Increasing 3D Printer Throughput with Automated Part Removal

IN PARTIAL FULFILLMENT OF A MAJOR QUALIFYING PROJECT
WORCESTER POLYTECHNIC INSTITUTE

Submitted by:

Nicholas Colucci (ECE)
Noah Donald (ECE/ME)
Alexander Johnson (ME)
Andrew Kacherski (ME)
Dante Mauriello (ME)
Owen Smallcomb (ME)

Project Advisors:

Prof. Pradeep Radhakrishnan (ME)
Prof. Joseph Stabile (ECE)

August 2019 – May 2020

gr-3D-Printer-Automation@wpi.edu

Copyright Notice

Abstract

A 3D printer is a powerful machine that allows for rapid realization of designs. Reducing the downtime between print cycles is essential for use in mass production or when multiple users are using the same printer. Presently, an operator manually removes the print from the machine and starts a new print. This project is an improved system from a previous Major Qualifying Project (MQP) to automate this process and eliminate downtime. The system consists of modular sub-systems, making it compatible with any i3 style printer. In the case of a printer with a removable bed, our system removes parts by lifting the bed from the printer and feeding it through a curved track that causes the bed to bend. This bend breaks the adhesion between the part and the bed, allowing the part to fall off. The system then feeds the bed back out, places it back on the printer, and starts the next print in the queue. For printers without a removable bed, the system utilizes a clamping mechanism and custom removable bed rather than the lifting component. The clamping sub-system secures the custom removal bed to the printer's bed during printing and releases the bed for removal when done. Print jobs are managed using a queue in OctoPrint on a Raspberry Pi. Upon completion of the print, the printer pauses while the finished print is removed, and resumes printing from the queue once the bed has been replaced.

Acknowledgements

The completion of this project could not have accomplished without the support of our advisors, Prof. Pradeep Radhakrishnan and Prof. Joe Stabile. We also thank the Electrical & Computer Engineering and Mechanical Engineering Departments. Thank you for your yearlong guidance and commitment to helping us reach our goals.

Executive Summary

Introduction

3D printing is a low cost, high speed manufacturing technology which is perfect for creating prototypes and low volume parts which would be cost and time prohibitive to be manufactured by subtractive means. 3D printing has a low initial process time compared to subtractive methods, allowing parts to be prototyped quickly after design completion. The most significant bottlenecks in the 3D printing process are removing completed prints from the machine and starting new printing jobs. On most available systems both jobs must be performed manually by a human operator. This requirement binds the throughput of the printer to the performance of the operator and restricts continuous operation.

A review of the current literature indicated that there are few available 3D printers which offer continuous operation, and there are no independent add-on systems which facilitate the repeated initiation and removal of prints. The objective of this project is to design a system which facilitates continuous 3D printing. Successful completion of this goal will result in a more efficient and productive 3D printing process.

Design & Manufacturing

Several design requirements were determined to define the success of the system based on the project goals. First, the system must automatically clear the print bed and removed the finished product once a print has been completed. This requirement solves one of the main bottlenecks of the 3D printing process by eliminating the need for a human operator to remove a finished print. Second, the system must handle multiple print jobs at once, printing and removing them in succession to achieve continuous operation. These first two requirements define conditions to achieve autonomous 3D printer operation: the system executes a print, removes it from the machine, and executes the next print, given that there is another pending print job. Furthermore, the system must require minimal to no modification of the printer it is installed on, which ensures that it is user-friendly and easy to install. Lastly, as the system was designed with the vision of several autonomous printers working simultaneously, the system must be both modular in design and scalable to include several machines.

The chosen design for print removal revolves around flexing the surface bed the print is printed on to break the surface adhesion between the part and the print bed. Once this adhesion is broken, the part can be removed much more easily from the bed using a scraper. To accomplish this behavior, the system removes the print bed from the printer and feeds the bed into a curved track that flexes the print bed. The bed is then fed further into the track, where a set of scrapers come into contact with the bed to fully remove the part from the print bed. Once the print is removed, the print bed is then fed back out of the bending track and is deposited back onto the printer so that the next print may begin.

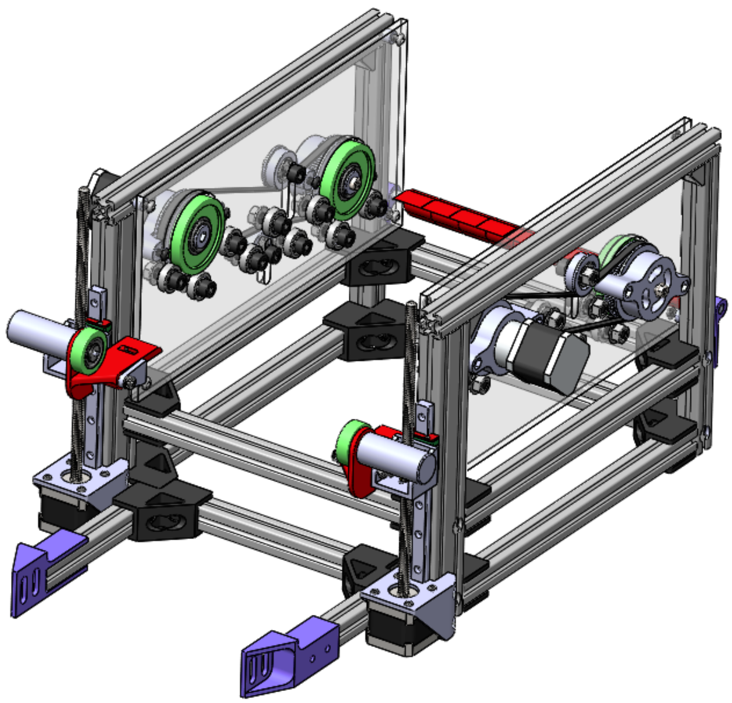


Figure i: Isometric view of print removal system

Testing & Analysis

We designed several tests to determine if the system met the initial design objectives. To test the print removal capabilities of the system, we printed squares of various sizes on the removable bed and then pushed the bed through the bending track. The bending track flexed the bed enough to break the adhesion between the print and the bed. The scraper then successfully pushed the detached prints off the bed. To test the automation of the removal system, we ran the system to ensure it could lift the bed, run it through the bending track, and then replace the bed on the printer. After some modifications and troubleshooting, the system successfully completes full print removal cycles without the need for human interaction. We thoroughly reviewed the design of the system to confirm scalability is possible and modularity was achieved. The aluminum extrusion that makes up the frame of the system can be replaced with larger or smaller pieces to fit to different size printers. The elevator system is independent from the bending track, so it can be adjusted for any printer bed height.

Our team also developed a set of future tests as a deliverable. These tests quantify the system's print removal capability and automation reliability. First, we designed an experiment that tests if a print's size or geometry could affect the system's ability to remove it. This test has the system try to remove small prints such as 1 in x 1 in squares, large prints such as the Prusa Mk3's maximum 8 in x 8 in square, and special print features such as brims and skirts. To test the reliability of the system's automation the system would initiate and remove prints continuously to simulate a real-world scenario. This test determines if the system can print over-night, over a weekend, and for an entire 1 kg spool of PLA filament. Over-night and over a weekend simulate a time when a human operator would likely not be readily available to fix a failure with the system. The system running continuously until the filament spool is empty would be the maximum efficiency of our system. If this is achieved, the system only requires a human operator to replace the spool every few days.

Applications

There are two main applications of this 3D printer automation system. First, it allows for a part to be produced multiple times one after another. This allows for a set of 3D printers to more effectively be used for mass production of printed parts. Also, because the system is scalable to multiple printers by attaching a removal system to each printer, this further expands on the ability to mass produce parts in a print farm. Another application of the system revolves around printing a set of parts in a queue. This feature allows for organizing a complete assembly of 3D parts to be printed in a single set of prints, which allows for better organization of parts that are printed.

Contents

Table of Figures

List of Tables

Authorship

Abstract	All
Executive Summary	All
1. Introduction	All
2. Previous MQP Work	Colucci, Kacherski, Mauriello, Smallcomb
3. Literature Review	All
4. Methodology	Colucci, Smallcomb
5. Preliminary Design and Testing	Johnson, Kacherski
6. Detailed Mechanical Design	Johnson, Kacherski, Mauriello, Smallcomb
7. Detailed Electrical Design	Colucci, Donald
8. Fabrication & Testing	Smallcomb
9. Discussion & Results	All
10. Conclusion	Donald, Johnson, Kacherski
Appendix A: Full FMEA	Mauriello
Appendix B: Full Electrical Shcematic and Individual Models	Colucci, Donald
Appendix C: Custom End-of-Print G-Code Script	Colucci
Appendix D: Assembly/Operation Instructions	All

Nomenclature

FDM: Fused Deposition Modeling

SLA: Stereolithography

SLS: Selective Laser Sintering

i3-style: An open-frame style of 3D printer modeled after the Prusa i3, where the print head moves to control x- and z-axis movements, while the print bed moves to control y-axis movements.

CoreXY: A style of 3D printer where the print head moves to control the x- and y-axis movements, while the print bed moves to control the z-axis movements.

1. Introduction

Throughout most of history, manufacturing was done through subtractive methods, in which a piece of material is cut and shaped until the desired geometry is achieved. Recently, however, a new method of part creation was invented, where material is added to form a part layer by layer. This process, popularly known as additive manufacturing 3D printing, has started a revolution in the way that parts can be manufactured.

1.1. History of 3D Printing

The first rapid prototyping device was developed by Hideo Kodama of Nagoya Municipal Industrial Research Institute in 1981. He developed a method where photopolymers were solidified in layers using Ultraviolet light, creating models [1]. In 1984, Charles Hull used photopolymers when he created SLA. His device used a laser to solidify liquid resin, creating the model. The development of this device shook the manufacturing world. Designers could design and test prototypes faster, and cheaper than using traditional methods such as machining.

From 1999-2010 serious progress had been made in 3D printing and it has entered unimaginable markets, such as the medical and culinary fields. In 2005, FDM printers started to gain traction in the consumer market and RepRap launched an open-source initiative to have printers be made of parts that could be 3D printed. In 2008, they accomplished their goal by releasing an open-source machine called "Darwin" [6]. This construct is one of the many reasons that the price of printers became more affordable to the everyday consumer. Early on, printers were selling for thousands of dollars. Now you can get one for as little as a few hundred dollars. This technology has changed how designers and manufacturers operate. This technology has even helped astronauts in space by printing tools on the International Space Station [2].

1.2. Comparison of Additive and Subtractive Manufacturing Methods

One of the main benefits of 3D printing, an additive manufacturing process, is the ability to create a prototype for a part faster than using traditional subtractive manufacturing processes such as machining. Printed prototypes can be made in hours, while CNC machined parts could take days to develop fixtures, tool paths, and processes.

Figure i: Subtractive manufacturing (left) and Additive manufacturing (right)[1]

Figure 1 shows the same part made via CNC milling (left), and 3d printing (right). The machined part starts from an aluminum billet, and the material is cut away until the final geometry is created. The printed part starts as a roll of plastic filament, the plastic is then extruded into many layers which creates the final geometry. The plastic is only placed in the shape of the part and there is no removal of material which means that 3D printing creates far less waste then CNC machining. The machine time for CNC machining the part in Figure 1 would likely be less than an hour while 3D printing could take a few hours. Despite CNC machining requires less machine time in this case, there is substantial process time needed before the part can be machined. Toolpaths must be generated for machining by either manually writing G-Code or by using a CAM (Computer Aided Manufacturing) software. This process can take hours or even days depending on the complexity of the part. After the toolpaths have been created the CNC machine has to be set up. This involves loading all the tools needed to create the features of the part, fixturing the billet of work material, and finally defining offsets for the work piece and tools. Complex parts may require these processes to be repeated as some features may need the work material to be fixtured in multiple orientations. In some cases, fixtures themselves need to be designed and machined to support the manufacture of the part. Conversely, 3D printing requires a small amount of process time. After a 3d model has been designed it is loaded into a special program called a slicer. The slicer allows the user to select printing parameters such as material, layer height and print orientation. It then “slices” the model into many thin layers and generates a G-Code file to print the model layer by layer. The slicing process usually takes a user no more than fifteen minutes. After slicing, the code is loaded onto the printer and the part is produced. CNC Machining requires an operator for the entire process while printing can occur in the background while the user is working on other tasks. Since there is less waste and processing time, 3D printing is a much better alternative to subtractive manufacturing for parts and prototypes that may need many revisions.

One of the main drawbacks for 3D printing is the part must be manually removed from the printer before the printer can be used to create the next part. This creates two issues. First,

[1] Reproduced as is from https://www.stratasys.com/explore/whitepaper/3d-printing-vs-cnc

while a part sits finished on the printer, that printer is no longer able to create more parts, and is thus not productive until said part is removed. Second, since the parts must be manually removed, there must be a person or team of people to continuously remove parts and start new prints to keep the system running consistently. These two issues are problematic for places which require high throughput. Examples include print farms such as the space located in the Foise Innovation Studio or even offices where many users have parts which need to be made from the same machine. Requiring workers to remove the prints is expensive and inefficient as there will always be some downtime between the completion of the print, and the worker resetting the machine. There are also possible health concerns for workers as some materials such as ABS are proven to off gas hazardous fumes during the printing process [7].

1.3. Manual Part Removal Methods

FDM 3D printers deposit material onto a printing surface referred to as the bed. The plastic filament adheres to the bed during the printing process to form the desired print. The print must be removed afterwards, otherwise the print bed is occupied and a new print cannot begin. Users can remove prints from the bed using several techniques.

1.3.1. By Hand

Firstly, users can remove prints by hand. This method is most effective for prints with a small contact patch on the print bed since the adhesion forces are weaker. This method is not recommended for parts with delicate geometries or large parts. Delicate parts are likely to be damaged from this method and large parts have a chance of transmitting potentially damaging forces through the bed and into the structure of the printer. This can occur if pulling on the adhered part leads to the any part of the printer bending out of place or even breaking in the case of glass beds.

1.3.2. Scraper

Another removal method is to use a scraping device like a razor blade, or a paint scraper. There are even specialized spatulas for 3d printing like the one made by BuildTak shown in Figure 2 below.

Figure ii: BuildTak print removal scraper[2]

The advantage of the BuildTak spatula is that the spade shape makes it easier to get underneath the part by providing a smaller initial contact area. The thin blade of the scraper pries the edge of the part away from the bed which starts to break the strong adhesion between the part and the bed. Scraping techniques include pushing against the edge of the part with constant force, as well as hitting the scraper with a heavy object such as a mallet to provide an impulse force. Users must take caution when removing prints with either of these methods as there is a risk of bodily injury from the sharp scraper and there is risk of print bed damage from overly aggressive scraping.

1.3.3. Removable Print Surfaces

Some printers feature removable print surfaces. Thermal shocking can be used on removable beds by submerging the warm bed/part in cool water. Since the part and the bed are made from different materials, they cool at different rates. This is the technique that the print lab in Foise Innovation Studio uses. The bed is heated to 60°C and the water is room temperature at 20°C. As the plastic and the bed contract unequally, the adhesion forces are weakened, and the part can be removed easily by hand. This method works best directly after a print has completed as the bed and the print have not fully cooled yet. This means that there is the maximum temperature differential between the bed and the water. The difference in thermal contraction rates between the bed and the part is what breaks the part adhesion.

Finally, some removable print beds come with a “flex” plate. A flex plate is a removable bed that can be bent to break surface adhesion between the bed and the part but will not plastically deform. After removing the plate, the user can gently twist the plate back and forth which pulls the plate away from the part, making it pop off. Examples of printers which use flex plates are the Prusa i3 MK3 and the Creality Ender 3. Figure 3 shows each of these printers, and Figure 4 shows an example of a flex plate being used to remove a part.

[2] Reproduced as is from https://www.BuildTak.com/product/BuildTak-spatula/

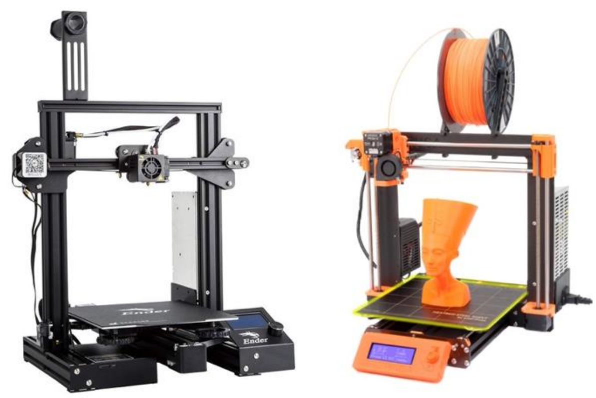


Figure iii: Creality Ender 3 (left), Prusa i3 (right)[3]

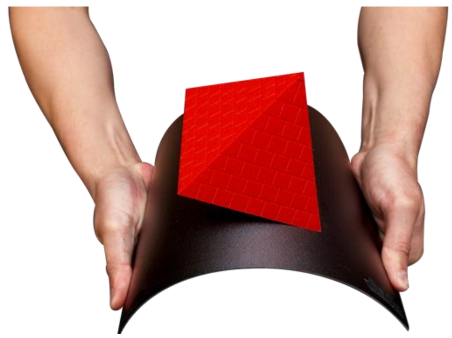


Figure iv: Removable flexible print plate[4]

Flex plates are considered one of the best print removal methods as they require the least amount of effort, are reliable, and risk of part damage is minimal.

1.4. 3D Printer Operation

Manual operation of a 3D printer requires a decent understanding of the entire 3D printing process for the print to be successful. Many 3D printers provide users with only the bare minimum for printer operation, including only a simple control panel with an LCD screen and some buttons. The firmware natively allows users to start/stop/pause prints, change print settings such as extruder and bed temperature, move the extruder manually, and troubleshoot problems or errors. To initiate a print, it is common for the user to save a G-code file on an SD card, then plug the SD card into the printer and select the desired file. If the user wishes to print a second object after the first, they must manually select a new file from the SD card, granted the G-code file is already stored there.

As 3D printer operation is clunky out-of-the-box, 3D printing enthusiasts have sought to improve the management of printers. Many computer applications have been developed for managing and interfacing with 3D printers, including OctoPrint, Repetier-Host, and AstroPrint. These applications run on computers and connect to 3D printers over a serial connection, and they provide much cleaner and more advanced user interfaces. However, while these software

[3] Reproduced as is from https://all3dp.com/2/creality-ender-3-pro-vs-prusa-i3-mk3s/
[4] Reproduced as is from https://filament2print.com/gb/accessories/883-BuildTak-flexplate.html

solutions do much to improve the user experience for 3D printing, they are limited in that they cannot natively handle multiple print jobs in a row, much like the printers themselves [1].

1.5. Project Goals

3D printers offer amazing prototyping and manufacturing potential, but their throughput is bottlenecked by the need for human operators to clear the bed and load new prints. The maximum potential of a 3D printer could be unlocked if there were a system which would remove completed prints from the bed and start new printing jobs. The goal of this project is to create a system that can be implemented onto various printers to allow for automatic queueing and removal of multiple prints. This system will facilitate continuous 3D printing, which in turn will increase the utility and value of existing machines.

2. Previous MQP Work

A design for an automated 3D printing system existed from a previous project completed in the 2018-2019 academic year. A thorough understanding of the existing design's strengths and weaknesses was invaluable in designing our own solution, so it was imperative that we learned as much as possible about the existing system. We analyzed the documentation and test results from the previous project as well as conducted our own tests, which are described below.

2.1. Analysis of Prior Tests

The first step in reviewing the existing design was to review the results that they had left behind. The previous team had provided several videos of their system in operation performing the tests that they had laid out. Before watching these videos, we defined the following criteria for successful print removal:

a. Adhesion between the print and the bed is completely broken.
b. The print is completely removed from the bed and print area.

We were able to test against these conditions by watching the videos they provided. The first video that we watched showed the scraper attempting to remove a 3" diameter PLA cylinder with a standard shaped raft, which took about one minute. A screenshot from this video is shown below in Figure 5, taken at the end of the removal process. The geometry of the blade slid underneath the part with ease and the depth of the blade allowed the part to sit on top of the top surface, satisfying condition (a). However, the part did not move horizontally until the nut at the end of the blade contacted it (circled in red in Figure 5), and the video did not show the part falling off the scraper. As seen below, the part was stuck on top of the scraper at the end of the process.

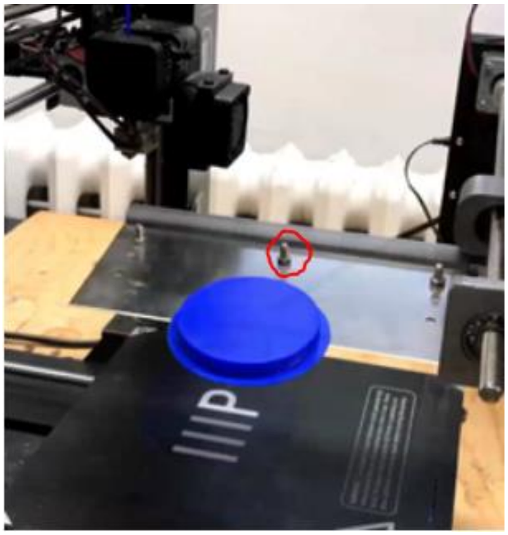


Figure v: The print left on the scraper after print removal[5]

[5] Reproduced as is from Hussain, M. et al.

Further review of the print removal videos revealed the print being stuck on the scraper to be a common theme, as it happened in nearly all tests. Given that the finished print remains stuck on the scraper in the common case and that the scraper must re-enter the print area to remove the next print, we considered this to be a failure under criteria (b). The print has the chance to fall back on to the print surface and interfere with future prints, and it is unclear how the finished product would be automatically removed from the scraper. More examples of the print being left on the scraper are shown below in Figure 6.

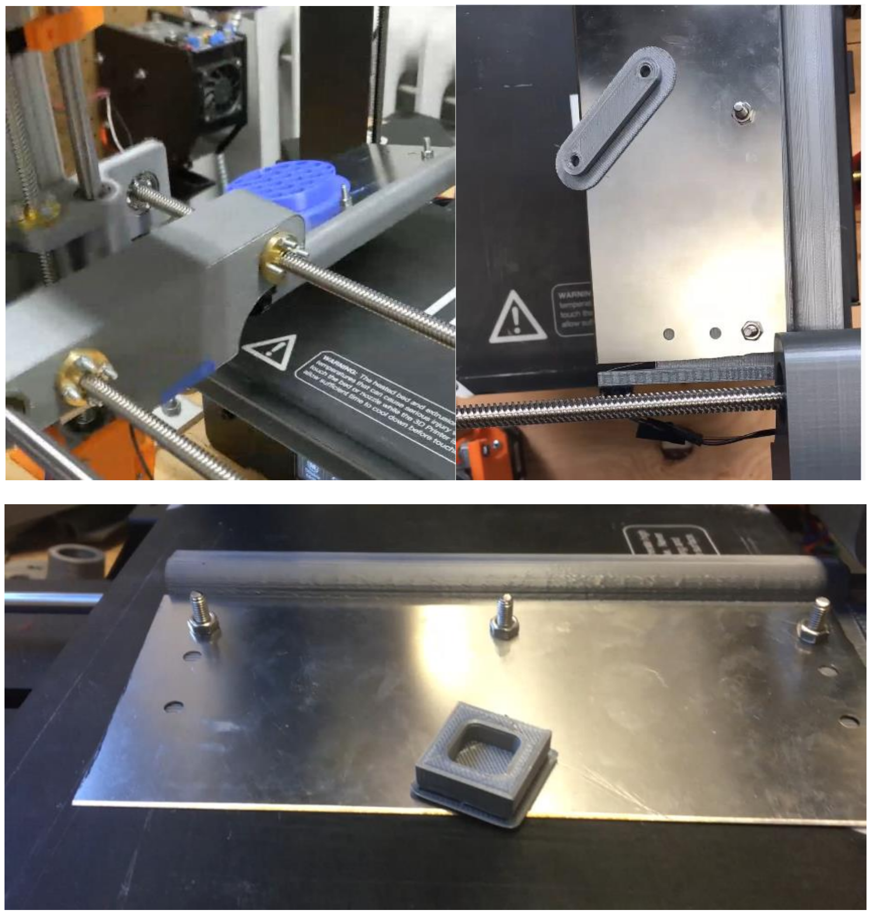


Figure vi: Examples of prints left on the scraper[6]

In a test for removing multiple parts at once, the team stated that the system successfully removed all three parts. Watching the video for this test, we determined the system failed for the same reasons as the previous videos. Figure 7 below shows the results of the multiple print test. While the square print did not get stuck on the scraper, it did not fully clear the print bed, and the circular and triangular prints did remain on the scraper. Thus, we defined this to be yet another failure under criteria (b). The parts going to different locations would increase the complexity of a sorting system for the printer.

[6] Reproduced as is from Hussain, M. et al.

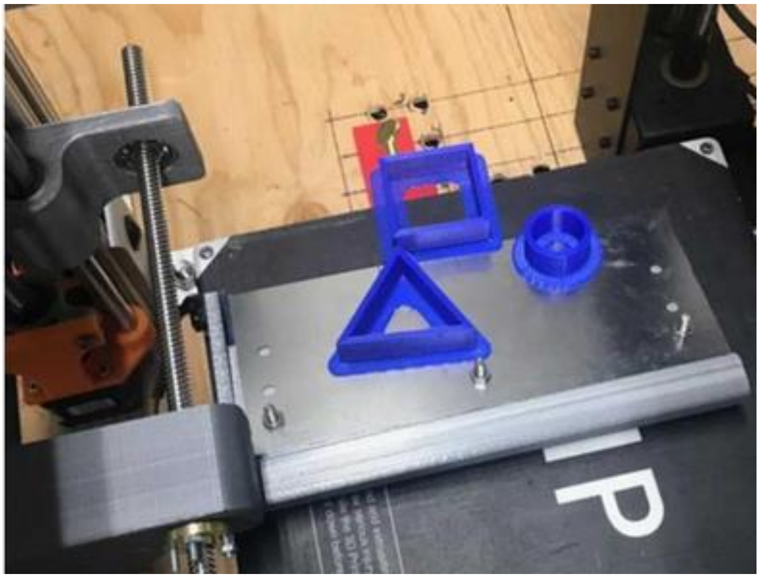


Figure vii: Failure to successfully remove multiple prints at once[7]

Out of the videos that we watched, the only print not to be lifted by the scraper after removal is a 5” diameter cylinder. Rather than remain stuck on the scraper afterward, the print fell off to the side of the print bed instead, shown below in Figure 8. Most of the circular print is hanging off the print bed and scraper, causing it to tip and fall off the side. While this behavior technically satisfies both criteria for a successful removal, it is inconsistent with the behavior of the system with smaller prints, which get lifted into the air by the scraper.

7 Reproduced as is from Hussain, M. et al.

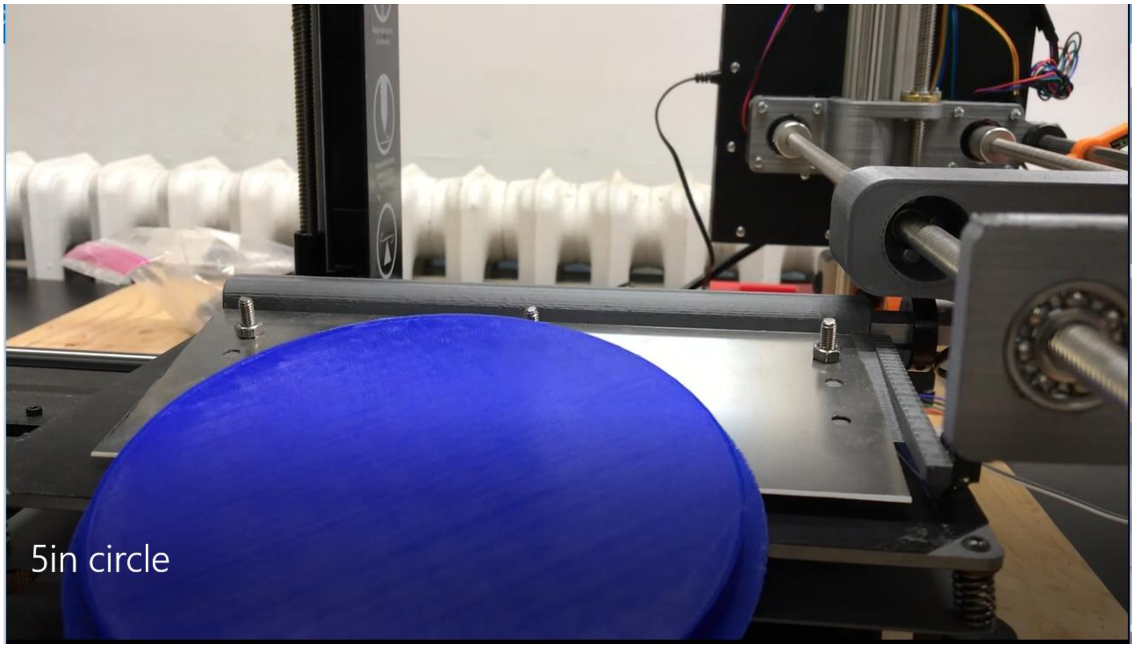


Figure viii: Large circular print falling off the side of the print bed[8]

Upon review of the team's documentation, it is unclear where removed prints were intended to go. The only design goals for the system are "Provide enough force to remove parts from the print bed" and "Produce accurate, repeatable motion." While the motion of the blade is accurate and repeatable, the end location of printed parts varies depending on size. This adds complexity into an automated system, as the system would need either a receptacle large enough to cover all possible outcomes or the ability to track where a removed print has ended up.

Another behavior we noticed from reviewing the test videos is that the scraper tended to deflect before breaking the adhesion between the print and the bed. Figure 9 below shows an example of the blade hitting the print, deflecting, and then snapping back to position. The leftmost image shows the scraper and print before contact, the middle image shows them on deflection, and the rightmost shows them after the adhesion is broken. The scraper bends backward as it attempts to break the bed adhesion, which puts strain on the arm and lead screws that drive the scraper, and it also indicates that the system is struggling to provide enough force to remove the print.

[8] Reproduced as is from Hussain, M. et al.

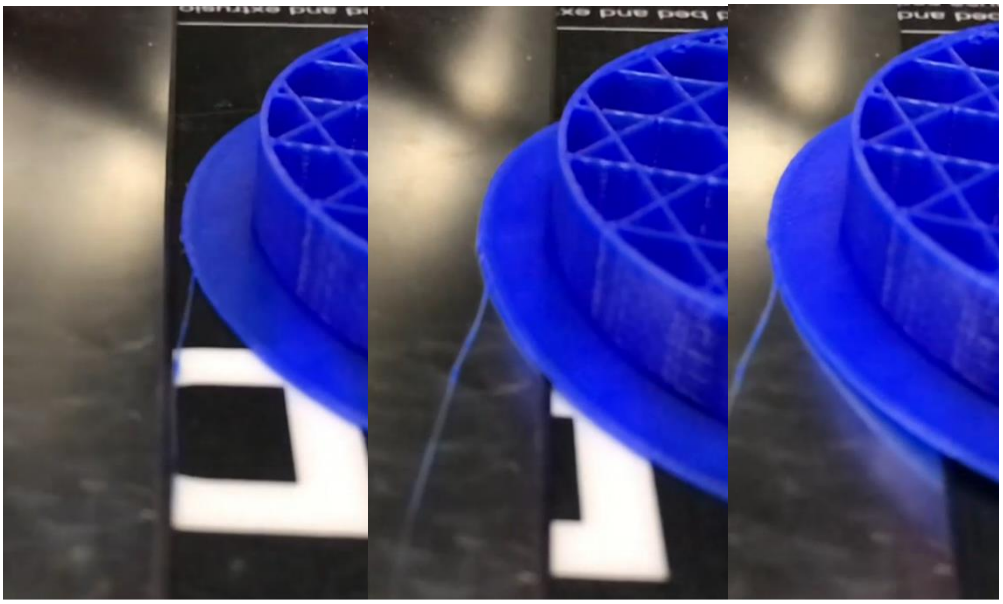


Figure ix: Scraper deflection before, during, and after breaking print-surface adhesion[9]

2.2. Further Testing of the Existing Design

To study the existing system further, we first had to reassemble the previous design, considering the improvements that they recommended. When we first got access to the system, all the electronics had been disconnected, and as a result, we spent a lot of time reassembling the electronics system. The most important problem that they had with the electronics was that the Arduino controller being used to run the stepper motors could not supply enough power to the motors to run them all simultaneously without reaching its current limit. We solved this problem by rewiring the circuit to power the motors using an external power supply rather than sourcing the current directly from the Arduino. After the circuit was modified, certain sections of the Arduino code had to be rewritten. The initial code included consideration for a relay to toggle power between the stepper motors; however, the feature was never implemented due to time constraints in the last project. Once the system was reassembled, we were able to run some of our own tests on it.

2.2.1. Additional Removal Testing

In testing their design, the previous team only used prints with rafts. A raft is a platform printed underneath the actual print, usually about 3 mm thick. Since rafts are removed from the final product after printing anyway, they can be damaged, and are intended to ease the removal of prints from the print surface. Printing with rafts makes it significantly easier to fit a blade between the print and the print surface. Though rafts do make it easier for users to remove their prints without causing damage, it is inconvenient to print a part with a raft, since the raft will need to be manually removed after the print is completed. Our first test was to test the capabilities of the system when removing prints that did not have rafts.

