. The final prototype was created with resilience and precision in mind. In earlier iterations, machining and build time were paramount in the overall design to ensure the concept could be tested. As such several changes were made in construction strategy of the underlying components.

6.1.1 Print Head

The print head has undergone more revisions than any other component of the print-module (Figure 22). The current design is an accumulation of all the innovations and insight gained through the development of the project. The print head is made of several sub-components. The workhorses of the print head are the linear actuators. These include the threaded rods that push the tiles out of the system, the motors that drive them and the instrumentation needed to in order to properly integrate the mechanical system with the controller.

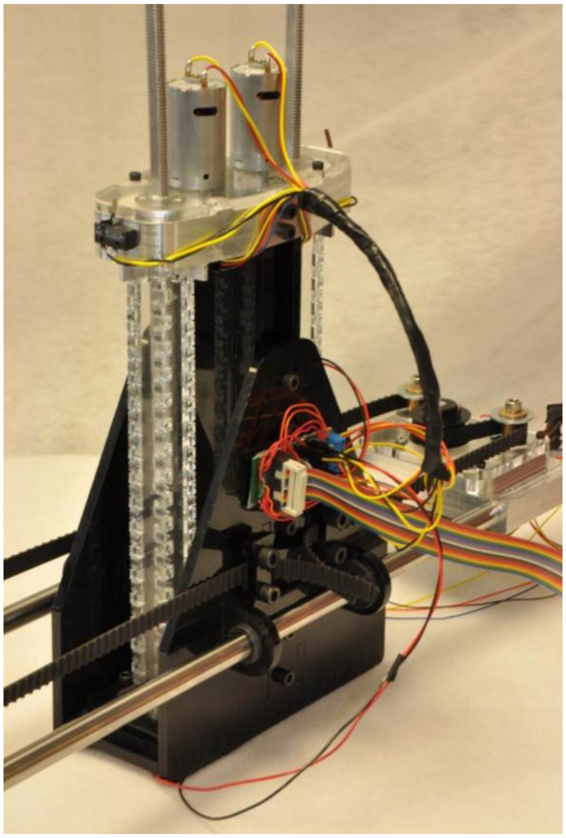


Figure 22: Completed Print Head

6.1.1.1 Linear Actuator Module

Of all the changes made between the initial iterations of the mechanical system and the final design, the linear actuator module's changes are the most profound. In an attempt to reduce the slop of the system, reduce the weight and improve its tolerances, the housing was switched from acrylic to aluminum (Figure 23). This change in material caused a shift the design mentality. Instead of designing parts as plates that would interlock with dovetails or standoffs, parts would be designed with unidirectional extrusions that would clamshell the internal components together with a top plate. By designing parts in this method, they would be machineable with two or three operations depending on how easy it was to fixture the part. Because these parts were machined on a CNC milling machine instead of a laser cutter, the precision and tolerance was greatly improved once feeds and speeds where correctly set. This design shift allowed the parts to become more form fitting so there was less unneeded material.

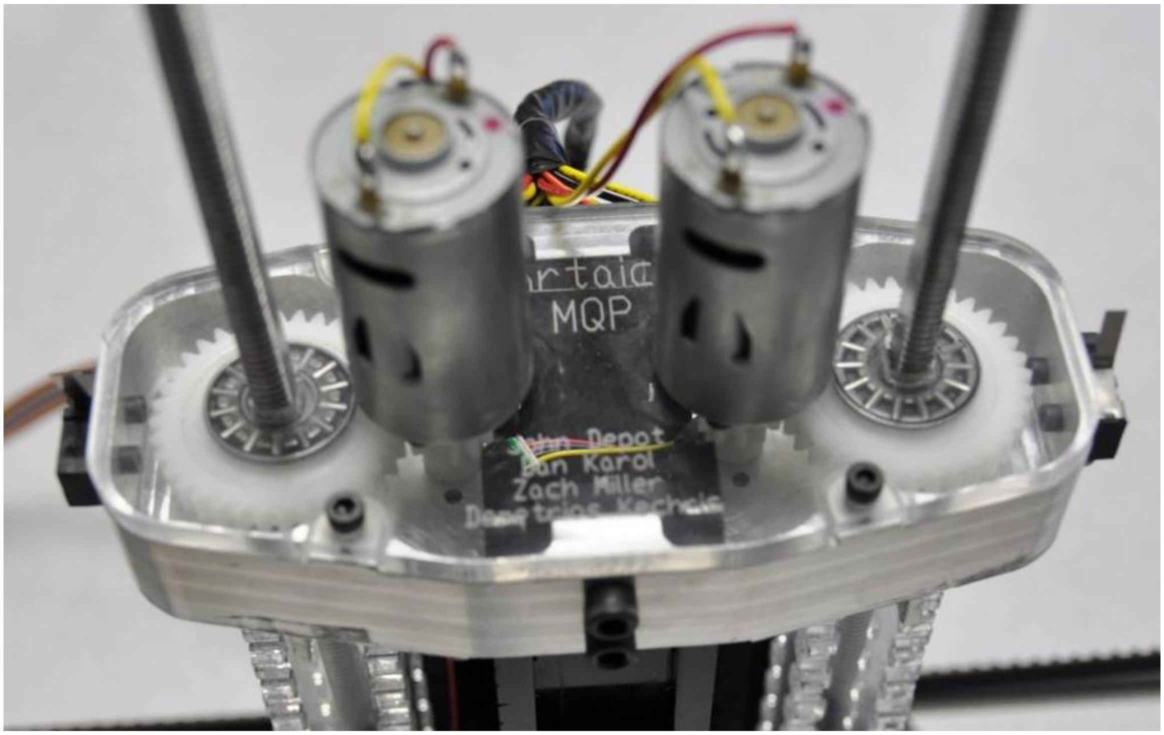


Figure 23: Linear Actuator Module

The new design prevented the original cartridge locking system from being used because it is impossible to machine a pocket with sharp inside edges. To adjust for this change, the design of the cartridge locking system was changed to a cross shape that would intersect where the threaded rod's hole was placed (Figure 24). This method of locking in the cartridge in place removed the need to have overhanging pegs to fix it in place and allowed the cartridges to be the same size as the housing they connected to. The only problem that arises from changing to this method of securing the cartridge tube is there is no longer any gate to easily remove the cartridges. To compensate for this change, the design was modified to enable the entire linear actuator module to be pulled off the top of the print head by removing two locking pins.

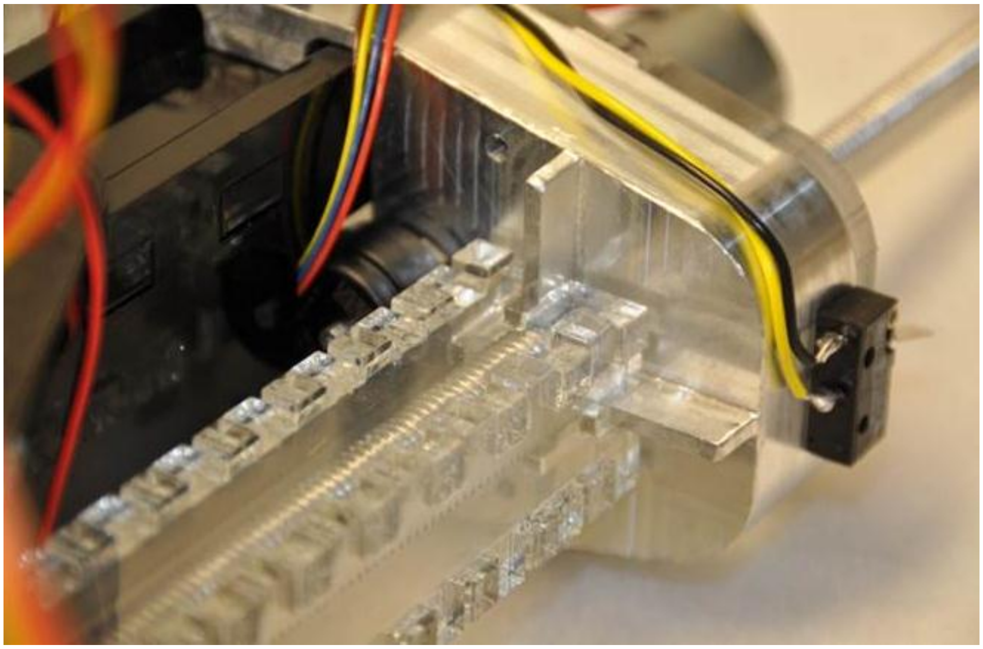


Figure 24: Cross Cartridge Locking Mechanism

Once the housing was improved, the tolerances of the moving components could also be addressed. Instead of cutting custom gears with a hexagonal axle bore for a hex nut on the laser, a gear was purchased from a supplier. To ensure the gear had a low coefficient of friction on the axle, the gear could have either been made out of brass or plastic. Due to the difference in cost, weight, lubrication needs and machining difficulty, plastic was chosen. The gears were threaded so they could be used directly on the threaded rod instead of using a hex nut. To ensure the gear doesn't move vertically or generate a lot of friction when it spins, two needle roller thrust bearings were added above and below the gear.

