Composite DLP Printer

A Resin Printer for Creating Composite Parts

Krutik Shah, Elias Timmons, Andrew Mazurek, Kenneth Powley, Keith Vitz, Matthew Ciocco

Rowan University

Advanced Materials & Manufacturing Institute

Composite DLP Printer

A Resin Printer for Creating Composite Parts

by

Krutik Shah, Elias Timmons, Andrew Mazurek, Kenneth Powley, Keith Vitz, Matthew Ciocco

Instructor: Joseph Stanzione, Ph.D
Project Duration: May, 2022 - May, 2023
Department: Advanced Materials & Manufacturing Institute, Rowan University

RowanUniversity
ADVANCED MATERIALS & MANUFACTURING INSTITUTE

Contents

1

Introduction

The Composite DLP Printer utilizes Digital Light Processing (DLP) technology, commonly used in resin printers, to print composite parts from resin and fiberglass. It utilizes a belt-drive system where a roll of woven fiberglass mat is fed through the build area where the shape of each layer is etched onto the belt.

This project was broken up into three phases of development: Structure & Layout, DLP Functionality, and Composite Developments. Refer to Figure 1.1 for more information.

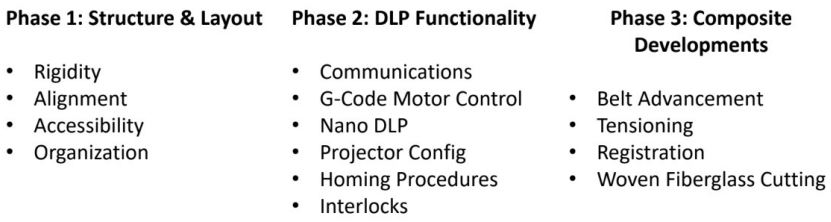

Figure 1.1: Three Phases of Development

There are various kinds of fibers that can be used for composite manufacturing, however for the purposes of this project, glass fibers are used. There are different types of fibers: continuous fibers, discontinuous fibers, particle fibers, and woven fibers. Continuous fibers are long fibers that are particularly strong throughout the entirety of a composite part, while discontinuous fibers are short and randomly aligned and are useful at a specific area of a part. Woven fibers are similar to continuous fibers except they are aligned by weaving the long fibers together, which will enhance the mechanical properties much more than continuous fibers. Embedding woven fibers throughout a resin part is a new feat in the field of additive manufacturing.

This project is developed under the Advanced Materials & Manufacturing Institute (AMMI) at Rowan University. This project is managed by Dr. Joseph Stanzione and Amit Dundhi, with consulting assistance from Karl Dyer, and with financial support from the U.S Army Cooperative Agreement W911NF-17-2-0227.

2

Design

The design and engineering of the printer is broken up into several sections: the mechanical frame, axes of movement, optics, belt-drive system, electronics, Marlin firmware, and NanoDLP.

2.1. Mechanical Frame

The printer utilizes long 40mm x 40mm aluminum bars that make up the mechanical frame. The frame has four leveling feet at each of the corners, which can be adjusted to level the entire printer. The printer has dimensions of 0.842 m x 0.842 m x 0.724 m. It is fully squared and leveled.

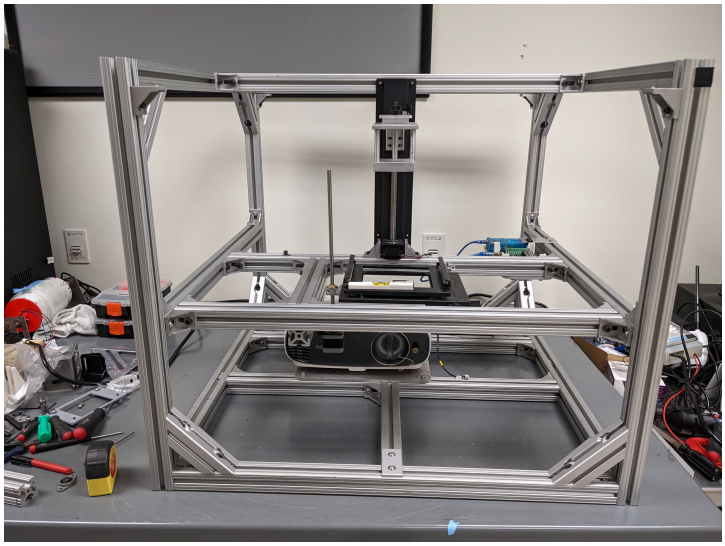

Figure 2.1: Mechanical Frame

2.2. Axes of Movement

There are a total of four axes of movement on the printer: the Z-axis, the Tilt axis, and the two belt rotation axes. In the Marlin firmware for the printer, the Z-axis is defined appropriately as the Z-axis, and the tilt axis defined as the Y-axis. One belt rotation axis motor is defined as Extruder 0, and the other motor defined as the X-axis.

2.2.1. Z-Axis

The Z-axis is the axis which purpose is to move the build platform vertically. The axis motor is a NEMA-23 motor, which rotates a ball screw. On the ball screw there is an attachment for the build platform mount. Rotating the screw will move the platform up and down.

2.2.2. Tilt Axis

The tilt axis is the axis where the vat tilts down to help release the part when printing. It is operated by a NEMA-23 motor, which attaches to the tilt mechanism. Figure 2.2 shows the Tilt axis mechanism.

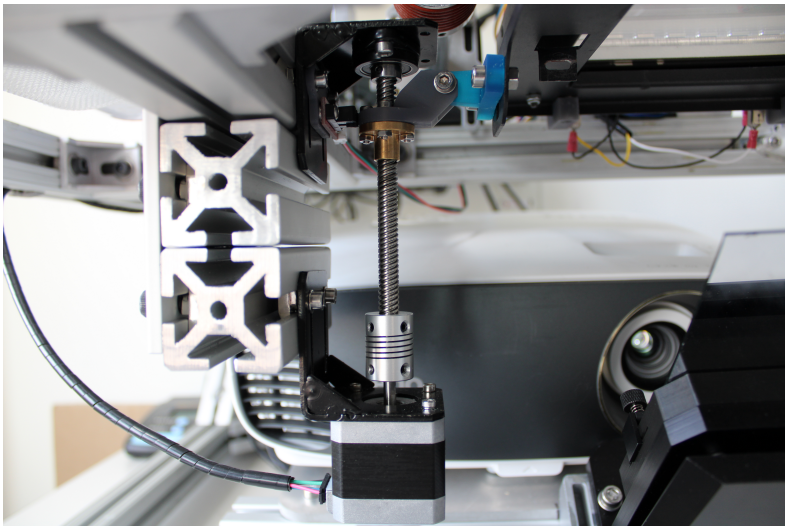

Figure 2.2: Tilt Axis

2.2.3. Rotational Axes

There are two rotational axes that individually control the belt-drive motors. The belt drive system utilizes NEMA-23 stepper motors, attached to worm drives. The worm drives provide higher torques than just with a stepper motor, and is meant to prevent slippage.

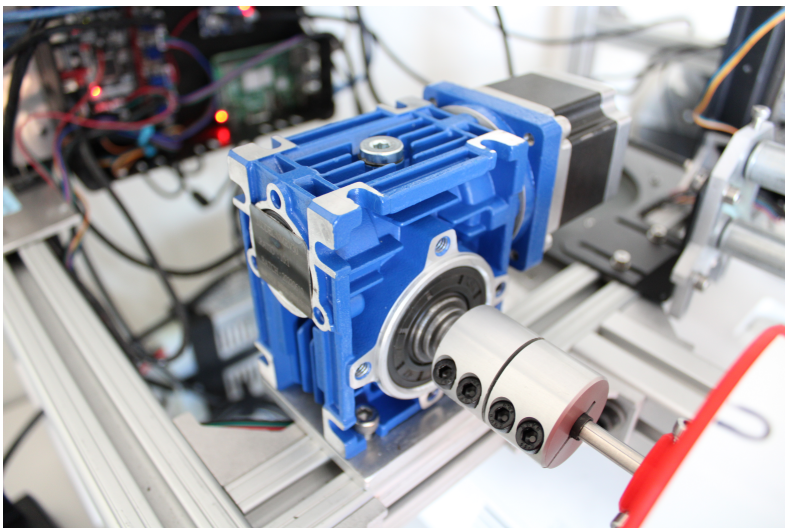

Figure 2.3: Rotational Axes

2.3. Optics

The optical subsystem includes the projector, which will project all the layers that need to be cured, the mirror, which will direct the projection into the clear vat surface and onto the build platform, and the laser cutter meant for cutting the fiber belt.

2.3.1. Projector

For the DLP printing functionality to work, the printer needs a projector to project each layer onto the build platform. This printer uses a consumer-rated projector: BenQ HT2550. The projector's user manual can be found on BenQ's website. This projector is highly advantageous to the purposes of this printer because it has a high output of ultraviolet light.

2.3.2. Mirror

The mirror mount uses a three-point fine adjustment to adjust the angle of projection onto the mirror. This fine adjustment should be calibrated to project a square image into the vat for an accurate projection. Refer to Section 3.2.4 for calibration procedure.