[9] Reproduced as is from Hussain, M. et al.

In our first test, we printed a large block with a skirt, no raft, and dimensions 8x3x1 in. This was a large print intended to push the limits of the previous design, and it took up about 30% of the print bed. The test can be seen below in Figure 10. The scraper system was not able to break the adhesion between the print and the print surface. Additionally, the scraper did not separate the skirt from the bed, indicating the potential for failures due to the print bed not being completely cleared. Significant deflection was also observed, as described in Section 2.1.

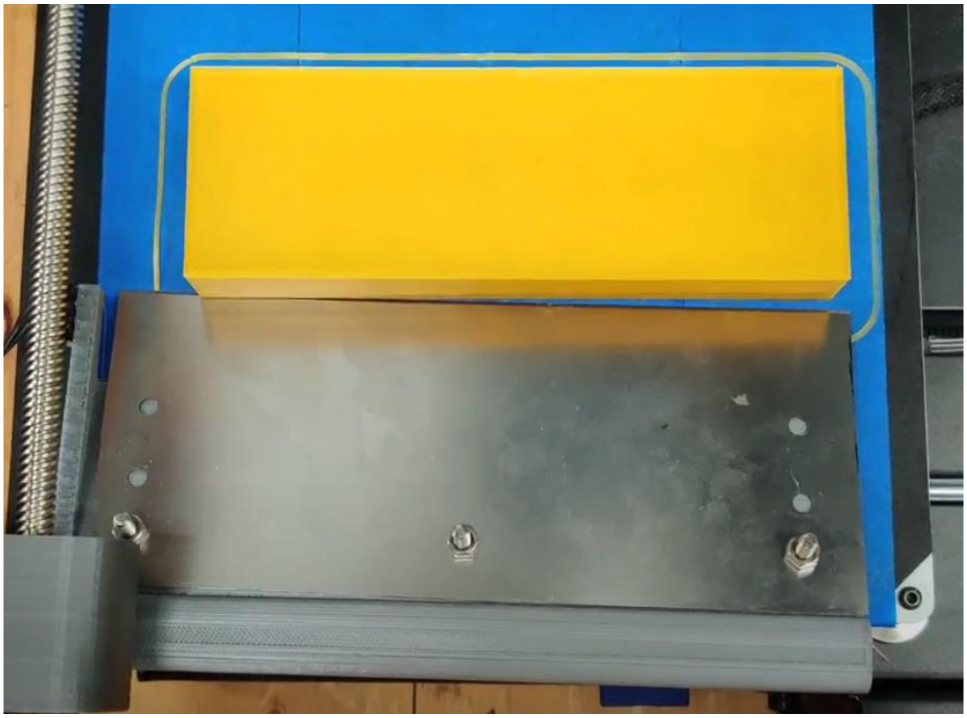


Figure x: Failure to remove a large print with no raft

To demonstrate that the previous failure was due to the absence of a skirt and not the large area of the print, we then tested a much smaller print, a star with dimensions of 25 mm between the points of the star. The test is shown below on the left side for Figure 11. In this case, the scraper also failed to remove the print, and caused slight damage to the leading point, shown on the right side of Figure 11. These tests demonstrated that the scraper could not remove any prints without rafts, which was a clear area for improvement.

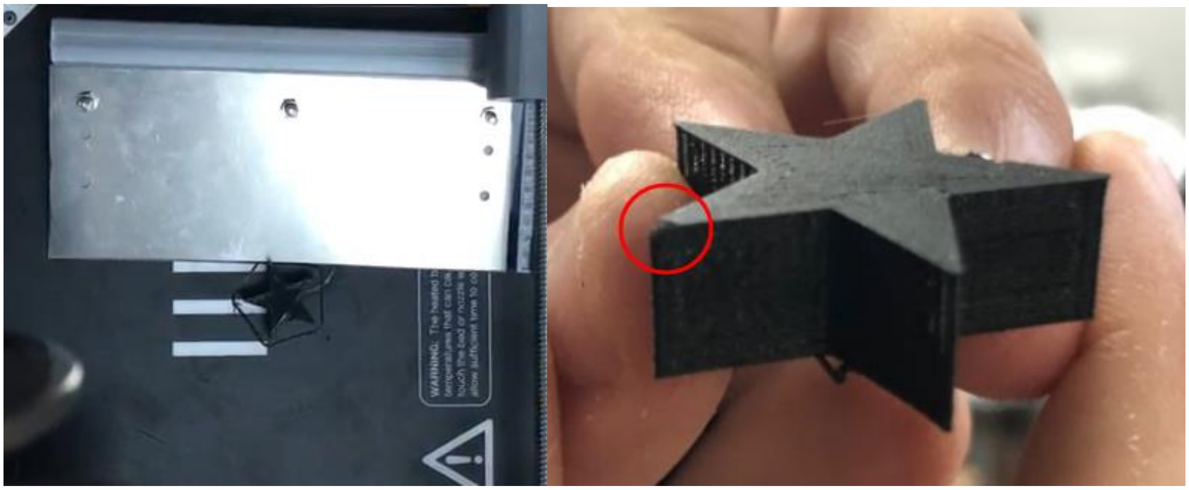


Figure xi: failure to remove a small print with no raft, and print damage

2.2.2. Testing for Bed Damage

While reviewing the previous team's tests, we were unable to find any tests that demonstrated the effects of the scraper on the bed. As such, we designed a cycle test to test the effects of many iterations of a scraper passing over the print surface. The test set-up is displayed below in Figure 12. Blue painters' tape, a theoretical print surface, was used on top of a flat, rigid surface as a control surface for testing bed damage. A weight was secured to the top of the scraper to simulate the force that would be applied downward into the bed during removal of a print. New code was written for the Arduino, which used stepper motors and limit switches to move the scraper back and forth across the bed.

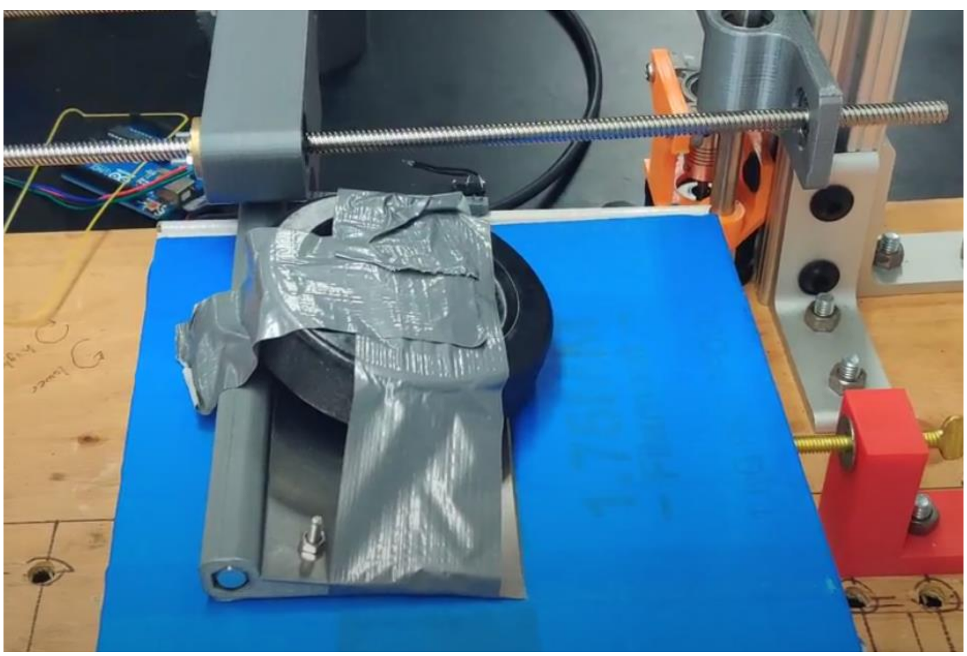


Figure xii: Setup of bed damage cycle test

The test was run for 250 cycles to simulate the life of a standard i3-style printer bed. This caused more damage to the print surface than anticipated. A picture of the resulting damage is shown in Figure 13. Large scrape marks are circled in red, showing the damage that repeatedly dragging a scraper blade over the print surface may cause.

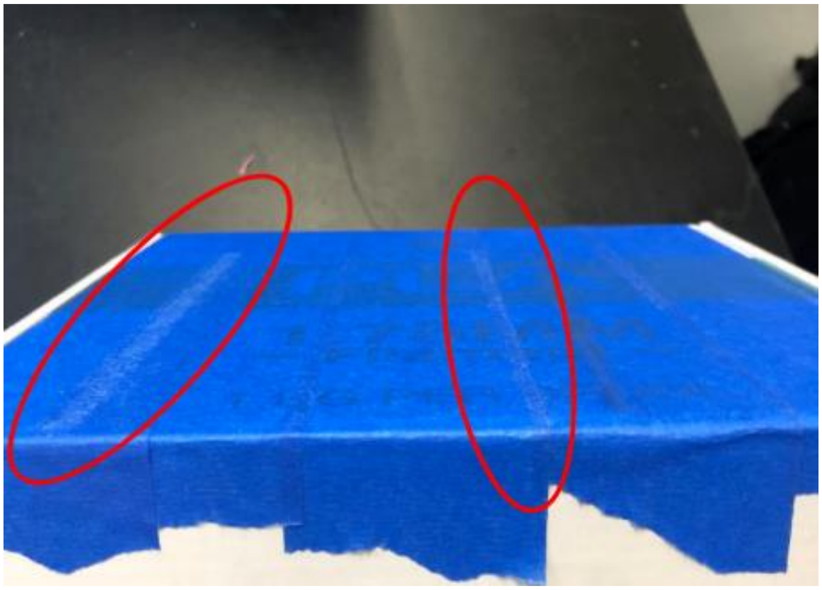


Figure xiii: Results of print surface cycle test

2.3. Limitations of the Previous Design

Many key limitations of the existing design were discovered combined with our analysis of the previous team's documentation and our own testing. The team stated that the removal system could remove a part with a maximum bottom surface area of 150 cm^2. The build area of the printer used for testing is 400 cm^2, meaning that when using the removal mechanism only 38% of the print surface can be utilized. The largest bottom surface area part observed in testing was the 5” diameter cylinder which was 126 cm^2 plus the area added by the raft. This condition significantly limits the capability of the printer. Additionally, the previous team’s report indicated that all prints must have custom raft geometry for the system to be able to remove it. Our additional tests confirmed these findings. The raft introduces waste material and increases the time it takes for parts to print. The team also stated that the orientation of prints was a significant factor in remove, as the scraper struggled to remove prints in cases where the initial contact edge was parallel to the scraper (much like the large block that we tested in Section 2.2.1). Instead they recommended that parts were oriented so that either corners or curved edges faced the scraper. Our team decided that the limitations of the previous design outweighed its benefits, and that a new system should be designed with these limitations in mind.

3. Literature Review

It is crucial to the success of the project to understand the history of the technology and how users interact with it. Researching previous attempts at automating the 3D printing process provided insight into which types of solutions could potentially work, and which will not work. It was also necessary to research some of the technical aspects of 3D printing to fully understand the process. Combined knowledge in these areas provided a solid foundation on which to base our design process.

3.1. Automatic Part Removal Methods

While most users remove their parts manually, there currently do exist some methods of automatic print removal. Some hobbyists are developing systems that will automatically remove their personal prints. Some establishments are also developing automatic removal systems to commercialize and sell with their printers.

3.1.1. Collision Removal

A common method for automatically removing prints is altering the g-code generated to use the print head to ram the print off the bed after its completed. After the model is sliced, and the g-code is generated, extra lines are added using a text editor that will move the x- and z-axis to be coincident with the print. The y-axis is then moved to cause a collision between the print and print head, knocking the print off the bed. This is demonstrated in Figure 14 below. The print head is pushing the completed part off the front of the bed.

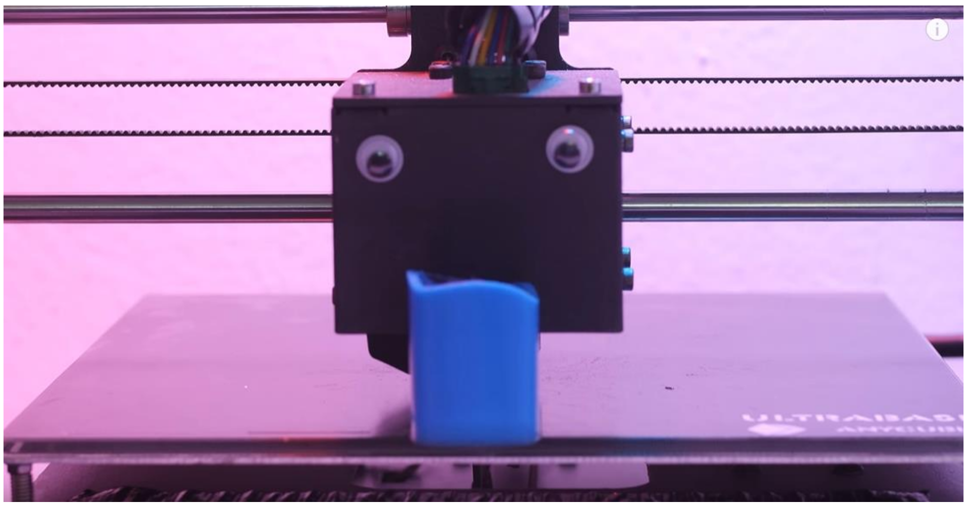


Figure xiv: Print removal by colliding the print head with the print[10]

This method of removal is very popular among hobbyists. Most users remove prints from the bed by just prying it off with their hands, in a very similar fashion to how this removal method works. It requires no printer modification or extra tools, just a few extra lines of g-code at the end of the file that goes into the printer. Since the lines are appended to the end of the print's g-code file, they will be executed as soon as the print is completed. The last lines of code

[10] Reproduced as is from https://www.youtube.com/watch?v=avlengYsJdw

for a print usually involve presenting the bed forward on the y-axis and homing the print head. Once these are done with print head will immediately be moved into the positions described above. At this point the heater for the print bed will have been turned off, but the bed will not have any time to cool and still be hot when these operations are executed. This method does however have its drawbacks.

This method of automatic print removal will not work with very small prints or prints with a low height. The print must clear at least the height of the extruder (as shown in Figure 14 above) or it will not be able to make enough contact to knock the print off the bed. While this height may vary slightly from printer to printer, a height of around 1 in is needed. It also becomes much harder as the surface area in contact with the bed increases. For example, a 3 in tall print with a 10 in^2 base is much harder to remove using this method than a 3 in tall print with a 5 in^2 base. Our removal system must be able to remove prints of all sizes and geometries. While used by very many people, this method is only effective under certain conditions.

3.1.2. Automated Scraper

Another removal method being used is an automatic scraper. The most notable example of this is that of New Valance Robotics (NVBots) on their 3D printers. Once a print is completed, the scraper travels from one end of the bed to the other, scraping the print off. The scraper then hits a wall and the print is pushed over the scraper and behind it, and the scraper reverses pushing the completed and removed print into a collection bin. The NVPro automatically initiates the scraping process after print completion, so there is no need for the user to append extra lines of g-code to the print file themselves. As a part of their software package, NVBots also has a queueing system that is linked to the printer that will start a print after the previous has finished. NVBots has a full patent on this system, which is pictured below in Figure 15.

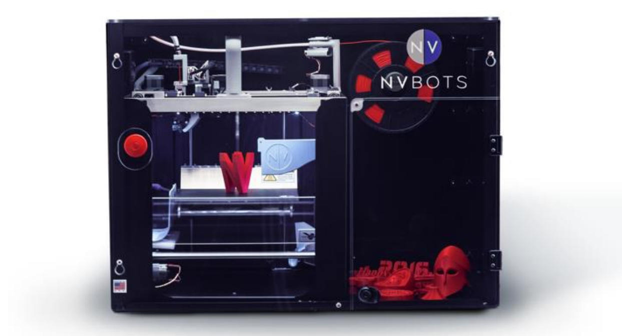


Figure xv: The NVPro system[11]

While this system uses the same conceptual method as last years' team of an automated scraper, the two systems have some key differences. The system designed by last years' team

[11] Reproduced as is from https://www.3printr.com/nvbots-announces-general-availability-nvpro-3d-printer-automated-part-removal-2937947/

was modular and meant to accommodate different models of printers in the i3 style. The NVBots system is completely integrated with their printer in one inseparable system. Also, the NVBots system is not i3-style but rather CoreXY (meaning the bed moves in the z-axis and the print head moves in the x- and y-axis).

This method has shown to be effective for prints of different sizes, and even for prints with large surface areas on the print bed. Because this method was proven to be effective, NVBots has made it commercially available as a part of their NVPro printer package. However, this system is completely integrated into the printer and is being professionally manufactured. This system could prove to be hard to make modular and work with other printers. Certain conditions regarding the print surface must be very accurate for this system to be effective. The bed leveling must be consistent to avoid the initial layer becoming more bonded to the print bed in certain areas, which would require much more force from the scraper to remove. The height of the bed must also be extremely consistent to ensure that the blade is contacting between the print and the bed, rather than two layers of the print, which would not scrape the part off the bed but rather be forcing the blade into the part. If a print is too hard to remove, the lead screw that drives the scraper will experience large amounts of torque and could damage the system. These are not issues with manual removal since human operators can account for any bed height or leveling inconsistencies. This scraper method cannot be used in tandem with manual removal or other removal methods as they may alter the bed conditions and lead to complications with using an automated scraper.

3.1.3. Disposable Bed Material

The "Continuous Build 3D Printer" by Stratasys [22] is a modular, automatic 3d Printer seen below in Figure 16. The system prints onto a thin polymer sheet which is stored on a roll at the rear of the machine. At the end of the print, the disposable material is advanced out of the front of the machine and a slicer separates it from the rest of the roll. The sheet falls into a bin at the front of the machine with all the parts adhered to it. The user than retrieves the sheet of parts from the hopper. The system (shown below) is 78 in tall, 31 in wide, and 35.5 in deep (including the collection bin). One unit consists of three printers, stacked one on top of another as pictured, and a blue collection bin attached in front of each. These machines come with Stratasys SkyLab, which includes a print queue system that interfaces with these systems. The user loads in their print files, and SkyLab directs them to the designated printers. These systems are designed for industrial use, not personal, and sold through Stratasys directly.

Figure xvi: Stratasys continuous build 3D printer[12]

3.1.4. Continuous Belt Printer

The final method that was researched was printing on a conveyor belt, and then rolling the print off the edge. This method is currently being developed in personal projects, but also on a commercial scale. The company BLACKBELT currently has a 3D printer that prints on a continuous belt, while holding the print head at a 45-degree angle to the belt. It has a max print volume of 340 mm wide and 340 mm tall, with no length limit due to the ability to print at a 45-degree angle. It is a desktop sized printer and sells on their website for £11,495.00, and spare belts sell for £356.95 [4]. The BLACKBELT 3D Printer is pictured below in Figure 17.

[12] Reproduced as is from https://www.stratasys.com/3d-printers/continuous-build-3d-printer

Figure xvii: BLACKBELT 3D printer[13]

STL files are loaded into BLACKBELT Cura, a slicer program that is meant specifically for this printer to slice parts at a 45-degree angle and generate g-code specific only to this machine. When the print is finished, it rolls off the end of the belt and is removed that way. Being rigid (in most cases, besides flexible filament), the prints cannot bend to match the contour of the roller, so the print peels away from the belt and is removed. Some examples of prints from the BLACKBELT are shown below.

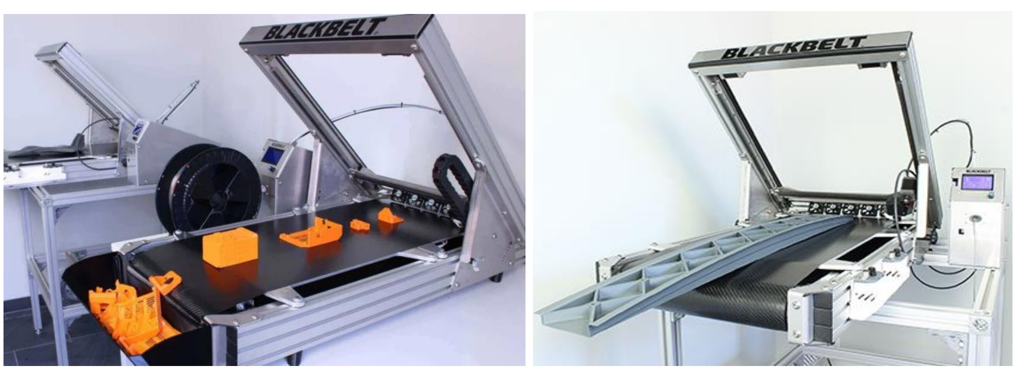


Figure xviii: Printing multiple parts consecutively (left) and infinite length printing (right)[14]

This removal method is also being used in personal projects. The White Knight 3D [15] printer is a project being worked on that follows almost the same structure as the BLACKBELT printer but is modeled as an open-source project that others can replicate.

[13] Reproduced as is from https://blackbelt-3d.com/blackbelt-3d-printer-desktop-version

[14] Reproduced as is from https://all3dp.com/blackbelt-3d/, https://hackaday.com/2017/05/12/another-printer-with-an-infinite-build-volume/

Swaleh Owais [2], a mechanical engineering student in Canada, has also developed a very similar system, with one major difference. Owais has developed his project not only to enable infinite length prints, but with more of a focus on a 3D printing "factory." His project uses the belt as an automatic print removal system that allows a part to be printed, rolled off the bed, and have another print initiated through the implementation of a print queuing software. The belt replaces the print bed and moves as the y-axis, and when the print is completed the belt simply rolls forward around one rotation or so to remove the print, and then the next print starts.

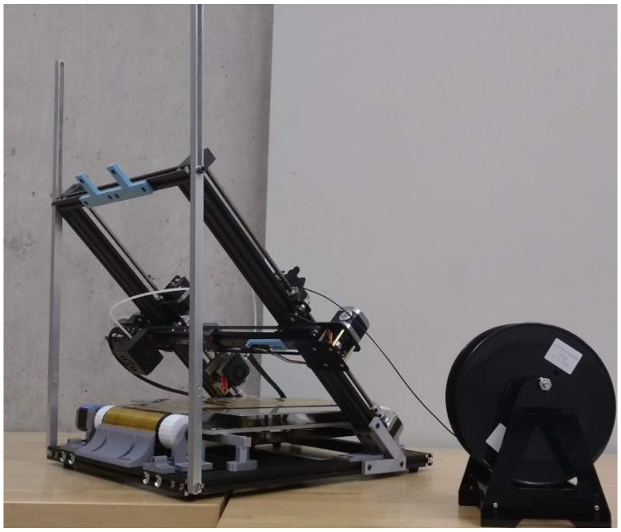


Figure xix: Conveyor belt 3D printer by Swaleh Owais[15]

There currently exist multiple methods for automatic print removal, however two of them are not modular, and one of them is very ineffective unless certain conditions are met. Ramming the print off the bed using the printer structure, while requiring no modification to the printer, requires prints to fall within geometric requirements, and will not remove smaller prints or skirts. The current scraper method is integrated directly into the printer, and an anomaly in a print could cause system complications. Finally, while the belt idea allows for reliable and effective print removal, if not well executed the quality of the print could be unreliable. All the current automatic removal methods have benefits and downsides, which were taken into consideration during brainstorming and design sessions, and ultimately factored into final design decisions.

3.2. Possible Materials for Printing

There are various filament materials that can be used to make parts with 3D printing. Each of these materials have different properties in terms of the temperature required for the extruder, temperatures required for the bed, as well as specific surfaces that each material adheres to well. On top of this, each material behaves differently in its difficulty of being removed from the bed, as well as its rigidity and fragility. The most common filaments used are PLA, ABS, PETG, TPU.

[15] Reproduced as is from https://hackaday.io/project/114738/gallery#36d727f76cc5dbdd7a12d6bba8c24b04

3.2.1. PLA

PLA (Polylactic Acid) is one of the most popular materials among hobbyists for a few reasons. It requires an extruder temperature of 205°C and a bed temperature of 40°C. These qualities allow for PLA to be used without a heated bed while still yielding good results. PLA adheres best to a surface of blue painter's tape [24] which is mainly made of polyacrylate [25], and is known for being odorless while printing, as well as being resistant to warping. The drawbacks to PLA are that the material is brittle and is generally less heat resistant than other filaments due to its melting point of 205°C. Being brittle will need to be considered for removal and sorting methods to ensure parts printed in PLA will not be damaged.

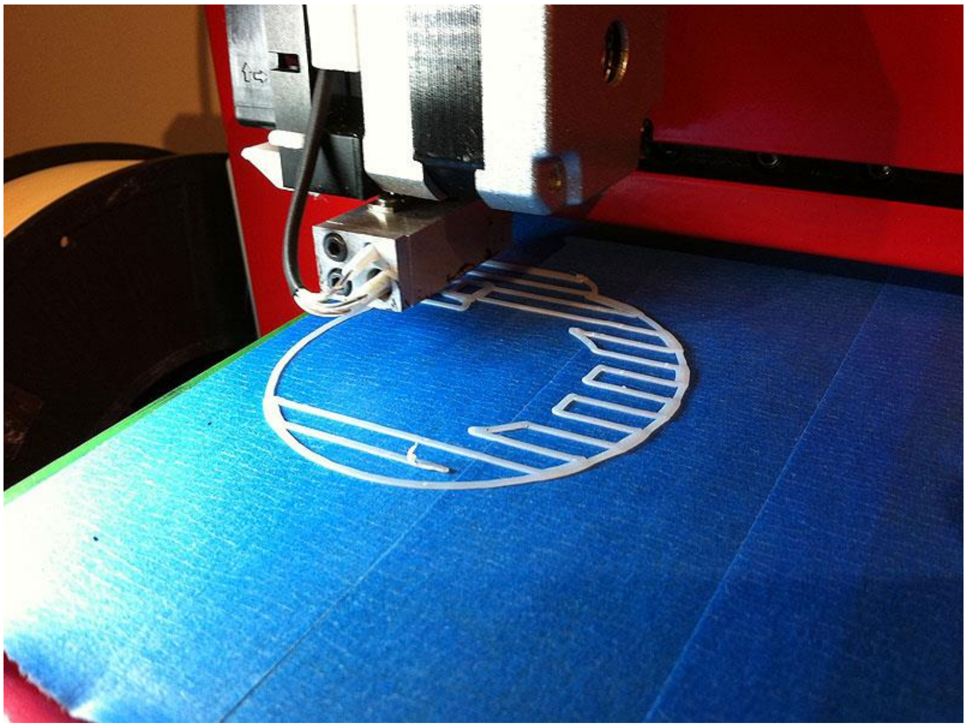


Figure xx: PLA print adhering to Blue Painter's Tape[16]

3.2.2. ABS

ABS (Acrylonitrile Butadiene Styrene) is another very popular material for many reasons. It requires an extruder temperature of 230°C and a bed temperature of 90°C. This means that although a heated bed is required to achieve good results, the parts will be much more heat resistant than parts printed from PLA because of its melting point of 230°C. ABS adheres best to a bed of Kapton Tape or hairspray [12], and has the benefits of being more durable and ductile compared to PLA but is more prone to warping.

---

[16] Reproduced as is from https://www.alibaba.com/product-detail/Cheap-Wallpaper-Blue-Painter-Masking-Tape_60098687164.html?bypass=true

Figure xxi: ABS part printed on Kapton Tape[17]

3.2.3. PETG

PETG (Polyethylene Terephthalate) is another popular filament for printing. It requires an extruder temperature of 245°C and a bed temperature of 60°C. Like ABS, PETG requires a heated bed for good results, but is more heat resistant than PLA or ABS because of its melting point of 245°C. PETG adheres best to a print surface of blue painter's tape, and is best known for its strength, and is also odorless while printing, warp resistant, and is food safe [12].

3.2.4. TPU

TPU (Thermoplastic Polyurethane) is another popular filament for printing. It requires an extruder temperature of 250°C and a bed temperature of 50°C. Like PLA, TPU is also able to be printed well without a heated bed. TPU differs from the other materials in that it is a much more elastic material, which allows it to be used for many different applications such as phone cases or stoppers [12].

Table 1 below summarizes the material specifications for the filaments discussed above:

Table 1: Material properties of the discussed filaments

Material	Melting Point	Ductility	Best Adheres to
PLA	205°C	Brittle	Blue Painter's Tape
ABS	230°C	Ductile	Kapton Tape / Hairspray
PETG	245°C	Ductile	Blue Painter's Tape
TPU	250°C	Ductile	Blue Painter's Tape

[17] Reproduced as is from https://community.ultimaker.com/topic/7140-glassbed/

3.3. Conventional 3D Printing Surfaces

Much like how the material being used to print the part is important to the end removal conditions, the surface in which it is printed also plays a significant role. There are many print surface materials used for 3D printing, each with their own strengths and weaknesses. A couple common surfaces used for printing are glass, polypropylene, as well as flexible surfaces like the Easy-Peelzy and the BuildTak FlexPlate.

3.3.1. Glass

One very common material for a surface to 3D print on is a flat glass plate. Glass is popular because it is gives parts it is printed on a very smooth and glossy finish. It is also very heat resistant and warp resistant, which makes it very useful for dealing with the conditions that 3D printing requires. However, glass is an extremely brittle material, and thus it can only be made into plates on which to print, which rules out belts if we decide to use it in our design. If a scraper is implemented, this could be an option for our system.

3.3.2. Polypropylene

Another very common material used for a print bed surface is polypropylene. Polypropylene is popular because it has good adhesion with most filament materials, can be very durable if cared for properly, and is very cheap. However, the material is easily scratched, and can be prone to warping. This makes it less feasible for our purposes because a low maintenance, durable material would be preferable for the print surface to minimize the human interaction with the removal system.

3.3.3. Easy-Peelzy

Easy-Peelzy is a flexible surface plate that has a magnetic bottom side. The plate is designed to go on top of the print surface, and once the print is done, the surface can be removed and bent to gradually remove the part by peeling the surface away. This method of removal is easier than using a scraper or pulling the part directly from the surface. This material has the potential for being used in a passive belt design, since the material can already bend around a radius to remove parts and can stay in place on any surface that is magnetic [8].

3.3.4. BuildTak FlexPlate

BuildTak FlexPlate functions very similarly to the Easy-Peelzy in that it is a flexible plate that is applied to and removed from the print surface to more easily remove parts [9]. The difference between this plate and the Easy-Peelzy is the lack of a magnetic bottom side for these plates. Because of this, a design involving the BuildTak FlexPlate over the Easy-Peelzy would have to have a different clamping system in place for the surface but would be able to function on printers with both magnetic and nonmagnetic surfaces.

3.4. Previous System Review

Hussain et al. developed a print removal system which utilized a mechanized scraper (Figure 22).

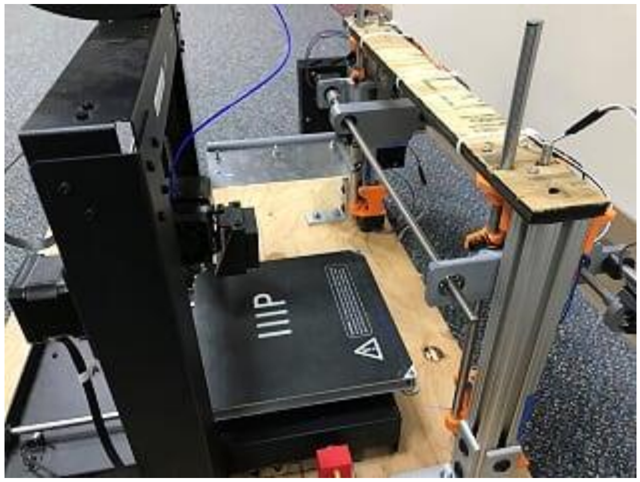


Figure xxii: Scraper design from the previous year

The final prototype was only able to remove prints which met a specific list of conditions.

1. Parts need to be printed on a raft
2. Maximum surface area of printed parts is $150cm^2$
3. Parts needed to be oriented such that a corner was pointed towards the scraper

The system also did not facilitate continuous printing as the removal system was unable to detect the end of the print. Instead, users had to enter the estimated print time into the removal system at the beginning of each print. The previous system also did not transport removed prints reliably, large prints would get pushed off the side of the bed while small prints would get stuck on top of the scraper.

3.5. Extra Print Conditions

When printing, many slicers have the options to add extra conditions to g-code such as a raft or skirt. As stated in the design requirements, the removal system should not limit the abilities of the printer. This means our system should reliably be able to remove these extra conditions.

3.5.1. Rafts

A raft is simply a platform printed underneath the part, usually around 3mm thick. The raft is printed using thicker layers than usual, sacrificing accuracy and appearance for speed. A very small air layer is left, and then the actual print is done on top of the raft. An example of a simple raft is shown below in Figure 23.