To ensure the linear actuator doesn't overrun its bottom bound, a limit switch was added to the top of the print head. The lever arm that hits the limit switch is threaded so the throw of the threaded rod can be calibrated.

To prevent the threaded rod from overflowing its top bound a quadrature encoder was connected to the driving motor axle. The quadrature reports 400 counts per motor revolution which equates to 6400 counts per vertical inch giving an accuracy of roughly one ten-thousandth of an inch per tick The encoders can handle speeds three times higher than the motor can produce.

When the previous iterations of the linear actuator where assembled and tested, they failed to extract the threaded rod from the diaphragm without stalling. Even though those tests were conducted with limited current draw, the motors would need to be selected for continuous operation without overheating. To adjust for these needs the motor used for the linear actuators was upgraded from the Banebots RS280 to the RS390.

The last noteworthy change to the linear actuator module was switching the threaded rods from 10mm diameter steel to ⅜" diameter aluminum. This change was made for two reasons: it is much faster to machine aluminum, and saves time when creating multiple print modules, and by switching to SAE sizes, all sizes and dimensions on the product were of the same standard.

After making these changes the overall weight of the print head was reduced from 1.8 pounds to 0.96 pounds, lowering the inertial resistance of the print head considerably.

6.1.1.2 Diaphragm

The print head cartridge requires a means of precisely dispensing a single tile from the stack, accomplished by a rubber diaphragm, seen in Figure 24. To center the tile underneath the diaphragm regardless of where it was in the cartridge, an 'X' shape was used. As the tile was pushed through the diaphragm, its corners slide in the grooves formed by the cross, centering it and squaring it. The corners of the cutout experienced large amounts of strain risking a tear when tiles were passed through the diaphragm so relief holes were added. Earlier designs of the diaphragm came to a point at the center of cross. These points would cause the tile to release with a side force and push the tile to the side. To fix this issue, the tips of the diaphragm where squared. The mounting holes are placed in the center of the sides to prevent the diaphragm from distorting in an undesired manner when loads where applied.

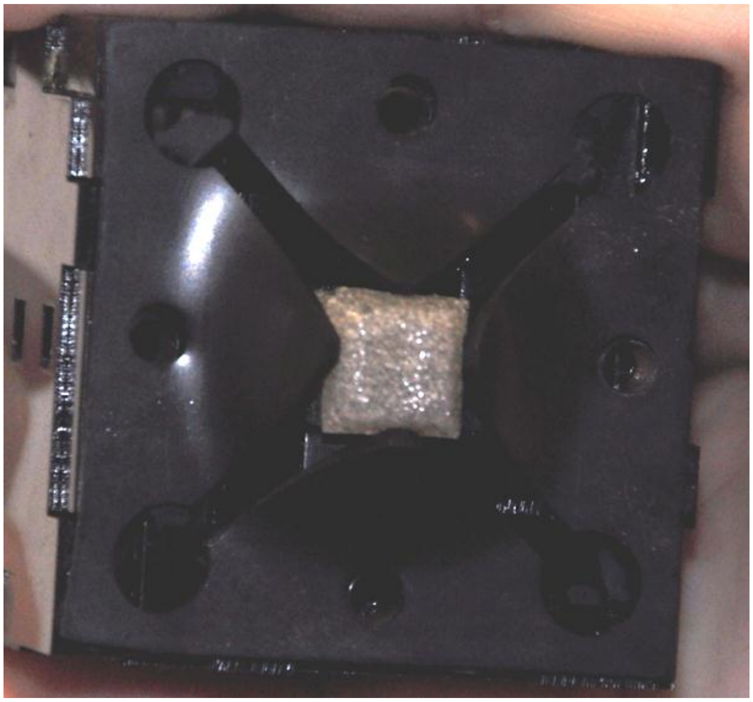


Figure 25: Diaphragm Mounted on a Test Rig

To test the effectiveness of the tile diaphragm design a test rig was designed to emulate a tile storage tube. The tests were carried out by hand, using a stack of tiles pushed through the test rig.

Diaphragms were originally cut by hand based on the CAD description. These prototypes were not accurate enough to provide usable data, but did prove useful for qualitative analysis. To achieve more reliable performance, a laser cutter was used to cut new diaphragms for testing.

In order to form a strong bond between the tile and substrate, the tile must hit with enough force. Furthermore, any inconsistency in the diaphragm edges or stack pressure could cause the tile to be released at an angle, reducing the accuracy. Through testing of different diaphragm heights, it was determined that the fall height should be as close to the substrate as possible without touching placed tiles. With these factors balanced, the diaphragm was able to place tiles within a tolerance of ±0.8mm at a height of 10mm from the substrate.

6.1.1.3 Broken Beam Detector

An optical detector was used to detect the tiles that exit the print head without disrupting their motion path. Most industrial beam detectors are very expensive, priced at around $250 per receiver and transmitter pair. In order to keep the project within budget, the beam detector used in this system was made using a laser module, designed for a laser pointer, aimed at a photoresistor (Figure 26). When no tile is present, the laser blinds the photoresistor. When a tile is present, beam is blocked, and not detected by the photoresistor. The beam detector system is mounted along a diagonal split in the diaphragm so the tongues of the diaphragm never interfere with the signal.

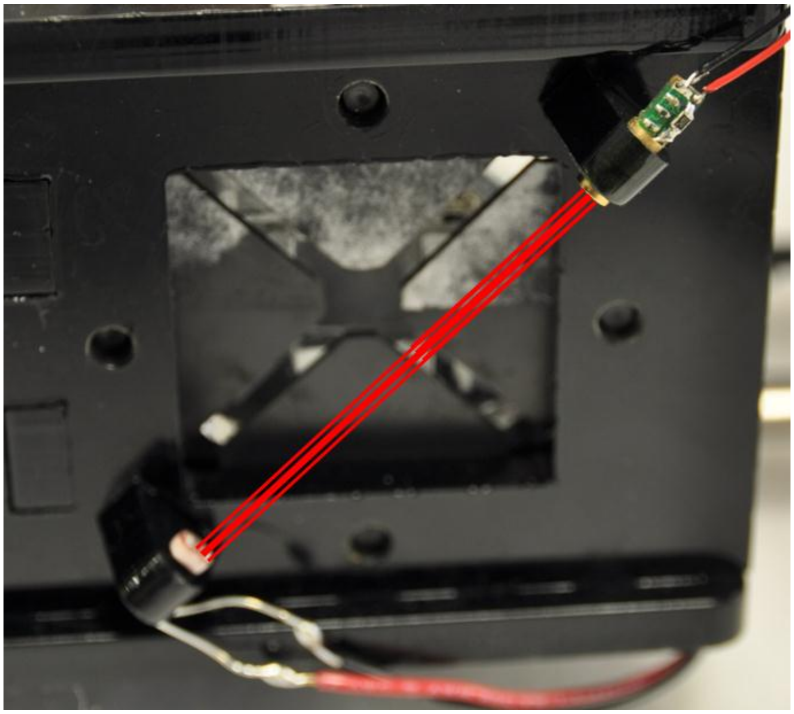


Figure 26: Broken Beam Detector with Laser Illustration

6.1.2 Support Structure

The main structure of the print module consists of the diaphragm support plate, central column and side plates (Figure 22). The support plate holds the cartridge, diaphragm and broken beam detector. The central column connects the support plate to the linear actuator housing. The linear actuator housing can be quickly removed from the support structure to allow for cartridge swaps. Two brackets on each side plate connect the print head to the steel rails. Acetel bushings mounted inside the brackets slide along the rails with minimal friction.

The rail system uses a belt clamped to one of the side plates to move the print head along the X-axis. The slop in the belt can be reduced by drawing up the slack before clamping it down. The belt must be kept taut for the rail system to work effectively.

6.1.3 Rail System

The rail system includes all of the components necessary to translate the print head back and forth along the X-axis of the machine. The rail system is comprised of four components: the set of precision rods that the print head rides along, the belt that holds the whole system together and drives the print head back and forth, the gearbox which serves as a support for the rails and drives the belt, and the end bracket that acts as a belt routing point and also supports the precision rods (Figure 27).