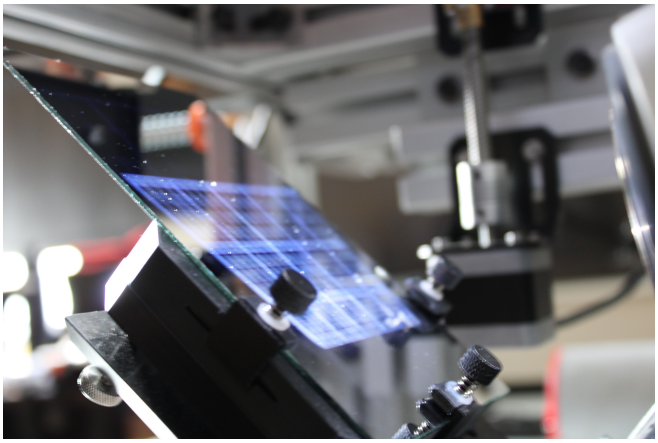

Figure 2.4: Mirror

2.3.3. Laser Cutter

This printer utilizes a diode laser rated for 20W. A diode laser uses a semiconductor to produce a laser. The laser cutter frame was reconstructed to be mounted on the printer. Some sample tests were performed with the diode laser to measure its cutting strength against the fiberglass belt. It was proven to cut through the fiberglass with minimal issues.

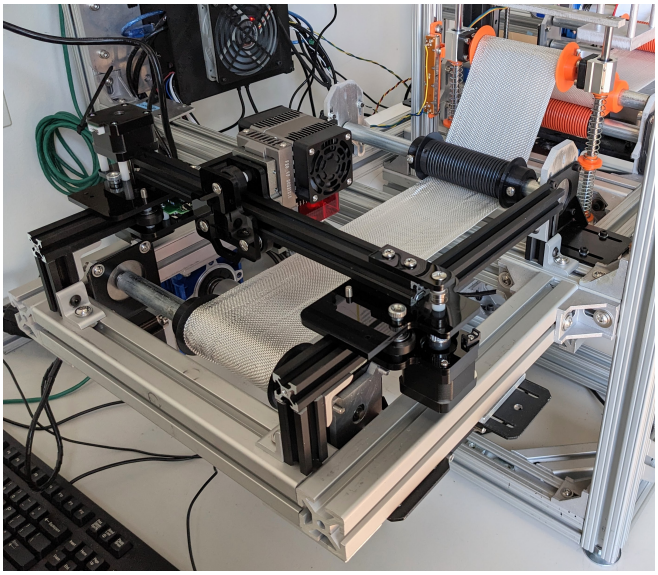

Figure 2.5: Laser Cutting Mechanism

There is also a second laser cutter, not mounted on the printer, that can be used to cut the fiber belt. This is a CO2 laser called OMTECH K40, rated for 40W. A C02 laser cutter is heavier duty and requires more materials. The CO2 laser is an electrically powered gas laser that has greater flexibility with its power strength. However this system is much larger. It comes with its own enclosure, mirror system, and cooling unit for the laser. This system requires two pieces of software called LaserDRW and INKSCAPE. One issue with LaserDRW is that it does not permit the SOLIDWORKS export file for laser cutting. This is solved by using INKSCAPE. This is another editing software that can create the .wmf file that LaserDRW takes. Once the SOLIDWORKS file is uploaded to INKSCAPE, the text can be deleted and the sizing adjusted before exporting as an .wmf file. Once exported to LaserDRW, the custom design can be cut by the K40 laser. The exact steps to go from SOLIDWORKS to LaserDRW can be found in Section 5.3.

2.4. Belt-Drive System

The belt-drive system uses a fiberglass woven belt. It is rolled up on the outtake roller and initially feeds through the first set of pinch rollers, then reaches across the vat and into the second set of pinch rollers, then under the tension roller and back up onto the intake roller where it will collect. Figure 2.6 is a full, top down overview of the belt-drive system, from the outtake roller on the left to the intake roller on the right.

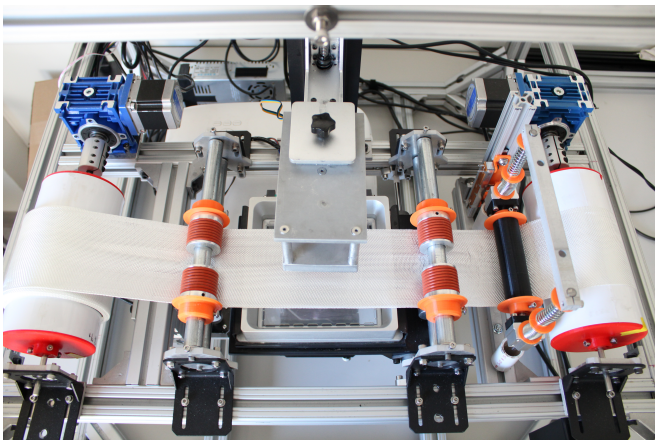

Figure 2.6: Belt Drive System

2.4.1. Tensioning

The purpose of the belt tensioning mechanism is to maintain constant tension across the entire belt. This is key to the function of the printer as for the new layer of fiberglass to be positioned correctly while being added to the print, the tension must remain close to constant. In order to do this properly, the design of the mechanism includes a roller and belt guides to keep the belt laying flat and in line with the print bed. The roller is able to freely spin so that the belt can freely move through the mechanism. The roller and guides are attached to two guide rods such that the roller can freely move up and down the guides based on the tension of the belt.

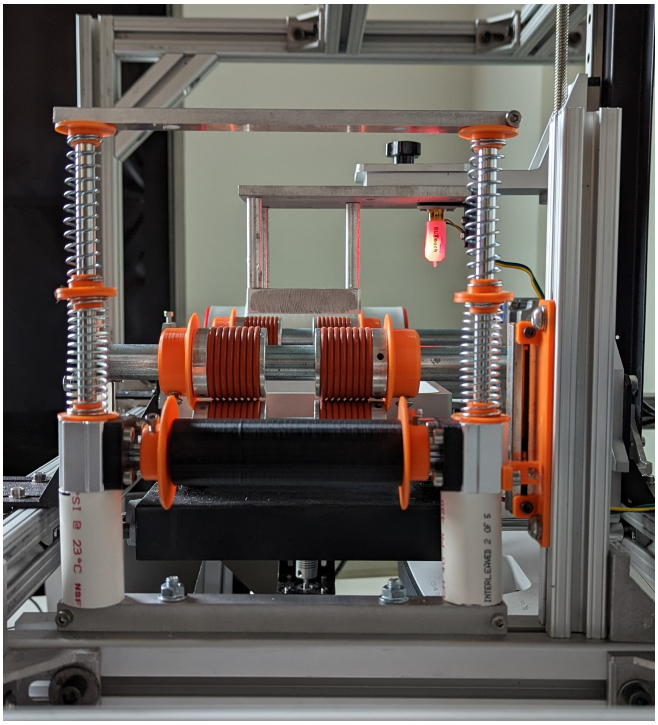

Figure 2.7: Tensioning Mechanism

Springs were added to the design so that the belt would stay in tension even if some slack was added due to the nature of the belt system. Then the printer can measure the distance traveled with a sensor, and using a Proportional, Integral, Derivative (PID) program we can automatically adjust the speed of the motors at either end of the belt system to keep the belt at an acceptable tension.

Several processes of the composite printer require precise measurements which become impacted due to fluctuating tension levels. Such functions include precise laser cutter cutouts for the printing layer and punching out this layer into the resin while printing the requested layer. A sliding potentiometer is connected to the tensioning mechanism via a clip that is attached to one of the springs. The back of the potentiometer is connected to a separate control board, SKR V2. The setup is shown in Figure 2.8 below.

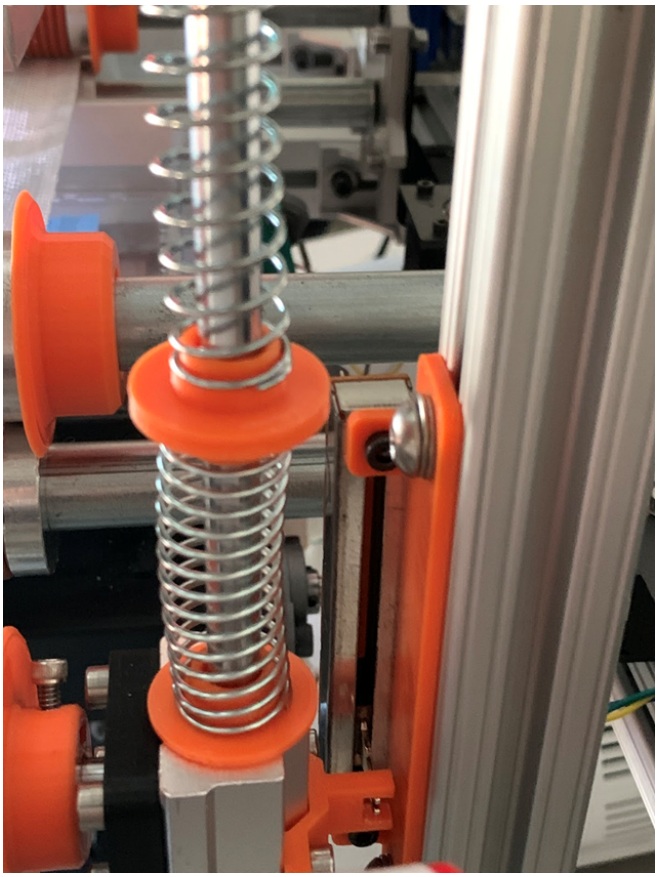

Figure 2.8: Tension Monitoring

The potentiometer is a variable resistor that helps determine the position of an object. The potentiometer in use has a voltage range of 0V - 5V, with the default positioning voltage being 2.5V. As the tensioning mechanism begins moving, the potentiometer will either increase its voltage if the belt becomes too taut, or will decrease its voltage if the belt becomes too loose. Depending on the position of the potentiometer, a signal will be sent to the control board which will convert the signal by using Analog to Digital Conversion (ADC). The control board will then process the digital signal and reposition the motors accordingly. The motor control will adjust the speed and direction of the motors until the suitable data value range for proper tension is returned. While the motors are adjusted, the belt system will then reposition the belt to the proper tautness. In turn, the tensioning system will return to its normal state and reposition the potentiometer to its default 2.5V position.