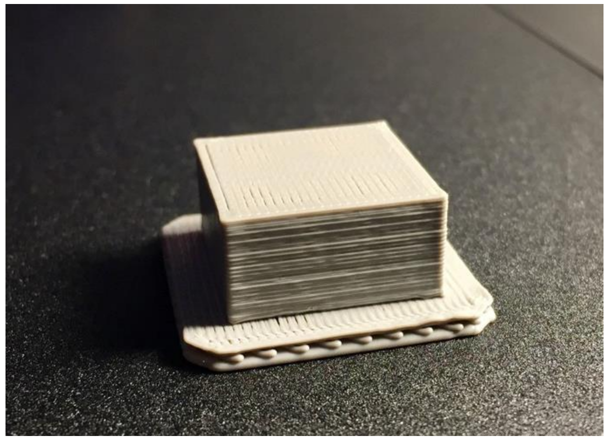


Figure xxiii: Part printed on a raft[18]

As pictured, the raft extends past the boundaries of the print. This increases the surface area in contact with the bed, helping with adhesion and preventing the edges of the print from peeling and warping. Rafts can be useful for parts with very thin geometries at different points since the raft will give the print more surface area at that point to adhere to the print bed. However, not all prints need rafts and if not needed it is usually recommended to print without one. Rafts can prove to be very tricky to remove and could even damage the part that was printed. If the part has a large surface area, the raft can take a lot of force and extra work to separate from the print after being pulled off the bed. The force it can take to remove the raft from a print is similar to that required to remove a print from the print bed. Sometimes rafts are stuck to the part so well, pieces of it break off and stick to the print instead of separating as intended. Smaller pieces of this can prove to be particularly tough to remove, particularly any pieces too small to be gripped with one's fingers and require the use of extra tools. Other times if the part is fragile or has very thin geometry, pieces of the part will break off because they are so well bonded to the raft. For these reasons, rafts should be avoided when possible. Lots of printers now come standard with heated beds, which prevents most warping or peeling issues anyways, further decreasing the need for a raft.

3.5.2. Brims

Brims are an intermediate condition between rafts and skirts (explained next). A brim is printed around, and coincident to, the part. Unlike a raft however, the brim is not printed underneath the part, and is only one layer thick. A brim is shown below in Figure 24**.**

[18] Reproduced as is from https://www.pinterest.ca/pin/411305378454450182/

Figure xxiv: Part printed with a brim[19]

Different slicers and settings will determine the width of a brim, but it is usually a few millimeters wide and always a single layer in height. Brims are used often, as they help with adhesion to the print bed and are easy to separate from the printed part. Brims are useful to prevent warping or peeling like rafts do, but they are much easier to remove. An automatic removal system may have trouble as it may be tough for a scraper to get underneath the single layer, and it may not peel from a belt since it is a single layer and will remain somewhat flexible.

3.5.3. Skirts

A skirt is an offset outline of the print that the printer lays down before starting the print, pictured below in Figure 25.

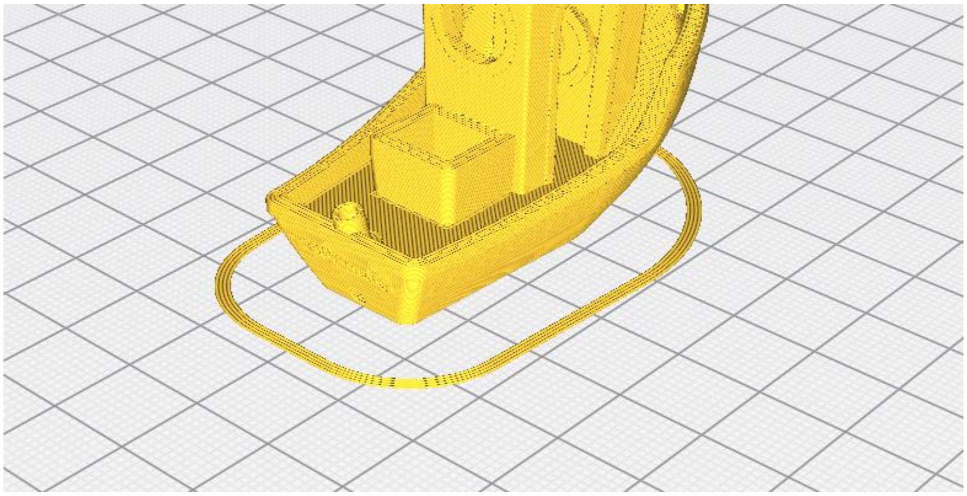


Figure xxv: Slicing software depicting a skirt

The skirts are only a single layer high, and while width varies depending on slicers and settings, are usually only a few passes wide. The purpose of the skirt, contrary to the raft and brim, is to establish a stable flow of filament through the nozzle before starting the actual print. While not

[19] Reproduced as is from https://all3dp.com/2/3d-printing-brim-when-should-you-use-it/

necessary, it can help ensure the print does not have any defects in the first few layers and adheres to the bed as intended. Our removal system must be able to remove the skirt as well. The skirt may prove difficult for an automatic removal system, since it is only a single layer. This means there is not much leverage for a scraper to get underneath it. It will also be flexible, meaning a belt system cannot remove it on its own. This will have to be taken into consideration when designing the removal system.

4. Methodology

The goal of this project is to increase the throughput of the 3D printing process by designing a system that facilitates continuous 3D printer operation. The main bottleneck in 3D printing is the downtime between prints. In most situations, a human operator must be available to remove completed prints from the bed and upload a new print job to the printer. Thus, our design seeks to eliminate the downtime caused by these two tasks, and the main objectives of our design are as follows:

1. The system automatically removes prints from the printer upon completion.
2. The system handles multiple print jobs continuously and sequentially.
3. The system requires little to no modification of the 3D printer itself.
4. The system is both modular in design and is scalable to include multiple printers.

This chapter describes our design process. The first section describes each of the four main objectives in detail and relates them to a set of system requirements that guided the design process. Once the system requirements were decided on, we began our initial design. The preliminary design underwent several iterations before the final design was completed. The last section describes the finalized design in detail.

4.1. System Requirements

Our team set system requirements based on the four main objectives. These requirements guided the decision making and development process. For each primary objective, there is a set of secondary objectives that must be met to fully achieve these goals. These objectives seek to make the system reliable, versatile, and user-friendly.

4.1.1. Automatic Removal

The main objective is for the system to automatically remove prints from the printer upon completion. To achieve this the bed is cleared for the next print without any interaction from a human operator. It is important that no filament remains on the bed when the next print starts, as this could cause the next print to fail. Because of this, we set a secondary goal that the printer does not limit the printer's capabilities. This means that any size print, and any special print settings, such as rafts and brims, can be removed. It is important that this secondary goal is met, because if a user was to queue a print that the printer can create, but our system cannot remove, a failure will occur.

4.1.2. Continuous Printing

The second main objective is that the system can handle multiple print jobs continuously. To achieve this goal, our system must be able to queue multiple prints, initiate the prints, and then remove them. Based on these requirements we set secondary goals. A secondary goal is to queue multiple print files. This means that print files can be uploaded and temporarily storage in sequential order. Another secondary goal is for the system to initiate the print jobs and initiate print removal. The system must be able to communicate with the printer to send files from the queue for printing and to signal for part removal upon print completion. We set three milestone goals for the number of consecutive prints our system can remove based on realistic scenarios. Our first goal is to initiate and remove for 12 consecutive hours. This simulates overnight printing, which is a time where a human operator may not be available to fix a failure. Prints can take anywhere from 30 minutes to several days to complete, but a realistic time for a typical print is two hours. This means our first goal is to automatically initiate and remove six prints consecutively without failure. Our next goal is 48 hours of printing, which would be 24 prints. This goal simulates the weekend which is a likely time when the printer will be unattended. The last milestone for our system is continuous automatic printing until the printer runs out of filament. This would be the limit of our system, as we are designing a print removal system, not a filament loader. Once the printer runs out of filament a human operator would need to replace the spool and resume printing. A typical two-hour print has a mass of 20 grams, so one 1 kg spool of filament can supply 50 prints. We may not reach each of the three milestones, but we can use them to measure our system's success.

4.1.3. Printer Modification

The third main objective is that the system requires little to no modification of the 3D printer itself. The purpose of this objective is to make the system easy to attach and remove from an existing printer without permanent modification to the printer itself. Tampering with the mechanisms of a printer can cause damage and would make repairing the printer or removal system more difficult. This objective allows typical 3D printers to quickly be automated and returned to their original state if necessary. It also allows a single removal system to be moved from one printer to another. This could be useful to an operator, as printers have different capabilities, such as print size and compatible filament types.

4.1.4. Modular Design

The fourth main objective is for system to be both modular in design and scalable to include multiple printers. The system being modular means that it's subsystems can be scaled, modified, or replaced to make the system compatible with different printers. Our specific design goal is a system that is compatible with any i3 style printer. i3 printers have a range of sizes and features such as removable beds which our system will be able to account for. All i3 style printers have the same axis layout and open access to the bed which provides us with a reasonable baseline for design and allows the system to utilize a universal removal approach. A secondary objective to making the system scalable to multiple printers is that it can communicate with more than one printer and can remove parts from more than one printer. The removal system must be able to attach to multiple printers and the print queue can send jobs to multiple printers. The purpose of this would be to connect multiple printers in a print farm configuration to achieve more throughput.

4.2. Design Process

Once the system requirements and background research were completed, it was possible to begin the design process, displayed in the flow chart in Figure 26. After deciding on a set of requirements, the team collectively brainstormed as many different design ideas as possible. From there, we selected the most viable options based on feasibility, reliability, and how well they met the system requirements described above. Next, we developed some rough designs to get a better sense of what would be required were we to move forward with them. From these designs, we were able to perform a detailed decision analysis, described later in Section 5.1.

These steps formed the brainstorming and design selection phase, represented in yellow in the chart. The next phase was preliminary design, represented by blue. We devised a detailed design based on the solution we had selected and tested its feasibility. Eventually, we realized that our initial design was not feasible, and that the system would need to be redesigned. This process is designed in section Chapter 6. The last stage, design finalization and assembly, is represented by green. In this last phase, we finalized our detailed design and assembled the system. This also included thorough testing after the prototype was completed to ensure that the design met all the requirements described above.

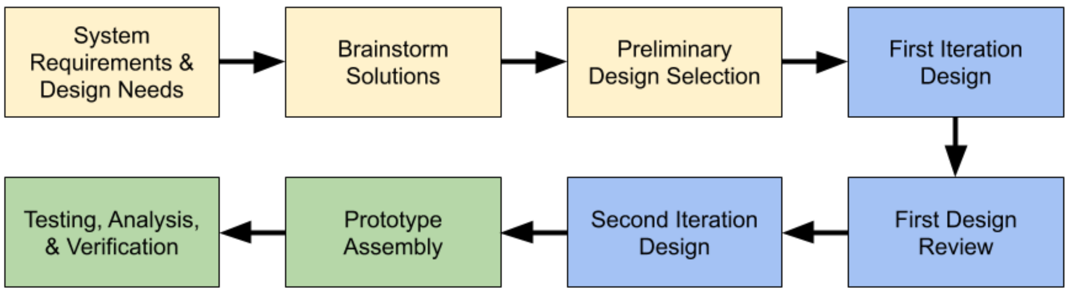


Figure xxvi: Design process flow chart

5. Preliminary Design and Testing

From our preliminary analysis, it was clear to us that we needed to redesign the system in order to meet our design requirements. To achieve this, we started with a brainstorming session. Brainstorming started with listing methods of automatic removal that we thought of, in a very general sense. These methods included the following:

- Scraper
- Belt
- Impulse Force
- Bed Separation
- Rapid Cooling

From this point, the two most reasonable and feasible categories were chosen: Scraper and Belt. The scraper was selected to use as a comparison for the previous removal system. The belt was selected because of the previous success shown by the devices listed above. Impulse force was rejected due to its issues with handling smaller prints and prints with large surface areas seen with the extruder ramming. Bed separation and rapid cooling were rejected because these methods were assumed to require heavy modification to the printer.

Brainstorming continued with more detailed focus on each category. There were a few different ideas for the scraper system. First, there was the linear system that last year's team used. Second, there was a rotating scraper that would sweep across the bed, which would be positioned near one corner of the print bed once the print is finished. Lastly, there was a scraper that included the impulse force idea, which would strike the part along the print bed to get underneath the part and lift it off the bed.

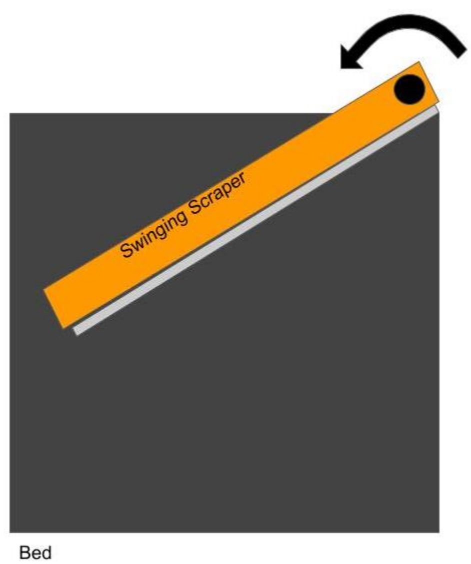


Figure xxvii: Rotating scraper

There were also a few ideas for different types of belt systems. First, there was an active belt that would replace the y-axis bed movement, as seen by the BLACKBELT design. Second was a passive belt that would simply clamp onto the bed until part removal is necessary. This belt would need to be slacked in order to avoid interfering with printer operation. Finally, there was the disposable belt where the part would be printed on a surface that would be rolled onto the printer bed, clamped down during printer operation, rolled off the printer bed with the completed part, and cut from the roll to release the part.

Each of these systems then underwent preliminary judgement through a group discussion. After considering the limitations we discovered from the previous team's scraper system we decided not to pursue a scraper design out of concern of being bound by similar limitations. Because of this, belt designs were the focus of these discussions. The main limitations for the systems go as follows: For the active belt, the main limitation was the amount of modification to the printer that would be required. For the passive belt, the main limitation was the amount of space available to attach the passive belt to the printer. For the disposable belt, the main limitation was the requirement to keep the material rolls stocked.

5.1. Design Selection

Grading criteria were created to be used in our design matrices. Four design ideas were put into the decision matrix to decide which idea we would be pursuing: the scraper, an active belt, a passive belt, and the disposable belt. Each of the designs were rated on a scale of one to four, for the criteria of adaptability, modularity, cost, maintainability, reliability, and size, with four being the highest rating. Each of the grading categories were also given a weight ranging from zero to one, with all six of the weights adding up to 1. The grading criteria used in the matrices are as shown in Table 2.

Table 2: Criteria for each score in the decision matrix

Criteria	Description for each grade				Weight
	4	3	2	1	
Adaptability	All components within the system can be installed onto any i3 style printer without need for printer specific customization	Most components within the system can be installed to i3 printers without need for printer specific customization	A few components within the system could be installed to an i3 printer without need for printer specific customization	System must be specially designed for each different model of printer.	0.2
Modularity	System is independent from the printer, and can be easily removed from the printer and replaced entirely	System is dependent on the printer but can still be easily removed from the printer and replaced.	System is dependent on the printer but can be removed from the printer and replaced with some difficulty.	System and printer are completely dependent on one another, and the system cannot be removed from the printer once installed.	0.2

Cost	< 25% of our budget	25-50% of our budget	50-75% of our budget	> 75% of our budget	0.05
Maintainability	System can be maintained without the need for disassembly	A few parts need to be removed for maintenance	many parts need to be removed for maintenance	Major disassembly is required for maintenance	0.15
Reliability	Always accomplishes the desired task regardless of any potential caveats with the part	Can reliably accomplish the desired task for normal parts, but may struggle with special conditions	Can only reliably accomplish the desired task for normal parts	Not always capable of completing the desired task	0.3
Size	System fits within the original envelope of the printer	System takes up a large amount of space in only 1 dimension	System takes up a large amount of space in only 2 dimensions	System takes up large amounts of space in all 3 dimensions	0.1
Total					1

We decided that the most important criterion was reliability. This system is intended to operate autonomously without any operator oversight, therefor the system must be reliable for it to function effectively. Modularity and adaptability were ranked as equally important, they determine the ease and compatibility of installing the system to different printers. Next in importance is maintainability, followed by size and finally cost.

The decision matrix used to choose a high-level design to be used is shown below in Table 3.

Table 3: Belt decision matrix

Design Idea	Reliability	Adaptability	Modularity	Maintainability	Size	Cost	Grade Total
Disposable Belt	4	4	4	3	3	4	3.75
Passive Belt	3	3	4	2	3	4	3.1
Active Belt	2	1	2	1	4	4	1.95

Disposable Belt

- Reliability: The disposable belt received a score of four for reliability. The printing surface is refreshed every print. If a print failure occurs it will not inhibit the following prints as the printing surface and failed print will be cleared from the system. The original motion of the printer bed is used during the printing process. The only components which require mechanical actuation are clamps to hold the disposable printing surface to the bed, a blade to cut the disposable printing surface and the roll of printing material to cover the bed after cutting.

- Adaptability: The disposable belt received a score of four for adaptability. It can be installed to many different i3-style printers. The Roll of material is situated at the rear of the machine and the blade is located at the front. The only component which would need to be physically connected to the printer are the clamps which hold the printing surface to the bed.
- Modularity: The disposable belt received a score of four for modularity. The system is comprised of three modules, the roll, the clamps, and the blade. The modules can easily be installed or uninstalled to the printer without major modification to the system
- Maintainability: The disposable belt received a score of three. The only moving parts in this system are the clamps, roll, and blade. The largest maintenance item for this system is replacing the roll which is why it received a three.
- Size: This design only extends in the z direction which is why it received a score of three.
- Cost: There are not many electrical components or sub systems needed which will keep costs low.

Passive Belt

- Reliability: The passive belt received a score of three for reliability. The reason for this is that it is possible for small remnants of prints to be left on the belt after removal. These remnants would interfere with subsequent prints
- Adaptability: The passive belt received a score of three for adaptability. Some parts within the design would need to be custom for different printers.
- Modularity: The passive belt received a score of four for modularity as the system can be easily installed or uninstalled to an existing printer
- Maintainability: The passive belt received a score of two for maintainability. The belt may receive uneven wear depending on the location of repetitive printing, making regular maintenance difficult to predict
- Size: The passive belt received a score of three for size as the system only increases the y dimension of the printer.
- Cost: There are not many electrical components or sub systems needed which will keep costs low.

Active Belt

- Reliability: The active belt received a score of 2 for reliability. The reason for this is that it is possible for small remnants of prints to be left on the belt after removal. Additionally, the active belt replaces the y axis of the machine so the actual printing of parts could be impacted by any faults with the removal system.
- Adaptability: The active belt received a score of one for adaptability. The active belt is replacing the y-axis of the printer and would have to be custom for different printers.
- Modularity: The active belt received a score of two for modularity as the printer needs heavy disassembly and modification to install the system.
- Maintainability: The active belt received a score of one for maintainability. Major disassembly of the system would need to occur to maintain the mechanism

- Size: The active belt received a score of four for size as the system does not increase the footprint of the printer in any dimension.
- Cost: The system reuses the electronics from the y-axis of the printer which keeps costs low.

As can be seen above, the disposable belt idea was chosen due to it outperforming the other designs in almost every category. The active belt design was the lowest-rating design due to its complexity, required interfacing with the machine, and difficult adaptation. The passive belt was in second place and while identified as a good option to pursue for the system, the disposable belt idea rated slightly higher in multiple categories. The disposable belt option scored the highest marks in all categories but maintainability and size, categories in which no designs scored a 4.

5.2. High Level Design: Disposable Belt

The team began working on a high-level design for the disposable belt removal system. Parts will print onto a sheet of material that will sit on top of and be clamped to the printer bed, upon completion, the part will remain attached to the material which will be removed from the printer.

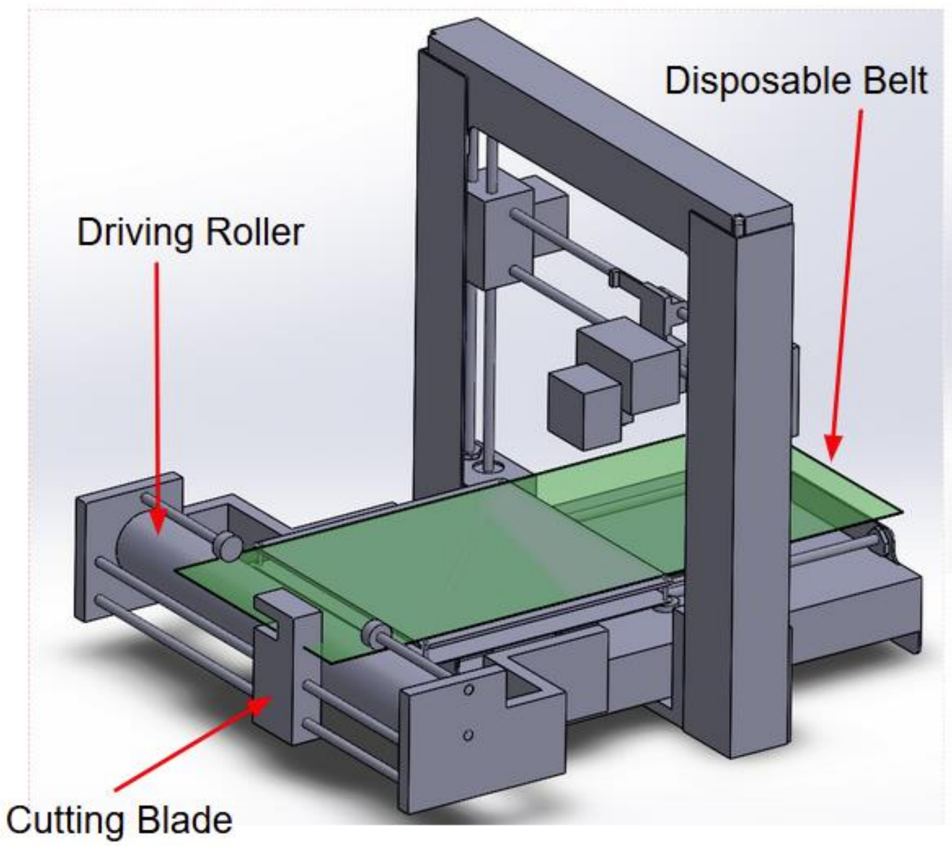


Figure xxviii: Preliminary CAD model of the disposable belt system

Figure 28 shows the disposable belt design. The design consisted of a large roll of disposable belt material behind the printer, and a linearly moving blade to slice this material.

There is a driven roller in front of the printer that will roll the material onto the bed, it will be clamped down, and when the print has completed the roller will pull the material forward and the blade will slice it. This puts a new layer of material on the bed and prepares the system for the next print.

5.2.1. Print Surface Material Testing

From our decision matrix shown above in section 5.1, we concluded that printing onto a belt would be the best option for our project. There are a lot of materials on the market that consumers have attempted to print on. The most common print surfaces are PEI, glass, and painters' tape. Each of these materials have very different surface properties. The surface roughness of glass [4] and painter's tape [5] can be seen in the table below.

Table 4: Surface roughness of common printing surfaces

Material	**Surface roughness (μm)**
Glass	0.5
Painter's tape	2.5

Since both materials are widely used print bed materials, the discrepancy in surface roughness led our team to believe that there is more to print adhesion than surface roughness alone. To try to understand what materials work well as print surfaces, we tried printing on materials with differing properties to try to find cheap, common materials that we could use as our bed. We tested plastic transparencies, aluminum foil, wax paper, cardboard, and present paper. We chose these materials because they are cheap (less than $0.13 per print) and are heat resistant. This price was calculated by calculating the price per square inch of the material being tested and multiplied it by the area of the print bed. We created a table to keep track of the materials and the quality of prints produced by each. This table can be seen below. The purpose of this experiment was to help us determine what types of materials we should be continuing to test in the future.

Table 5: Test results for printing on various disposable materials

Material	Did the initial print layer adhere?	Comments
Stock Print Bed (control)	Yes	Stuck well, some curling on first layer, overall bottom has good definition
Wax paper Dull (WSD)	Yes	First layer stuck up more than just bed, not as defined as the control. Stuck well. Easier to remove than control
Wax Paper Shiny (WSU)	Yes	First layer stuck better than 2 predecessors, harder to remove than WSD but easier than bed
Aluminum Foil (non-shiny)	No	Skirt stuck, great surface finish, part warped and came off after multiple layers. Requires more testing
Transparency	Yes	Great surface finish, stuck well, need to test further to confirm bed adhesion. Produced the best surface finish, seemed too easy to remove. Requires further testing
Cardboard	Yes	Stuck well, very hard to remove print, cardboard and print became fused (could not separate print from paper)
Present Paper	Yes, too well	Stuck, ripped the paper off with it. Little bit of warping.

We found that the transparency gave us the best finished-part quality and provided a surface that was very east to remove the part from. The part stuck, but the adhesion did not seem as sturdy as the control. The parts stuck well to the paper, but the paper became fused with the first layer of the part.

6. Detailed Mechanical Design

While conducting material testing, our team decided that it was unacceptable to leave behind remnants of paper infused on the first layer as shown in Figure 29.

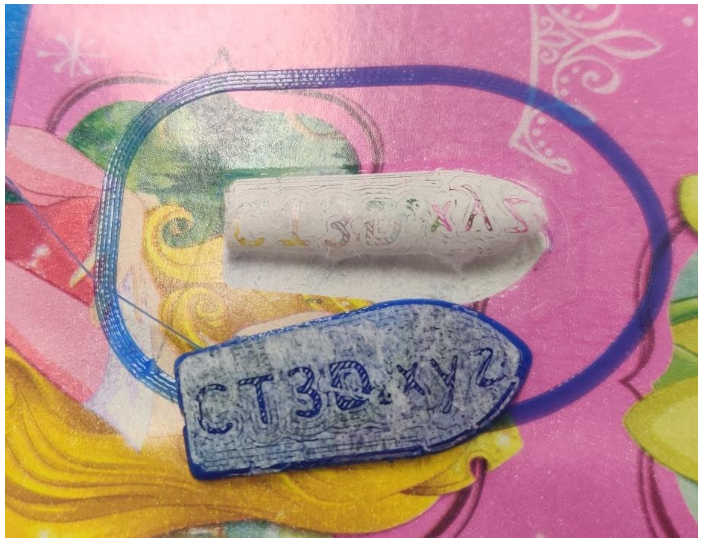


Figure xxix: Paper residue from printing on wrapping paper

When discussing potential alternatives, printing onto a flexible sheet was proposed as a solution. The idea was simple in that the printer would print onto a sheet metal plate, and then the plate would be flexed (like how current flexbeds are used) using our device. As discussed in the introduction section two printers which use flexbeds are the Creality Ender 3 and the Prusa i3 MK3.

6.1. Flex Bed Design Overview

We liked the bed flex idea because there would be minimal maintenance, it is completely autonomous, and is non-damaging to the part. There is minimal maintenance since there are no consumable materials in the design. To confirm our assertions, we compared the Disposable Roll model to the new Bed Flex model It would be completely autonomous using our sensing and queueing system. It would also be non-damaging to the part because the system would mirror how human operators remove parts from printers using a flex plate. The decision matrix can be seen below in Table 6. To help our team consider the electrical components in our design, we added a sensor ability section. We replaced "Cost" with this new section since we found that our designs required similar components such as stepper motors and aluminum extrusions making them similar in price to build.

Table 6: Decision matrix of disposable belt vs. bed flexing

Design Idea	Reliability	Adaptability	Modularity	Maintainability	Size	Sensor ability	Grade Total
Disposable Roll	3	3	3	4	3	2	3.1
Bed Flex	4	3	4	4	4	3	3.75

The Bed Flex system scored higher than the disposable roll for Reliability Modularity, Maintainability size and Sensor ability. During the testing of our disposable belt surfaces we found that some parts were warping from poor bed adhesion. The Bed Flex design has a higher reliability rating because parts would print on the stock printer surface. The Bed Flex design is more modular as it simply sits in front of the machine instead of needing to have a clamping system attached to the bed. It scored better on size as it only sits in front of the printer. Finally, the belt flexing system had more areas which we could attach sensors such as limit switches which scored it a higher sensor ability score. Since, the Bed Flex model outperformed the disposable roll design in most categories, so our team shifted focus to refining its design and beginning to build a prototype.

6.2. FlexTrak High Level Design

Our system must cover all the principles described in our design requirements which are automatic removal, continuous printing, printer modification, and modular design. We found that there are five subsystems necessary to cover these requirements, they are the 3D print controller, microcontroller, bed removal system, bending track, and scrapers. The system architecture diagram in Figure 30 shows the interactions between the subsystems in our design, the "FlexTrak".

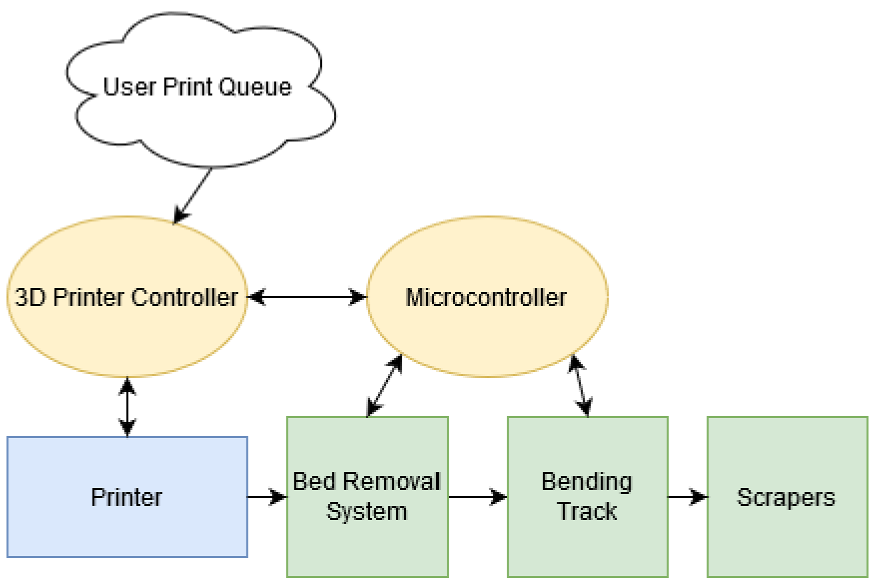


Figure xxx: System architecture diagram

The user interacts with the system by adding files to the print Queue which is hosted by the 3D printer controller. The 3d printer controller communicates with the printer to initiate prints and to the microcontroller to initiate print removal. The microcontroller controls the behavior of the bed removal system and bending track. The bed removal system removes the bed from the printer and introduces the bed to the bending track. The bending track flexes the bed which breaks the adhesion between the part and the bed. Finally, the scrapers break any final adhesion and remove small pieces of plastic from the bed which were not removed from the bed during flexing. The process is reversed to load the print bed back onto the printer. The bed removal system, bending track, and scraper system are discussed in greater detail in the following sections, the 3D printer controller and microcontroller are discussed in Chapter 7, Detailed Electrical Design.

6.2.1. Bending track

The first subsystem we designed was the bending track. The bending track needed to provide a way to bend the print bed in order to reliably break the adhesion between the part and the bed. We produced a CAD model which can be seen in Figure 31 which shows a preliminary design for a bending track. The track has a vertical displacement of 50 mm over a length of 270 mm.

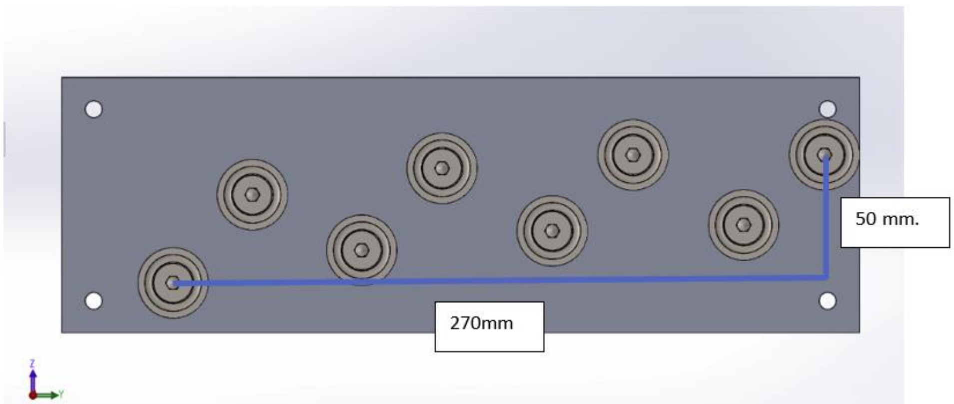


Figure xxxi: Preliminary design of the bending track

The sheet would be fed through the bearing track, causing the sheet to bend. This bend would break the surface tension of the part. The bearing track is the best way to flex the bed because we can bend the material using the linear drive of a readily available motor. Not shown in the model is how we are going to feed the sheet through the track. It will be fed through with two stepper motors with a 1:5.18 planetary gear reduction attached in the factory. These motors were chosen due to their availability and inexpensive cost, and the 1:5.18 gear reduction provided 2 Nm of torque, which was more than enough for our applications. Figure 32 shows the bed is fed forward (represented by the green arrow), the top set of bearings on each side of the system will force the bed to deflect downwards (represented by the blue arrow). The bottom set of bearings on each side will support the bed and ensure it bends along the prescribed track. The bed will be 14mm wider than the default print bed. This allows for the 7mm thick bearing on either side to engage completely with the bed without reducing the printers print volume.