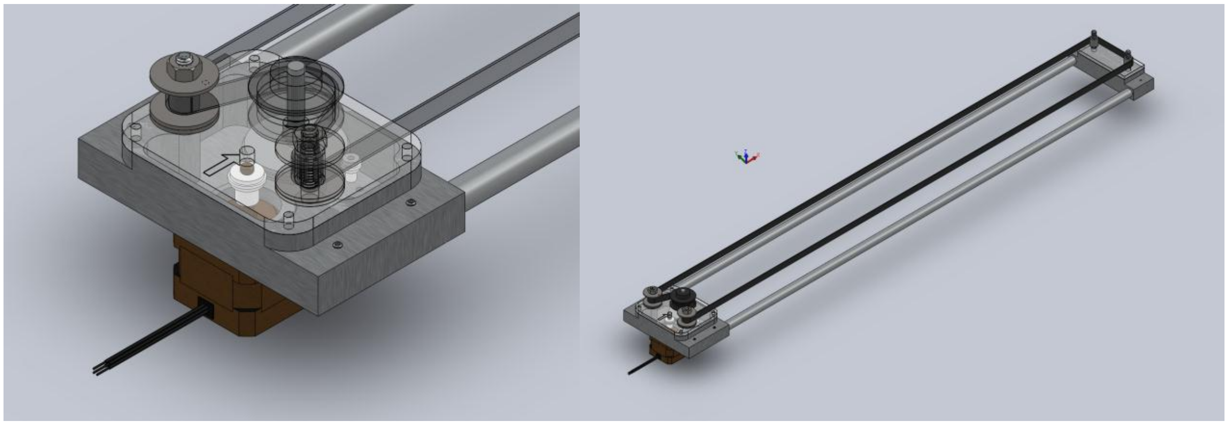


Figure 27: The X-Axis of the Print-Module

The objective of this system was to be able to position the print head at any position along the rail within 0.1 mm. To reach this target precision, a steeper motor was used in concert with an encoder to move the print head. To support the weight of the print head without noticeable deflection, two steel rods where used. With a five pound print head, laden with tiles, the maximum deflection experienced by the rods is 0.26mm (0.01in) as seen in Figure 28 , small enough to not affect the tolerance of the tile exiting the print head.

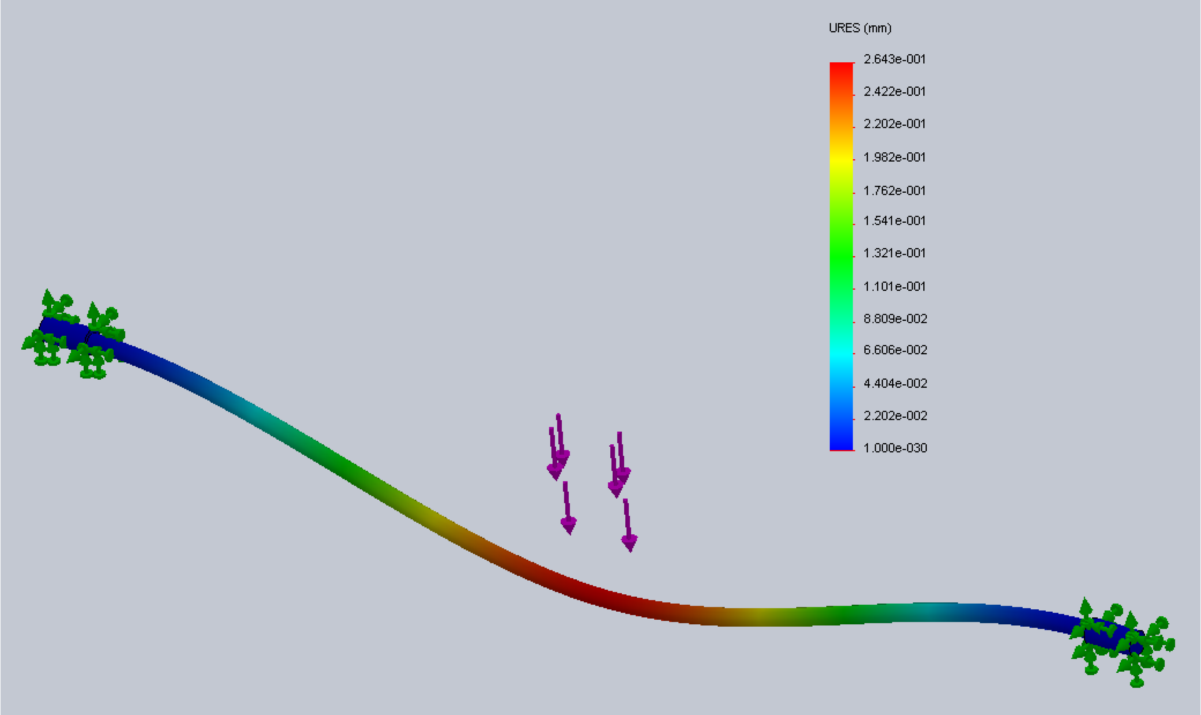


Figure 28: Finite Element Analysis of the Rods

Due to their fine step size of 1.8* and tight tolerances of +/- 5% of the step size, a stepper motor lends itself to the task. Even with the best steppers and driver circuits there is some step loss in the system, especially when driving the motor at full speed. A quadrature encoder was used to keep the print head from accumulating systematic error. Unfortunately, the stepper motor has twice the resolution of the encoder so the encoder cannot be mounted coaxial with the stepper without sacrificing resolution. To compensate for this discrepancy, the encoder was connected with a one to two gear ratio to balance out the number of ticks and counts on both devices. Due to space limitations, the two gears could not directly mesh with one another, so instead they both meshed with a much larger idler gear that was used for the output. With the output gear three times the size of the driving pinion the overall step size can be described as:

Equation 2: Step Distance

$$Step\ Distance = (\frac{Step\ Degrees}{360})(Sprocket\ Diameter * \pi)(Gear\ Ratio)$$

$$Step\ Distance = (\frac{1.8}{360})(16.2mm * \pi)(1/3) = 0.085mm = 0.0033\, in$$

This satisfies the target positioning tolerance, assuming no sources of error. When the sources of error are accounted for, the repeatability of the system can be described as:

Equation 3: Precision

$$Precision = (\frac{1.8 \pm 0.18}{360} \pm \frac{.2.5}{360})(16.2mm * \pi)(1/3) = 0.105mm\ to\ 0.042mm = 0.0041in\ to\ 0.0017in$$

In an attempt to make the system easy enough to move via stepper motor, the print head was mounted on Acetal bearings which have a low coefficient of friction. The amount of force the stepper motor needs to apply to move the printhead can be described as:

Equation 4: Motor Force

$$Motor\ force = Print\ Head\ Mass * Coefficient\ of\ Friction * \frac{1}{Belt\ Bearing\ Efficiency}$$

$$Motor\ force = 5lb * 0.15 * \frac{1}{0.97} = 0.77lb$$

The amount of force that can be provided by the gear box can be described by the following equations:

Equation 5: Motor Torque & Force

$$Motor\ Torque = 0.23Nm = 0.17\ foot * lb$$

$$Gearbox\ Output\ Force = Gear\ ratio * \frac{Motor\ Torque}{Pinion\ Diametrical\ Pitch} * Gear\ Efficiency$$

$$Gearbox\ Output\ Force = 3 * \frac{0.17\ ft * lb}{(0.637\ in \div 12\ ft/in)} * 0.97 = 9.32\, lb$$

The stepper motor can apply over nine pounds of force and the print head should slide with less than one pound of force giving the X-axis a factor of safety of roughly twelve. Any failure of the gearbox to move the print head could be due to set of bearings not operating at the advertised efficiencies.

Finally, to ensure the system can fully calibrate itself, limit switches were added to the end brackets that bind the rails together and contain the gearbox. These limit switches were positioned so their housings could never contact the print head or any moving part expect for their lever arm to prevent damage. One limit switch was necessary for each side because the active tensioning spring will change the distance from the two tensioning brackets.

6.1.4 Reloading System

The reloading system allows the robot to run continuously without stopping for operator intervention. The reloader is formed of four sub components; a print head dock, a linear actuator module, a removable cartridge and a support column with a limit sensor mounted near the ground.