2.4.2. Belt Registration

The belt registration is a method used by the printer in order to determine the current position of the fiberglass belt and shift the belt into the correct position in order to print. It acts as a safety method in order to ensure that the cutout used in the printing layer is in the right position before being cut out. This system consists of a laser cutter, an optical sensor, the fiberglass belt, and a control board. The optical sensor used includes a beam between two heads that sends a signal when either the beam is broken or is re-established.

Belt registration starts when the on-printer laser cutter cuts out a small rectangle on the bottom left corner of the fiberglass belt in addition to the cutout design of the printing layer. As the belt is being fed through the optical sensor, the sensor will detect this mark or lack of belt. The sensor will then send a signal to a control board that will stop the fiberglass belt. Once stopped, the belt will then be repositioned to be above the build plate. This is done through the control board. The set distance from the middle of the build plate to the current position of the cutout is the same distance as from the registration mark to the middle of the printing area that the cutout is in. This distance will be demonstrated by a value ”x”. The control board will control the motors to position the belt

”x” distance forward. This distance will allow the cutout to then be centered in the build plate and ready for the printer to print out the layer.

2.5. Electronics

The electronics on the printer are enclosed by a 3D printed case, with a Noctua NF-P12 fan on the front blowing air in to cool all of the components (especially the motor drivers, which often get hot). The enclosure can be found at the Printables website. There are openings on the sides for wires to come in as well as for airflow. Since the 120mm Noctua fan is rated for 12V, there is a buck converter inside the enclosure to step down from 24V to 12V. The figure below shows the enclosure.

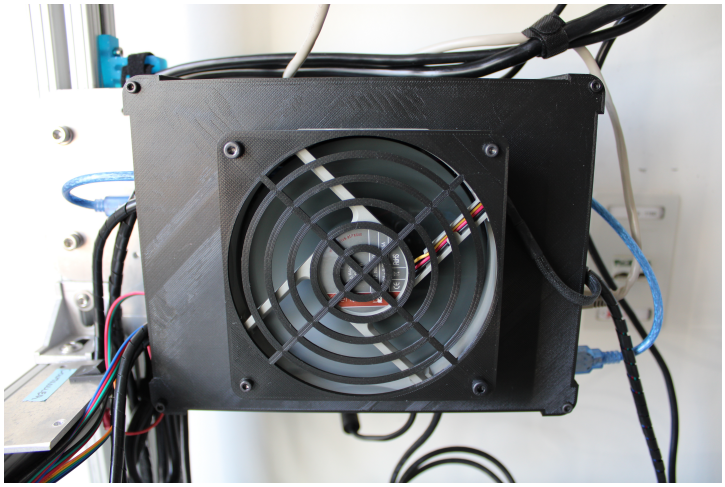

Figure 2.9: Electronics Enclosure

Inside the enclosure, there are 2 main boards: the Raspberry Pi 4B, which runs the control interface, and the SKR V2 which runs the firmware for the printer. Figure 2.10 shows all of the electronics inside the enclosure.

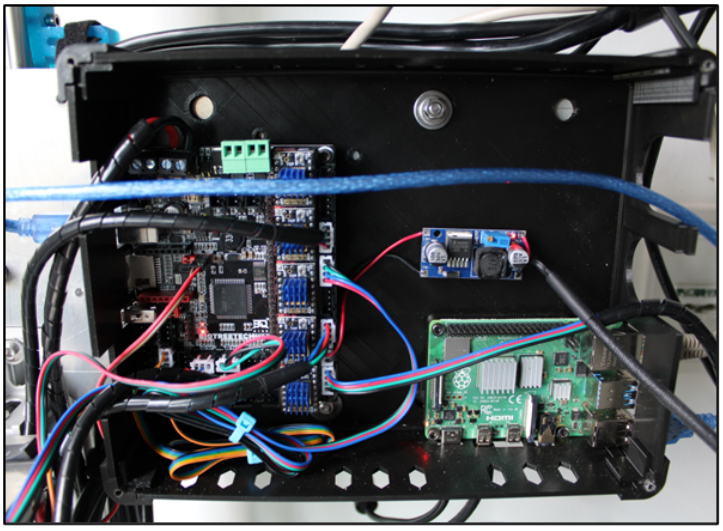

Figure 2.10: Electronics Boards

2.5.1. Raspberry Pi

The printer utilizes a Raspberry Pi 4B to run the NanoDLP server. The image for this server is sourced directly from the NanoDLP website. The simple installation is used, which involves downloading the image file and flashing to the SD card. This image file includes the Raspberry Pi OS with NanoDLP already installed, however it does not include a desktop, just the command line, which is why there is a second Raspberry Pi. The Raspberry Pi 4B is connected to the router to access NanoDLP from its network. The router SSID is ”Custom 3D Printer” and the password is ”pstanzione”.

There is a Raspberry Pi 3B+ that runs a Raspberry Pi OS desktop that is plugged into the Lenovo ThinkVision monitor with keyboard and mouse. It is mounted to the back of the monitor, and the ethernet port is connected to the router. The purpose of this Pi is to act as a desktop computer for the printer, for convenient access to NanoDLP without a separate computer.

2.5.2. BTT SKR V2

The BigTreeTech (BTT) SKR V2 is the control board that runs the firmware for the printer. It is where all the motors, sensors, and other peripherals connect to. This board requires a 12/24V power supply (this printer uses a 24V power supply). Refer to Figure 2.11 for a pinout diagram of the board, with all necessary connections color coded.

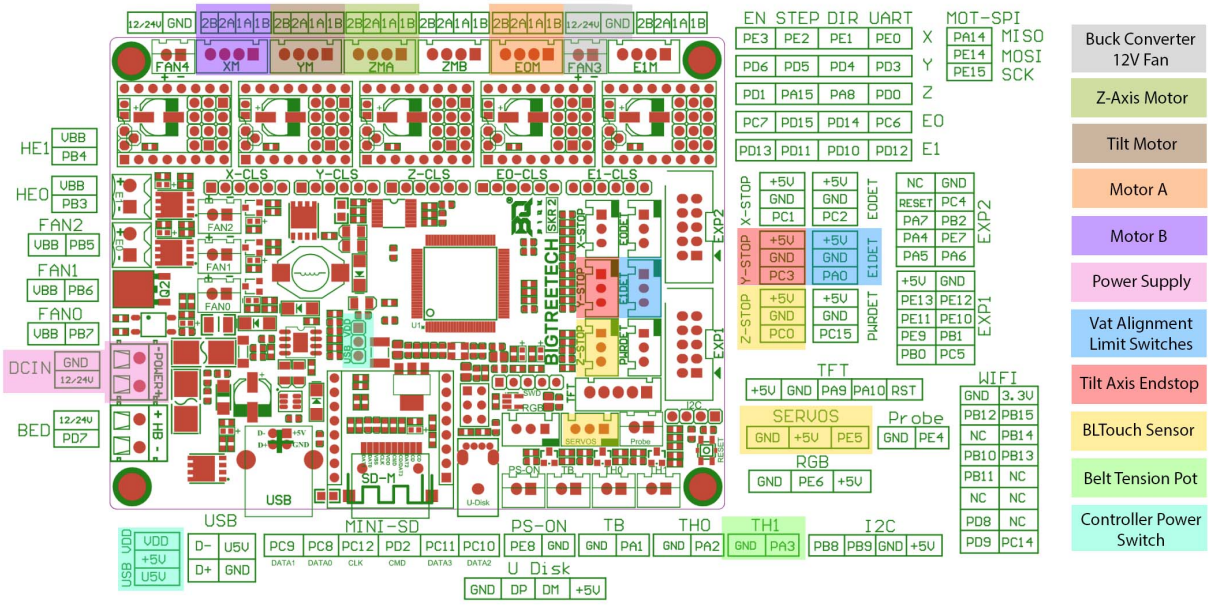

Figure 2.11: SKR V2 Pinout Diagram

2.5.3. Auto-Tensioner

A PCB was originally designed through DipTrace to support the active tensioning system. The PCB directly connects to the stepper motor on the printer to control its speed and position. Components included a resistor, capacitor, the TMC2209 driver mount, motor pin headers, and junctions to have to ability to switch between 5V and 3.3V.

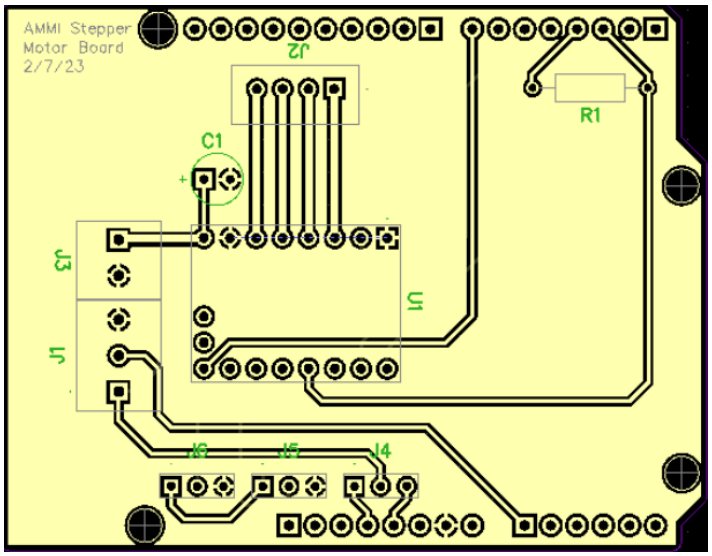

Figure 2.12: Auto-Tensioner PCB

An Arduino board with a custom stepper motor control library was used in combination with the PCB. This library was able to control the speed and direction of the motor based on the voltage of the potentiometer. Three ranges were made based on the voltages to determine the tautness level. Whenever the voltage dropped from the mid-

range, the Arduino code told the motor to run or stop along with accelerating or decelerating. Eventually, the Arduino was able to create the feedback loop to keep the system tensioned.