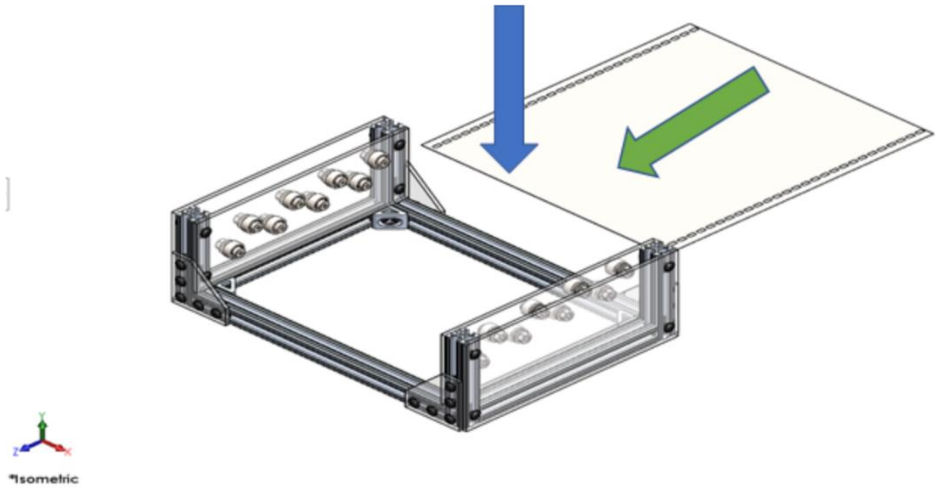


Figure xxxii: Bending track

6.3. Bed Removal System

The bed removal system is responsible for removing the print bed from the 3D printer and introducing it to into the bending track discussed in the previous section. This subsystem helps facilitate the design requirements related to automatic print removal and continuous printing. The bed removal system also helps to remove the need to modify the printer in support of the printer modification requirements. The bed removal system underwent two design iterations, the sprocket system and the elevator system.

6.3.1. Sprocket System

The First design iteration utilized a sprocket which would drive a custom fabricated bed into the bending track. We were initially concerned that using a smooth roller to grip and drive the bed would leave the possibility for slipping or wearing down over time, and not be able to reliably drive the bed into the track. To avoid this, we wanted to use a mechanism that had a more reliable engagement method. The operation of the sprocket is very similar to how the sprocket and chain work on a bicycle. The bed has cutouts that coincide with the spacing of teeth on the sprocket. As the sprocket turns, the teeth engage these cutouts and drive the bed forward, just how a bike sprocket drives the chain. This way the torque from the stepper motors is reliably transmitted through the sprockets onto the bed. The initial sprocket design was inspired by a bicycle sprocket and created from scratch. The sprocket had 10 teeth and had an overall diameter (including tooth length) of 51 mm. The diameter was decided upon based on the dimensions of the motor mount. The stepper motor mount was 49 mm tall, so making the diameter 51 mm would allow for 1 mm clearance on either side of the mount. This provided enough length to properly engage the bed cutouts but were not so long as to prevent proper engagement and disengagement.

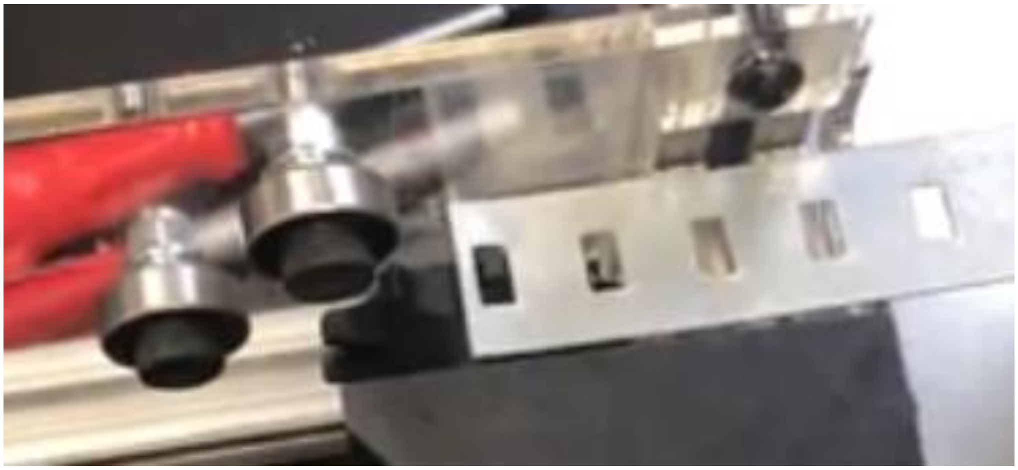


Figure xxxiii: Sprocket-bed interface

This design was based around i3 style printers that did not have an integrated magnetic bed, and it required a custom bed to be cut (to include the line of rectangular cutouts) for each printer being used with the system. Some printers come with a removable bed that attaches to the print surface via magnets. The original design was centered around a MonoPrice Maker Select Plus, but it was soon changed and designed around a Prusa i3 Mk3. The MonoPrice had a manual bed leveling system and was difficult to get consistently level under normal operation, let alone after a custom bed was added that may experience some slight deformation after repeated flexing. For these reasons the decision was made to change to the Prusa i3 Mk3 to produce a more effective and reliably system. The Prusa had an integrated magnetic bed meaning there was no need for a clamping system and had auto bed leveling as well as features such as filament runout detection, crash detection, and functionality which allows the printer to resume a print after power loss. The additional features of the Prusa carried the potential to make the system more reliable than the MonoPrice machine was capable of. Reliability is the most important design criterium for the system so the team decided that this switch was necessary.

To avoid having to make a custom bed to replace the removable bed the printer comes with, the choice was made to switch from using sprockets (shown above) to rubber wheels (shown below as the green wheels in Figure 34). Using the included Prusa bed meant less cost and complexity that came with manufacturing a new bed. It was also much simpler than ensuring the sprockets engaged correctly each time. The 30-durometer rubber provided sufficient traction to the bed and transmitted 2 Nm of torque from the planetary steppers to drive the bed through the track. Figure 34 shows the front and back of the bending track assembly. One of the wheels was directly powered by a planetary stepper, the power was transmitted to the other wheel with a GT2 synchronous belt so that both wheels are powered and driven by the same motor.

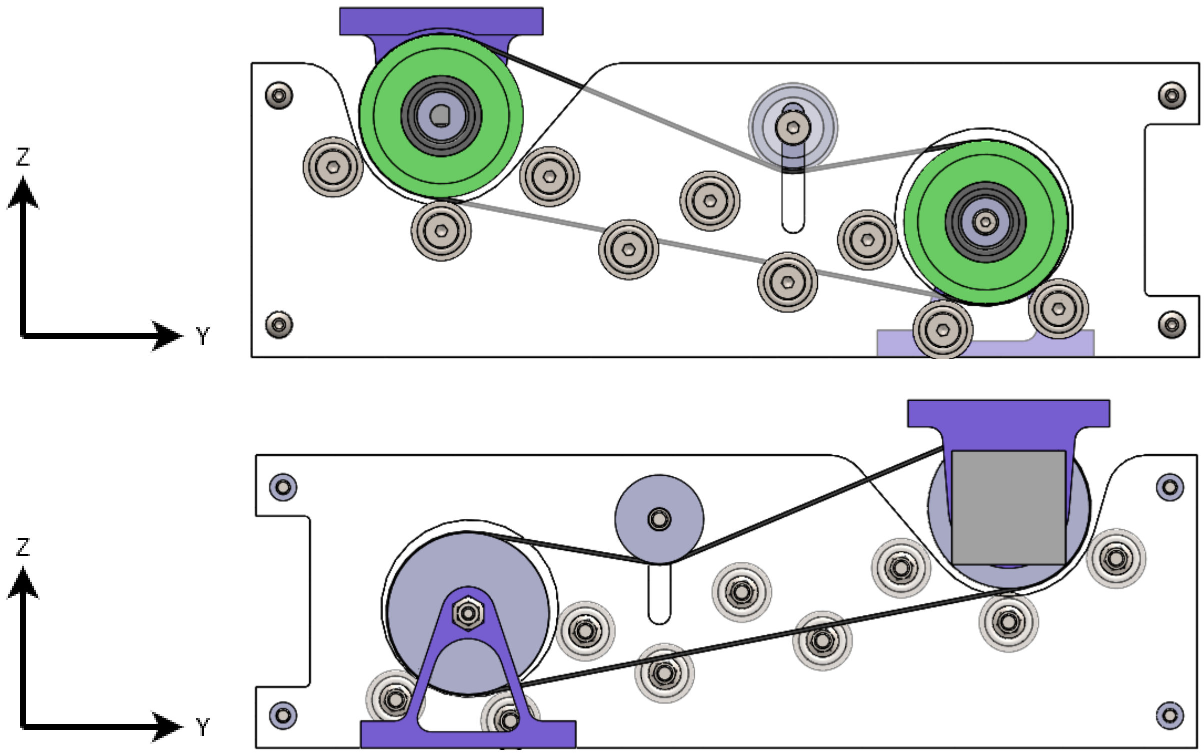


Figure xxxiv: Front (top) and rear (bottom) views of the bending track assembly

The above images show the rubber wheels and their corresponding mounts. The mounts were custom designed to fit the frame of our system and then 3D printed out of PLA material. The mount for the live roller connects to the 20/20 frame via t-slot nuts and bolts and supports the rubber wheel and stepper. The non-live roller is connected to the mount via another shoulder bolt and lock nut, and the mount is attached to the 20/20 frame in the same manner as the other mount. Also pictured in the above image is the belt tensioner. This is adjustable and ensures there is proper tension in the belt that links the two rollers so there is no slippage.

The integrated magnetic bed retention system on the Prusa, meant that the bed could not be pulled directly off the front of the printer as the original design intended. Instead, an elevator system was developed to first break the magnetic force, then feed the bed into the system.

6.3.2. Elevator System Overview

Our force measurements of the magnetic bed revealed that 16N of force needed to be applied normal to the bed to overcome the magnetic force holding it down. After the edge of the bed has been lifted, 6.76 N of force is required to move the bed horizontally. Figure 35 shows free body diagrams with the results from our force testing.

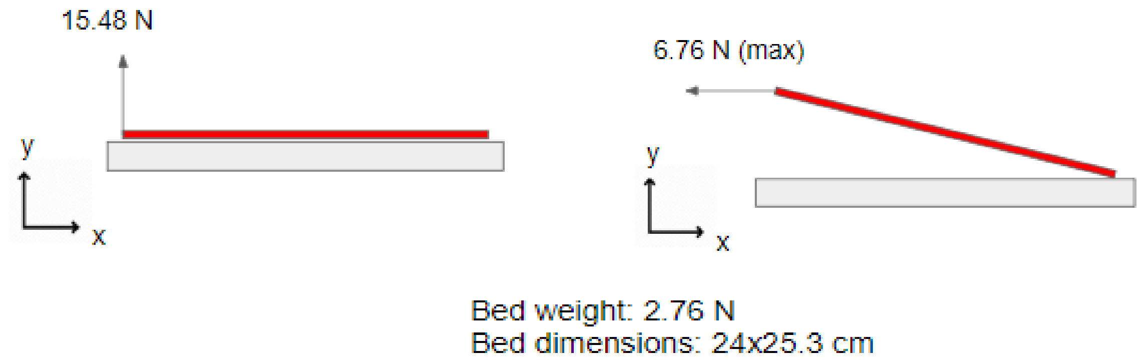


Figure xxxv: Free body diagrams of the initial bed lift

To provide this force and motion we designed an elevator system which is composed of a Nema 17 lead screw stepper motor, a linear rail, and the bed clamp assembly. Figure 36 shows the elevator system is placed on either side of the Prusa bed. The payload of the elevator is the "bed clamp assembly" which is designed to grip the bed and introduce it into the bending track after the elevator reaches the correct height.

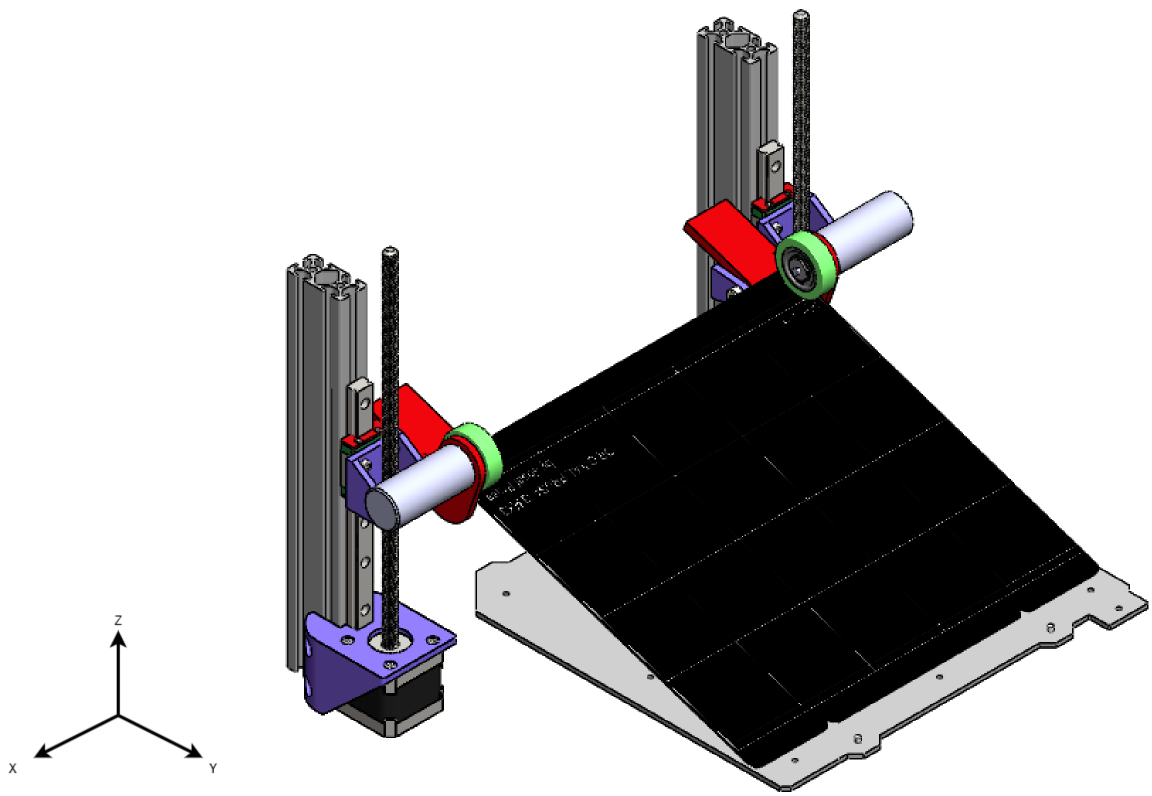


Figure xxxvi: Isometric view of the elevator system lifting a print bed

6.3.2.1. Elevator System: Lead Screw Stepper

The lead screw stepper was chosen because it is the most compact way to provide linear motion as the motor and lead screw are axially in line with each other. We had considered the use of a timing belt driven by a stepper motor to lift and lower the carriage on the rail, but the lead screw was deemed the better option since it is not back drivable and could be made more compact. The neoprene rubber material of the timing belt is also more susceptible to wear over time or damage that could cause a failure, while the stainless-steel lead screw is more robust and reliable. The maximum force output specification listed for the lead screw stepper is or 122N, meaning it will be sufficiently powerful to effortlessly lift the bed from the magnetic base. We wanted the motor to be provide more force than is required to lift the bed since the weight of printed parts was not factored into the force testing.

6.3.2.2. Lead Screw Force Analysis

During the lifting phase, the lead screws drive the slider up in the positive z direction. The slider is attached to the lift, which grabs the bed. The system must break the magnetic force of the bed, which was measured at 15.48 N. The following free body diagrams show the interaction between the forces and the parts. The system has symmetrical sides containing each of these parts, so the 15.48 N load is split equally between the two sides.

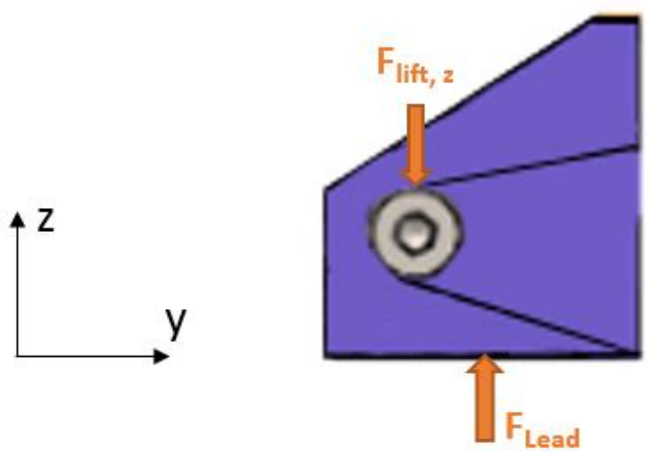


Figure xxxvii: Free body diagram of the slider

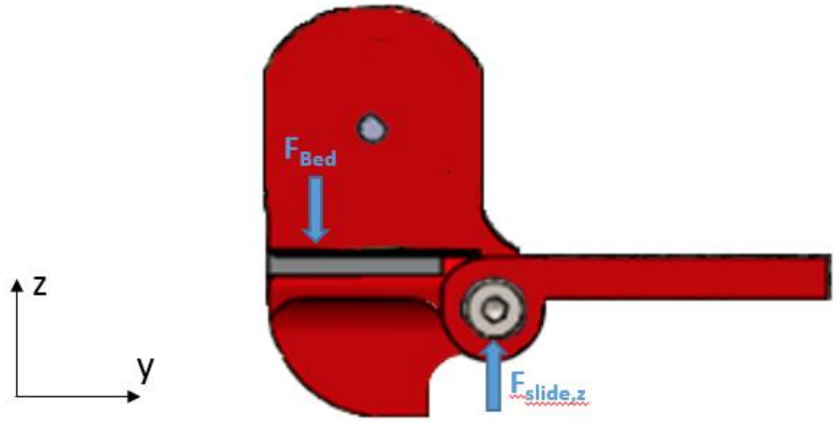


Figure xxxviii: Free body diagram of the lift

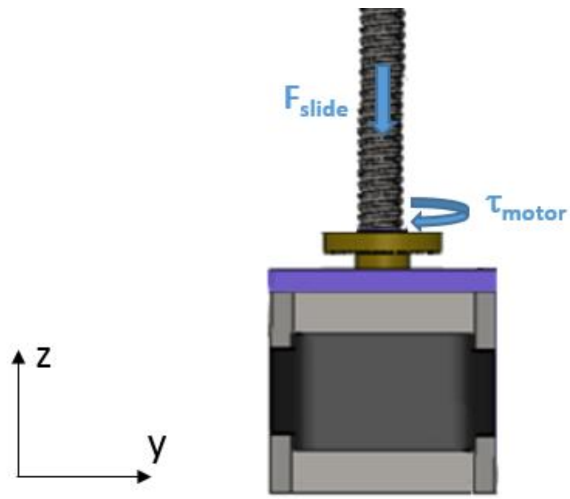


Figure xxxix: Free body diagram of the lead screw

From these free body diagrams, we get the following equations, which allows us to solve for the force exerted by the lead screw.

$$F_{Bed} = -7.74\ N$$

$$F_{Bed} = -F_{Slide}$$

$$F_{Slide} = -F_{Lift}$$

$$F_{Lift} = -F_{Lead}$$

$$F_{Lead} = 7.74\ N$$

We then solved for the torque output of the motor required to lift the bed. d is the mean diameter of the lead screw, which is 8 mm. The coefficient of friction for a steel lead screw is 0.15, taken from *Mechanical Engineering Design* (7th Edition), Shigley (2003).

$$\tau_{Motor} = \frac{F * d_m}{2} * \left(\frac{1 + \pi * \mu * d_m}{\pi * d_m - \mu * 1}\right)$$

$$\tau_{Motor} = \frac{7.74\ N * 8\ mm}{2} * \left(\frac{1 + \pi * 0.15 * 8\ mm}{\pi * 8\ mm - 0.15 * 1}\right)$$

$$\tau_{Motor} = 5.91 * 10^{-3}\ Nm$$

The lead screw motors need to output a torque of $5.91*10^{-3}$ Nm to initially lift the removable off the of the printer. We selected stepper motors with a maximum torque of $4.0*10^{-2}$ Nm, to meet the lead screw's demand.

6.3.2.3. Elevator System: Bed Clamp Assembly

During printing, the bed clamp assembly sits at the bottom of the elevator as shown in Figure 40. After a print completes, the machine homes the bed toward the bed clamp assembly. Interfacing geometry on the lift (red part) shown in Figure 42 aligns with the cutout on the magnetic base. After the printer has homed, the elevator drives the assembly vertically to a height of 52 mm Figure 41. Finally, the green rollers drive the flex bed into the bending track. The rollers are powered by a 52-rpm gear motor which produces 2Nm of torque.

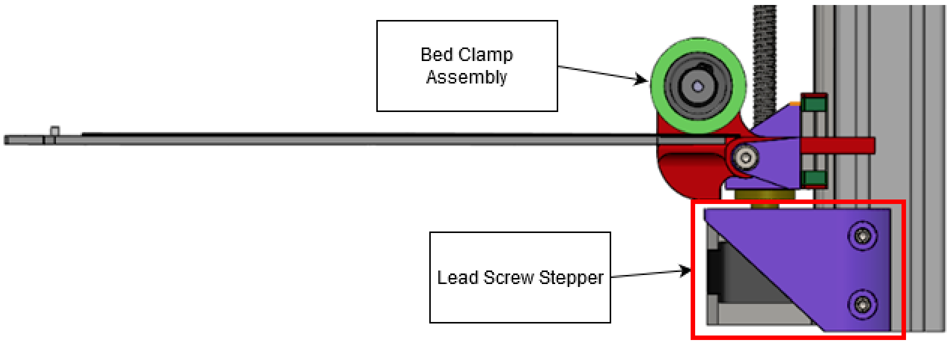


Figure xl: Elevator at home position

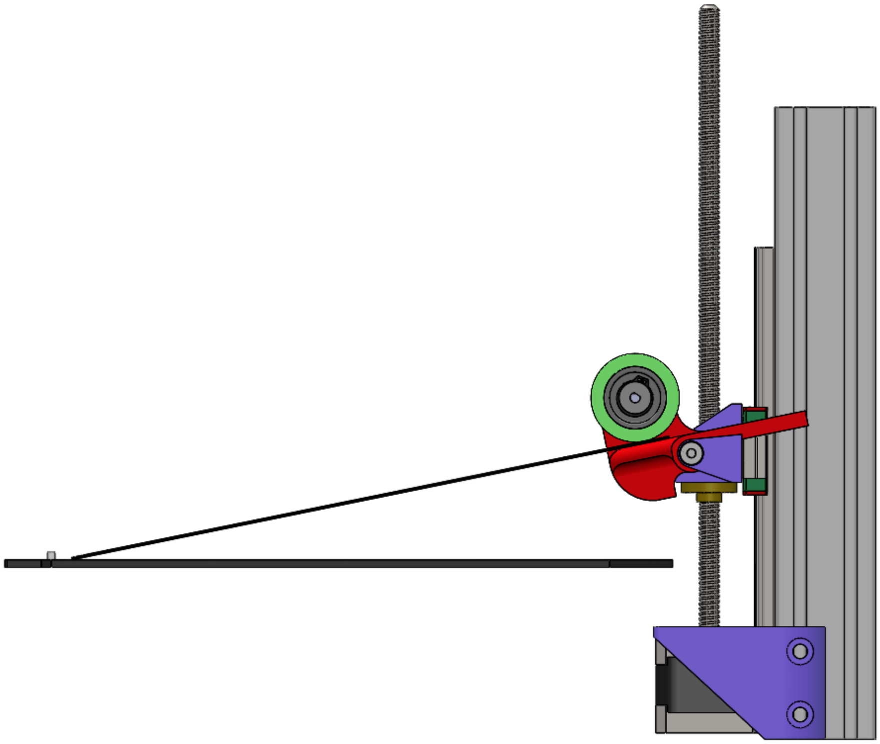


Figure xli: Bed lifted 52 mm from the magnetic base

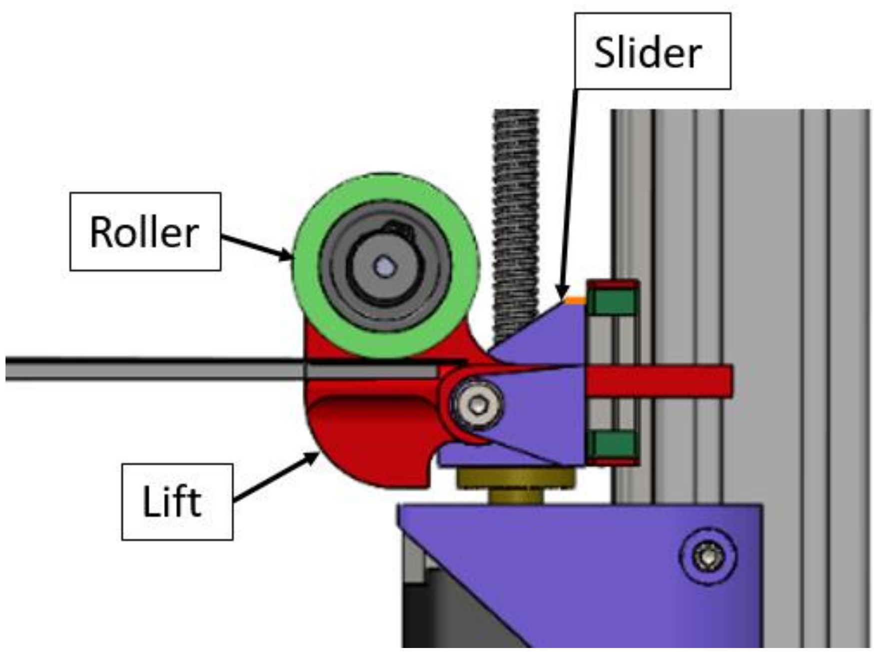


Figure xlii: Bed clamp assembly

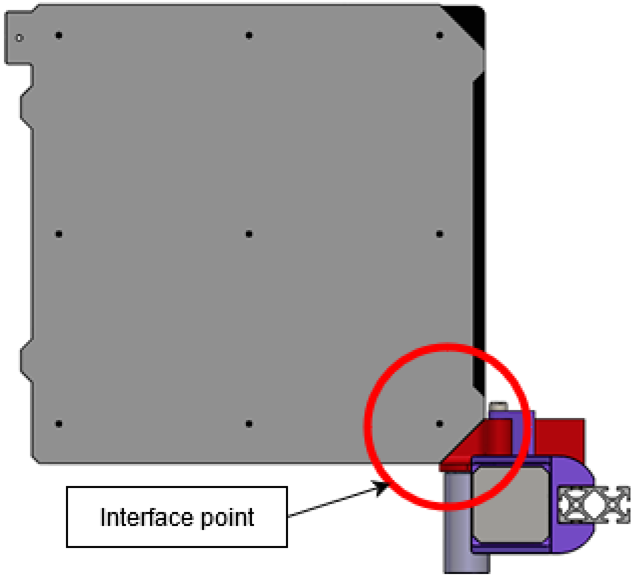


Figure xliii: Lift interface geometry (red) aligns with cutouts on the magnetic baseplate (grey)

6.3.2.4. Roller Force Analysis

The roller pulls the removable bed from the printer after the elevator system has lifted the edge of the bed 52 mm. At this elevation, the lift is rotated bout the slider 11.8°. Below are the free body diagrams of the elevator system during this stage.

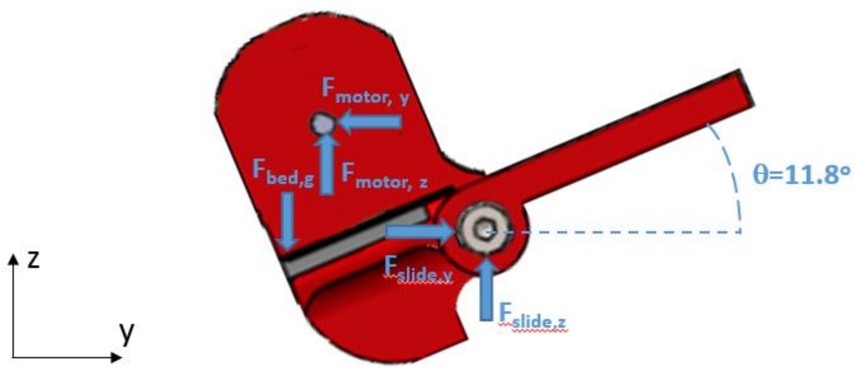


Figure xliv: Free body diagram of the lift

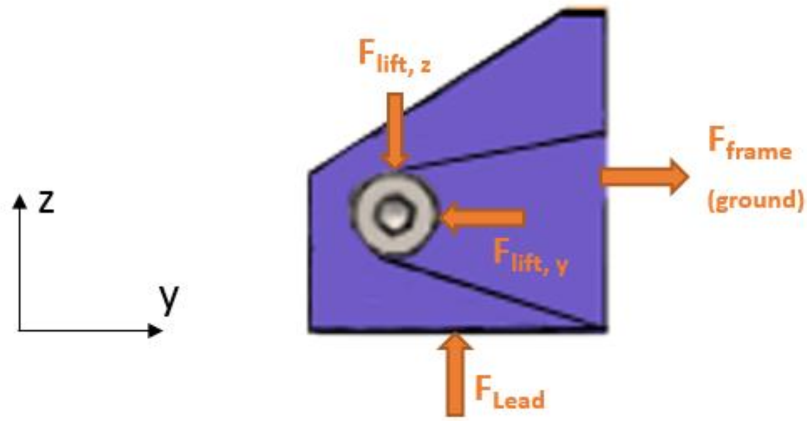


Figure xlv: Free body diagram of the slider

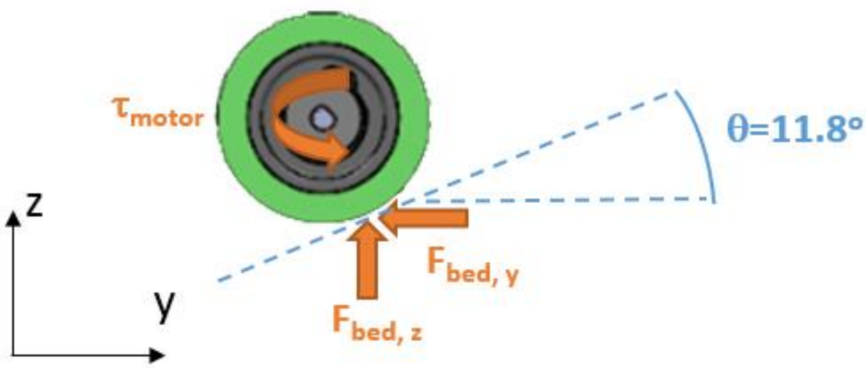


Figure xlvi: Free body diagram of the roller

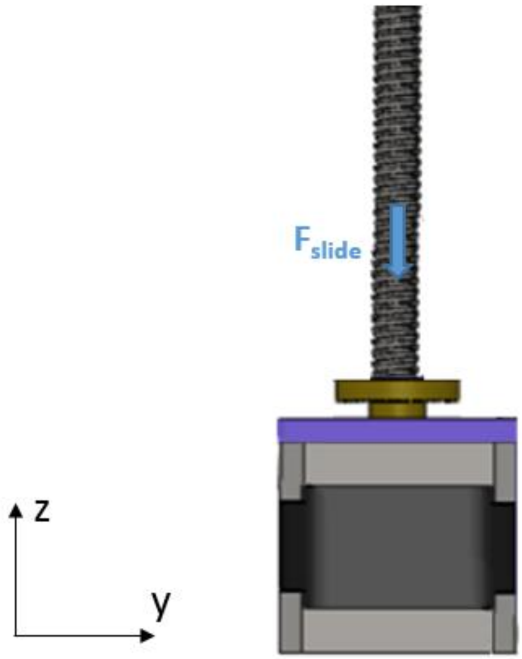


Figure xlvii: Free body diagram of the lead screw

The system has symmetrical sides containing each of these parts, so the 6.76 N force needed to pull the bed is split equally between the two sides.

$$F_{Bed,y} = \frac{6.76\ N}{2} = 3.38\ N$$

The lift is angled 11.8° during this phase, as it has been raised 52 mm in the z direction by the lead screw. Using this angle, we found the tangential force required on the roller to pull the bed.

$$F_{Roller} * \cos(\theta) = F_{Bed,x}$$

$$F_{Roller} * \cos(11.8^o) = 3.38\ N$$

$$F_{Roller} = 4.69\ N$$

The roller we selected has a radius of 1.75 cm. Using the radius of the roller and the force required by each roller to pull the bed, we solved for the torque required of the roller's motor.

$$\tau_{Motor} = F_{Roller} * r$$

$$\tau_{Motor} = 4.69\ N * 1.45\ cm$$

$$\tau_{Motor} = 6.8 * 10^{-2}\ Nm$$

The required torque from the roller's motor to pull the bed is $6.8*10^{-2}$ Nm. We selected motors with a maximum torque of 2 Nm to safely achieve this.