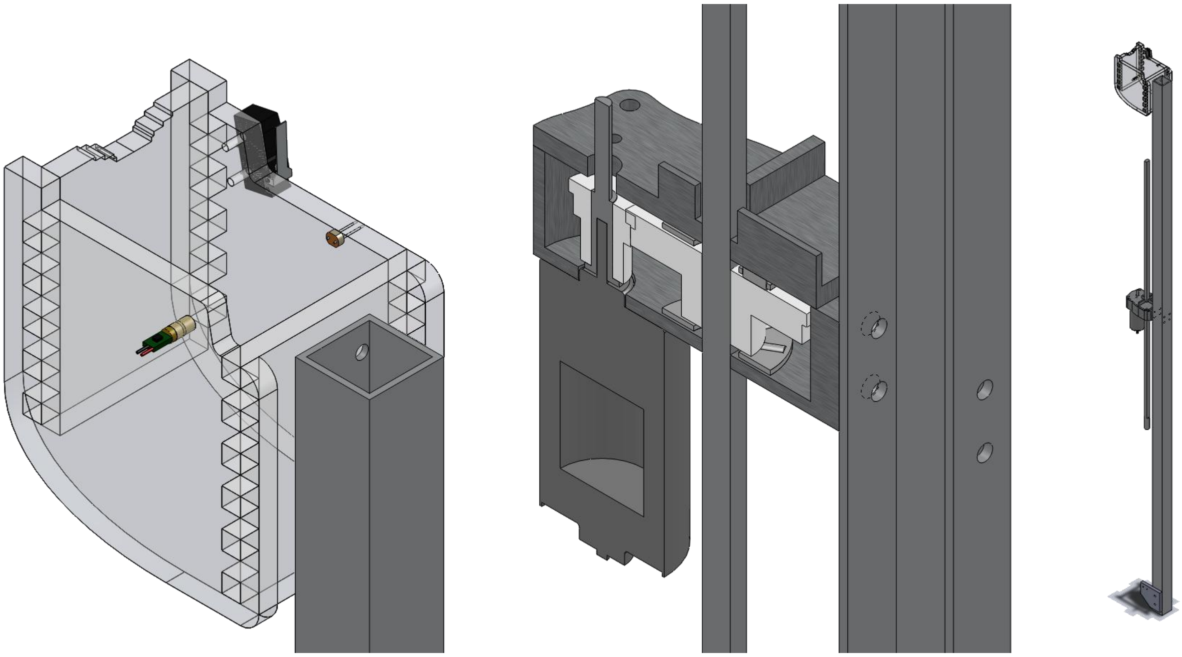


Figure 29: Reloader Module

Many of the innovations made during the final revision of the linear actuator module on the print head were transferable to the reloading linear actuators. Like the print head, the linear actuators on the reloading system where made out of aluminum instead of acrylic. This change allowed for the housing to more easily support the weight of the larger reloader cartridge and the higher quantity of tiles. The form fitting shape of the reloaders allows for a larger number of housings to be cut out of a single block of aluminum, lowering their cost. Also like the print head, the reloading system now uses the same cross system where the removable cartridges connect to the print head.

Previous designs called for a gate to be raised to allow the cartridge to be removed. On the final design, to remove the cartridge, it is pulled upward through the reloading dock to remove it. This change removes an unnecessary part form the system increasing its stability. Unlike the print head, the limit switch that is used to calibrate the system contacts the threaded rod only when it’s in the bottom most configuration. This allows the print head to know when it is too full to accept any more tiles and allows it to calibrate right before it moves to reload. Every other action by the reloader can be performed without the use of the limit switch.

The unique innovations to the reloader come in the form the dock. Instead of accepting tiles partway up the side of the cartridge and using a gate to prevent them from spilling back out when they are pushed into the print head, the new system accepts them above the height of the print head. This shift removes the need for a locking door and makes the reloading process easier on the operator so they no longer need to pry open the locking door. In addition to moving the tile acceptor, the dock now conforms to the shape of the print head so the two devices will line up when the print head is in the docked position.

6.2 Electrical system

6.2.1 Microcontroller Allocation

The Sanguino microcontroller selected for this project uses an Atmel ATMega644P as the main CPU (Figure 30). The 644P has thirty-two I/O pins that can be used for a variety of purposes, specifically serial communication, PWM signal generation, analog voltage reading, and digital inputs. On each module, there are four DC motors and four corresponding quadrature encoders, as well as a stepper motor with encoder. Each motor and encoder combination requires four I/O pins, using twenty available. Six limit switches and four broken beam detectors bring the pin count up to thirty. The two remaining pins are used for serial communication.

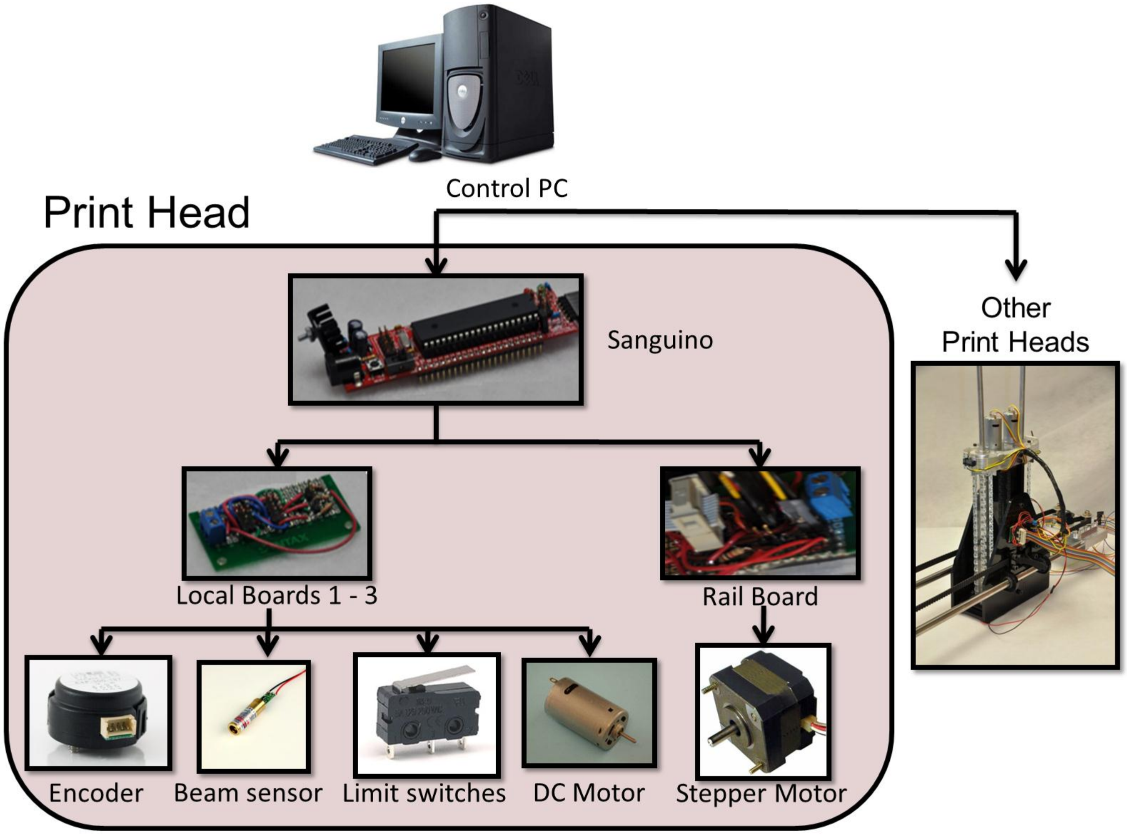


Figure 30: Interaction between Systems

6.2.2 Control Circuitry

H-bridges will be used to control power to the four DC motors (Figure 31). Each SN754410 IC contains four half-h drivers, and can be used to drive two motors. The number of pins required for each motor was reduced from three to two by having the direction of the motor controlled by a single digital output and an inverter. The digital output was sent to one h-bridge and its inverse was sent to the other. The speed of the motor was controlled by a PWM signal connected to the enable pin of the h-bridge.