2.5.4. LCD / HMI Interface

A Nextion NX4832T035 LCD display was implemented on the printer for a user friendly approach for operation. The LCD display is 3.5'' that supports NanoDLP. The LCD is mounted on the front left of the printer right below the belt tensioner. This screen is a convenient means of operating the printer through NanoDLP and as well as to see the status of the print. The LCD is connected to the GPIO pins on the Raspberry Pi 4B, specifically the the 5V, GND, and GPIO15 (UART RXD) pins. In the machine settings of NanoDLP, the Nextion Display Port Address is set to /dev/ttyS0.

Figure 2.13: LCD Interface

NanoDLP created their own display package for the Nextion board. The download can be found on their website. In order to install the package, The board must be powered by a 5V connection and then flashed with the files by a micro-SD card. The LCD is touchscreen and displays features such as printing time and layer number in addition to features for auto-homing the axis, temperature gauges, and projector options. Instead of connecting a laptop to the printer or using the monitor, this LCD allows the user to interact with the printer with only their finger.

2.6. Marlin Firmware

Marlin firmware is a open-source firmware available for 3D printers, generally used for FDM printers. Marlin is heavily customized and repurposed for composite DLP printing. Most of the process in configurating Marlin is commenting out features that are not needed and defining features that are. Other more complex configurations required creating new code, however this is avoided as much as possible by finding ways to manipulate the original Marlin code enough to accomplish the goal that is intended to be completed.

The full list of changes in the firmware for the purposes of this project can be found at the repository, linked here.

2.7. NanoDLP

NanoDLP is a open-source control interface and slicing software used for DLP and SLA (Stereolithography) printing technologies. There are various settings within the platform that need to be set up, from resin profiles to the display engine, before attempting to print. It also utilizes python scripts to cut the proper shapes of layers from the laser cutter.

NanoDLP can be accessed by connecting to the network router (SSID and password found in section 2.5.1).

2.7.1. Resin Profiles

Most of the resin profile settings can remain default. The most important changes for this printer is the after-layer GCODE, under the After-Layer Exposure section in the resin profile settings, and the Normal Layer Cure Time, under the Normal Layer Exposure section.

In the After-Layer Exposure section, the Code After Each Layer field is modified to include GCODE that allows tilting the tray down after each layer is cured, releasing the build platform off of the belt, and moving the belt forward. This code can be found below.

Listing 2.1: Code After Each Layer

```
G0 Y-20 ;                                          Tilt down the build vat, value of 20mm
G1 Y20 Z[[ZLiftDistance]] ;                        Lift build plate and zero the build vat
[[WaitForDoneMessage]] ;                           Wait for process to complete
G28 X ;                                            Advance belt to next registration mark
M206 X+10 ;                                        Set belt axis offset for next movement
[[Delay 1.0]] ;                                    Wait 1 second
G0 Z-{[[ZLiftDistance]]-[[LayerThickness]]} ;      Move build plate down into build vat
```

Under the Normal Layer Exposure, the field labeled Normal Layer Cure Time is set to 17.5 seconds. This means that each layer will cure for 17.5 seconds before it proceeds to the next layer. The normal layer thickness is set to 200 μm.

The following sections describe the different resin profiles that currently exist for both experimental and production stages of this printer.

1A Manual Fiber Placement

The purpose of this resin profile is to allow the user to manually place layers of fiberglass in between a specified number of layers of resin. The way it works is that in the after-layer procedure, it will run a JavaScript code to determine whether the layer that was just printed will need a fiber layer. If it does, then it will raise the build platform (Z axis) and then pause the print. The user then can manually stick the fiber layer underneath the last layer that was just cured, under the build platform. Once this is done, the print can be resumed and it will continue to print until the next fiber layer is ready to be inserted. The after-layer code (which includes the JavaScript code, inside of the [JS][/JS] brackets) can be found below:

Listing 2.2: 1A Manual Fiber Placement, After-Layer

```
1  G0 Y-20
2  G1 Y20 Z[[ZLiftDistance]]
3  [[WaitForDoneMessage]]
4  [[Delay 1.0]]
5  [JS]
6  if ((([[LayerNumber]]+1) % 2 == 0) {
7  output = "G91\nG0 Z75\n[[Pause]]\nG0 Z-75\nG0 Z-{[[ZLiftDistance]]-[[LayerThickness]]}\n[[
       WaitForDoneMessage]]";
8  }
9  else {
10 output = "\nG0 Z-{[[ZLiftDistance]]-[[LayerThickness]]}";
11 }
12 [/JS]
```

In this case, this code indicates a pause every other layer of resin. To change the frequency of fiber layer placement, change the ”2” on line 6 to the desired number.

1A Python Testing

The purpose of this resin profile to test the automated process of laser cutting and placement of fiber during printing. There are two python scripts that execute at the start of the print and in the after-layer procedures. This after-layer code can be found below. To find the start of print GCODE procedure, refer to Listing 2.4: Start Print Code.

Listing 2.3: 1A Python Testing, After-Layer

```
G0 Y-20
G1 Y20 Z[[ZLiftDistance]]
[[WaitForDoneMessage]]
G90
[[ExecReturn python3 /home/pi/printer/Scripts/afterLayer.py [[LayerNumber]]]]
G91
[[Delay 2.0]]
G0 Z-{[[ZLiftDistance]]-[[LayerThickness]]}
```

To learn more about the python scripts, refer to Section 4.5: System Integration.

2.7.2. Machine Settings

In NanoDLP's Machine Settings, various specifications of the printer are set, as below.

Display Engine

The resolution is set to 1920 x 1080, as per the specifications of the projector. For the fields labeled "X/Y Resolution" and "Y Resolution", refer to Section 3.2.1. These fields essentially define the pixel dimensions, and will need to be calibrated. The baud rate is set to 9600. All other fields are left as default values.

Axis/Movement

In this section, there are various settings that need to be adjusted. The movement positioning must be set to relative. Additionally, there is a field labeled "Shield Axis Direction" which specifies which way the Z-axis will move for position zero, and it must be set to "Zero at bottom". The start of print code (will initiate after a print job is submitted to the printer) is as follows:

Listing 2.4: Start Print Code

```
G90 ;           Absolute Positioning
G0 Y0 ;         Ensure Tilt Axis position of zero
M226 P0 S0 ;    Ensure Build Vat is secured in place
G0 Z0 ;         Move Z-Axis (build platform) to position zero
G91 ;           Relative Positioning
```

Another field to fill in is called "Resume Print Code" which is initiated once a paused print is resumed. This GCODE can be found below:

Listing 2.5: Resume Print Code

```
G91 ;           Relative Positioning
M226 P0 S0 ;    Ensure Build Vat is secured in place
G1 Z-75 ;       Move build platform back down into vat
```

The stop print code (after a print job is finished) is as follows:

Listing 2.6: Print Stop Code

```
G91 ;           Relative Positioning
G1 Z75 ;        Move build platform up
```

All other settings in this section remain default, and are overridden by the Marlin firmware.

RAMPS/Controller

The only field that needs to be changed in this section is the baud rate. In Marlin Firmware, this is defined as 115200. The field labeled "Bootup Code" can be found in Appendix A. The fields labeled "Start of Print Code", "Resume Print Code", and "Print Stop Code" can be found in Listings 2.4, 2.5, and 2.6, respectively, found in the subsection above.

Code/GCODE

In this section, the code is already defined in this document. For "Bootup Code", refer to Appendix A. For "Start of Print Code", refer to Listing 2.4. For "Resume Print Code", refer to Listing 2.5. For "Print Stop Code", refer to Listing 2.6. For "Manual Movement Code Template", refer to Listing 2.7 below.

Listing 2.7: Manual Movement Code Template

```
G1 Z[[Position]]
```

Hardware
The field ”Printer Type” is set to Projector/LCD. ”Shield Use” is USB/Serial. ”Wait GPIO” is disabled. ”Stop/Shutdown Physical GPIO” is disabled. ”Stepper Driver Fault Detection GPIO” is disabled. ”Fault GPIO Default State” is set to Low.

HMI (Human-Machine Interface)
For the Nextion display interface, the display port address is set to: /dev/ttyS0

Print Quality
None of the fields in this section are changed.

Slicing
The ”Horizontal Resolution” is set to 1920, and the ”Vertical Resolution” is set to 1080. The ”X/Y Resolution” is set to 83.8 μm, and the ”Y Resolution” is set to 83.8 μm, after calibrating using the instructions from 3.2.1.

Network
The ”TCP Port to Listen” is set to 80.

Camera
None of the fields in this section are changed.

Appearance
None of the fields in this section are changed.

Shutter
None of the fields in this section are changed.

3

Calibration Procedures

3.1. Tilt & Z Axis

The purposes of this calibration process is to align the two linear axes, tilt and Z axes. A video procedure can be found here.

3.1.1. Tilt Lead Screw

Ensure that the lead screw is fully seated in the tilt motor coupler. It should be sitting up right against the motor shaft. Once it is there, tighten the set screws.

3.1.2. Vat Holder

Ensure the springed leveling screws for vat holder are set with equal compression, parallel to tilt platform.

1. Use a spacer on each corner and check for equal compression on all sides.
2. If any of the corners need to be adjusted, use a nut driver on the nuts beneath the vat holder, and check the compression again with your spacer.