6.3.3. Scraper

Although most of the surface adhesion of the part on the bed may be broken after flexing, some of the print may still be attached to the bed. This is where having a passive scraper is necessary, to fully remove the part. Because the scraper is passive and does not change position on the system, the design was kept as simple as possible. The original design (pictured below left, Figure 48) was a printed part that had a beveled scraping edge and a weighted extrusion 90 degrees from the leading edge and pivoted around a rod at the corner. The parts were printed out of PLA with a 15% hatch infill. As the bed is fed through the track it would collide with the vertical hanging extrusion, pivoting the scraper and putting the leading edge in contact with the bed to scrape off the remainder of the part.

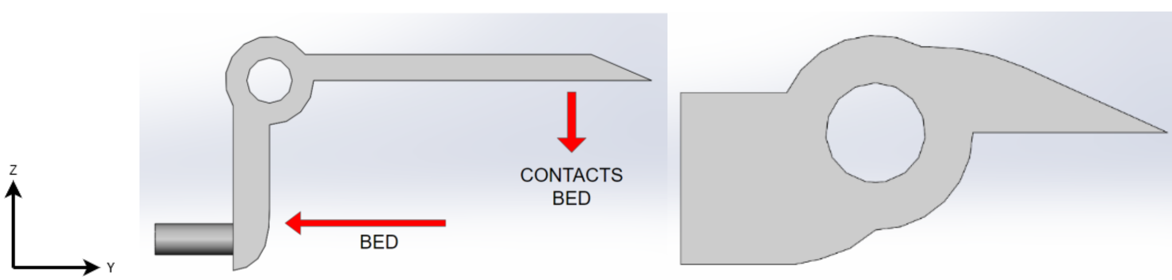


Figure xlviii: Initial (left) and final (right) scraper designs and their operation

This scraper was 15mm too large did not engage the bed early enough. It would pass over prints that were printed at the front of the print area due to its long arm. After some revisions the final scraper design (pictured above right, Figure 48) was settled upon. The length of the arm was shortened, and a chamfer was added to allow the bed to continue moving past. This scraper uses the same mechanism and works in the same way as the original but is more effective and engages the bed much sooner. Each scraper component is 42.5 mm from end to end, is 25 mm thick, and pivot around an 8 mm diameter rod. These dimensions were decided upon through multiple iterations. Each iteration the length of the leading edge was reduced as well as the back extrusion, and the ratio of lengths was altered until successful dimensions were met. 9 of these scrapers are placed side by side on the rod as to span the full width of the system.

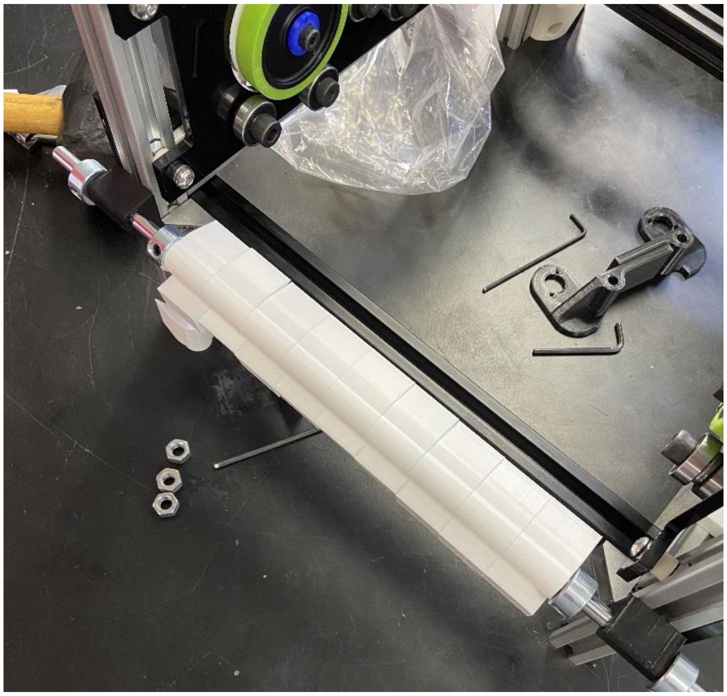


Figure xlix: Final scraper design

The above image shows how the scrapers are integrated into the full system. As can be seen, the 9 components are aligned on the rod at the rear of the system. As the bed emerges from the bearings pictured at the top of the image, it travels underneath the leading edge of the scraper but collides with the rear extrusion of the scraper, triggering the mechanism described above. The number of scrapers can be adjusted based on the width of the print bed. Doing this allows for each scraper component to move independently, accounting for any sleight height differences in the bed due to bending to ensure every point on the bed is covered by the scraper. If any component is damaged at any point, due to a failed removal or misuse of the system, a new one can be printed and added on the rod to replace the other component. Because most of the surface adhesion has already been broken, the scraper serves more to push the part off of the bed and therefore does not need a metal blade with a very sharp angle like is used to remove parts in the traditional method. An acute angle as pictured above is sufficient to get under the printed parts and remove them from the bed.

6.4. System operation Steps

1. Lift the bed a 52mm upwards by activating the elevator stepper motors to break the magnetic forces on the bed.
2. Pull the bed securely onto the lifting mechanism by activating the DC feed motors for 2 seconds.
3. Continue lifting the bed via elevator stepper motors until both limit switches located at the top of the lead screws are activated, indicating that the bed has been raised to the proper height.
4. Feed the bed into the bending track by activating the same DC motors again.

5. Pull the bed through all the way through the track by activating the horizontal stepper motors for a predetermined number of steps.
6. Feed the bed back through the track to the beginning of the elevator by reversing the direction of the horizontal stepper motors and DC motors.
7. Reverse the direction of the elevator stepper motors to lower the bed until it is a short distance from its lower limit.
8. Reactivate the DC motors to roll the bed back onto the printer, while continuing to lower the elevator until it reaches its lower limit.

6.5. FMEA

An FMEA (Failure Mode and Effects Analysis) was conducted of the mechanical system to identify possibly areas of failure and evaluate their severity. The FMEA was conducted at the component level, and each component was analyzed to identify different failure methods and respective effects. The failures were rated on severity level from 1 to 3, with 1 being the most severe and 3 being the least severe. The same scale was used for risk level, indicating how likely the failure is to happen, with 1 being the most likely and 3 being highly unlikely. Many of the failure modes were catastrophic and would hinder all operation of the system, but they are highly unlikely. They few failure modes that are very severe and likely involve the failure of printed parts if they were to fracture or crack. For the full FMEA see Appendix A.

7. Detailed Electrical Design

The mechanical design described in Chapter 6 is accompanied by a set of electronics to allow it to achieve its full functionality. These electronics consist of controllers, actuators, and sensors, that all work in tandem to produce the motion and forces required by the mechanical system. This chapter describes the various requirements of the electrical system and how components were selected in order to satisfy those requirements and develop a functional design.

7.1. Electrical System Requirements

Several electrical design requirements were derived from the overall system requirements, which guided the design of the electronics system. These requirements described how the electrical system would power, control, and monitor the various parts of the mechanical system. The top-level requirements are as follows:

1. The system must supply power to all sensors and actuators.
2. The system must control the various motors and sensors of the removal system.
3. The system must determine that a print has been completed.
4. The system must indicate that it is ready for a new print to begin.

The first two requirements were derived directly from the mechanical design, as the system could not function without the electrical system providing consistent power and control for the mechanical system. The third and fourth requirements were derived directly from the general system requirements described in Chapter 4.1. Specifically, these electrical system requirements addressed the general system requirements that the system should a) automatically remove prints from the print bed upon completion and b) handle multiple prints continuously and sequentially. The functionality to determine that prints were ready for removal and determine that print removal had been completed was deemed necessary to automatically removing multiple prints sequentially.

Based on these requirements, the electrical subsystem was divided into two subsystems: removal system control and printer control. Removal system control was responsible for powering and controlling all the actuators and sensors required to drive the mechanical system described in Chapter 6, satisfying requirements 1 and 2. The printer control system was responsible for interfacing with the 3D printer itself to ensure that it behaves as one with the removal system, satisfying requirements 3 and 4. The high level electronics system block diagram is shown below in Figure 50. The printer controller interfaces with the printer, the removal system controller controls the sensors and actuators for the removal system, and the two controllers communicate with each other. Each subsystem is described in detail in the following sections.

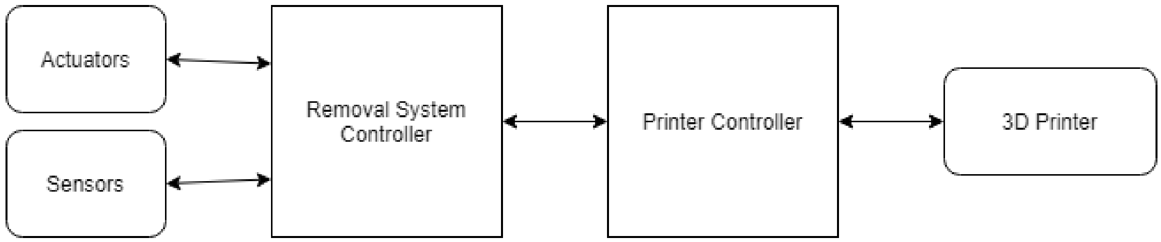


Figure l: Top-level electronics block diagram

7.2. Removal System Control

The control for the print removal system comprises actuators, sensors, and a controller that drive the mechanical print removal design. The design of this system with such components was based around the first two of the main electrical system requirements, which stated that the system must power and control all required actuators, motors, and sensors of the removal system. The low-level block diagram in Figure 51 shows the different electrical components and their connections together in the system. The microcontroller serves as the controller for the removal system. It utilizes two limit switches as its sensors. Two different motor drivers are used by the microcontroller to drive two different types of actuators: DC brushed motors and stepper motors. The power supply provides the required power to all components. The following sections described all these components in further detail in the following sections.

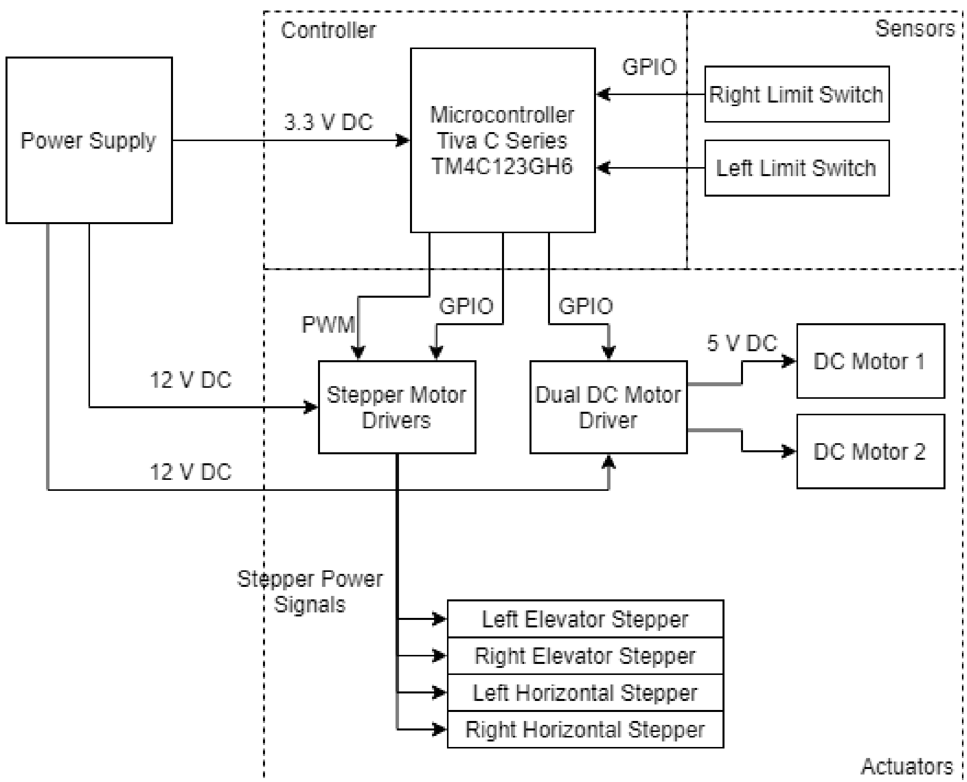


Figure li: Low-level block diagram of removal system electronics

7.2.1. Actuators

According to the mechanical design specifications, there are two types of actuators for this system: DC motors and stepper motors. There were two types of stepper motors that were used in the system. The first type of stepper motor was the Usongshine Nema 17 Geared Stepper Motor, which was used for the horizontal driving rollers on the flexing track. The second type of stepper motor was the Iverntech NEMA 17 Stepper Motor with Integrated 100 mm T8 Lead Screw, which was used for the elevator mechanism. The DC motors used were the 52 RPM Premium Planetary Gear Motor w/Encoder from ServoCity and were used on the lifting mechanism to feed the bed off the printer and into the removal system. Table 7 below lists the specifications for these motors.

Table 7: Motor specifications

Motor Name	Voltage	Stall Current	Gear Ratio	No load speed	Stall Torque
Usongshine Nema 17 Geared Stepper Motor	12 VDC	1.68 A	5.18:1	18.10 RPM	2 Nm
Iverntech NEMA 17 Stepper Motor	12 VDC	1.5 A	1:1 (No gearing)	93.75 RPM	0.40 Nm
52 RPM Premium Planetary Gear Motor w/Encoder	12 VDC	4.9 A	231.22:1	52 RPM	2.06 Nm

Each of these motors required electronics to drive them, and so two types of motor drivers were chosen to implement control of these motors. The BigEasyDriver ROB-12859 by SparkFun Electronics [19] was used to drive both sets of stepper motors. Stepper motors require alternating pulses of current on their coils to cause the motor to turn. The ROB-12859 is based around the A4988 chopper driver. The driver generates the required pulses of current when it detects a change from low voltage to high voltage on its input pin. This greatly eases the burden of motor control for the removal system controller because it allows the controller to advance the stepper motor one step in the desired direction simply by causing a rising edge on the input of the motor driver. The controller can also control the direction of rotation by writing either low voltage (0 V) or high voltage (3.3 V) to the direction pin on the driver, adding extremely simple directional control. The power supply to the stepper motor can also be toggled on and off by the controller using the enable pin on the driver. In short, using specialized drivers for these steppers motors greatly simplified the control of the motors.

The DF-MDV Dual H-Bridge Motor Driver by DFRobot [6] was used to control both DC motors. This motor driver had two independent channels, which enabled control of the two DC motors with only one driver. The DC motors were enabled by applying high voltage (3.3 V) to the input pin on the controller corresponding to the desired motor. The speed of the motors could be adjusted based on the duty cycle of the signal supplied to the input pin, with 0% (always low voltage) turning the motors off and 100% (always high voltage) turning the motors on at full speed. Therefore, the controller could change the speed of the motors by adjusting the duty cycle, a useful feature that improved control of the system. Much like the stepper motors drivers, the direction of rotation could be toggled by toggling the value on the direction pin for the corresponding motor, another useful feature.

In both cases, the motor drivers isolated the motor power supply from the microcontroller, allowing the controller to control the motors without being required to source the current and voltage itself. Instead, the required voltage and current were supplied by the power system, described in more detail in Section 7.2.4.

7.2.2. Sensors

As mentioned previously, sensors were used to improve the controller's control over the motors. The main sensors implemented were normally open limit switches capable of handling voltages up to 125 V. These switches were used to determine when the elevator motors had raised the elevator to its maximum height. One limit switch was used for each side of the elevator, and this was done to protect against the possibility of the lifters starting and ending at different heights after multiple uses of the removal system. Figure 52 shows the circuit implementation of one of the limit switches. The switch is represented by SW1. It is connected to the LaunchPad GPIO, with an input pin configured to use a weak pull-up resistor (WPU). When the switch is open, the LaunchPad input reads high voltage (V_{DD}), and when the switch is closed, the input reads low (ground).

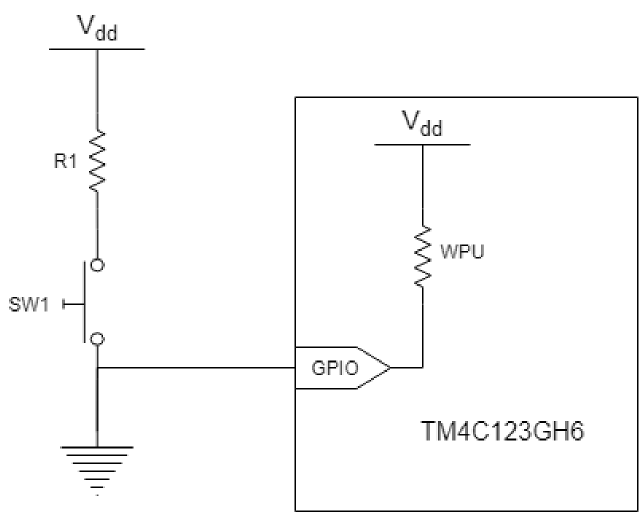


Figure lii: Limit switch circuitry

7.2.3. Controllers

A logic controller was required to control the behavior of the actuators and sensors to achieve the desired system functionality. This unit was programmed to power the various motors for finite intervals and in the necessary directions. It also utilized the sensors to determine that the motors were behaving as anticipated. This section details the selection and implementation of an appropriate controller.

7.2.3.1. Controller Selection

At first it was clear that some type of logic controller would be required to deliver the required functionality with the actuators and sensors; however, there are thousands of different types and models of logic controllers, such as microcontrollers, programmable logic controllers (PLCs), and FPGAs. We chose to use a microcontroller because they can be easily programmed through custom software to perform much more complex tasks, while PLCs and FPGAs are generally restricted to using digital logic, which is significantly more limited in terms of processing power and arithmetic capabilities. Further, microcontrollers generally come with on-board peripheral devices, such as general-purpose timers. PWM modules, and serial communication chips, which would be invaluable in system control.

Once it was clear that a microcontroller would be required, we had to choose between many different models. It was important that our microcontroller supplied enough general-purpose digital I/O (GPIO) pins and PWM pins to connect to the two limit switches, the four stepper motors, and the two DC motors. Table 8 below displays the number of GPIO pins and PWM pins required to use each component, as well as the total number of pins required.

Table 8: Output pins by type required for the microcontroller

Component	GPIO Pins per Component	PWM Pins per Component	Total GPIO Pins Required	Total PWM Pins Required
Stepper Motor (x4)	2	1	8	4
DC Motor (x2)	2	0	4	0
Limit Switch (x2)	1	0	2	0
Total	**5**	**1**	**14**	**4**

The additional peripheral devices on chip were also an important consideration when selecting a microcontroller. General purpose timers would be required to keep track of the system time, as well as control the duration of motor actuation. One dedicated timer was required to keep time, and another two timers would be required for motor timing (one for the pair of elevator system motors and one for the pair of bending track motors, as described in Chapter 6). Therefore, our microcontroller required at least 3 general-purpose timers. Additionally, a serial communication interface, such as UART, I^2C, SPI, or USB was required to communicate with the printer controller.

After considering the many different possibilities, we observed that many commercially available microcontrollers greatly exceed these requirements [7]. Therefore, we had to consider other factors as well in choosing a microcontroller. Ultimately, we selected the TM4C123GH6 from Texas Instruments because of ease of implementation. Our team had prior experience using this specific microcontroller and the TI software development toolchain, so selecting this microcontroller eliminated the time it would take to learn how best to use the microcontroller. We purchased the Tiva C Series LaunchPad EK-TM4C123GH6 [7]. This evaluation board comes with an in-circuit debugger to make testing software significantly easier. It also has headers to access all the I/O pins, making implementation in a circuit much simpler. The TM4C123GH6 supplies sufficient PWM pins, with two PWM modules and 4 independently programmable generators in each module. It also has 5 GPIO ports with 8 pins each and one GPIO port with 4 pins, all of which are capable of driving interrupts. Some functionality is multiplexed on the same package pin (for example, a pin may be capable of GPIO, PWM, analog input, etc., and the user must choose which function they wish to use in the software), so this does limit how many GPIO pins/PWM pins may be usable at the same time. However, with 44 pins available compared to the required 14, it is highly unlikely that we are limited by available pin space. Finally, there are 4 general-purpose timers, and 8 ADC channels for future sensor implementation. For clarity, the microcontroller is referred to as the “LaunchPad” in the following sections when discussing our implementation.

7.2.3.2. Controller Software

Once the LaunchPad was selected to control the print removal system, it had to be programmed to implement the functionality of the motors and limit switches. This task included developing specific functions for powering the motors in controlled durations and directions, detecting the activation of limit switches, and performing tasks in a consistent and organized manner. The custom software was written in embedded C code developed in TI’s software development kit Code Composer Studio [Texas Instruments].

At the system level, the LaunchPad software is designed as a basic state machine. The program runs a continuous loop with a switch-case construct at the top. The switch checks the value of a global variable, gSysState, which represents the state of the system, and then enters the appropriate routine based on the value of gSysState. Figure 53 shows a pseudocode representation of the basic code structure.

```
while (1) {
    switch(gSysState) {
        case STARTUP:
            <Initialize device>
            gSysState = READY;
            break;
        case READY:
            <Wait for signal to remove print>
            if (<remove print signal received>) {
                gSysState = REMOVE_PRINT;
            }
            break;
        case REMOVE_PRINT:
            <perform remove print routine>
            if (<done removing print>) {
                <send ready signal to printer controller>
                gSysState = READY;
            }
            break;
    }
}
```

Figure liii: Pseudocode representation of LaunchPad state machine

A graphical state diagram is shown in Figure 54 below, which shows each of the three main states, and the system behavior during each state. In the STARTUP state, the system initializes all required peripherals and I/O to their proper settings. It the idle state, the system waits for the “remove print” command to be received from the printer controller. Once the command has been received, it initiates the print removal procedure described above. On completion, a message is sent back to the printer controller, and the system returns to the idle state. The state machine is represented in the flow chart in Figure 54.

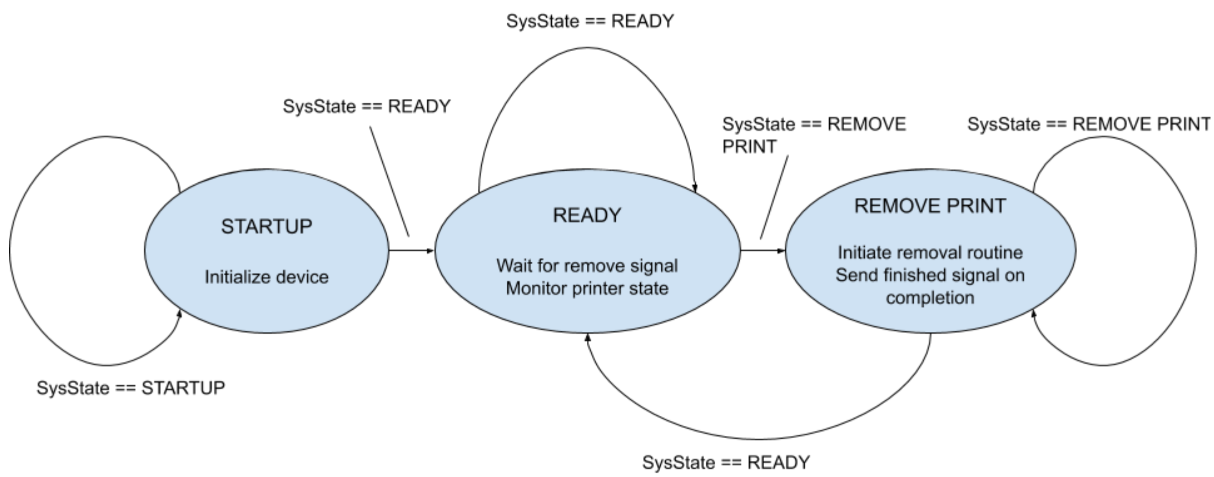


Figure liv: LaunchPad software state diagram

To control the actuators, the BigEasyDriver ROB-12859 was used to drive the stepper motors and the DF-MDV dual motor driver was used to drive the DC motors as described in Section 7.2.2. The BigEasyDriver significantly simplified motor control; the LaunchPad could advance the stepper motor one step by supplying a rising edge to the STEP input of the driver. To keep the motor running continuously, the STEP input must be repeatedly supplied rising edges. Thus, we configured PWM modules to generate square waves of constant frequency and constant duty cycle. Through experimentation, we found that 5 kHz was the fastest usable frequency before the motor would not respond to step inputs and fail to turn. Thus, the PWM modules were configured with a constant frequency of 5 kHz and duty cycle of 50%. The direction of rotation was controllable by a single digital I/O pin, with high turning the motor clockwise and low turning it counterclockwise. A general-purpose timer was configured in one-shot mode to turn the motor for a specified duration. The LaunchPad used the ENABLE pin on the motor driver to turn it on and then configured the timer to count down to zero from a starting value equal to the number of steps the motor should turn. When the timer ran out, it triggered the LaunchPad to disable the motor. Control of the DC motors was much like control of the stepper motors; however, the DC motors only needed a voltage applied to them to turn rather than a PWM signal. The LaunchPad software was able to control the rotation of the DC motors by simply writing high to the enables pin and writing either high (clockwise) or low (counterclockwise) to the corresponding direction pin.

The limit switches were implemented as described in Section 7.2.2 to improve control over the motors. Each limit switch was connected to its own pin on the LaunchPad. They were each connected to their own GPIO pin, both of which were configured to generate interrupts on falling edges. The pins were configured as inputs with pull-up resistors, so that they read high voltage when open, and transitioned to ground when closed. This generated the falling edge necessary to trigger the interrupt. An interrupt service routine, which was called whenever the interrupt was triggered, disabled the motor corresponding to the limit switch that was pressed.

Each motor-limit switch pair was operated independently, so that a triggering of one switch did not turn off the other motor. This behavior allowed both side of the elevator system to rise all the way to the top, ensuring that the bed was in the correct position to be introduced to the bending track.

7.2.4. Power

In order to properly use each of these components, a few different types of power converters were needed. First, the main power supplied to both the stepper motor drivers and the DC motor drivers is 12 VDC. This was done using a desktop variable power supply that was available in the lab. This piece of equipment could be replaced with a 12 VDC power supply given that it can output at least 3 A, which was the upper limit on the desktop power supply. In order to prevent overdrawing current from the power supply during operation, the stepper driver currents were limited to 0.3 A on each individual stepper motor by adjusting the current limit potentiometer on the driver board. In order to supply the Raspberry Pi with power, a dedicated 5 VDC Micro USB power supply was used. For the microcontroller, a 3.3 VDC USB power supply was used.

7.3. Printer Control

While the LaunchPad-based system provided sufficient control over the print removal system, another controller was required to interface with the 3D printer. This control system was responsible for satisfying the third and fourth electrical system requirements, stating the system must detect that a print has been completed and indicate that it is ready to begin a new print. These are complex challenges, which require background knowledge of 3D printer firmware implementations that our team did not possess. Thus, A design for printer control was developed with both effectiveness and ease of implementation in mind.

7.3.1. Printer Controller Selection

Much like we selected the LaunchPad for the removal system, it was necessary to select an appropriate device to control the 3D printer. 3D printers behave much like traditional printers, in that they are designed to be connected to a user's computer, which is running some software that controls it. Such an application is far too complex to put on a simple embedded device like a microcontroller due to restraints on code space and processing power. The next step upwards in terms of capability is Raspberry Pi, which offers a unique hybrid between embedded device and personal computer [17]. Raspberry Pi runs a simplified operating system, which allows the user to install software and applications (such as what we may need to interface with the printer), while also providing digital I/O to interface with the LaunchPad. By using a Raspberry Pi, we could significantly ease the burden on the team by installing software to interface with the 3D printer so that we did not have to build it from the ground up. We selected the Raspberry Pi 3 to control the printer for its combined capabilities in high-level software (for interfacing with the printer) and low-level electronics (to communicate with the removal system).

7.3.2. Printer Controller Software

Next, it was necessary to select the software to run on the Raspberry Pi that interfaced with the 3D printer. Some examples of 3D printer management software are mentioned in Section 1.3.4. While there are many applications designed to provide a user interface to a 3D printer, we ultimately selected OctoPrint. OctoPrint provided two significant advantages, the first being it is an open-source software so we could modify it to suit our needs, and the second being there exists a distribution of OctoPrint specifically for Raspberry Pi called OctoPi [12]. This custom Raspbian (a Linux-based operating system for Raspberry Pi) with OctoPrint installed significantly simplified the task of interfacing with the printer.

OctoPrint serves two primary functions. First, it serves as the user interface for the system. The user can connect to OctoPrint by connecting their computer to the Raspberry Pi via Ethernet, and then navigating to “https://octopi.local” in a web browser. There, they find a control panel from which they may upload print jobs, monitor the status of the printer, and issue manual commands to the printer through a terminal. Figure 55 below shows a screenshot of the OctoPrint user interface main page. The printer status is displayed in the left panel, while diagnostics (in this case a graph of the extruder and bed temperature) are shown in the main panel.

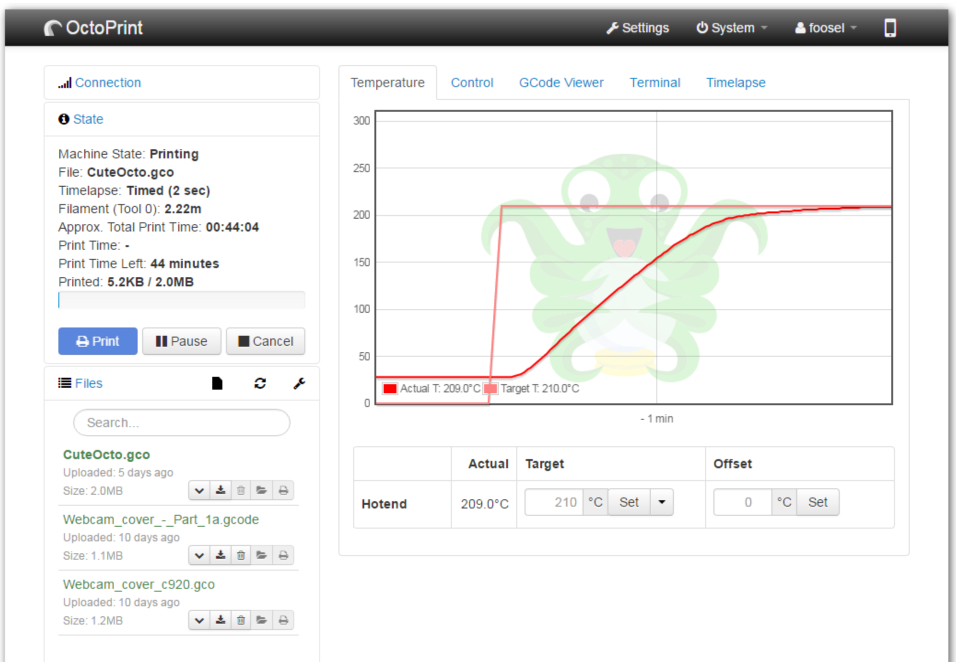


Figure lv: OctoPrint main page

7.3.3. Communication Between OctoPrint and the LaunchPad

The main benefit of OctoPrint was that it was an open-source project, allowing us to modify it slightly to meet our design requirements. Because it monitors the status of jobs on the 3D printer, OctoPrint by nature is aware of when a print has completed. It was possible to take advantage of this functionality to alert the print removal system that a print was ready to be removed.

A means of communication between the Raspberry Pi and LaunchPad was required to coordinate print removal with the operation of the 3D printer. A simple two-wire serial connection was used to send messages back and forth between the two devices. The Raspberry Pi’s Linux terminal (dev/tty0) is exposed as a UART serial communication bus on two of its

GPIO pins. This UART module was connected to one of the built-in UART peripherals on the LaunchPad, allowing the Raspberry Pi to send and receive data to and from the LaunchPad at any time. A diagram of this connection is shown below in Figure 56.

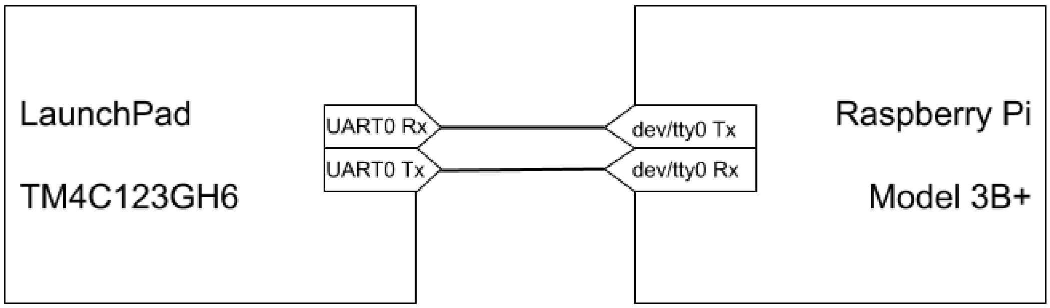


Figure lvi: UART communication lines between the LaunchPad and Raspberry Pi

In software, the OctoPrint code was modified to send a message to the LaunchPad whenever a print was finished. An event listener function was added to listen for OctoPrint's "End of Print" event; this kind of function is called whenever a print is completed. This function sent a message out over its serial terminal, which was received by the LaunchPad. On the LaunchPad side, the interrupt for the UART module was enabled, which triggered whenever there was data available to be read (i.e. a message had been received). This interrupt triggered a routine that read all the data on the bus, parsed the message, and updated the system state variable gSysState to move the printer from the READY state to the REMOVE_PRINT state to initiate print removal. This satisfied the design requirement that the system should detect when a print is ready to be removed.