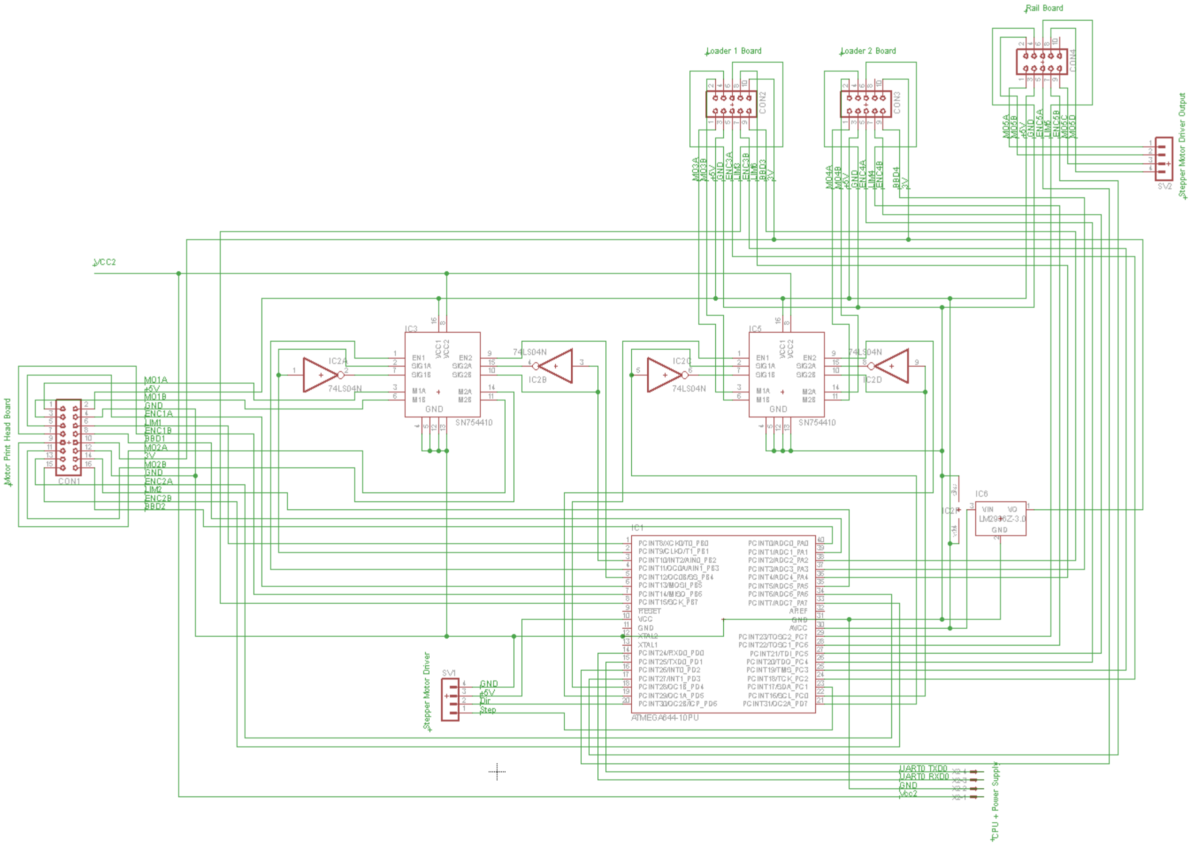


Figure 31: Daughter Board Schematic

The stepper motor was controlled by a dedicated stepper driver board. The broken beam detectors are simple photoresistors detecting changes in light from a laser. The photoresistor circuit was designed with voltage inversely proportional to intensity using a 10k resistor to give a threshold of 0.83 volts. The lasers require between 2.7 and 3.2 volts requiring the use of a three volt linear regulator.

6.2.3 Expansion Boards

After choosing the microcontroller, several breakout boards were designed to connect the Sanguino to the print-module’s subcomponents. There is one print head board which connects the two motors, encoders, limit switches and broken beam detectors to the Sanguino. There are two loader boards that control the motor, encoder, limit switch and broken beam detector of the bottom feeder. There is also a rail board that controls the motor, encoder and two limit switches for the print head’s horizontal position.

The print head board has a 16-pin ribbon cable connecting it to the Sanguino board while the other three boards are connected by a 10-pin ribbon cable. The ribbon cables supply 5 and 3.3 volt power for the sensors, as well as motor power and signal lines. The six extra pins on the print head cable are for the second motor, encoder, laser, and limit switch.

Initially, there were different boards for the loaders and rail. The difference between the three of them, however, was merely a limit switch or a broken beam detector. During prototyping, it became apparent that these three boards could be identical to each other, leaving the unused sensor lines unused. By each having 2 limit switches and the broken beam detector on each board, any board can be used in any position without fear of overloading or missing a sensor.

6.3 Control System

The control system for the TileJet was described in three major software components: the host control software, the robot control firmware, and the communications protocol connecting the two. The host controller was written in Java, and ran on a Linux-based PC (Figure 32). It communicates over USB-serial connections to the print heads, and can support up to 255 heads at a time. The print heads use Sanguino microcontrollers to control the hardware of one printing unit, including the mechanism to drop and detect tile, and the motor to move the print head along the X-axis. Once the responsibilities for each system were defined, each subsystem was further divided into smaller, more manageable sections.

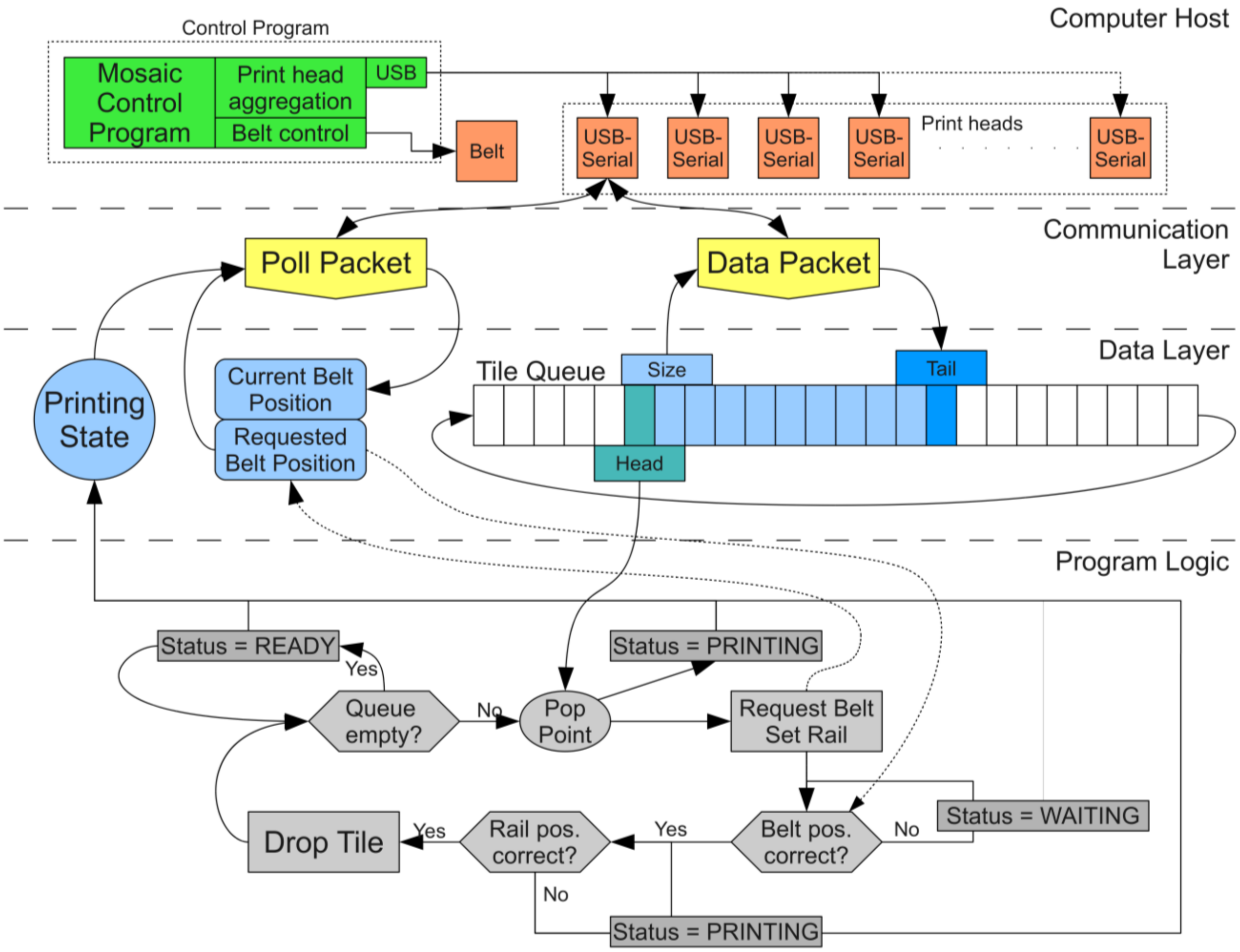


Figure 32: Software Overview

6.3.1 Master Controller

The responsibility of the host control system is to process a mosaic file, distribute points to print heads, and move the belt when the print heads are ready to move.