3.1.3. Z Platform Level

Check the Z platform level by:

1. Place a spacer of known size on top of the build platform
2. Ensure that the vat and the build platform are the same level
3. Manually bring Y tilt axis to visually level
4. Lower Z axis to just above spacer
5. Use a gauge block, shim stock, or a piece of paper of known thickness in between the spacer and build platform. There should be enough space for this to slip through but you should also feel some friction or resistance on the material.
6. Now test the left and right side of the spacer for equal resistance. Adjust the tilt axis to fix the uneven side.
7. Bring Z axis down slowly, incrementally, while periodically testing the fitment of the gauge material.
8. Once the highest side of the spacer is in contact with the gauge material and friction between the 3 parts is felt while moving the gauge material, move the Y tilt axis as needed to raise the low side of the spacer.
9. Adjust Z axis as needed to avoid crashing the build platform while raising the tilt axis.
10. Move back and forth between adjusting Tilt and Z axis and testing gauge material fitment until equal friction and spacing is reached across the X dimension of the build platform.
11. If the Y dimension is uneven, then this indicates a problem with the Z axis, please note that this is not a normal part of the calibration procedure and must only be done if absolutely necessary. Adjust the top 4040 bar holding down the Z axis by loosening the screws and move the bar along the top.
12. Once this is reached, it is known that the build platform and vat surface are parallel. Zero the tilt axis at this position by sending “G92 Y0”.

13. Add the spacer and gauge material thickness together and zero the Z axis using the spacer(s) as the offset from the current position (ex: “G92 Z40.1” using 40mm spacer and 0.1mm gauge material).

3.1.4. Water Vat Test

Utilize the water vat test to determine if the Z axis comes into contact with the vat surface evenly and completely without compressing springs.

3.2. Optics Calibration

This calibration procedure involves fine-tuning the projector and optics. A video procedure can be found here.

3.2.1. Pixel Dimensions

Define the size of the pixels in NanoDLP's display engine (setup → machine settings → display engine)

1. Zoom in the projector as much as possible while focusing. *Note: you cannot zoom in all the way because, so you must back off the zoom a little bit while still retaining focus.
2. Then, the projection image on the Y-axis must be measured, and then divided by 1080 pixels. That is the size of the pixels in microns. That value must be entered in the fields labeled "X/Y Resolution" and "Y Resolution", in NanoDLP's Display Engine settings.

* Both fields need to be the same because the pixels are square. If the values are not the same, it will result in a squished projection where a square image may come out as a rectangle.

3. Additionally, the projector resolution needs to be entered in the fields labeled "Horizontal Resolution" and "Vertical Resolution".

3.2.2. Calibration Grid

The calibration grid is a useful tool to detect possible issues with projected images. In NanoDLP, this grid can be displayed through setup -> projector calibration -> display dynamic calibration. Figure 3.1 shows what the calibration grid will look like.

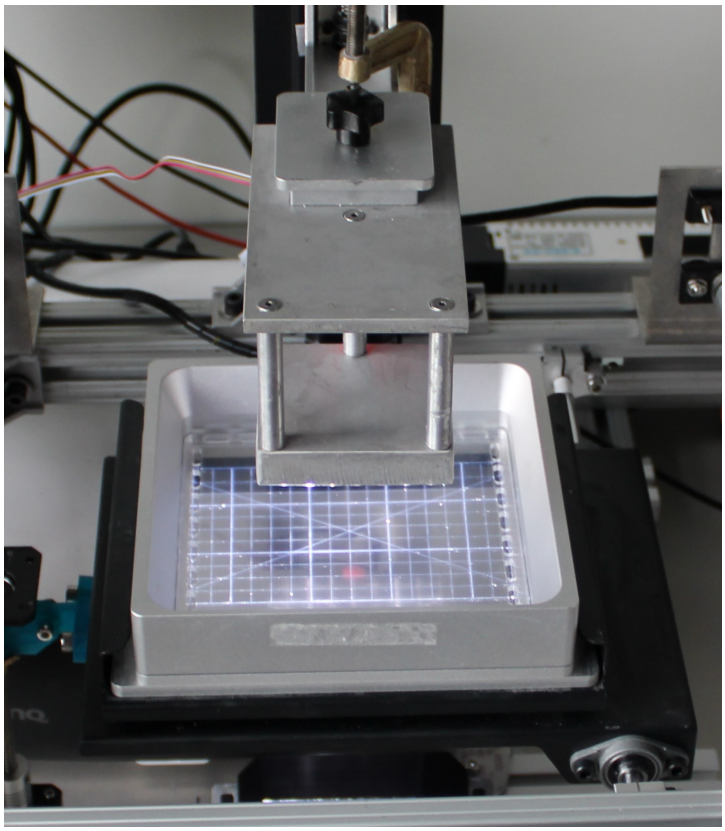

Figure 3.1: Calibration Grid

3.2.3. Projector Angle

To calibrate the projector angle, the angle of projection must be as horizontal as possible.

1. Remove the mirror mount setup, and attach a long 4040 bar, centered in reference to the lens of the projector. The bar should be parallel and perpendicular to the rest of the frame.
2. The calibration grid must be displayed, from NanoDLP. On the 4040 bar, a square piece of acrylic must be aligned as close to the projector. If the grid is not visible on the acrylic, the projector must be refocused to get as clear of an image as possible. The center of the projection is a cross. The marking on the acrylic must be aligned with the center of the projection, at the X.

3. The acrylic must be moved back 100 cm on the bar, and the projector must be refocused again. If the X does not line up with the edge of the tape, then the angle of the projector must be adjusted.
4. If the X is too high or too low (> 5mm difference) from the edge of the tape, then the angle of the projector must be further adjusted with the rear feet of the projector. This readjustment must repeat until proper alignment is achieved.
5. After the projector angle is calibrated, the mirror mount must be put back into place, with a proper placement of the mirror at 45 degrees from the horizontal.

3.2.4. Mirror Mount

There are three fine adjustment knobs on the mirror mount, which can be used to produce a square image in the vat. The knobs can be adjusted so that the grid lines are parallel to the edges of the vat. Figure 3.2 shows the three fine-adjustment knobs.

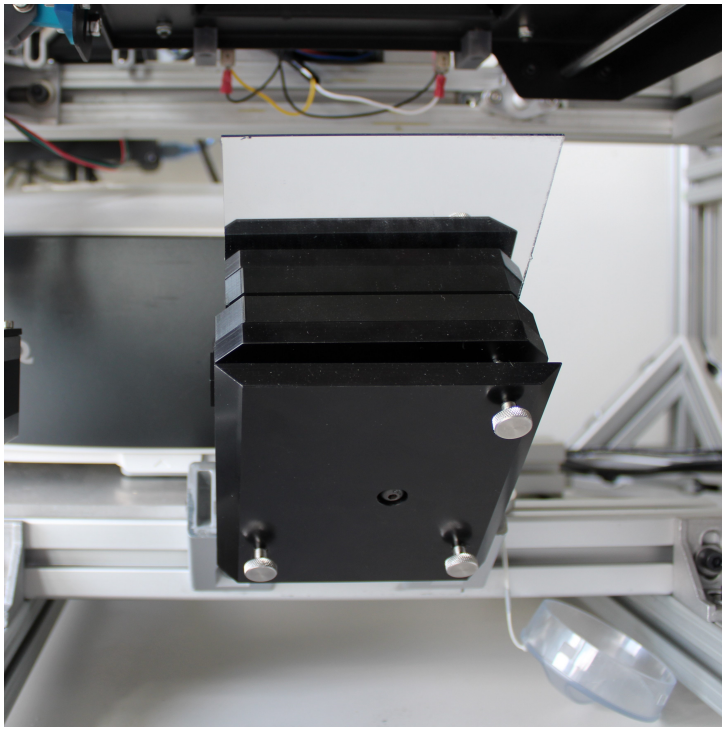

Figure 3.2: Mirror Mount

float

4

Experimental Processes

4.1. Print Properties

Mechanical and physical properties of the resins used have a vital role in the operation of the composite printer, primarily due to resin bonding and penetration into the glass fiber belt. For this reason, multiple resin properties will be selected and tested for in order to improve the final mechanical properties of a composite part. Preexisting resins were first utilized in order to set a baseline for DLP resin printing, after which point specific properties can be developed and tested for.

4.1.1. Resin Synthesis

Resins synthesized in the chemical engineering lab were developed with the following procedure.

1. Note weight measurements of monomers/curing agents in lab book
2. Collect monomer bottles in one place on lab bench
3. Set up near weighing balance
4. Have a spatula, glass beakers, 50mL capacity, to use to pour monomers
5. Use a ThinkyMix container to prepare formulation
6. Place ThinkyMix container on a weighing balance
7. Tare before each chemical
8. Pour first chemical into glass beaker
9. Pour chemical slowly from beaker into the ThinkyMix container to the required weight
10. Tare again
11. Repeat steps 8-10 until all chemicals are added
12. For TPO use the weighing balance with increased precision
13. Take a weigh boat and tare it on the balance
14. Add required TPO from bottle (toxic handle with extra care)
15. Add weighed TPO to ThinkyMix container
16. Measure gross weight of ThinkyMix container and set counterweight in ThinkyMix
17. Add ThinkyMix container to the ThinkyMix mixer
18. Set mixing parameters, Mixing 10 minutes, Defoaming 5 minutes and run
19. Repeat step 18
20. Take 3 FTIR samples of liquid resin

4.2. Dimensional Accuracy

Dimensional accuracy for printed parts is vital for precise and repeatable curing, which will be required for complex parts or those with critical outside dimensions.