Once the print removal process was initiated, OctoPrint waited for a return signal from the LaunchPad that the system was ready to accept a new print. Meanwhile, the LaunchPad incorporated code into the end of its print removal process that sent a "done" message back to OctoPrint, signifying that the removal was finished. Another event listener was added, this time to listen to a new event that was added, the "End of Removal" event. This event was triggered whenever OctoPrint received the LaunchPad's "done" message. The new event listener was configured to send a "resume operation" G-code command to the 3D printer to begin its next print, satisfying the requirement that the system detect when the print has been removed. The process of beginning the next print once the previous one has been removed is described in more detail in the following section.

7.3.4. Queueing of Multiple Prints

Although the system met the four main objectives of the electrical system, it still did not meet the overall design goal that it must be able to automatically handle multiple print jobs sequentially. It was necessary to add a print queue feature into the software, allowing it to process several print jobs automatically in first-in, first-out order.

The queueing feature was achieved by leveraging OctoPrint's vast library of plugins. These are open-source add-ons that can be installed in OctoPrint, adding new features on top of the base functionality. The OctoPrintPrintQueue plugin was used in our system to add an additional page to the OctoPrint browser page. In this tab, the user could upload several G-code files at once to be printed automatically in succession. The OctoPrintPrintQueue readme file specifies that the plugin is intended to be used with some method of automatically clearing the print bed, which is what our system is designed to do.

The queueing plugin allows the user to provide a G-code script to be run at the end of each print so that print removal may occur. For our print removal system to begin, we created a short G-code script that first moved the extruder up to its maximum height and all the way to the near side of the printer. The print surface was then moved all the way forward to its maximum y-limit so that it could be fed into the elevator system. Finally, the script issued a pause G-code command to the printer, which stopped all printer operation until a resume G-code command was issued. The event listener for the "End of Removal" event in OctoPrint (as described in the previous section) was modified to issue the resume command, so that the printer would being the next print in the OctoPrint queue. This process was key in satisfying the design requirement that the system must handle multiple prints automatically and sequentially. The end of print G-code script that was provided to OctoPrintPrintQueue is found in Appendix D.

7.4. Full System Layout

The subsystems described above combine to form the full electrical system layout, displayed in the block diagram of Figure 57. This diagram shows the detailed electrical connections of the print removal system, and its connection to the printer through the communication link between the LaunchPad and Raspberry Pi. After the full design of both the electrical and mechanical systems, the project proceeded into full assembly and testing. Detailed schematics for all circuit elements are provided in Appendix B.

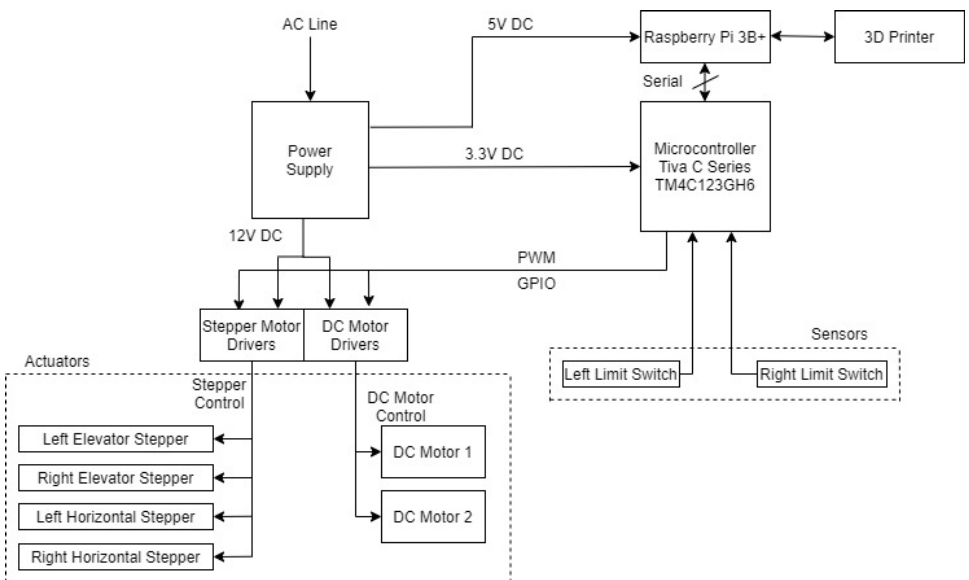


Figure lvii: Full system detail block diagram

8. Fabrication & Testing

Once this design was approved, we started constructing a prototype. The frame was constructed using 20/20 metric extrusion, cut to length and connected using t-nuts and 90° brackets. This production method for the frame was chosen because of its simplicity and ease of modification. Frame components had to be cut to length, but after that was done, they could be assembled and moved around without any other permanent modification to frame components. As our design changed slightly during the prototyping phase, we could use the same frame components and slightly adjust their placement since we were not using permanent fasteners. For the sides of the bending track. 0.25 in acrylic was laser cut to the design, and skateboard bearings were attached to it using shoulder bolts and washers. Acrylic was chosen because it could be laser cut easily and quickly, and if modifications were made to the bearing track another panel could be laser cut and attached to the system within the hour. Other materials like polycarbonate were considered, but none were as readily available or easily manufacturable as acrylic. Skateboard bearings were used in the bending track due to their easy availability and standard sizing, making them inexpensive components. The washers were placed between the acrylic and the bearings to ensure only the inner races of the bearings were restrained, and the outer races could spin freely. Lock nuts were used to secure the shoulder bolts so that they would not become loose during normal operation of the system. Many of our parts, such as various motor mounts, were custom designs and thus were 3D printed out of standard PLA. As parts were assembled, we conducted a series of tests to ensure that our system would work. By isolating and testing the subsystems, we are able to isolate and solve problems.

8.1. Removing Prints Through Manual Operation

The first system our team built was the bending track. Before continuing with the automation process, we wanted to ensure the track could effectively remove prints from the bed. We printed test parts and manually pushed the bed through the bend track to demonstrate its ability to remove the print. We found that the track can break the adhesion between the print and the removable bed. With the prints free from the bed, they slid off without any resistance. However, breaking the adhesion was not enough to remove the print from the bed completely; the prints often stayed on top of the bed unless pushed off manually after being run through the track. This led us to the conclusion that the scraper at the end of the track is necessary to push the removed part off the bed and through the bottom of the system.

Figure lviii: Manual testing of the bending track

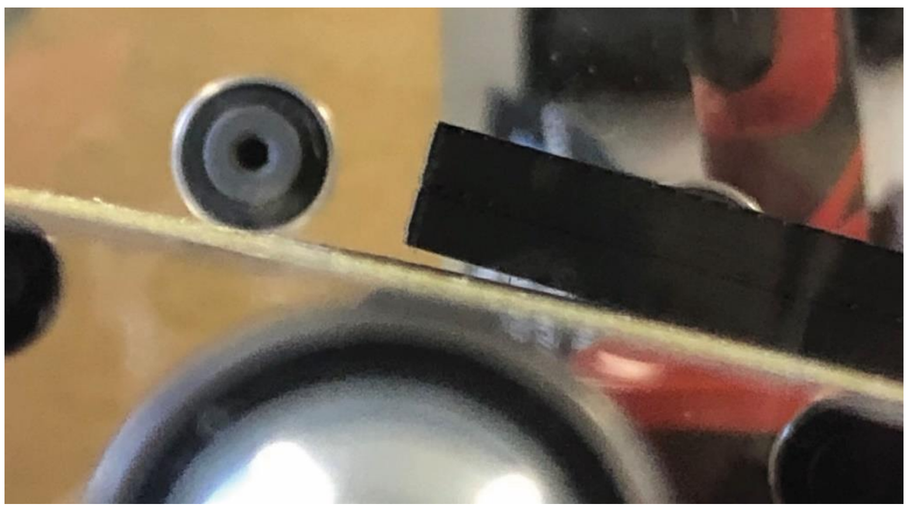


Figure lix: Close-up of the adhesion between print and bed being broken by track

8.2. Automated Use

Upon completion and testing of the bending track and motor controls, the full system was assembled and attached to the printer. Initial testing sought to demonstrate the system running through the print removal process described in section 6.2.1. There were several technical issues with the circuitry on first pass. The first issue was caused by a power shortage to the stepper motors; as a result, the stepper motors would make hissing and whining noises, and would not turn. This problem was solved by turning down the current limits on the motor drivers to ~0.3A each, resulting in about 1.2A to the stepper motors from the DC power supply. The second issue was a bug related to the digital I/O controlling the stepper motors of the elevator system. On raising the elevator, occasionally only one motor would turn. The problem was eventually traced to a faulty I/O pin, as the voltage level would not correctly be pulled "low" to enable the motor. Altering the software, to use a different digital I/O pin, fixed the issue, so we believe there must have been a problem with the pin. After troubleshooting these errors our team ran the full print removal cycle without a print on the bed. The system successfully performed all the required operations automatically without need for a human operator. The next test we had planned was to test the removal of several prints successfully. Photographs of the active removal process are shown below.

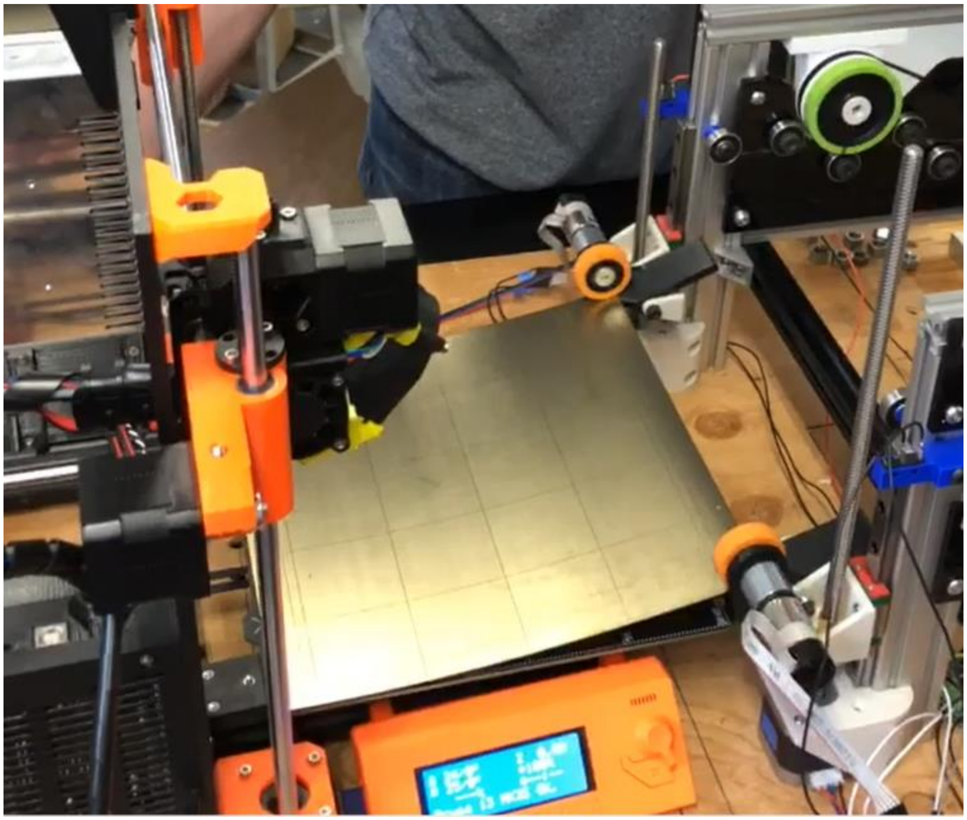


Figure lx: The elevator system automatically lifting the removable bed from the printer

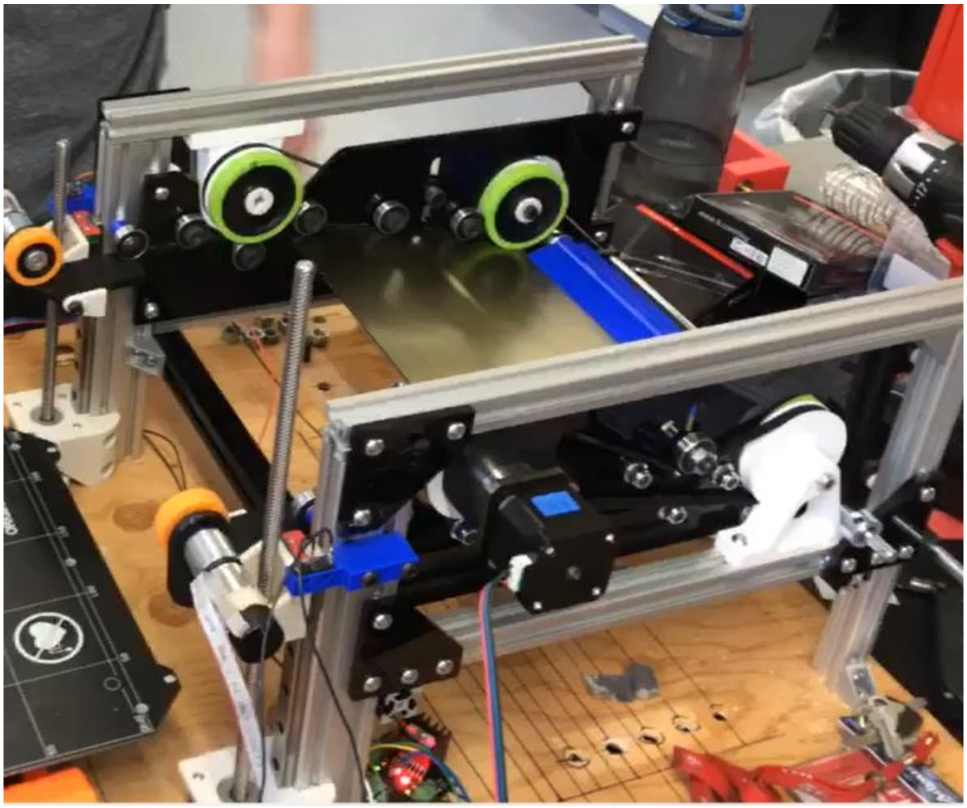


Figure lxi: The rubber rollers pus the bed through the bending track, engaging the scrapers

9. Discussion & Results

WPI restricted student access to campus spaces at the beginning of D-term due to the Covid-19 pandemic, which prevented our team from accessing the project and the lab space. At the end of C-term we created a functioning removal system, but there was still room for improvement on some of the subsystems. We have made minor changes to the CAD model that could not be implemented due to restricted lab access. We have made suggestions for software implementation as well as testing processes to be conducted on the system. Finally, there are feature recommendations which will improve the functionality of the system.

Another change is an improved layout of the acrylic walls of the bending track to reduce belt friction and better protect the motors, belt and bearings. These changes require access to the MQP lab as well as Washburn Shops and therefore were halted by COVID-19.

9.1. Completing the Updated System

We recommend updating the prototype to the newest design iteration. This includes the updated frame, bending track assembly, and mounting system. Figure 62 shows the full CAD model of the improved design.

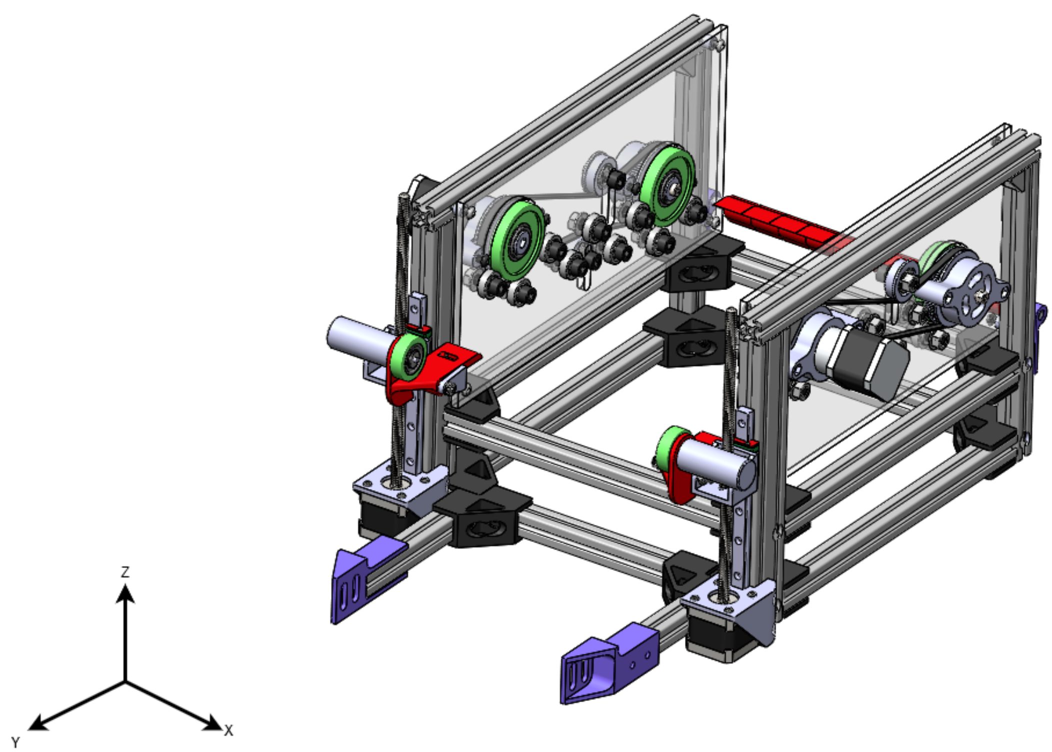


Figure lxii: CAD model of the design with the recommended improvements

9.1.1. Updated Frame design

The most significant change is an improved frame. The current system requires precise levelling and alignment to function properly. The updated frame has a second set of supports and includes bores in the extrusion to ensure the bending track is leveled and ready to operate upon assembly. Figure 63 shows the redesigned frame component which is used twice to provide rigidity to the frame.

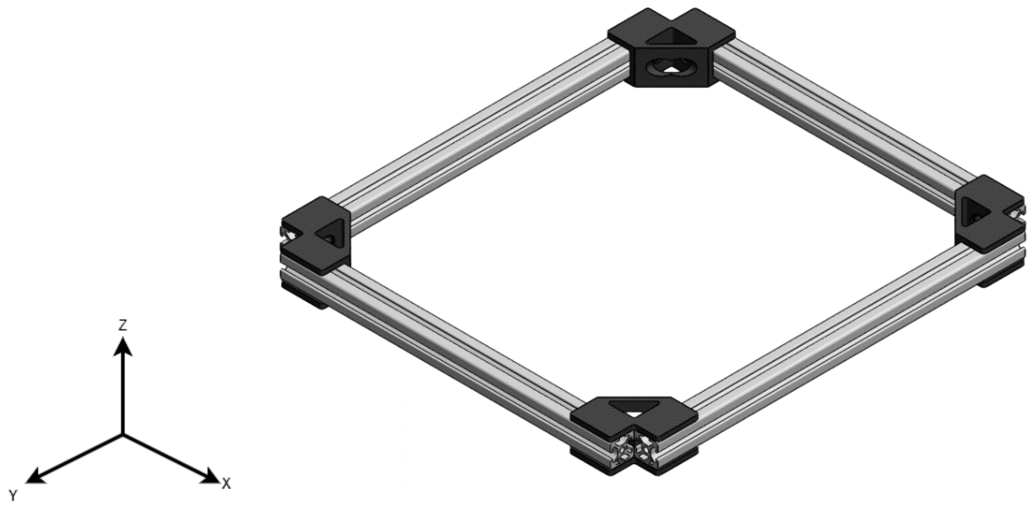


Figure lxiii: Updated frame design

9.1.2. Updated Mounting System

The updated mounting system replaced the temporary wooden mounting device we used for preliminary testing. The new mount secures the removal to the printer for a reliable connection between the removable bed and the elevator system. A common material for 3D printers to be made of is aluminum extrusion. This is due to its low cost and ease of construction. Since our device also utilizes aluminum extrusion, our team can utilize extrusion hardware for mounting our device to 3D printers. There is a type of hardware commonly called drop in hardware, in which locking T-nuts can be “dropped” directly into the slot of the extrusion without having to slide it in from the end. This kind of hardware would allow us to mount directly to the exposed slots of the printer without the need for its disassembly. The only addition to our system to mount it to a printer would be two arms that would bridge the gap between the device and the printer. Figure 64 shows a possible design for these arms. This design is made to attach the system to the Prusa i3 MK3 3d printer. The brackets feature vertical slots to fasten to the extrusion of the printer so that precise alignment between the printer and removal system is not needed.

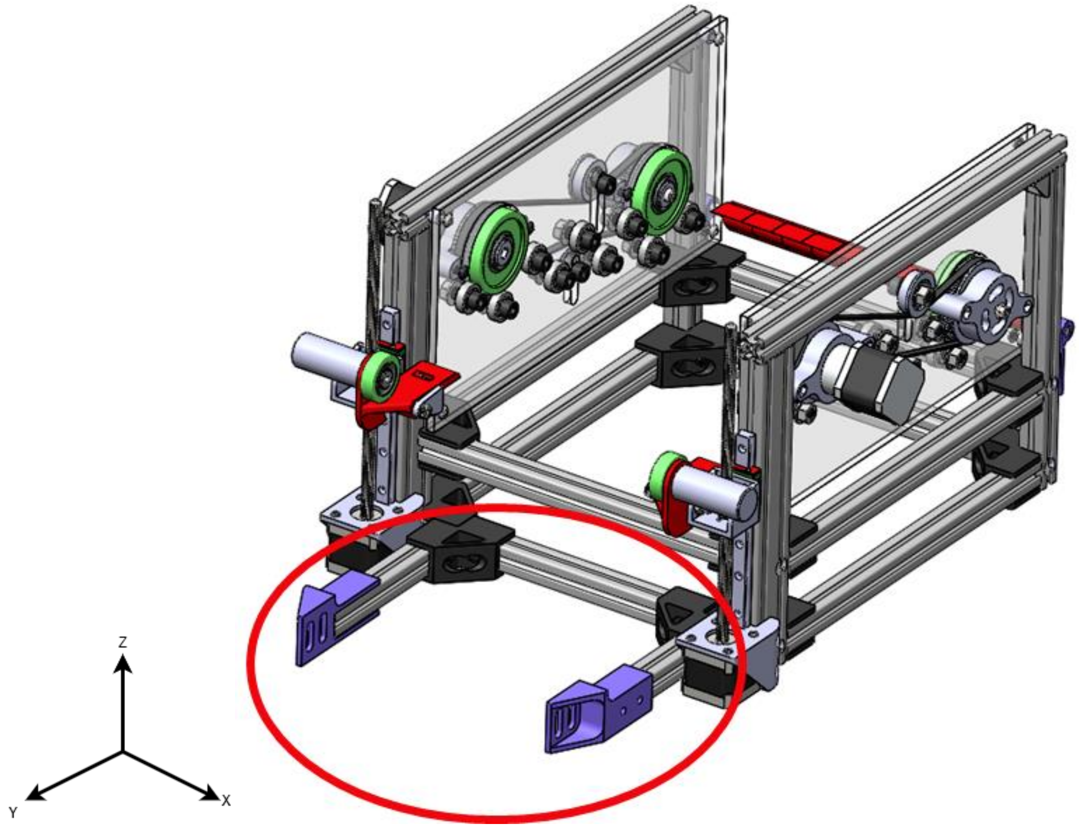


Figure lxiv: Design for attachment arms

It is very important that these arms sit lower than the elevator system at its lowest position to ensure proper functionality. There are some printers on the market that do not utilize aluminum extrusion for its frame. In that case our system will have to be mounted to the table surface using right angle brackets. These brackets are low cost and are often used for supporting right angles created by extrusion. By securing one end in a slot, the other hole is available to drive screws, bolts or other fastening hardware into the table surface to secure it.

9.1.3. Updated Bending Track

The design of the acrylic plate of the bending track was updated to behave as a base component. In the previous design the motor and wheel mounts were mounted separately from the plate on the aluminum extrusion. This made assembly difficult as fine adjustments needed to be made to these parts in order to properly locate them. Figure 65 shows front and back of the updated design. Everything related to the bending track mounts into the same base component. This simplifies the design by reducing the amount of fine adjustment needed to assemble the bending track.

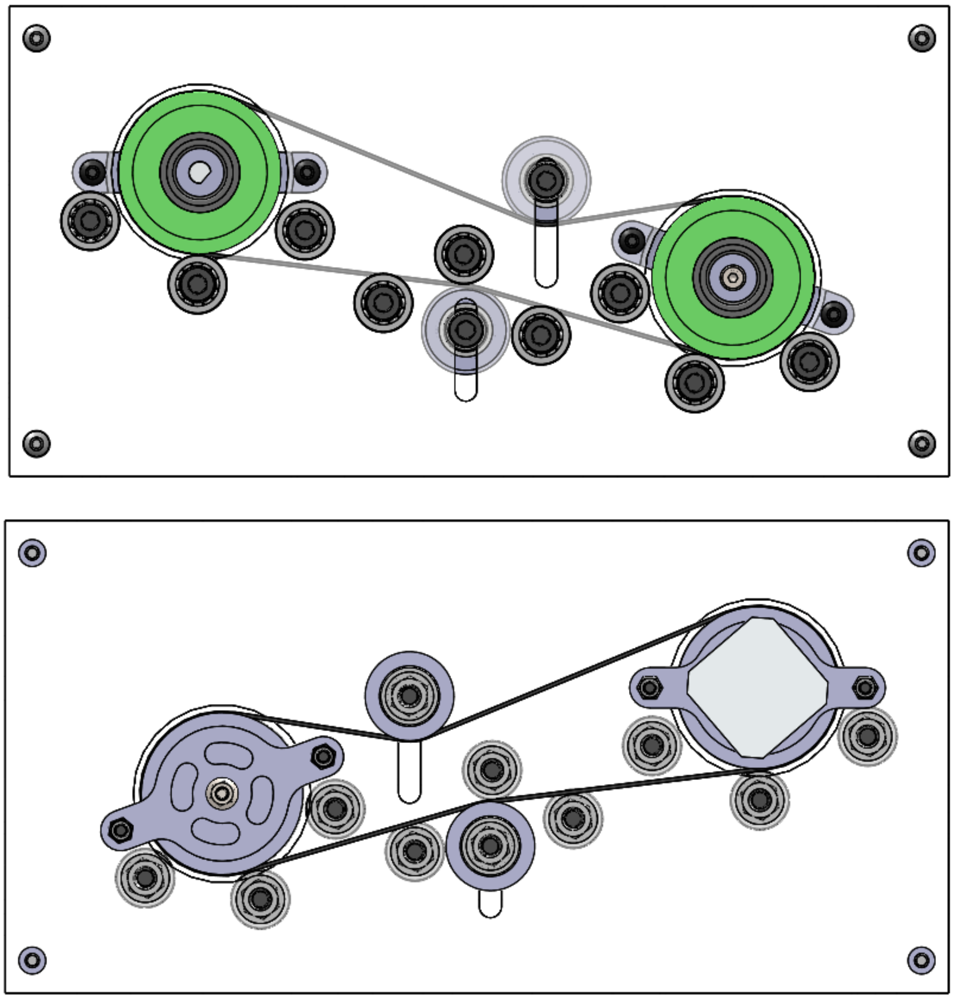


Figure lxv: Front (top) and rear (bottom) views of the updated bending track assembly

9.1.4 Extra Sensor Integration

Other places the limit switches were planned on being used was in the lifters themselves to determine how far the DC motors had to pull the bed onto the lifters after the initial lift, as well as how far the lifters had to roll the bed out for replacing the bed. On top of this, a current sensor was also planned on being implemented into the different actuators in order to determine the power consumed by the different actuators throughout the process.

9.2. Full Software Integration

Although much of the software exists already, the system remains in a highly developmental build state. There are a few steps left that must be taken to fully implement the software. The first and most significant hurdle is that the custom OctoPrint build must be installed on the Raspberry Pi. To do this, the OctoPi must be configured to build with our modified OctoPrint code, rather than the default OctoPrint version. The CustomPiOS library is used to build the OctoPi disk image for Raspberry Pi, so the CustomPiOS scripts must be modified to use the custom OctoPrint build. Instructions for doing this have been included in the FlexTrak User's Guide, located in our project Google Drive.

Once the custom OctoPrint version is installed on the Raspberry Pi, the UART connection between the Raspberry Pi and LaunchPad must be tested. First the basic functionality must be tested; the Raspberry Pi should send a message out over UART on print completion, and then command the 3D printer to resume printing. The LaunchPad should enter its UART interrupt on receiving data on the UART Rx line. In this interrupt, it should parse the message, and update a global variable if the message is to begin print removal, causing the state machine to enter the "Remove Print" state. Once the print removal is done, it should send another message back to the Raspberry Pi. This communication system is simple in its current state and should be expanded upon in the future. Perhaps most importantly, OctoPrint should incorporate a method to periodically poll the LaunchPad for its status; this would prevent the system from printing while the LaunchPad is in a faulty state, and would also allow OctoPrint to notify users that there is an error.

The LaunchPad state machine also needs to be completed. Currently, there are 3 implemented states: Startup, Idle, and Remove Print. However, there are other states which can be added. The "Calibration" state should be implemented in order to calibrate the system sensors after startup. Additionally, a "Sleep" state can be implemented to put the LaunchPad into sleep mode after being idle for a long period of time. In this case, the control flow must be modified to have the LaunchPad exit sleep mode when a "remove print" command is received. Finally, an "Error" state would add robustness to the system if something in the electronics were to fail. The LaunchPad could alert the Raspberry Pi of the error, where it could be attended to by the user.

9.3. Further Testing

We prepared a series of tests that would show the reliability and limitations of our system. The first set of tests involve printing many parts in a range of sizes, shapes, and infills. These tests are to identify any print parameters that could cause a removal failure. The second set of tests provide information on the automation of our system.

9.3.1. Removal Capability Testing

These tests will determine if the system met our objective to automatically remove prints from the printer. The test is a success if the part is removed from the bed, and the bed is completely clear of filament. The test is a failure if the bed does not return to the printer or returns with filament stuck to it. For this test, a printer failure during the print is not considered a removal system failure.

The print size test shows if surface area or height of a print affects the system's ability to remove it. A rectangular block print with a standard 20% infill is queued. Only one part at a time is set to be printed and removed. The bed temperature and extruder temperature are to be kept constant between tests. The same spool, or a similar spool, of PLA is to be used for all tests. This process will be repeated with a 1 in x 1 in square, a 4 in x 4 in square, and an 8 in x8 in square, each with a height of 1 in. These sizes were selected because 1 in x 1 in is to test very small prints, 4 in x 4 in is larger, typical print size, and 8 in x 8 in is the maximum print dimensions of the Prusa Mk3 printer. We want to record multiple data points for each size, but the prints take several hours and use up filament. To achieve these data without being wasteful, we will test each size three times. With each print, we will record whether the system was able to remove the print and any issues the system encountered. To test the effect of height, the same test will be run again, but with print heights of 0.5 in and 2 in. Taller prints are typically more rigid, and often have some warping at the base of the print where thin prints are more flexible. These variables could impact the bending track's ability to remove the print. To test the effect of infill, the same test will be run again, but the infills of 10% and 40%. Infill affects the rigidity of the print and may influence the bending track's ability to remove the print.

The second removal test is to determine if special case prints affect the system's ability to remove the print. In this test we will queue a print with a brim, a print with a skirt, and a print with multiple contact points with the bed. Each print will have 20% infill. Only one part at a time is set to be printed and removed. The bed temperature and extruder temperature are to be kept constant between tests. The same spool, or a similar spool, of PLA is to be used for all tests. Each print will be tested three times to record multiple data points.

9.3.2. Automation Testing

The next test looks at the reliability of the automation. We want to see how our system handles a large queue of prints. As stated in the objectives, our milestone goals are printing without failure for 12 hours, 48 hours, and a for a full spool of filament. This would allow the system to operate during times where the printer is not supervised at all by a human operator. With this goal in mind, a test time of 12 hours is appropriate. The first step is to queue six parts that take 2 hours to print. This is to ensure several parts are printed that are each of substantial size. The system is then left to run until all the prints are completed or a failure occurs. The system will be video recorded, so if a failure does occur, we will have evidence as to what went wrong. After this entire process is repeated 3 times, there are two outcomes. The system had a failure during one or more of the tests, or the system completed all 3 tests without an issue. If a failure occurred, the problem(s) shall be located, and potential solutions will be implemented. After the potential solution has been implemented, more tests will be administered. In the case that the system did not fail during the 3 tests, the test duration will be increased to 24 hours and run again. This process of increasing test time can be repeated until we find a consistent rating for how long the system can operate without failure.

9.4. Recommended Future Improvements

To continue our project, our team has proposed three areas that should be of focus to a future team: a clamping mechanism, dedicated software, and a sorting system.