Upon starting, the program first attempts to initialize the print heads on the belt. The print heads are managed in aggregate form by a PrintHeadManager, which performs other functions as well as initialization. It starts a thread that is used to discover the print heads by searching for USB-serial ports. It is assumed that the computer used for this system is dedicated, and does not have any other USB-serial devices. The host asks each printer for its ID, which is stored in EEPROM on the controller. The ID is used to keep track of where each print head is (to calculate offsets), and determine what tile colors each print head is using. This data is saved to a configuration file, and the data can be changed while the program is running, to swap out colors.

The program reads in mosaic data from a mosaic source, which can either be a file to parse, a test mosaic generated by the program, or potentially a data store on a server. To avoid re-implementing a mosaic generator program, the Tessera program in use by Artaic is used to create source files. Once the mosaic has been parsed into TileCoordinates, each containing a position vector and a tile type, the coordinates are sent to the PrintHeadManager (Figure 33). The manager keeps PrintHead objects for each print head, which contain a link to the physical device. The manger delegates points to specific print head objects, which keep track of all points that they is capable of printing.

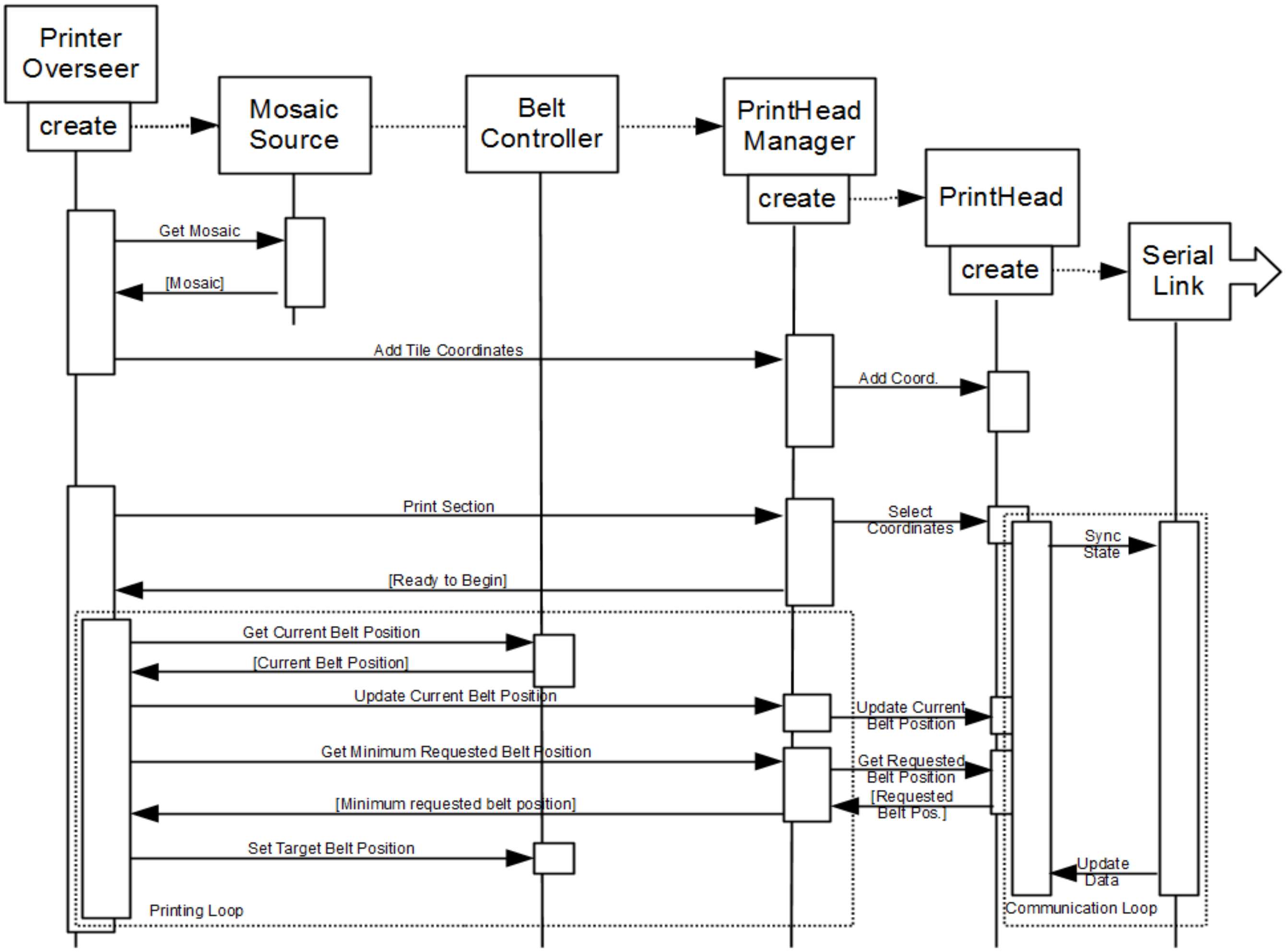


Figure 33: Sequence Diagram

When the system is instructed to begin printing, the mosaic is divided into vertical columns, and these rectangular sections are sent to the print head manager which has each PrintHead select a subset of points in that rectangle. A specialized pathfinding algorithm (See Appendix A for details) was used to determine the fastest order in which to print the tiles. Coordinate data is then sent to the print heads over serial in the order to be printed. When the print heads report that they've received data, the main printing loops begin.

When printing a mosaic, the purpose of the master controller is to keep the print heads fed with data, and to move the belt when the heads are ready for it. To do this, three RPCs are used with the slave device: a poll request to keep cached information up to date, a data request to give the device more instructions, and an initialization request to reset the device. These are described in detail in a later section. Each print head requests a belt position based on the point it is attempting to print next. The print head manager aggregates these requests, and returns the lowest one to the main controller, which moves the belt to this position. While a print head is printing at a location, it will continue to request the same position, which will keep the belt in place. If a print head has no points left to print, it requests the belt's position at positive infinity, effectively removing itself from the loop. When all print heads indicate that they are finished, the main loop reads positive infinity from the print head manager, and knows that this section has been completed. It can then select another section to print, or complete the mosaic.

6.3.2 Slave Controller

The controller on the print head occupies itself with motor control and sensory interpretation. It uses two 8-bit timers as PWM signal generators for motor speed control, and one 16-bit timer to control the stepper driver. Quadrature encoders are used with pin-change interrupt detection to count ticks, stored as signed 64 bit numbers. It also uses a 10-bit ADC to detect tiles passing in front of a broken beam detector, and digital inputs with pull-up resistors to read limit switches. The controller maintains a serial link with the host computer for communication.

The program running on the microcontroller separates itself from the communication system by means of a data layer. The data layer contains a queue of points to be printed, as well as the main belt's current position, and desired position. Once initialized, the firmware enters a printing loop, represented by a state machine. The loop attempts to pop a point from the print queue, and move to that position. If there are points in the queue, it sets its desired belt position to the point's Y coordinate. If not, the desired belt position is set to positive infinity, and the print head continues to read the queue until it receives a point. Upon popping a point off the queue, the print head moves to its X coordinate on the rail, and checks the position of the belt and rail. When both axes are within tolerance, it drops a tile by pushing the piston downward until detecting a tile drop with the broken beam detector. After dropping, it returns to pop a point off the queue. At no point in this loop is the program ever directly waiting on a response to a network call. All network requests come from the master, and are acknowledged by the slave.

The two RPCs used to interact with the print head serve two different functions. The poll RPC is called periodically, and is used to keep track of the desired position of the belt and the number of points on the print queue. It also updates the current belt position, as if it were a sensor on the print head. The data RPC queues another point onto the end of the print queue. The network protocol interacts with the main loop only indirectly through these data structures.

6.3.3 Communication Layer

The communication layer links master and slave, and consists of three RPCs: poll, data, and initialization.

6.3.3.1 Poll

The poll RPC synchronizes belt information with the print head, and keeps track of the print head's queue. The master sends a byte identifying the packet as a poll packet, followed by a 64-bit unsigned integer representing the current position of the belt. The print heads do not use floating point representations of tile coordinates, as the processor does not have a FPU, and the overhead cost in converting the values on chip outweighs the flexibility gained by using floating point numbers. For belt control, zero is the start position of the belt, and the maximum 64-bit value is considered "positive infinity." The number itself is of arbitrary scale, and is actually the belt encoder tick count, but remains consistent with the values given for the Y-position of the tile coordinates.

Upon receipt of the poll packet, the print head sends a byte containing a poll response identifier, a 64-bit unsigned integer representing the desired belt position, and an 8-bit number indicating the number of empty slots in the print queue, or how many more points the print head can hold. This combination of packets ensures that the current and desired positions of the belt are maintained, and that the print head is kept well supplied with tile coordinates.