4.2.1. Power Density

Power density consistency across the build platform is vital to appropriately cure all portions of a print and reduce warping or dimensional issues post-cure. Power density measurements in W/cm^2 were taken with a radiometer in order to determine the intensity of light being projected to each part of the build platform. These measurements included grayscale values from 255-5 in increments of 10 at the center of the build platform. Results are shown and graphed in Figure 4.1 below.

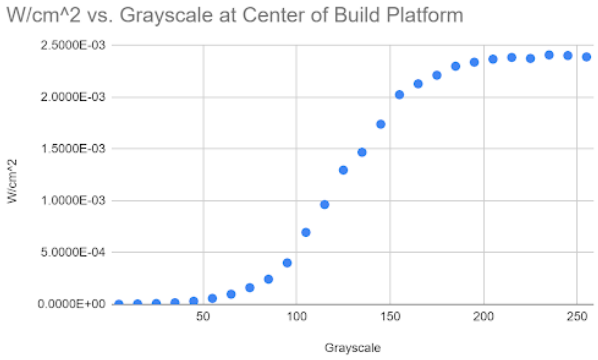

Figure 4.1: Light Intensity Control Curve Measured at Center of Build Platform

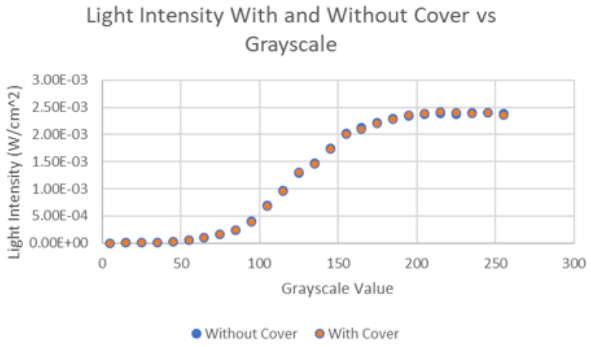

Figure 4.2: Light Intensity Measurement With and Without Shield

Following that experiment grayscale measurements were then taken with a cover over the build platform. This was done to see if the ambient light of the room affects the power density of the build platform, which affects the rate that resin will cure. It was found that at all but the lowest grayscale values (15 and 5) the percent difference between the covered and uncovered build platform was within +/- 2%. The comparison of light intensity with and without the build platform shielded from surrounding room light can be seen in Figure 4.2 above.

Power Density measurements in W/cm^2 were also taken across four DMA bars projected onto the build platform at seven different positions one cm apart, which can be seen in Figure 4.3 below.

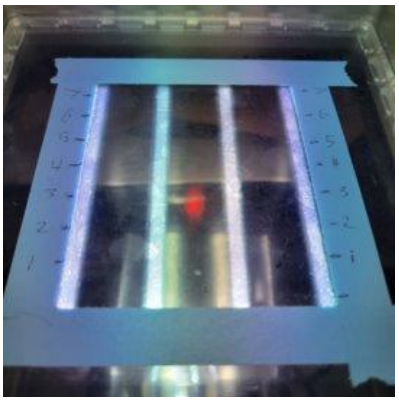

Figure 4.3: Measurement Point Grid Across Build Platform

This was done to compare light intensity across the positions of the four different DMA bars and see if there was a significant loss of power or light intensity that might affect cure times and material properties. This was done at three different grayscale values, 255, 130, and 5. It was found that at higher grayscale values such as 255 and 130 there was a difference of less than 10% of the value projected at the origin across all position of the build platform, however at the 5 grayscale value there was a larger variation of close to 20% of the origin value. Heat maps of the light intensity lost can be seen for each grayscale value below in Figures 4.4, 4.5, and 4.6 below.

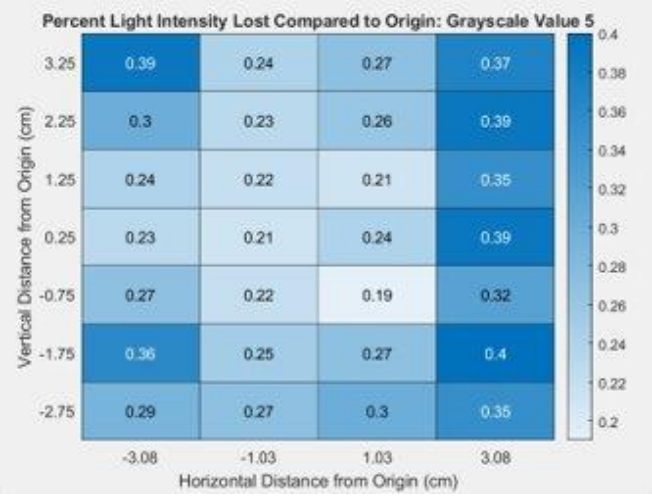

Figure 4.4: Heat Map of Light Intensity Across Build Platform Area for Grayscale Value of 5

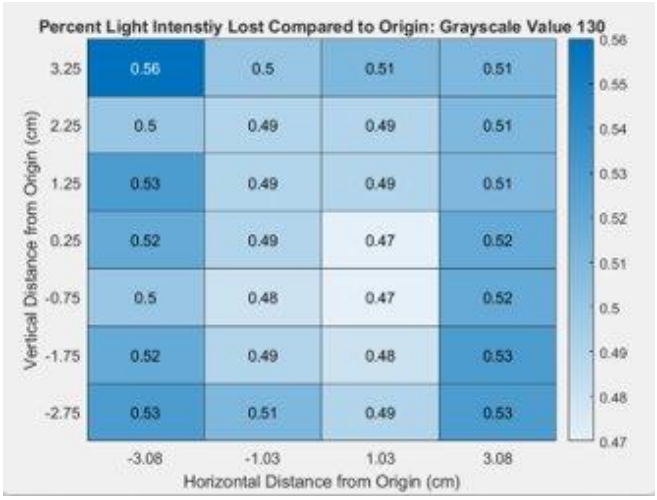

Figure 4.5: Heat Map of Light Intensity Across Build Platform Area for Grayscale Value of 130

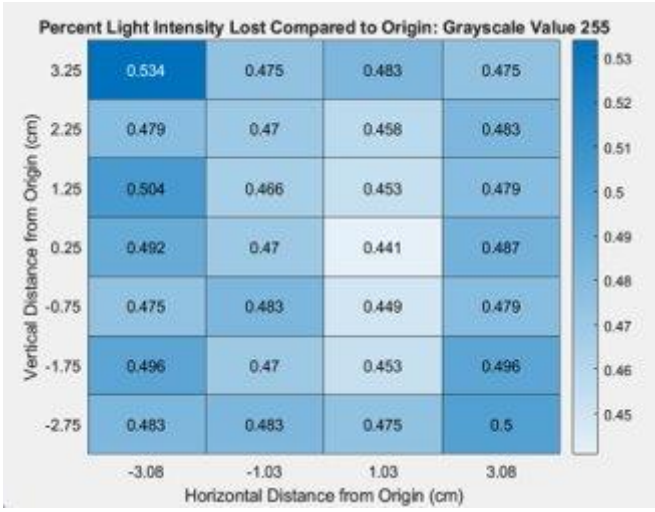

Figure 4.6: Heat Map of Light Intensity Across Build Platform Area for Grayscale Value of 255

Dimensional analysis was performed on DMA bars printed using the DA-2 resin formulation early on to practice and test the dimensional accuracy of the printer. Various cure times were tested ranging from 10-20 seconds with the results seen below. The height of the bars was not measured due to the supports adding extraneous material to the bars that needs to be ground down. The width of the bars was the shortest dimension with a desired 2.5 mm and has the greatest error. The length of the bars was the longest dimension with a desired length of 35 mm and has the smallest error. It is worth noting that the width had a slightly larger dimension than desired and the length had a slightly smaller dimension than desired in all cases. The error in the X and Y dimension across various tested cure times can be seen in Table 4.1 below.

Cure Time (seconds)	**Average % Error Width**	**Average % Error Length** 10
-4.40	-2.33 15	2.00
-1.53 17.5	3.80	-1.20 20
4.50	-0.96	

Table 4.1: Comparison of Dimensional Error vs. Cure Time

4.3. Working Curve Experiments

Three different working curves studies were performed on DA-2 resin with varying concentrations of TPO (0.1wt%, 0.5wt%, 1wt%). In order to do the working curve calculations, one must have a print file with multiple prints per layer (the three disc file at 255 grayscale value was used for these calculations), enough resin of the desired concentration to fill the build platform and make numerous small prints (about 150 mL of resin was used in each of these tests), a plastic scraper to remove the delicate prints and stir the resin vat after each print, and finally a micrometer to take accurate measurements on a scale from microns to several millimeters. To perform a working curve test, this can be found in Section 5.4.

4.3.1. Working Curve 1: 0.1 wt% TPO

Table 4.2 is the working curve data using 0.1wt% TPO. Additional data including the intercept, slope, and critical energy of the working curve is detailed in Table 4.3. Finally, Figure 4.7 is the final working curve with 0.1wt% TPO.