9.4.1. Clamping Mechanism

A preliminary clamping system was designed for use on printers that did not have magnetic beds. The design utilized a 20kg servo, which would transfer force through a linkage to provide 376N of clamping force to the bed. This system worked well when it was not attached to the printer, but we quickly realized its limitations when we attached it. The first limitation was that the servo was too large to fit under the bed and clear the z-axis support rails. If we were set on using these servos, then a custom bed mount plate would have to be fabricated to connect the bed to the y-axis. The servo arm would protrude 3-5mm from the bed, which was just long enough to hit the z-axis support and limit the y-motion of the printer. The second limitation was the linkage. The prototype linkage was 3D printed to allow for quick changes. The force transmitted through the linkage would severely bend and bow the links. Further testing and calculations must be done to design a system that will not bow when the clamping system is engaged. To conclude, we have designed a linkage that will transmit the force from a servo and amplify it to clamp a piece of material to a print surface. Further research must be done to make it more compact and rigid.

9.4.2. Dedicated Software

Throughout the course of the project, we realized that the software for this system could be extended far beyond what we would be capable of building, especially given the lack of dedicated programming experience on the team. Our team found it possible to implement enough electronics to control the print removal system and the necessary communication between the 3D printer and the print removal system. However, even this proved difficult to complete, and the finished product is not nearly as well-made as it could be. We recommend that in the future, a dedicated team of students should be assembled to build a custom software suite for this system. A dedicated team of students could design a complete application to take the place of OctoPrint. While OctoPrint works well in a setting with one printer, it is not easily scalable to a print farm setting, which is the long-term vision of this project.

Our current set-up requires a Raspberry Pi for each printer, which each run their own instance of OctoPrint; the user must choose a printer to connect to and then upload their g-code file to that specific printer. A more scalable model would be to have one central application that acts as the hub for accessing all printers. Users access the portal and upload their g-code files. The system then decides which printer to send the job to and notifies the user of its decision. We also recommend replacing the current two-controller design with one controller that controls both the printer and the print removal system. Implementing printer control would require research into 3D printer firmware but would be helpful in improving system control because multiple devices would not have to communicate with each other about their statuses. This software would be much more complex than the current LaunchPad software, however, so it would likely be necessary to implement the software in a RTOS setting on a microcontroller.

9.4.3. Sorting System

During the beginning of the project, the inclusion of a sorting system was discussed as a possibility, but it was eventually left out due to scope and time constraints. In order to fully design and realize a sorting system, the following are recommended in order to accomplish this goal.

First, a team should find a way to determine who submitted a part to the system, and to have a way for that information to remain with the part after it has been removed and is ready for sorting. If possible, the team could also focus on a way to notify the user that their part is ready for pickup after it has been sorted. Second, they should build a system that can move parts around to their sorted locations. This design should be easily scalable in order to fit any number of printers on the system, as well as be able to deal with any size part that could be printed.

10. Conclusion

There is a fear among society that automation will take jobs from humans. This fear is brought on by short-sighted thinking. When the industrial revolution happened in the 1800's people's jobs were initially lost to machines. However new jobs emerged repairing and working with these machines. We feel we are in the same scenario. While our device may initially replace the workers at the print farm whose job it is to remove and start new prints, we foresee this device opening new jobs. This could be in maintaining these devices or people opening new print farms due to the now lower labor costs. Lower labor costs mean cheaper manufacturing which could accelerate engineering projects and lower our dependency on foreign countries for manufacturing. Our device would increase the power consumption of the building in which it is in. This would increase the overhead cost of running the business. But with the increase in efficiency of renewable energy resources, these costs could easily be supplemented with things like solar panels that would both decrease electricity costs and reduce environmental impacts.

Our success in this project is due to the extensive engineering background provided to us by WPI in our times as students. Throughout this project, our team found ourselves contributing to our design with experiences from prior course work. We first began by breaking down customer needs, developing design ideas and weighing them against other ideas to determine the best design to choose, as taught to us in project courses such as Introduction to Engineering Design (ME2300) or Advanced Engineering Design (ME4320). With a design in mind, we began modeling our device and performing beam deflection calculations in SolidWorks, which we were first introduced to in Introduction to Computer-Aided Design (ES1310). To verify the deflection values given to us by SolidWorks, we used our knowledge of stress analysis, courtesy of ES2502-Stress Analysis, and MATLAB, learned ECE2010-Introduction to Electrical and Computer Engineering, to manually calculate the theoretical deflection and stresses our system would undergo. In selecting the electronic components, we first had to determine the force that the motors would need to provide to our system. Design of Machine Elements (ME3320) proved invaluable for us in this step of the project. To make our design cheaper and easier to assemble, we drew upon skills learned in Design for Manufacturability (ME5441). As the design was finalized and we began manufacturing, skills from other courses came forth. Cutting materials to the appropriate dimensions and hole patterns in Washburn Shops was possible due to the skills learned in ME1800.

Previous course work also provided extensive preparation in terms of the electrical and computer components of the project. In terms of working with the microcontrollers and understanding how they work, the courses Embedded Computing in Engineering Design (ECE 2049) and Electrical and Computer Engineering Design (ECE 2799) were incredibly useful. These classes provided crucial information about a microcontroller's architecture as well as experience with and methods for debugging. In some cases, extra circuitry was needed to be built for a system to work properly, and in these cases, the Microelectronics classes (ECE 2201 and ECE 3204) provided good insight for creating this circuitry since it generally involved using transistors, MOSFETS, or diodes.

While we drew on numerous previous experiences during this project, we also learned new skills and how to use new machines on campus. One obvious machine was the 3D printer. Our team came in with various levels of experiences with 3D printing ranging from very experienced, to no experience at all. When our team left campus in March, everyone was very comfortable both operating and repairing 3D printers. To allow for further functional controls of a 3D printer, members of our team learned how to write G-Code (the language which controls the printer). By manually inputting controls through a terminal, our team could tell the printer which functions to perform, such as preheating or moving the bed, via serial connection. This serial connection in conjunction with OctoPrint allowed our team to control the functionality of the printer from anywhere, which was crucial to the meeting the customer requirements. Another machine that was learned early on was the Scanning Electron Microscope (SEM). This machine was used to determine the surface roughness of the paper that we were considering printing on. This was to experimentally find if there was a correlation between print surface roughness and print adhesion force.

Seeing that this device's main goal was to automate a process normally completed by humans, a significant amount of time was spent analyzing the safety and ethics of our device. Since our device was not designed to work in conjunction with a human, our device is safe. If our device does break, it can simply be removed from the front of the printer. This will allow the printer to continue its normal operation and someone can perform service on it without the fear of getting pinched by its moving parts.

Sources

1. All3DP. (2020, April 20). 2020 Best Free 3D Printing Software. Retrieved May 17, 2020, from https://all3dp.com/1/best-free-3d-printing-software-3d-printer-program/#repetier-host
2. Automatic Infinite 3D Printer. (2018, April 2). Retrieved May 17, 2020, from https://hackaday.io/project/114738-automatic-infinite-3d-printer
3. Bala O, Arisu HD, Yikilgan I, Arslan S, Gullu A. Evaluation of surface roughness and hardness of different glass ionomer cements. Eur J Dent. 2012;6(1):79–86.
4. BLACKBELT 3D Printer Desktop Version. (n.d.). Retrieved May 17, 2020, from https://blackbelt-3d.com/blackbelt-3d-printer-desktop-version
5. BuildTak FlexPlate System. (n.d.). Retrieved May 16, 2020, from https://www.BuildTak.com/product/BuildTak-flexplate-system-4/
6. DFRobot. (2015, February 10). MDV 2x2A DC Motor Controller (L298N). Retrieved February 15, 2020, from https://www.dfrobot.com/product-66.html
7. DigiKey. (n.d.). Microchip Technology. Retrieved November 5, 2019, from https://www.digikey.com/products/en/integrated-circuits-ics/embedded-microcontrollers/685
8. DigiKey. (n.d.). EK-TM4C123GXL. Retrieved November 5, 2019, from https://www.digikey.com/product-detail/en/texas-instruments/EK-TM4C123GXL/296-35760-ND/3996736
9. Easy-Peelzy (square 20 cm / approx. 8 inches). (n.d.). Retrieved May 16, 2020, from https://3d-easy.xyz/products/easy-peelzy
10. Goldberg, D. (2018, December 21). History of 3D Printing: It's Older Than You Think [Updated]. Retrieved from https://www.autodesk.com/redshift/history-of-3d-printing/.
11. Häußge, G. (n.d.). Download & Setup OctoPrint. Retrieved May 18, 2020, from https://octoprint.org/download/
12. Hussain, M., King, B., Kirven, C., Sayers, B., Secino, B., & Strasburger, B., (2018) *Increasing 3D Printer Throughput with Automated Part Removal*. Worcester Polytechnic Institute
13. Loff, S. (2015, February 25). International Space Station's 3-D Printer. Retrieved from https://www.nasa.gov/content/international-space-station-s-3-d-printer.
14. Malengier, B., Hertleer, C., Cardon, L., & Van Langenhove, L. (n.d.). 3D Printing on Textiles: Testing of Adhesion. Journal of Fashion Technology & Textile Engineering. doi: 10.4172/2329-9568.S4-013. Retrieved from https://biblio.ugent.be/publication/8568361/file/8568362.pdf
15. NAK3DDesigns. (n.d.). NAK3DDesigns/White-Knight. Retrieved May 17, 2020, from https://github.com/NAK3DDesigns/White-Knight
16. Paper Roughness. (2018). Paper Roughness. Irvine, CA.
17. Raspberry Pi. (n.d.). Teach, Learn, and Make with Raspberry Pi – Raspberry Pi. Retrieved November 7, 2019, from https://www.raspberrypi.org/

18. "RepRapOneDarwin." *RepRap*, . 23 May 2015. 30 Mar 2020. Retrieved from https://reprap.org/mediawiki/index.php?title=RepRapOneDarwin&oldid=148986>.
19. Schmalz, B. (2015, February 13). Big Easy Driver. Retrieved February 22, 2020, from https://www.sparkfun.com/products/12859
20. Shigley, Joseph E.; Mischke, Charles R.; Budynas, Richard Gordon (2003), Mechanical Engineering Design (7th ed.), McGraw Hill, ISBN 978-0-07-252036-1.
21. Texas Instruments. (n.d.). Code Composer Studio (CCS) Integrated Development Environment (IDE). Retrieved November 5, 2019, from http://www.ti.com/tool/CCSTUDIO
22. The Factory of The Future: 3D Demonstrators. (n.d.). Retrieved May 17, 2020, from https://www.stratasys.com/demonstrators
23. Zhang, Q., Wong, J. P. S., Davis, A. Y., Black, M. S., & Weber, R. J. (2017). Characterization of particle emissions from consumer fused deposition modeling 3D printers. *Aerosol Science and Technology*, *51*(11), 1275–1286. doi: 10.1080/02786826.2017.1342029
24. 3D Printer Filament Comparison. (n.d.). Retrieved September 17, 2019, from https://www.matterhackers.com/3d-printer-filament-compare
25. 3M Scotch. (2009, April 6). *Blue Painter's Tape* [Material Safety Data Sheet]. Retrieved September 20, 2019 from http://www.aoc.nrao.edu/engineering/ElChemInventory/Merged%20Files%20BC/3M%20Scotch%20blue%20painter's%20tape.pdf

Appendix A: Full FMEA

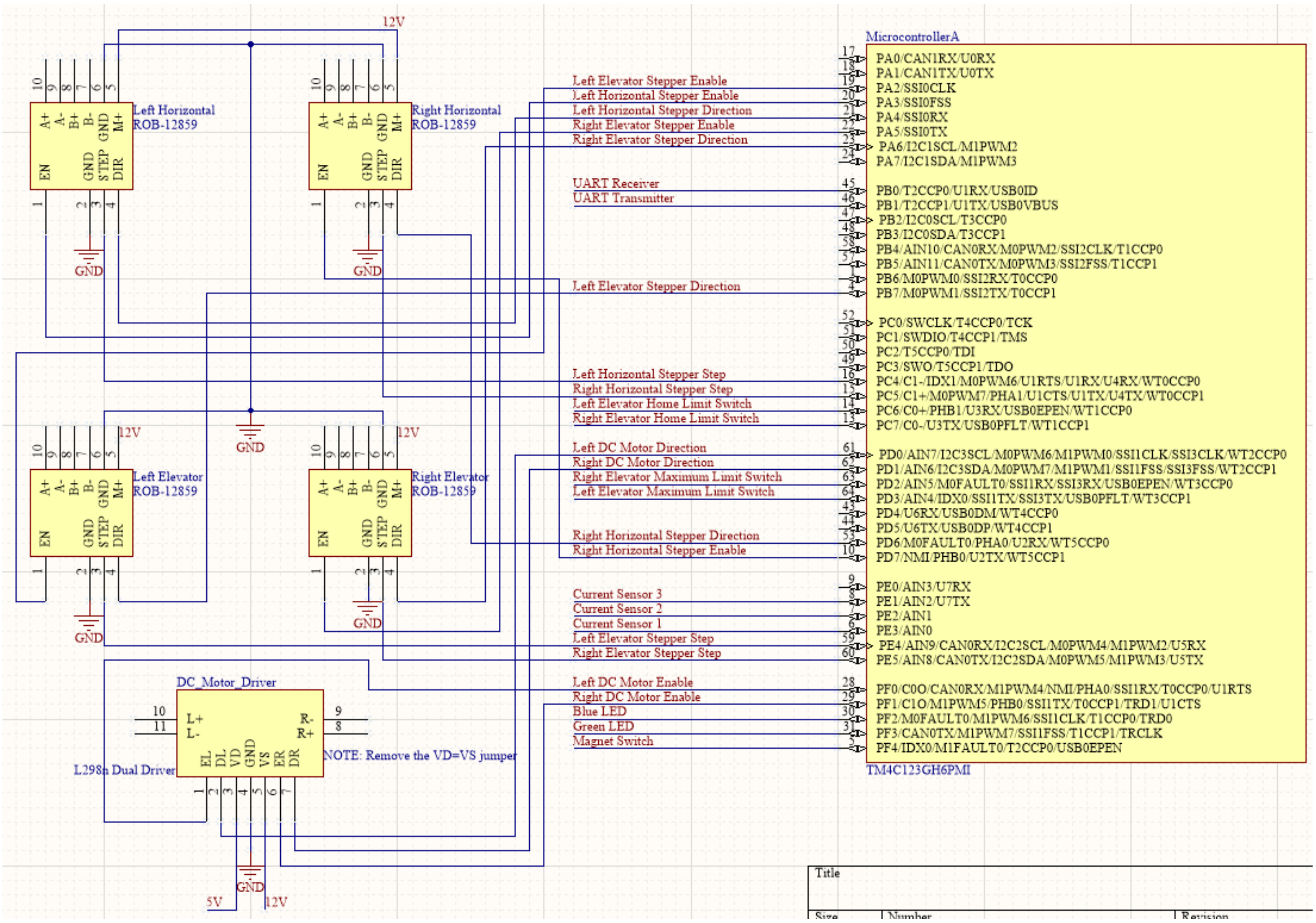
Item Number	Component	Function	Failure Mode	Failure Mechanism	Local Effects	Higher Level Effects	End Effects	Detection Method	Prevention Methods	Severity Level (1-3)	Risk Level (1-3)	Remarks
1.01	Lifting Tab	Indexes with bed corners to lift it from magnetic platform	Component failure	Fracture	Component breaks and needs to be replaced	Lift system fails to lift bed from magnetic platform	Component needs to be replaced	Scheduled checking for cracks	Correct material selection	1	1	
1.02			Operation failure	Tab slips from bed	Component doesn't index properly with bed and slips	Lift system fails to lift bed from magnetic platform	Tab must be at least re-seated	System monitoring with webcam	Test tab surface and geometry	1	2	As long as the roller has sufficient pressure on tab, this shouldn't be an issue
2.01	Lead Screw	Drives lifting system	Operation failure	Lead screw binds with screw nut	Lead screw binds and turning no longer lifts bed	Lift system fails to lift bed from magnetic platform	User must manually un-bind lead screw	System monitoring with webcam	Ensure lead screw is properly lubricated	1	3	
2.02			Operation failure	Incorrect height when bed feeding out of flex track	Lead screw and system (one or both sides) at incorrect height and bed cannot reverse back onto lift system and onto printer	System binds up as rollers trying to reverse bed out of flex system	Bind must be manually cleared and height of elevator system manually fixed	Webcam monitoring	Home steppers (at top) before reversal of flex track system after each print	1	3	
3.01	Driving Rollers	Drives bed through flex track	Operation failure	Rollers slip	Rollers slip and spin without driving bed forward	Bed not driven through track or driven unevenly	If not auto-corrected component needs to be realigned	System monitoring with webcam, scheduled checking of roller surface	Ensure rollers have sufficient friction with bed	2	1	
3.02			Operation failure	Pulley/belt slip	Second set of rollers will not spin	Bed only driven by first set of rollers or driven unevenly, and won't go through full track	If not auto-corrected component needs to be realigned	System monitoring with webcam	Correct belt tensioning	2	2	
3.03			Component failure	Belt breaks	Second set of rollers will not spin	Bed only driven by first set of rollers or driven unevenly, and won't go through full track	Component needs to be replaced	System monitoring with webcam, scheduled checking of belt	Correct belt tensioning and proper belt material selection	1	3	
4.01	Lifting Rollers	Drives bed into flex track	Operation failure	Rollers slip	Rollers slip and spin without driving bed forward	Bed not driven into flex track or driven unevenly	If not auto-corrected component needs to be realigned	System monitoring with webcam, scheduled checking of roller surface	Ensure rollers have sufficient friction with bed	2	2	Shouldn't be an issue as long as roller is under sifficient pressure
5.01	Lifting and driving roller hubs	Connect rollers to motors	Operation failure	Set screw loosens, hubs slip	Motors turn but wheels stationary	Bed not driven into/through flex track	Hub set screws need to be tightened	Webcam monitoring, scheduled checking of hubs on all specified rollers	Ensure all set screws sufficiently tightened, possibly thread locker on set screws	2	3	
6.01	Stepper Motors	Controls lead screws, driving rollers	Component failure	Stepper(s) burn out	Stepper no longer works	System binds up and cannot operate	Stepper needs to be replaced	Webcam monitoring	Test stepper motors and ensure correct operation	1	3	
6.02			Operation failure	Stepper(s) skips step(s)	Stepper skips step(s)	Affected system will bind or become misaligned	Stepper needs to be manually adjusted	Webcam monitoring	Test stepper motors and ensure correct operation	2	3	
7.01	DC Motors	Controls lifting rollers	Component failure	Motor(s) burns out	Motor no longer works	Bed will not be driven into flex track	Motor needs to be replaced	Webcam monitoring	Test motor and ensure correct operation	1	3	
8.01	2020 Hardware	Holds together frame, and secures components to frame	Operation failure	Harware loosens	Hardware loosens	Frame members or components come loose/fall off	System needs to be reassembled /recalibrated	Webcam monitoring, scheduled checking of components	Ensure hardware properly tightened, reduce system vibration, possibly thread locker	1	3	
9.01	Flex track bearings	Guide bed and reduce friction	Component failure	Bearings bind up	Bearings no longer turn	Increased resistance to bed moving through flex track	Bearing needs to be replaced	Scheduled checking of bearings	Ensure bearings spinning properly before final assembly	3	3	
10.01	Scraper	Removes parts after surface tension broken	Component failure	Plastic breaks	Scraper breaks, possible deformation to blade	Parts are not removed	New scraper piece needs to be printed	Webcam monitoring	Extensive testing of system before final assembly	1	3	
10.02			Operation failure	Scraper stuck under part	Not enough surface tension broken, scraper gets stuck under part	Parts are not removed	Part needs to be removed manually and system reset	Webcam monitoring	Testing different geometries before final assembly	1	3	

11.01	Raspberry Pi	Interfaces with printer through Octoprint	Component failure	Raspberry Pi burns out	Raspberry Pi no longer communicates with system(s)	Queue cannot operate and printing stops	Pi needs to be replaced and new one needs to be recoded	Webcam and Octoprint monitoring and investigation upon ceased system	Monitor during testing and add fans if necessary	1	3	
11.02			Operation failure	Octopi malfunctions/corrupts	Octoprint no longer communicates with printer	Queue cannot operate and printing stops	Pi needs to be restarted or possibly re-flashed	Octoprint monitoring and investigation upon ceased system	N/A	1	3	
11.03			Operation failure	Pi becomes physically disconnected from other system(s)	Pi no longer communicates with system(s)	Queue cannot operate and printing stops	Pi needs to be reconnected	General system monitoring	Minimize human interaction with system	2	3	
12.01	Arduino	Controls electrical systems	Component failure	Arduino burns out	Arduino no longer controls removal system	Removal system cannot operate, but queue system may attempt to continue prints	Arduino needs to be replaced and new one needs to be recoded	Webcam monitoring	Monitor during testing and add fans if necessary	1	3	
12.02			Operation failure	Arduino malfunctions/corrupts	Arduino no longer controls removal system	Removal system cannot operate, but queue system may attempt to continue prints	Arduino needs to be restarted or possible re-flashed	Webcam monitoring	N/A	1	3	
12.03			Operation failure	Arduino becomes physically disconnected from other system(s)	Arduino no longer controls removal system	Removal system cannot operate, but queue system may attempt to continue prints	Arduino needs to be reconnected	General system monitoring	Minimize human interaction with system	2	3	
13.01	Electrical System	Connects all systems and controls	Component failure	Electrical system fails due to short or other physical malfunction	Electrical system no longer works	Removal system cannot operate, but print may finish	Electrical system needs to be partially or entirely replaced	General system monitoring	Minimize human interaction with system	1	3	
14.01	Bed	Surface that is printed on	Operation failure	Surface tension not broken	No surface tension broken and part doesn't peel away from bed	No area for scraper to get under, does not engage, part may not be removed	Part has to be manually removed	Webcam monitoring	Test different sizes and geometries and adjust flex track severity if necessary	1	2	

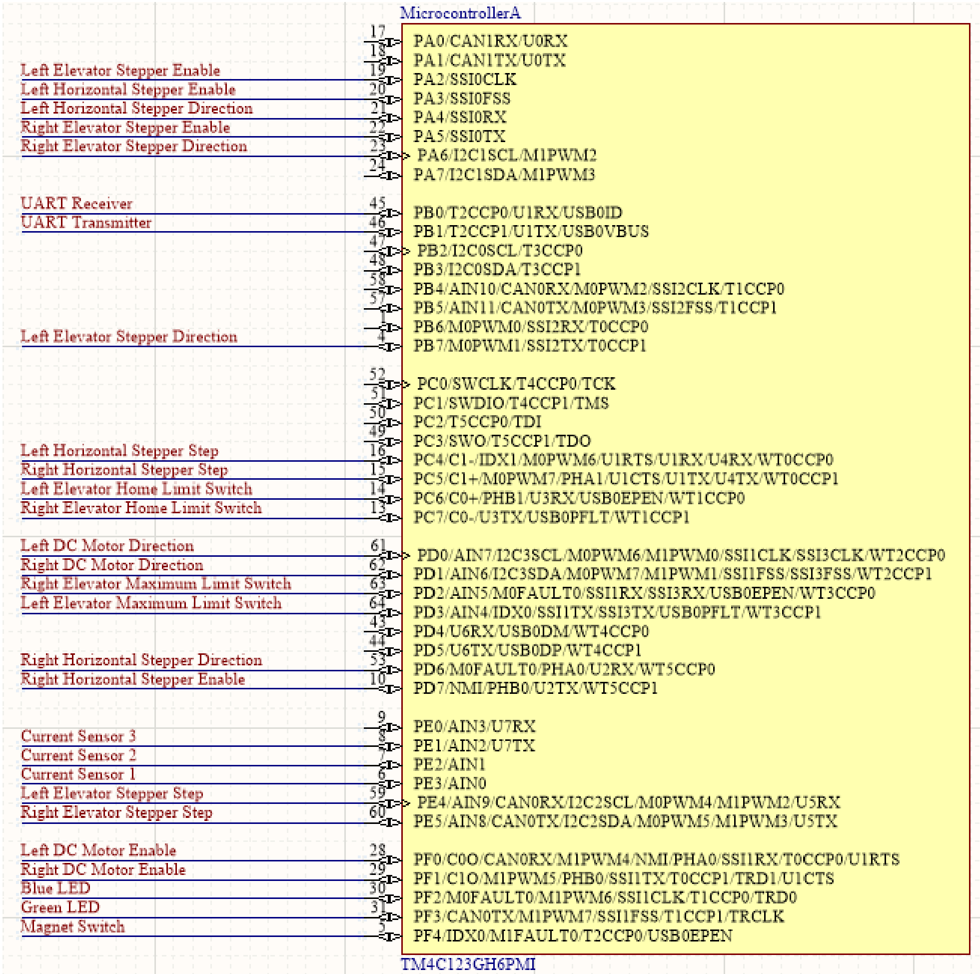
Severity Levels	Explanation	Risk Levels	Explanation
1	System ceases to operate entirely and components need to be fixed/replaced /reset	1	Of failures, likely to happen over others
2	System ceases to operate correctly and has to be reset, chance of self correcting	2	Somewhat likely
3	Does not affect system operation, superficial failure	3	Highly unlikely

Appendix B: Full Electrical Schematic and Individual Models

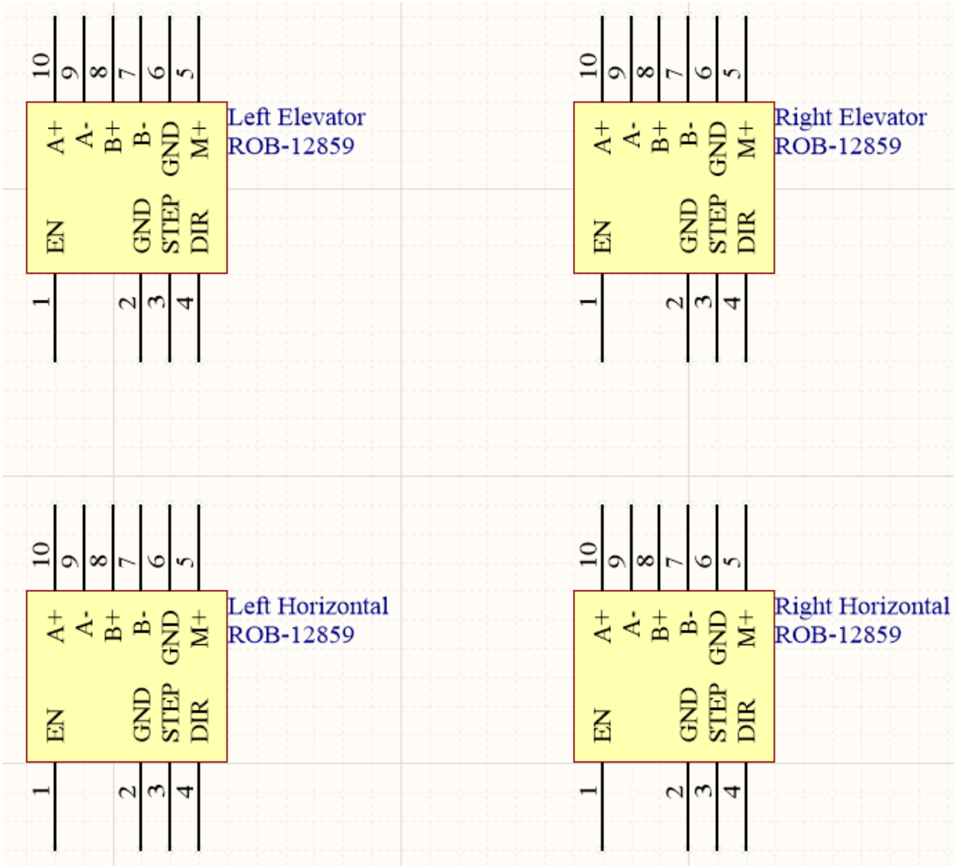
Full Schematic



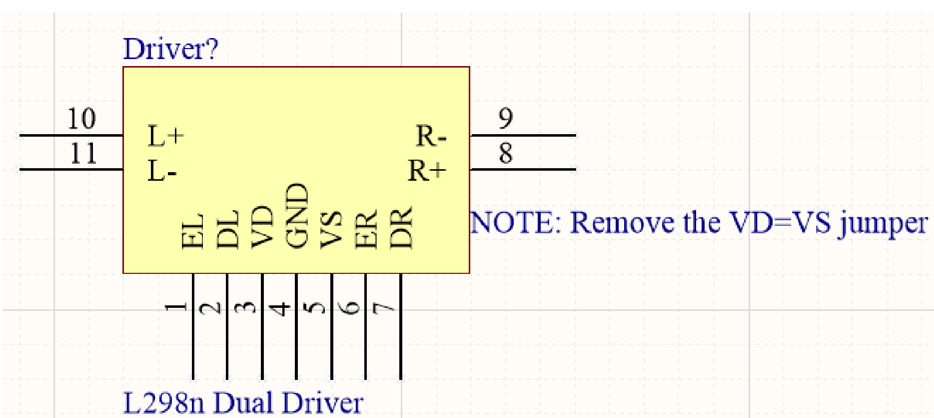
Microcontroller



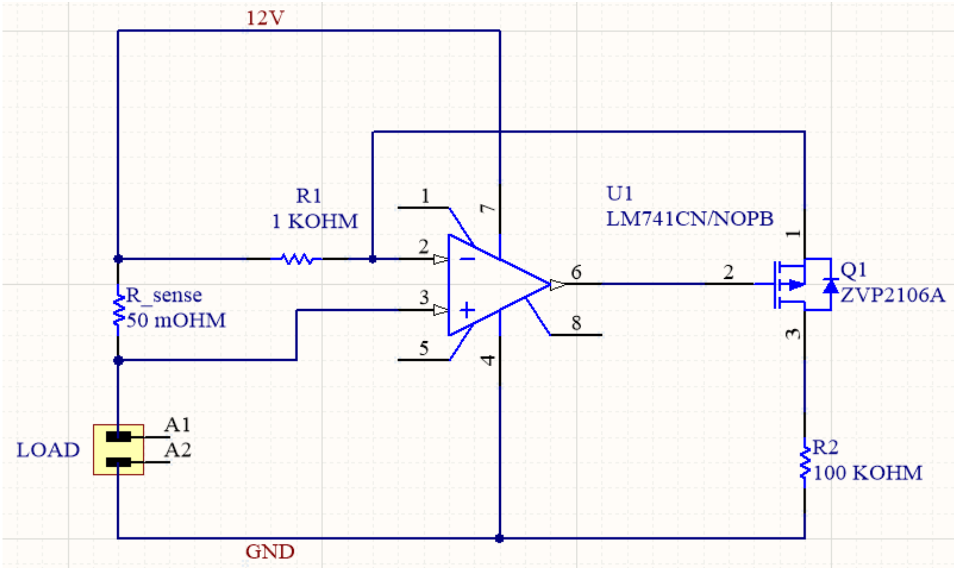
Stepper Motor Drivers



DC Motor Driver



Current Sensor



Appendix C: Custom End-of-Print G-Code Script

```
G21                ; sets units to mm

G1 X0 Z200 F3000 ; Bring extruder to side and up
G1 Y210            ; Bring bed forward

M600               ; stops printer action

M149 C             ; sets temp units to C
M104 S210          ; keeps nozzle temp at 210
M140 S60           ; keeps bed temp at 60
```

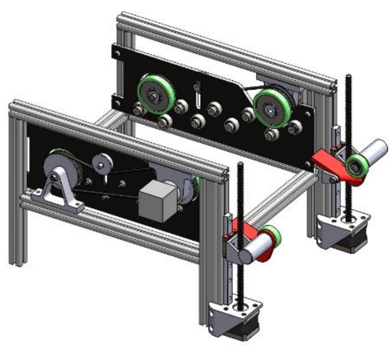
; the above g-code will present the finished print after completion (if not already done), will stop all printer action so the next print is not initiated, but will then immediately keep the bed and nozzle hot so there's no delay in starting the next print. operation will resume once next print is sent to machine

Appendix D: Assembly/Operation Instructions

FlexTrak

Automatic 3D Print Removal System

Instruction Manual



(FOR PROTOTYPE ONLY)
May 2020

EN

Preface

Thank you for your use of the *FlexTrak* automatic 3D print removal system. This system is designed to allow continuous 3D printing through the use of a bed flexing track integrated with a g-code file queue using OctoPrint. To ensure the product is assembled and used properly, please read and follow these instructions carefully.

3D printing is a rapidly growing field and print farm setups are becoming increasingly more common in corporate, educational, and recreational settings, and *FlexTrak* is a great system to enable high-output printing with little to no human intervention. The *FlexTrak* attaches directly to a Prusa Mk3 printer with a removable magnetic bed, but the system will soon be able to accommodate a wider variety of i3 style printers. As the print finished, the bed is fed through a curved bearing track to flex the bed and remove the print, the bed is fed backwards and re-seated onto the printer, and the software tells the printer to start the next file. The only human intervention required is to reload filament and empty the print collection bin.

Please enjoy the *FlexTrak* system, and remember, print responsibly!

NOTE: This instruction manual is for the *FlexTrak* prototype only.

For Your Safety

⚠ WARNING

Please read the following safety information carefully to prevent damage to your *FlexTrak* and/or injury to yourself and/or others.