6.3.3.2 Data

The data RPC adds a point of tile data to the print queue. Its packet contains an identifying byte header, followed by two 64-bit unsigned integers for the X and Y coordinates for the tile. The X coordinate is measured in 1024ths of a millimeter with zero as the leftmost (closest to the stepper) possible printable location. It also contains a byte indicating whether the point is to be printed with the left or right color.

The print head responds with a data response packet, containing an identifying header byte, and another byte indicating if the point was added successfully to the queue. If the master accidentally sends a point to a full print head, it will make sure to confirm that the point got there, and if not, will re-send the point as the next data point.

6.3.3.3 Initialization

The final RPC is an initialization call. This larger packet is sent to ‘reset’ the print head. It contains values for the PID constants used in motor control loops, as well as a printer ID number used to identify this printer. If the ID number is nonzero, the master is instructing the print head to change its ID to the one given; otherwise, the print head should retain its ID stored in EEPROM.

Upon receiving this packet, the print head sends a response back with an identification header and its stored ID. If the master does not receive this response back in a short amount of time, it considers the print head inactive and removes it from the system. The print head recalibrates its rail system upon receiving an initialization packet. It will still respond correctly to poll and data RPCs, but will not drop tiles until it finishes calibrating. An initialize call also clears the print queue.

7 Results

The final product of this MQP is a tile placement robot design. As proof of concept, a prototype print module has been fabricated. The goals of this MQP are to improve the quality of the mosaics produced, increase the overall throughput of the system, and expand upon Artaic's tile versatility. Improved mosaic quality is achieved by improving the placement tolerance of the system to under 0.9 millimeters. The production time of the system is decreased from Artaic's original 15 minutes per square section to theoretically 9 minutes per section for 12 print modules. The cartridge and diaphragm system allow for Artaic to accurately utilize any shaped tile within the bounds of the diaphragm, and the number of usable tile colors can easily be increased by adding more print modules. The prototype represents merely one part of a whole system; a system capable of quickly, consistently, and accurately producing tile mosaics.

7.1 Throughput

The most important task was to create a system with an increased throughput compared to Artaic's current system. This was done by utilizing multiple print heads in tandem, acting in much the same way as those in an ink jet printer. Each print module is responsible for printing two colors and only needs to be informed as to where it is located on the x and y coordinates of the grid below it. The belt is controlled by the main processor, and will only increment when all of the print heads are waiting to place a tile. The parallel print heads allow for a constant throughput of the system, regardless of how many print modules are in use. While having more print heads results in a longer time for a section to pass through the system, the rate at which they are inserted, and thus removed, remains fairly consistent. The belt increments depending on how quickly each print head completes its rows, meaning that more print heads of distinct colors will not reduce the time to assemble a section. If the system has duplicates of the most common tile colors then the time it takes for a section to pass through the system could decrease drastically.

7.2 Accuracy

A high rate of production, however, is useless without a high placement accuracy. To achieve this high level of accuracy, a new cartridge system was designed with a diaphragm that centers each tile. The diaphragm would bend as the tile was passed through, applying a force on each side of the tile. These forces would move the tile until they are all in balance. At this point, the tile has been adjusted so the center of the tile is in line with the center of the diaphragm and is ejected. This takes into account any imperfections in the tile. Tiles that are bigger or smaller than the advertised size are still centered properly because of how the diaphragm is designed. The diaphragm is split into four sections, one for each side of the tiles. This ensures that the sides apply only parallel and perpendicular forces on the tile. The vertical forces applied by the diaphragm are negated by force of the linear actuator pushing down on it. The placement accuracy of the diaphragm is eight tenths of a millimeter for every centimeter of height from the bottom of the flaps to the belt. The positional accuracy of the print head module is one tenth of a millimeter, obtained through the use of a stepper motor. Combined with the tolerance of the diaphragm, the placement accuracy of the print module ranges from 0.5 to 0.8 mm. This variance comes from differences in the thickness of the tile being placed, with the thinnest tile and thickest being five and ten millimeters, respectively.

7.3 Adaptability

This system has to be capable of not only handling Artaic's current tile geometries, but also any future tile geometries. With this in mind, the whole system was designed for modularity. The cartridge and diaphragm are designed to lock into a designated location on the print head through easily accommodating slots in the cartridge. The locking mechanism would ensure that the cartridges and diaphragms, both present and future, would be able to be quickly substituted. Future cartridges need only need to have slots machined in them, and because of this system, the time associated with changing the type or color used by the system is greatly decreased. The system is also designed to be modular in its activity. If a print head fails during operation, the system will continue running without that color. Sections that are being produced will still be completed, but they will do so without the colors of the inoperable print head. The system is designed so that the whole print head can be removed from the system for maintenance and assembly can continue. New or repaired print heads can be added in place of the missing print head, or at the end of the end of the system for extra capabilities.

8 Conclusions

With the completion of this project, the students have created a new mosaic assembly system, designed to enhance Artaic Innovative Mosaic's product and productivity. The original goals have been met within this new system, though future development is necessary to create a final, testable product.

8.1 Accomplishments

At the conclusion of the project, several solutions were developed to solve issues encountered during the course of this project. One of the greatest problem was the tiles themselves as their variability and low quality required robust solutions. A stacked storage cartridge was created that could be easily redesigned to accommodate new lines of tiles with varying tolerances. A system was designed to convert the Artaic stored tile, which was stored flat in tubes, to the vertically stacked cartridge using a linear actuator system. To complement this, diaphragms were created that would center a tile as it was being placed using a low cost linear actuator. The control system was designed to scale to a large number of print heads, and uses a relatively simple paradigm to do so. The student team is proud of what they have accomplished during this project and look forward to seeing future projects based on their effort this year.

8.2 Social Implications of the TileJet

Robotics as an industry has often scared people. Movies, video games and books have been made about the good and evil that comes from the automation of society. Almost all fears of robotics boil down into one of three categories. The most common fear of automation is being destroyed or harmed by the creation. Along similar lines, Curt Vonnegut's player piano was written about a society where engineers designed away the necessity for the human workforce and made people obsolete showing the second fear about technology. The final fear is more recent and includes everything that is physically built, not just robotics. That fear being the effect on the environment.

Unlike some other robotics MQPs, the TileJet printer does not have excessively powerful motors driving towards pinch points or move around the room on its own. In fact, the only pinch point where someone could accidentally get caught is the rail system, and it is only capable of applying less than ten pounds of force which is not enough to damage a hand. The other moving systems perform their movement when encased within cartridges. The TileJet is also programmed to obey the Three Laws (Asimov), and would never consider causing damage to any living thing, least of which a human.

The TileJet is designed to revolutionize a market dominated by artists that create custom mosaics by hand or have their design outsourced to overseas sweatshops. The success of the TileJet will allow artists to have their custom mosaics created in much shorter periods of time without the need to do business overseas to get results. This project only threatens the lively hood artists that are too stubborn to switch to a new, more efficient system for mosaic assembly and the sweatshop workers overseas who manually assemble mosaics all day who could easily switch to assembling anything else as it is an untrained position.

The last major social concern for the TileJet is its environmental impact. Each print head houses 5 motors, and a usable system contains roughly twenty print heads for a grand total of 100 motors in the system. At any given time there could be up to eighty motors moving in tandem, both reloaders refilling themselves and the print head popping out two tiles at a time. This situation is unlikely but there could be a rather large amount of power used. To compensate for the power draw, small lightweight motors where selected that would be able to operate a lower power levels. Furthermore, because power isn't wasted moving a print head unless it is going to print a tile there is relatively little waste, especially compared to other systems that are required to move a large arm, or table. The materials selection of the print head could be better; it uses large pieces of plastic that are in no way friendly to the environment. In future models the plastic components could be made out of extruded aluminum or aluminum plating.

8.3 Future Work

With further work this system can be brought to the production level where it will be producing high quality mosaics with a constant throughput, much greater than Artaic's current system. If future projects were to be based on what we have accomplished during this MQP this section will lay out what still could be accomplished and what we feel is the best solutions for problems we had yet to encounter.

8.3.1 Refill

To allow continuous production, the system must be able to refill the print head; to accomplish this, a docking and refill system for the print head has been designed. To create a functional refill mechanism the dock system needs to be tested to ensure the print head refilling method works appropriately.