Exposure Time (s)	Dose (mJ/cm^2)	Dose (J/cm^2)	ln(dose)	Measured Thickne	
			1	2	3 [5pt] hei
0.1617	5.09		0.742	[5pt] height65	168
5.12	1.036	1.294	1.123 [5pt] height70	181	0.18
1.315	1.475	1.38 [5pt] height75	194	0.194	5.27
1.619	1.269 [5pt] height80	207	0.207	5.33	2.19
2.002 [5pt] height90	233	0.2328	5.45	2.676	3.00
259	0.2587	5.56	4.097	4.065	3.34 [5pt] h
0.2846	5.65	4.131	4.271	4.045 [5pt] height	

Table 4.2: Working Curve with 0.1wt% TPO

Intercept	-33.04000254	-31.54660221	-28.99297454 [5pt] heightDp (s
6.601014041	6.358935961	5.825571005 [5pt] heightCritical Energy (Ec)	149.200638
142.7346858	145.0163973 [5pt] height		

Table 4.3: Working Curve with 0.1wt% TPO: Further Analysis

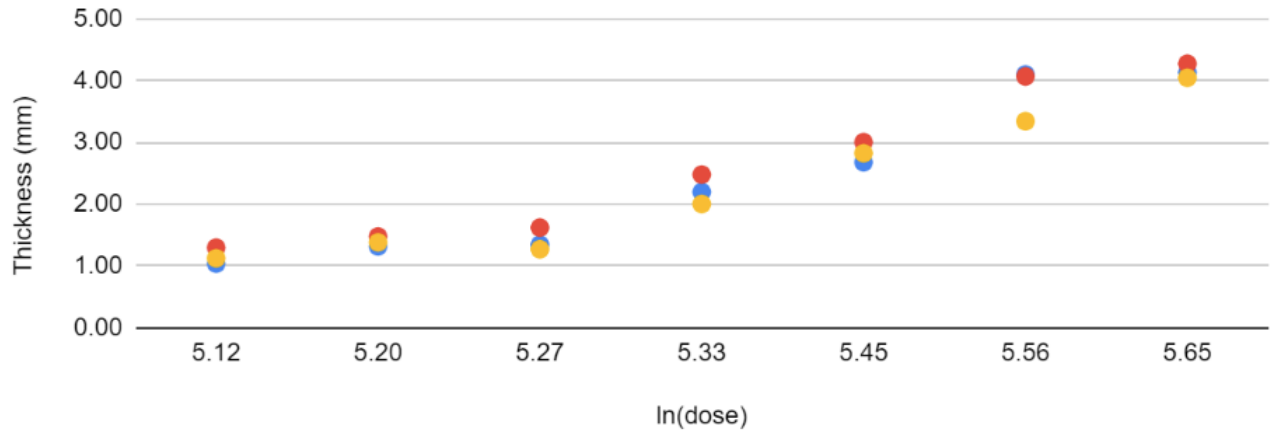

Figure 4.7: Working Curve with 0.1 wt% TPO

4.3.2. Working Curve 2: 0.5 wt% TPO

Table 4.4 is the working curve data using 0.5wt% TPO. Additional data including the intercept, slope, and critical energy of the working curve is detailed in Table 4.5. Finally, Figure 4.8 is the final working curve with 0.1wt% TPO.

Exposure Time (s)	Dose (mJ/cm^2)	Dose (J/cm^2)	ln(dose)	Measured Thickness	
			1	2	3[5pt] he
0.0388	3.66	0.338		[5pt] height17.5	45
3.81	0.375	0.41	0.445[5pt] height20	52	0.05
0.521	0.592	0.509[5pt] height25	65	0.0647	4.1
0.718	0.74[5pt] height30	78	0.0776	4.35	0.95
1.036[5pt] height40	103	0.1035	4.64	1.355	1.4
129	0.1294	4.86	1.766	1.868	1.805[5pt]
0.1552	5.04	2.132	2.32	2.3446[5pt] height80	20
5.33	2.274	2.403	2.524[5pt] height		

Table 4.4: Working Curve with 0.5wt% TPO

Intercept	-4.47736861	-5.11746429	-5.338166169 [5pt] heightDp (sl
1.275241011	1.430911424	1.481360866 [5pt] heightCritical Energy (Ec)	33.48166502
35.74344603	36.7285929 [5pt] height		

Table 4.5: Working Curve with 0.5wt% TPO: Further Analysis

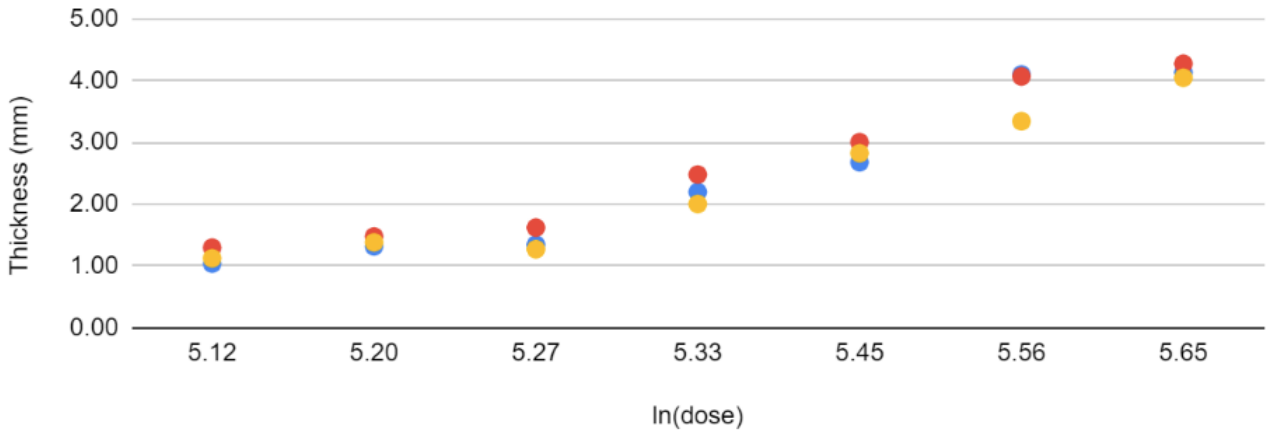

Figure 4.8: Working Curve with 0.5 wt% TPO

4.3.3. Working Curve 3: 1 wt% TPO

Table 4.6 is the working curve data using 1wt% TPO. Additional data including the intercept, slope, and critical energy of the working curve is detailed in Table 4.7. Finally, Figure 4.9 is the final working curve with 1wt% TPO.

Exposure Time (s)	Dose (mJ/cm^2)	Dose (J/cm^2)	ln(dose)	Measured Thickness (mm	
			1	2	3 [5pt] height11
0.0285	3.35	0.323	0.315	[5pt] height12.5	32
3.48	0.326	0.338	[5pt] height15	39	0.0388
0.431	0.493	0.487 [5pt] height20	52	0.0517	3.95
0.664	0.702 [5pt] height25	65	0.0647	4.17	0.855
0.91 [5pt] height30	78	0.0776	4.35	1.01	1.103
103	0.1035	4.64	1.278	1.364	1.397 [5pt] height
0.1294	4.86	1.657	1.78	1.796 [5pt] height	

Table 4.6: Working Curve with 1wt% TPO

Intercept	-2.680994668	-2.935900874	-3.476584905 [5pt] heightDp
0.8630172314	0.9381812149	1.063476418 [5pt] heightCritical Energy (Ec)	22.34352702
22.85920131	26.28703075 [5pt] height		

Table 4.7: Working Curve with 1wt% TPO: Further Analysis

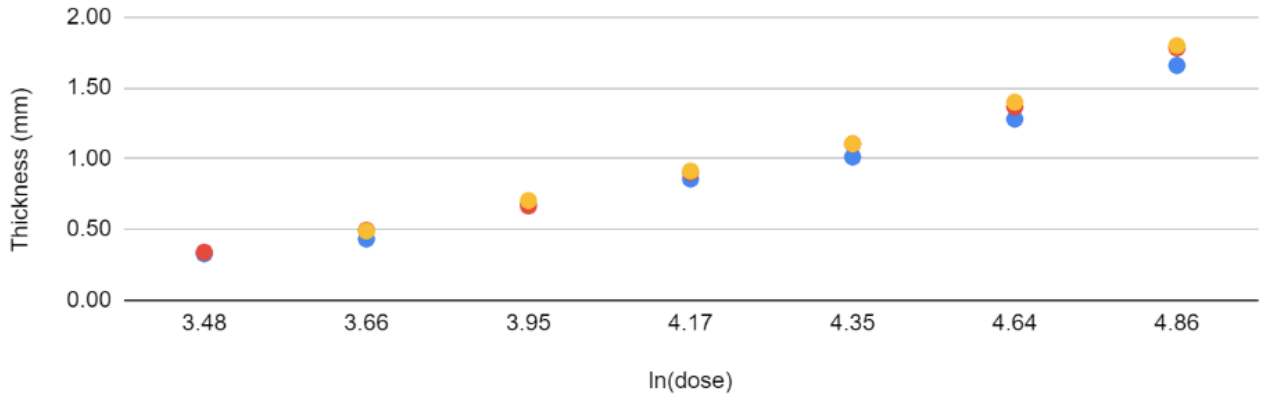

Figure 4.9: Working Curve with 1 wt% TPO

4.4. Grayscale Testing

Using the light intensity measurements taken at each grayscale level and the calculated Ec (critical energy) from the 0.5 wt% working curve study, the minimum amount of time was calculated for gelation at various grayscale values.