Do Not Disassemble

DO NOT DISASSEMBLE
Removal of any part(s) and/or improper reassembly could result in system damage or injury. Contact a qualified repair service in the event of a malfunction.

KEEP OUT OF REACH OF CHILDREN
This product is not designed to be used by children. It may fall on a child. Small parts present a choking hazard. Failure to observe this precaution could result in injury.

DO NOT USE IF UNDER THE INFLUENCE
Use while under the influence of drugs or alcohol could lead to improper operation and result in system damage or injury.

REGULARLY INSPECT THE PRODUCT FOR DAMAGE OR WEAR
Failure to observe this precaution could result in system damage or injury.

KEEP FINGERS AWAY FROM MOVING PARTS
There are moving parts and pinch points in this system. Placing extremities close to these parts could result in injury.

Using Your *FlexTrak*

Please read the following instructions carefully before using your *FlexTrak*. Observe all safety precautions listed in the *For Your Safety* section. Failure to observe these precautions could result in damage to your product and/or bodily injury. **It is very highly recommended that you also view our instructional video.**

Before Use

Please make sure the *FlexTrak* system is fully assembled according to the provided instructions. Ensure all parts move as intended and there are no interferences or binding parts that could hinder normal operation. Ensure all wiring matches the provided diagrams to avoid damaging electrical components. Ensure all parts of the system are resting on a secure platform level with the printer, and that neither the system or printer are at risk of moving or falling during operation. Double check that all parts are in working order and there are no signs of damage.

Ensure that all g-code files that will be loaded into the queue are sliced correctly for the Prusa Mk3 i3 printer, and that there is no geometry or conditions that will cause the print to fail. Also make sure all files are sliced for the same filament profile to prevent failure. Load the selected filament into the printer, and ensure it flows cleanly and is not clogged.

Queuing Prints

Gather the selected g-code files to be printed. Connect the computer where the files are stored to the Raspberry Pi via the serial cable, and navigate to the local OctoPrint address https://octoprint.local. In the main panel, navigate to the "Print Queue" tab. Add g-code files to be queued using the "+" button. Optionally, change the number of copies of each g-code file.

Starting Prints

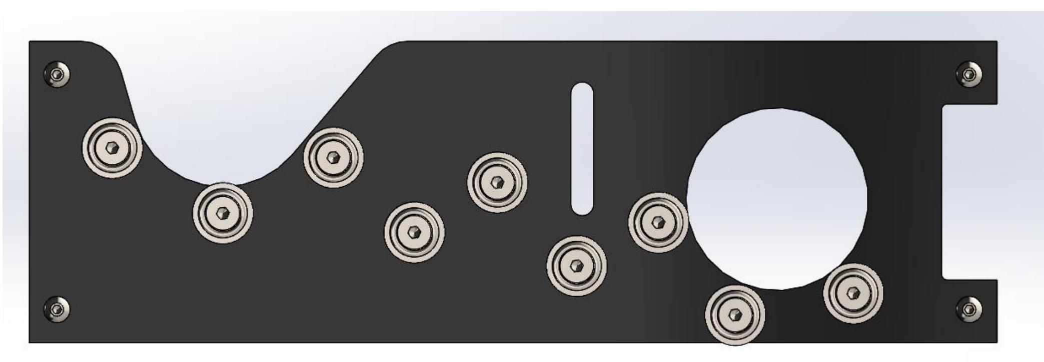
Make sure the printer is on and communicating with OctoPrint by checking that the printer information and "vitals" are being transmitted in the console tab of the main panel. Once everything is ready, click the "Start Print" button on the left side of the interface, and the system will take care of the rest. Observe the system for the first few layers of the first print to make sure the printer is set up and working properly.

FlexTrak Assembly

Before assembling the FlexTrak system, please check with the Bill of Materials and ensure there are no components missing. Below is a detailed set of text instructions on how to assemble the FlexTrak system, but please also refer to the provided images to clear up any remaining ambiguity and ensure assembly is correct.

Track Assembly

Gather all parts of the track: M8 shoulder bolts, M8 lock nuts, washers, and acrylic track pieces. Place one bearing onto the shoulder bolt, followed by a washer. Then slide the shoulder bolt into the track, and secure with a lock nut. Tighten the nut, the washer will rest against the acrylic plate and inner race to ensure they still turn. Repeat the same process with a shoulder bolt in the slot cut into the acrylic plate, but do NOT tighten this lock nut all the way yet. This bearing serves as a tensioner once the timing belts are in place. Also, this bearing will go on the OPPOSITE side of the plate as the other bearings since the timing belt will sit on the outside of the system. The result should look like the diagram below.



Scraper Assembly

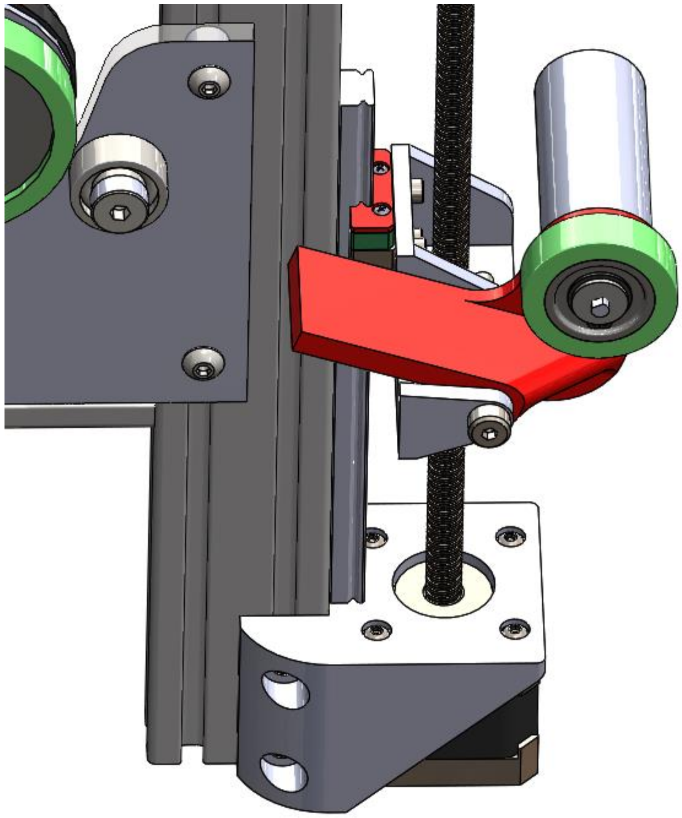
The scraper consists of 9 printed pieces that slide onto an 8mm chrome rod. Make sure they are all aligned in the same orientation, then attach a lock collar on either side, leaving about 1mm clearance on either side of the scraper pieces. Slide one scraper rod mount on either side and secure them with another lock collar on either side, leaving the same 1mm clearance.

Elevator System Assembly

Gather all parts for the elevator assembly *PER SIDE*: one geared DC motor, one small orange roller, one small roller hub (includes lock ring), one linear rail and carriage, one stepper with integrated lead screw (includes threaded bushing), one printed stepper mount, one printed stepper mount, one DC motor mount, one M8 shoulder bolt and lock nut set. Start by securing the stepper motor to the stepper mount with the included M3 screws. Next, attach the threaded bushing to the elevator base with the included short M3 screws. The bushing will be press fit into the base FROM BELOW (see image below for clarification), and then screwed in for security. This should be done with the bushing NOT threaded onto the lead screw.

Attach the DC motor to the DC motor mount using the included M2 screws. There is a small protrusion on the DC motor that will press fit into the mount, and then screwed in for security. Attach the small roller hub to the shaft of the DC motor using the included set screw, ensuring the screw is in contact with the flat portion of the shaft, and tighten to ensure it will not come loose. The flanged portion of the hub should be facing the mount, with about 1mm clearance between the two pieces. Then, slide on the small roller and secure using the included lock ring (lock ring pliers should be used to do this).

Connect the DC motor mount to the elevator base using the M8 bolt, with the threads of the bolt facing the lead screw, and the head facing what will be the inside of the system. The body of the DC motor should be adjacent to the lead screw and the roller facing what will be the inside of the system. See below image for clarification.



CAREFULLY thread this set onto the lead screw, and attach the elevator base to the linear rail carriage using the included M3 screws. See below image for clarification.

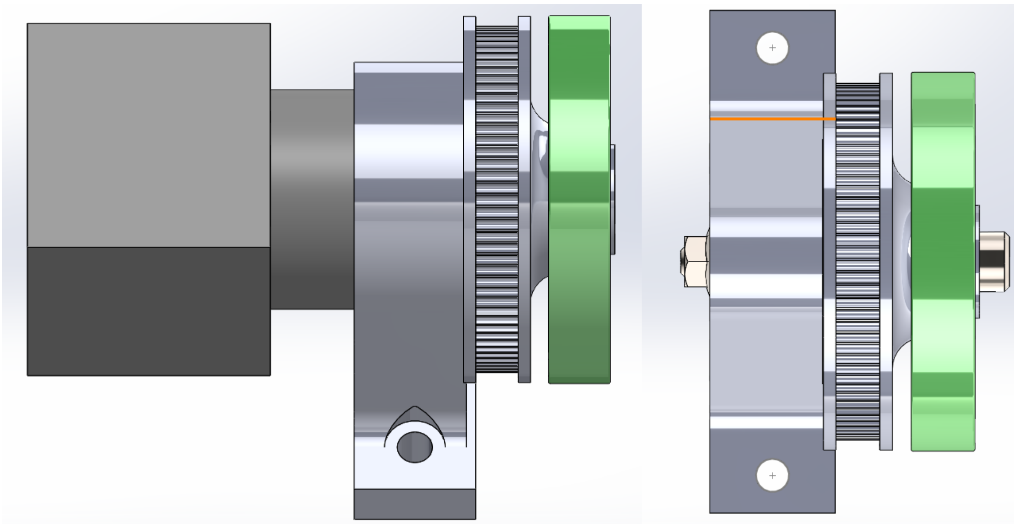
Driving Rollers Assembly

Gather all parts of the driving rollers assembly *PER SIDE*: one geared stepper motor, two large green rubber rollers, two large timing gears, two roller mounts, one keyed bushing (for the geared stepper motor), one smooth bushing (for the passive roller), and one M8 shoulder bolt and lock nut set (for the passive roller). First, assemble the geared stepper motor roller. Secure the mount to the motor using the provided M3 screws. Slide the large green roller onto the hex bushing so it is resting against the flange of the bushing. Then, slide on the timing gear with the raised hub resting AGAINST the roller (flat side facing AWAY from the roller). These parts will be press fit. Then, fit the NON-FLANGED side of the bushing onto the stepper shaft, aligning the cutout in the shaft with the flat spot in the bushing.

Next, assemble the passive roller. Slide the hex bushing onto the shoulder bolt, with the flanged side of the bushing against the head of the bolt. Slide on the large green roller and timing gear in the SAME CONFIGURATION as above.

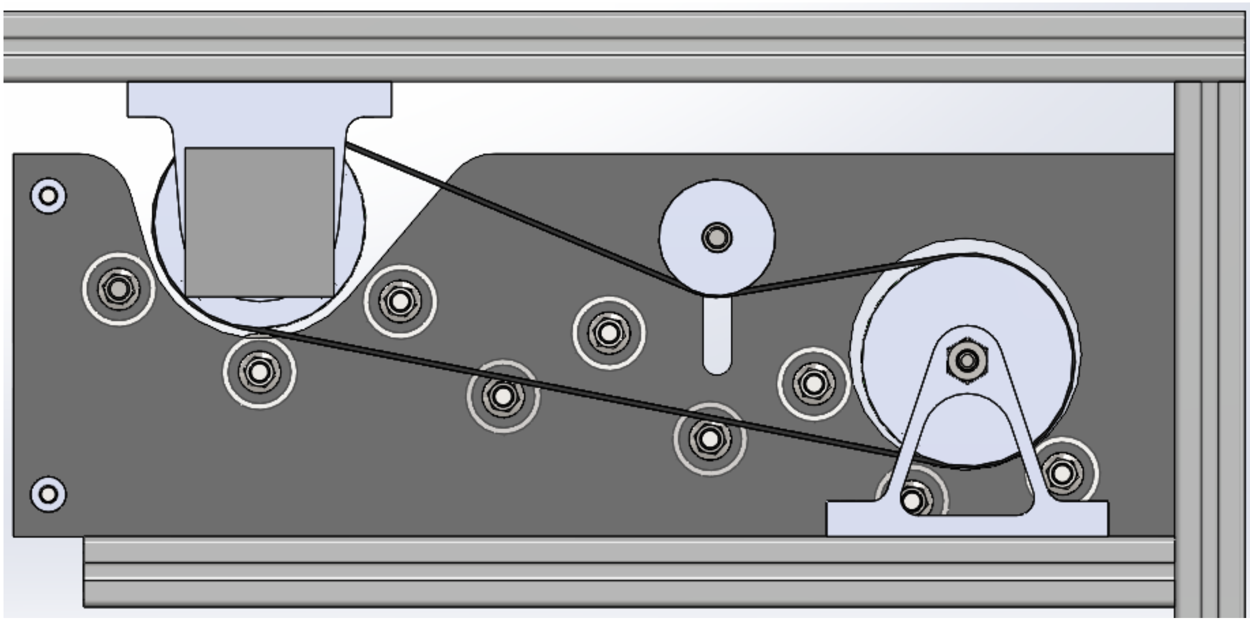
Slide the bolt through the mount, and then add the lock nut. Tighten the lock nut all the way, tolerances on the parts should allow the roller to spin freely on the bolt.

Repeat these steps in a MIRRORED fashion for the other side of the system. See images below for clarification.



Frame and Final System Assembly

Gather all parts of the frame: 20/20 beams, M5 T-slot nuts, 90° connectors, acrylic corner brackets, and M5 screws of designated lengths. Use the screws, t-nuts, connectors, and corner brackets to assemble the frame according to the diagram below. The t-slot nuts slide into the 20/20 beams before they are connected with the corner brackets and connectors. Assemble the frame and all above subassemblies as indicated in the below diagram. First build the sides of the frame and attach the track assembly plates. Next, attach the subassemblies for the driving geared stepper motors to the top bars in these frame sides, and the passive roller subassemblies on the bottom bars in the frame sides. Follow the image below for further positioning details.

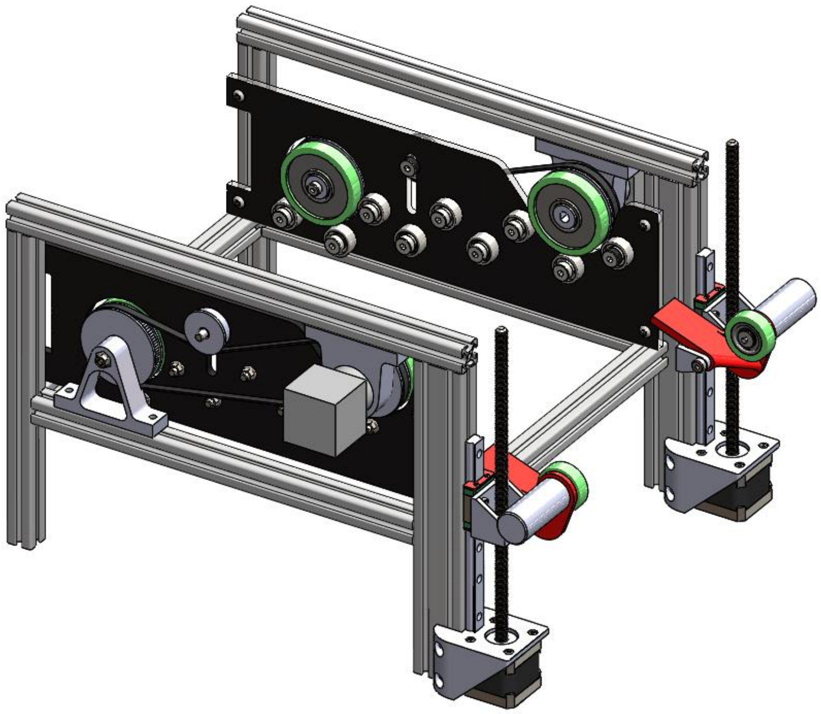


Place the timing belt around the timing gears, slide the tensioning bearing until the timing belt is taught, and tighten down the lock nut.

Next, connect the two sides with cross-beams and add in the scraper subassembly. This subassembly is held in place with a single M5 screw in each of the printed mounts. Do not fully tighten the screws for these supports as a height adjustment will need to be done later.

Finally, attach the elevator subassemblies. These attach to the bottom of the front two side beams via four M5 screws and t-slot nuts each in the lower printed mounts, and then the linear slide rail is attached to the front two side beams using M3 screws and t-slot nuts.

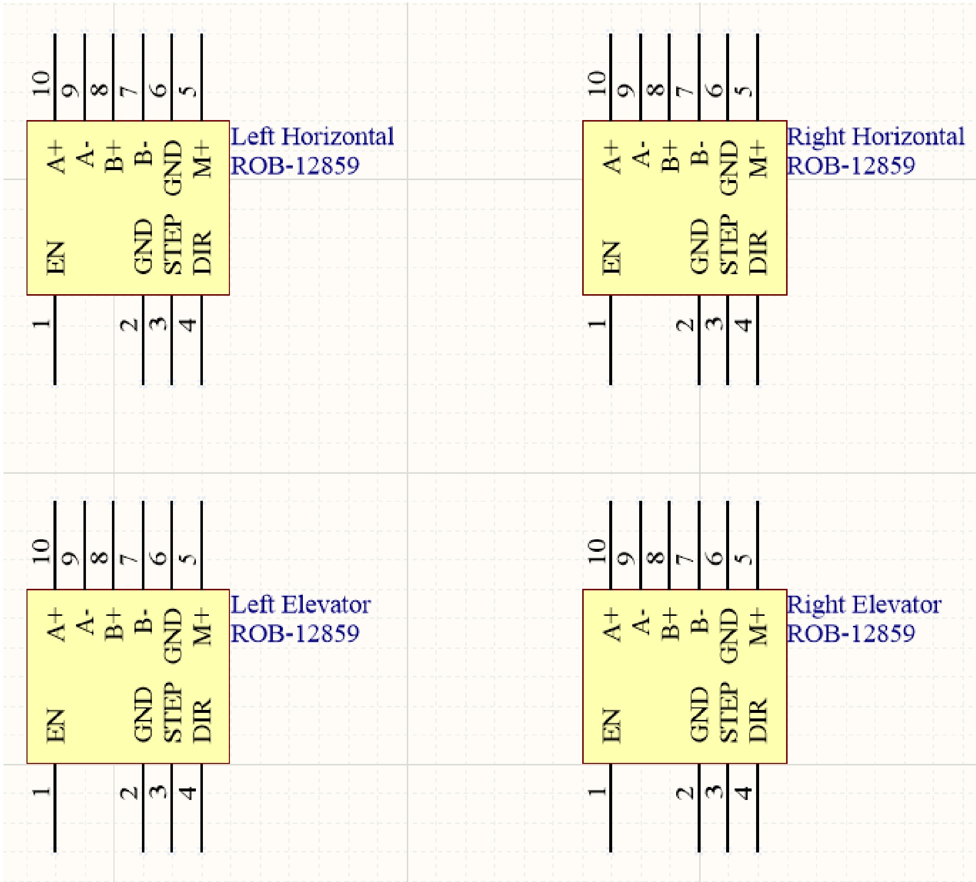
Refer to the diagram(s) below for clarification on any of the above steps.



Wiring

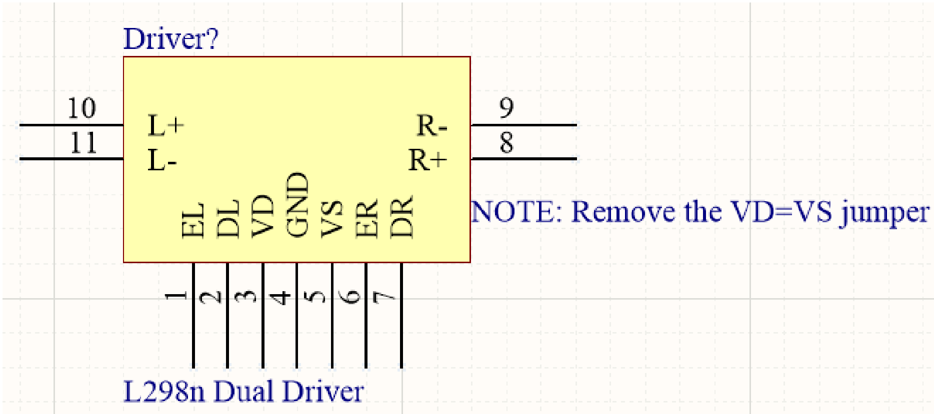
There are four main components to wiring the Flex-Trak system: The microcontroller, the stepper motor drivers, the DC motor drivers, and the current sensors.

For the stepper drivers, and conversely the stepper motors in the system, are arranged in this orientation when looking from the side at which the printer bed enters the system:



The M+ wire of each stepper driver is connected directly to a 12V DC power source, and the 2 GND pins for each driver are connected to ground. For the elevator stepper motors, the connections of the motor from left to right goes A+, A-, B+, B-, and as such, the driver connections in turn follow this from left to right. On the other hand, the horizontal stepper motor connections go from left to right as A+, B+, A-, B-, so the middle two wires are flipped for these connections. The remaining 3 pins: EN (enable), STEP(step), and DIR(direction) are covered in the microcontroller section for the ports they are connected to.

Next there is the DC motors and the DC motor drivers. The layout for the DC motor driver can be seen below:

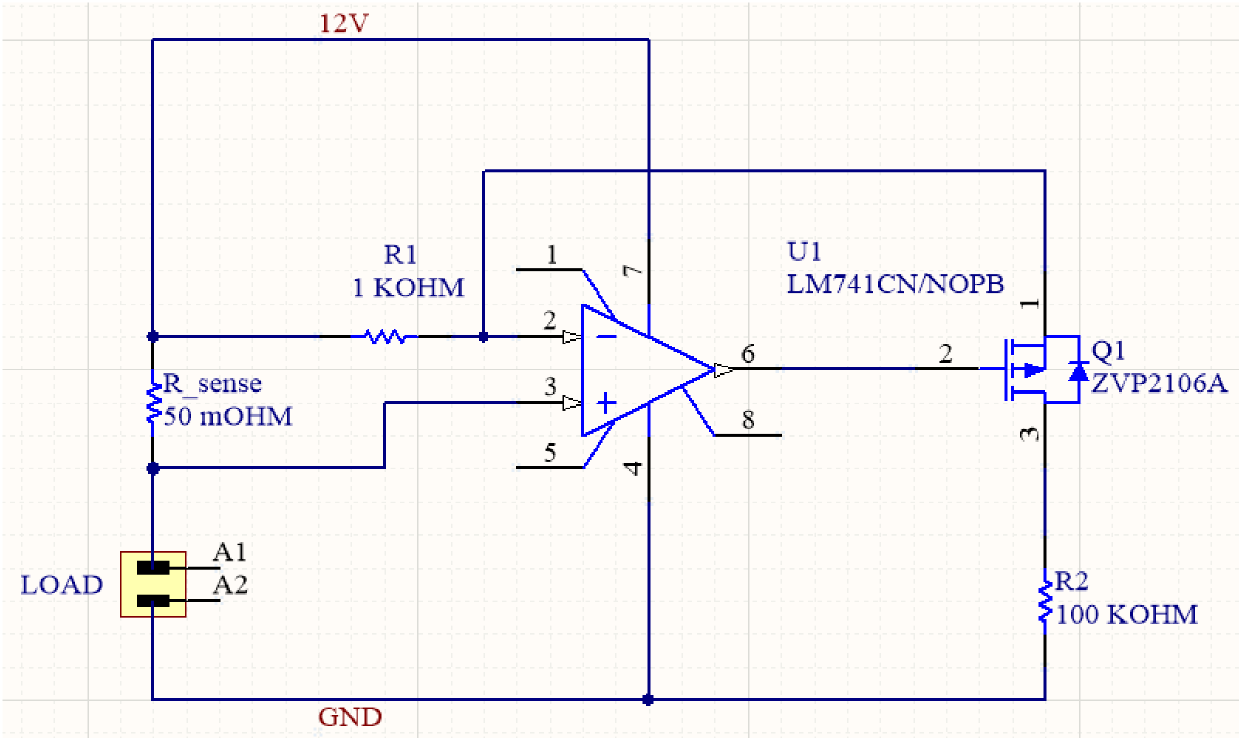


The L+ and L- wires connect to the left DC motor's positive and negative connections respectively. Likewise, the R+ and R- wires connect to the right DC motor's positive and negative connections. The VD pin is connected to the controller power (5V DC) and the VS pin is connected to the motor power (12V DC). It is important to note that there is a jumper on the board that sets the controller power equal to the motor power, and this jumper must be removed before use. The GND pin is connected to the system ground. Finally, the EL (left enable), ER (right enable), DL (left direction) and DR (right direction) connections are shown on the microcontroller.

The next piece of the wiring involves the microcontroller. The full microcontroller pin connection layout goes as follows:

Each wire label is placed above its respective wire in this diagram. Also, the first 3 letter-number combination describes the location the wire is connected to, where PX stands for Port X (examples: PA = port A, PB = port B, etc.), and the number describes the pin on that port to use (PA1 = port A pin 1, PC4 = port C pin 4, PF0 = port F pin 0). The numbers on the outside of the box are used in the case where the microcontroller is separated from the evaluation board, and in this case, the pins go counterclockwise from the first pin, designated in the corner with a dimple marking on it.

Finally, are the current sensors. The current sensor circuit follows the following schematic:



This circuit is added onto a different component to measure the current draw of said component. In order to add this circuit, connect the main power inputs of the component to the load section, with the 12V going to A1 and the ground going to A2 (For the stepper drivers, this would be M+ and GND). The output of this sensor is the node between pin 3 of the MOSFET (ZVP2106A) and R2 and increases with voltage as current increases through the load.

Interfacing with Printer

The printer is controlled by the OctoPrint 3D printer managing software. This software suite runs on a Raspberry Pi, and interfaces with the 3D printer over a USB serial connection. More details on installing, configuring, and using OctoPrint are described in the following sections.

Software Configuration

FlexTrak uses its own custom software to implement control and management of print jobs, queueing of multiple prints, and continuous operation. Two controllers exist in the system: a Tiva C Series LaunchPad (EK-TM4C123GXL) based around a TM4C123GH6 microcontroller for controlling the FlexTrak print removal system, and a Raspberry Pi 3 model B+ for controlling the printer itself. The Tiva C LaunchPad runs our own custom software, while the Raspberry Pi runs a modified version of the open-source printer management software OctoPrint. This section provides instructions on how to install these software packages and how to set up development environments for this system.

Developing for TI LaunchPad

FlexTrak uses the Texas Instruments IDE, Code Composer Studio, for programming and debugging the LaunchPad system. Download the latest version of CCS for free from TI's website: http://software-dl.ti.com/ccs/esd/documents/ccs_downloads.html. After choosing an installation directory in the installer, select full installation to easily install support for all devices and debug probes, which requires about 4 GB of free space on disk. Alternatively, you may elect for a custom installation, at which point the installer will prompt you to specify which devices and debug probes you would like to install; in this case, select "TM4C12x ARM Cortex-M4F core-based MCUs" in the "Select Components" window, and select all three debug probes in the "Install Debug Probes" window. Additionally, the software makes use of the TivaWare driver library, which provides APIs for using the various peripheral devices on the LaunchPad. TivaWare for TM4C can be downloaded for free from http://www.ti.com/tool/SW-TM4C; select the first entry in the table to get the latest version. Once downloaded, first close CCS if it is still open, then run the .exe file to install the TivaWare library. After selecting a workspace, CCS should alert you that it has detected new products; click install. After CCS fully opens, navigate to Project > Preferences > Build > Predefined Symbols. Under the top box, you should see the path to your new TivaWare installation (most likely C:\ti\Tivaware_C_Series_x.x.x.xxx). If not, select the option to add an include path, and find the TivaWare directory.

CCS is an Eclipse-based IDE, and will ask for a workspace upon startup, which is where the FlexTrak project will be stored. On the first startup, create a directory in the desired location to act as the workspace, and on subsequent startups, be sure to select this directory when prompted. To import the code into Code Composer Studio, clone the GitHub repository into the desired location on your machine. Open CCS, and navigate to File > Import, which will open the import wizard. Select General > Existing Projects into Workspace. In the next dialog box, choose the directory created above as the search directory in the top drop-down menu. The FlexTrak project will appear as an option to import. Uncheck "Copy Projects into Workspace" and choose finish.

Programming and Debugging the LaunchPad

Once CCS is set up, you may begin programming and debugging the LaunchPad. The main software state machine and all interrupt service routines (ISRs) are implemented in main.c. All definitions, includes, and pin designations are held in flextrak.h. Motor and actuator control and system initialization functions are implemented in motorctl.c and config.c respectively. Definitions related specifically to motor control are held in motorctl.h. The TivaWare driver lib APIs can be accessed under the includes tree in the project, and the User's Guide is available at https://www.ti.com/lit/ug/spmu298d/spmu298d.pdf.

To compile the project, select the build icon  from the top toolbar. To program the device, switch the slide switch at the top of the board to the "debug" position, and connect the device to your PC via the topmost micro-USB port. Then, select the debug icon  from the toolbar. In debug mode, the code can be resumed, suspended, terminated, or restarted from the toolbar. Hardware breakpoints can be added and removed at a specific line by double clicking in the left margin of the code at the appropriate line. While debugging the software, you have full control over code execution. Once the device is disconnected from debugging, however, it will continuously run its code while powered. **Note: Always terminate a debugging session before disconnecting the device from your PC to avoid damage to the device.**

Creating a Custom OctoPrint Raspberry Pi Distribution

To interface with 3D printers, FlexTrak implements a slightly modified build of OctoPrint on a Raspberry Pi 3 Model B+. As the Raspberry Pi lacks its own storage, it uses a MicroSD card to store its file system. Developers need an SD slot and MicroSD to standard SD adapter to program the Raspberry Pi.

The software uses three open-source GitHub libraries, FlexTrak's modified build of OctoPrint, CutomPiOS (https://github.com/guysoft/CustomPiOS), and OctoPrintPrintQueue (https://github.com/michaelnew/Octoprint-Print-Queue). CustomPiOS is used to configure a Raspbian disk image, the operating system for the Raspberry Pi, with OctoPrint installed on it. The readme file provides detailed instructions for creating custom disk images with CustomPiOS. To summarize, the developer must clone the CustomPiOS git repository to their machine, then modify the config file to clone the FlexTrak OctoPrint build rather than the standard library using the lines in the figure below. The URL specified in OCTOPI_OCTOPRINT_REPO_SHIP should be changed to use the URL for the custom OctoPrint build.

In chroot_script:

```
gitclone OCTOPI_OCTOPRINT_REPO OctoPrint
```

In `config`:

```
[ -n "$OCTOPI_OCTOPRINT_REPO_SHIP" ] || OCTOPI_OCTOPRINT_REPO_SHIP=https://github.com/foosel/OctoPrint.git
```

CustomPiOS will build a full disk image file for install, which must be written to the microSD card. It is recommended to use a program such as Balena Etcher to flash the disk image to the SD card. A few lines should also be added to the chroot_script file to clone the OctoPrintPrintQueue library and pack it as a .zip file to a known location in the file system. The process for installing this plugin is explained in the next section.

Configuring Raspberry Pi and OctoPrint Settings

Once the operating system has been installed, insert the microSD card into the Raspberry Pi and power it on. It is possible to log into the Rapsberry Pi via SSH by connecting to it over Ethernet from a PC; the default address is octopi.local. The default username for the Raspberry Pi is "pi" and the password is "raspberry" (without quotation marks). Once logged in, run the command "sudo raspi-config" to open the Raspi-Config tool to configure the Raspberry Pi settings as follows:

1. Change the device name and password -- the naming convention is FlexTrak<device number>, where device number starts at 1 and is unique for each FlexTrak system that is built.

2. Enable the serial module but disable the Linux console on serial -- this routes the Raspberry Pi's UART module to the GPIO header, Rx and Tx on pins 14 and 15 respectively.
3. Configure an internet connection, if desired.

Once the settings are configured, close the Raspi Config tool, stay connected to the Raspberry Pi via Ethernet, and navigate to https://<device name>.local in a web browser to access the OctoPrint terminal. Navigate to the settings page by clicking the button in the top right. Configure the settings specific to the printer being used. Then navigate to the plugins section of the settings tab, and select "install from an archive file." Install the OctoPrintPrintQueue plugin by selecting the .zip file that was prepared in the previous section.

Contact Us

At FlexTrak, we always value your input and feedback. **If you have any general questions, comments, or concerns, please contact:**

Prof. Pradeep Radhakrishnan
Email: pradhakrishnan@wpi.edu

If you have questions or comments about mechanical product design, assembly, or usage, please contact:

Alex Johnson: adjohnson@wpi.edu
Dant Mauriello: dgmauriello@wpi.edu
Andrew Kacherski: akacherski@wpi.edu
Owen Smallcomb: odsmallcomb@wpi.edu

If you have questions or comments about electrical and software systems, please contact:

Nicholas Colucci: ndcolucci@wpi.edu
Noah Donald: ndonald@wpi.edu