Table and Substrate Belt

Another aspect that has not been designed is the conveyor system, to which the print heads and reloaders can be attached. This attachment should allow the print heads to be vertically adjustable, to account for the varied thicknesses of tile as well as adjustable along the Y axis to allow spacing between print heads along the table. This table must include a substrate belt management system. This system needs to include a tensioning system that prevents the substrate from scrunching or stretching. The belt must also be indexed properly to maintain tolerance in the y axis. To keep the finished product from being destroyed, further sub-systems should be considered such as a cutting, labeling and stacking system at the end of the print table.

8.3.2 Control System

Many aspects of the control system, although designed and documented, remain unimplemented. Future work on the control system will focus primarily on the communication protocol and device firmware.

8.3.2.1 Communication Protocol

The communication protocol is largely unfinished. The main objects, such as the packet interpreter and handlers on both sides of the software have been implemented, but most of the handlers for individual packet types have not. A handler for poll and poll-response packets has been implemented by the host, and other packet types will need similar implementations. The packet handler for the firmware is also largely complete, only needing a few packet interpretation modules to be complete.

8.3.2.2 Print Head Firmware

Since the hardware was never functional during the course of the project, the firmware primarily consists of device drivers and components. These drivers and components have been tested individually, but never within the full system. These devices will need to be tested when the full system is complete, and integrated with the communication protocol and main printing loop to complete the firmware.

8.3.2.3 Control Program GUI and Refinement

There is no user interface for the control program as of yet. It asks for a mosaic, and begins printing immediately. A user interface to specify colors for print heads, as well as their offset and status should be implemented.

Refinement and New Tile Development

Once the functionality of the full system has been proven, the system could be further refined by optimizing the system for material costs and manufacturing needs. Further development is necessary to accommodate new tile varieties (storage tubes, diaphragms and refilling considerations).

9 References

"Custom Mosaic Design And Fabrication." Artaic Innovative Mosaic. Web. 29 Oct 2010. <http://www.artaic.com/b/>.

Asimov, Isaac. I, Robot. Spectra, 1991. Print.

"IRB 360." ABB. ABB, 2011. Web. 2 May 2011. <http://www.abb.com/product/seitp327/cf1b0a0847a71711c12573f40037d5cf.aspx>.

"Linea Assemblaggio Mosaico - Mosaic Assembly Line (EXACTA) ." TecnemaMaranello (YouTube). Web. 2 May 2011. <http://www.youtube.com/user/TecnemaMaranello>.

"Tilemyzer." Productivity Through Automation. Tilemyzer, 1999. Web. 6 Nov 2010. <http://www.tilemyzer.com/>.

10 Appendices

10.1 Appendix A: Print Head Optimization Algorithm.

One of the factors affecting printing speed is the order in which tiles are placed by the print heads. The simplest algorithm would have the print heads always start on the left, and move across to the right dropping tiles, before returning to the left and starting again. This printer would spend much of its time returning to the left, when it could be using that time to drop more tiles. Another algorithm would have the print head drop tiles left-to-right, then right-to-left, saving the time it would take to return to the start before printing another row of tiles, but this process can get more intelligent.

Mosaic patterns are not always consistent squares of color. Moving the print head in one direction in an early row affects the direction of printing in rows far ahead of it. We can make a few assumptions about printing that will assist in development of the algorithm. The printer can only do one row of tiles at a time, and cannot move backward in the Y direction. The printer will also always start at one end of the tile row and print tiles in sequence to the other end of the row, that is, will always move in the same direction once starting. With this in mind, the question is: which end to start printing at for each row?

These decisions are formed into a graph, and the order determined with a path findng algorithm. The tiles in the mosaic are divided into "rows" by grouping them according to the Y-axis tolerance of the system, and the furthest left and right X coordinates are considered. Each node in the graph (Figure 34) represents either starting at the left, or

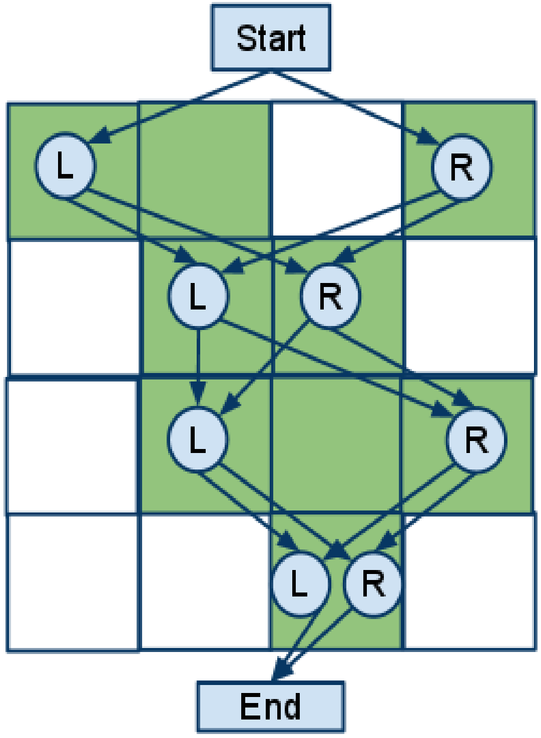


Figure 34: A run-time graph of a small mosaic.

starting at the right to print a row of tiles. Each row then has two nodes, and every row is connected to the row above and below it.

The edges between nodes describe how long it would take to get from the end of one row to the start of the next, taking into account both the X and Y distance. For some rows, the time it takes the Y-axis to move will take longer than moving to either start positions of the X axis, so the edges are identically weighted.

Using a graph with the structure illustrated in Figure 34, and considering that a node accounts for **all** of the tiles in its row, any path from start to finish will contain all of the tiles. Using Djikstra's algorithm, it is possible to find the shortest path from start to finish, which will represent the fastest order in which to print the tiles.

10.2 Appendix B: Run-time Estimates.

Just as with a car assembly line, in which a single car may take 3 hours to be constructed but the assembly line can output a new car every 30 seconds, the throughput of the TileJet is dependent on the time interval between two sections being completed. The print head module with the most tiles to print is the limiting factor of this interval. Assuming a Gaussian distribution of tiles per color, then for optimal module usage the color with the highest number of tiles should be paired with the color with the lowest number of tiles. This will allow all the print modules to have to print the same number of tiles: two times the average number of tiles per color. In this case, since there are 729 tiles per section and forty colors are being used, each module will need to print 2*(729/40), which is approximately 36.45 tiles. With print head usage standardized, the limiting factors each section become how fast the belt can increment and how quickly the print head modules can print. Assuming the speed of the stepper motor is three revolutions per second, then with the gearbox's three to one ratio the output speed is one revolution per second. The radius of the gear attached to the pulley is .625 inches so the linear translation speed of the print module is 1.963 inches per second. Assuming the worst case scenario that the print module has to travel the full length of the rail, 36 inches, to print each tile then the module has 14.26 seconds of travel time.

After the print module gets to the print location, it needs to print a tile. The BaneBots motors have a no load rpm of 12180, which equates to 203 revolutions per second. Because the full stack of tiles and linear actuator weighs 5 pounds, we can run the motor at 74% efficiency, 150 rps. Each motor revolution corresponds to 1 thread traveled on the linear actuator, and there are 16 threads per inch. With the motors running at 150 revolutions per second, the linear actuator can move 9.375 inches per second. The thickness of a tile averages five mm, .1965 inches, meaning that the motor will have to run for only .021 seconds to eject a tile. In comparison to the 14.26 seconds of travel time, the print time is negligible, since it is only .15% of the travel time.

An optimized system has each print head module printing the same number of tiles, 36.45 for a system with 20 modules, thus the system will produce at a constant rate. This rate is simply the time it takes a print head module to print all of its tiles plus the time it takes the belt to increment twenty seven times. If we assume that the belt speed moves at the same speed as that of the print head, then the belt also has a speed of 1.963 inches per second. The one foot by one foot section is divided up into twenty seven even spaces, where each space is 4/9 of an inch. It takes the belt just over .226 seconds to increment once, and just over 13.75 seconds for twenty seven times.

The total time is the number of times each print head needs to print added to the time it takes to increment the belt. The 14.26 seconds of travel time is multiplied by the 36.45 average tiles to print, for a total travel time of approximately 520 seconds. Including the belt movement time increases the time between two completed sections increases to 533 seconds or 8.89 minutes. Doubling the number of print modules to 40 almost halves the time between completed sections, decreasing it to 4.56 minutes. This time, however, is with the worst case of the print head module moving the length of the rail for every tile it needs to print. A reduction in the average length the print head has to move will proportionally decrease the time between sections. This is particularly interesting in systems with multiple colors. If an algorithm can be formed that calculates and reduces the average movement distance to eight or even four inches, then the difference between completed mosaics could be as low as 1.5 minutes for 20 modules.