Grayscale	Light Intensity (W/cm^2)	Light Intensity (mW/cm^2)	Average E_c from Experiment (mJ/cm^2)	Minimum Exposure Time (s) [20pt]
255				
2.39E-03	2.39E+00	35.32	14.78 [4pt]	2.40E-03
245				
2.40E+00	35.32	14.7 [4pt]	2.41E-03	2.41E+00
235				
35.32	14.66 [4pt]	2.38E-03	2.38E+00	35.32
225				
14.87 [4pt]	2.39E-03	2.39E+00	35.32	14.81 [4pt]
215				205
2.37E-03	2.37E+00	35.32	14.91 [4pt]	2.34E-03
195				
2.34E+00	35.32	15.1 [4pt]	2.30E-03	2.30E+00
185				
35.32	15.36 [4pt]	2.21E-03	2.21E+00	35.32
175				
15.96 [4pt]	2.13E-03	2.13E+00	35.32	16.58 [4pt]
165				155
2.03E-03	2.03E+00	35.32	17.44 [4pt]	1.74E-03
145				
1.74E+00	35.32	20.3 [4pt]	1.47E-03	1.47E+00
135				
35.32	24.04 [4pt]	1.30E-03	1.30E+00	35.32
125				
27.23 [4pt]	9.64E-04	9.64E-01	35.32	36.63 [4pt]
115				105
6.95E-04	6.95E-01	35.32	50.8 [4pt]	4.01E-04
95				
4.01E-01	35.32	88.1 [4pt]	2.43E-04	2.43E-01
85				
35.32	145.1 [4pt]	1.61E-04	1.61E-01	35.32
75				
219.09 [4pt]	9.84E-05	9.84E-02	35.32	358.89 [4pt]
65				55
5.79E-05	5.79E-02	35.32	609.98 [4pt]	3.07E-05
45				
3.07E-02	35.32	1,149.67 [4pt]	1.58E-05	1.58E-02
35				
35.32	2,233.90 [4pt]	8.46E-06	8.46E-03	35.32
25				
4,175.68 [4pt]	4.70E-06	4.70E-03	35.32	7,522.45 [4pt]
15				5
1.22E-06	1.22E-03	35.32	29,068.23 [4pt]	

Table 4.8: Calculated Minimum Gelation Time at Different Grayscale Values

Resin was then tested at these minimum times in order to determine when the minimum gelation time actually appeared experimentally and when the first stable print that could be removed from the resin vat occurred, yielding the results seen in Table 4.9

Grayscale	Average E_c (mJ/cm^2)	Calculated Minimum Time (s)	Experimental Gelation Time (s)	Experimental Stable Formation (s) [5pt]
35.32	15	17	21 [4pt]	35.32
15	19	23 [4pt]	35.32	16
20	22 [4pt]	35.32	24	30
34 [4pt]	35.32	88	109	123 [4pt]

255
215
175
135
95

Table 4.9: Minimum Gelation Time

4.5. System Integration

Attempting to integrate the DLP system and the laser cutter system is a difficult task because they are separate machines. It is important for them two communicate to each other to ensure the composite printing process is fully automated.

The laser cutter is controlled by the same control board as the DLP printer (BTT SKR V2). The laser cutter utilizes the X and B axes of the control board, and the laser itself will be controlled by a PWM connection (fan connection). These axes also have limit switches plugged in to X endstop and E1 endstop, respectively.

The way the two machines work together is through python scripts. NanoDLP allows one to execute python scripts between layers or even before or after a print, or also during bootup. This is done through any of the custom GCODE fields in resin profile settings or machine settings. To execute a python script in NanoDLP, the command(s) [[Exec]] or [[ExecReturn]] can be executed in any of the custom GCODE fields, with the latter allowing the python script to return the output to the NanoDLP RAMPS terminal.

For this printer, there are 2 python scripts that are executed. One is done before the start of the print, called ”start.py”. To execute this python script, the command found below is executed in the before print GCODE process. After ExecReturn, python is called, followed by the directory of the start.py file. Additionally, the number at the end (2 in this case) is the number indicating the frequency at which to cut the layers of fiber. So in this case, after every other layer of resin the laser cutter will cut.

```
[[ExecReturn python /home/pi/printer/Scripts/start.py 2]]
```

The next python script is executed after individual layers. This is called ”afterLayer.py” and its command can be found below. Similar to the previous command, except the [[LayerNumber]] is a variable from NanoDLP (indicating which layer the printer is currently on) and it is passed in to the script.

```
[[ExecReturn python /home/pi/printer/Scripts/afterLayer.py [[LayerNumber]]]]
```

4.5.1. start.py

The purpose of the start.py script is to examine an existing GCODE file with the name ”test.gcode” (found in the same directory as the scripts folder as indicated in the commands). This GCODE file is generated through a FDM slicer that sliced the same part that NanoDLP intends to print. There exists a PrusaSlicer profile for the laser cutter, so it is recommended to use that software when slicing the part. After slicing the part, place the GCODE file in the same directory as the script (/home/pi/printer/Scripts/). This can be done through FTP protocol or using a software like WinSCP.

The script will essentially create a modified GCODE file, called ”output.gcode”. The script will look in ”test.gcode” and find every instance of ”;LAYER_CHANGE” (which indicates layer change) and it will place all of the ”G1” or ”G0” movement commands in between each layer change into the new output.gcode file. Additionally, it will replace the ”Y” axis indicators to ”B” because the Y axis in a traditional Cartesian system is really relabeled as the B axis for the laser cutter. Also, it removes all instances of the extrusion commands. Finally, it will add layer change separators to the output.gcode indicating the layer number and whether that layer needs to be cut by the laser cutter (determined by the frequency number passed into the script). This layer separator flag is used by the afterLayer.py script.

```
import sys
import re

```

```
def generate_output_gcode(file_path, frequency_number):
    output_file_path = 'output.gcode'
    with open(file_path, 'r') as input_file, open(output_file_path, 'w') as output_file:
        layer_number = 0
        for line in input_file:
            if line.startswith(';LAYER_CHANGE'):
                layer_number += 1
                output_file.write(";" + "-" * 15 + "Layer {} (Will Cut: {})".format(layer_number,
                    layer_number % frequency_number == 0) + "-" * 15 + "\n")
            else:
                if line.startswith('G1 X') or line.startswith('G0 X'):
                    x_value = re.search(r'X([-+]?[0-9]*\.?[0-9]+)', line).group(1)
                    y_value = re.search(r'Y([-+]?[0-9]*\.?[0-9]+)', line).group(1)
                    cleaned_line = "G1 X{} B{}\n".format(x_value, y_value.replace("Y", "B"))
                    output_file.write(cleaned_line)

        # Return the frequency_number to the console
        sys.stdout.write("M118 E1 " + "Frequency Number: {}\n".format(frequency_number))

file_path = 'test.gcode'
frequency_number = int(sys.argv[1])  # Access the external variable through command-line argument

generate_output_gcode(file_path, frequency_number)
```

4.5.2. afterLayer.py

The afterLayer.py script will examine the output.gcode file. Since this script runs after every layer, it must return to the NanoDLP terminal the commands for the laser cutter to move and power the laser, which is what the output.gcode is intended to do. In the execution command for this script, the layer number is passed into the script from NanoDLP. The script will match that [[LayerNumber]] with that found in the layer change separators found in the file. If it is found, then it looks whether or not if that layer is flagged for cutting. If it is, then the GCODE that follows (up until the next layer change separator) is sent to the RAMPS terminal of NanoDLP. As a result, the cut layer is the same shape as the layer that will be printed and the belt can be advanced until the fiber layer lines up to the build platform. If the layer is not flagged for cutting, then nothing will occur and the next layer will proceed to print.

```
import sys

def extract_gcode_for_layer(file_path, layer_number):
    with open(file_path, 'r') as input_file:
        layer_found = False
        extract_gcode = False
        current_layer_number = 1
        for line in input_file:
            if line.startswith(';--'):
                if layer_found:
                    break  # Stop extraction if the previous layer was found and processed
                layerSep = line
                layer_info = line.strip().split(' ')
                current_layer_number = int(layer_info[1])
                will_cut = line.find("True")  # Check if the layer is marked for cutting
                if current_layer_number == layer_number and will_cut != -1:
                    layer_found = True
                    extract_gcode = True

            elif extract_gcode:
                sys.stdout.write(line)  # Output the GCODE line

                if line.startswith(';--'):
                    extract_gcode = False  # Stop extraction after the next layer change separator

# Usage example
file_path = 'output.gcode'
layer_number = int(sys.argv[1])  # Access the external variable through command-line argument

extract_gcode_for_layer(file_path, layer_number)
```

5

Operation

5.1. Startup Process

Marlin utilizes startup GCODE that runs when the printer starts. The main goal of the startup GCODE is to home and set offsets of the tilt and Z axes, as well as resetting the position of the rotational axes. It will also ensure that the vat is fully inserted into the holder before doing any tasks. Refer to Appendix A: Startup GCODE to analyze the step-by-step process.

To start the printer, simply press the green button above the red emergency stop, and the startup procedure will begin. The projector may display the bootup screen. Because of this, it is important to start the printer with no resin in the vat to ensure that the resin does not get cured from the bootup process.

5.2. Starting a Print

Note that the current state of the printer does not involve printing with the fiber yet, and this is procedure to print a pure resin part. To start a print, the printer must be on and the bootup sequence must have been completed prior. Then follow the steps below.

1. Once the 3D model is complete and ready to print (must be .stl, .obj, or .slc file type), you can either put it on a USB flash drive and plug it into the desktop Raspberry Pi behind the monitor, or you can remote into NanoDLP interface by logging into the network. The network SSID is ”Custom 3D Printer” and the password is ”pstanzione”.
2. Once in NanoDLP, click on Plates tab.
3. Click the plus symbol ”+” to add a new plate, and select the 3D model. For the profile drop-down, select ”1A BACKUP, DO NOT CHANGE”.
4. Click ”Submit” and let the software slice the 3D model. Once done (may need to refresh the page), click ”Print Now”.

5.3. SOLIDWORKS to LaserDRW

The main issue with using LaserDRW is that it does not natively support exported files from SOLIDWORKS. The workaround for this is found below, which requires an additional software called INKSCAPE.

1. Create a file of the shape that is being cut and save it as a .dxf
2. Upload the dxf into INKSCAPE
3. Delete text boxes and manually adjust page dimensions .1’’ larger than .dxf
4. File -> document properties -> page
5. Save As .wmf or .emf files
6. Go to laserDRW -> file -> open -> .wmf/.emf
7. Adjust scaling to mm and move to the top left of drawing space.
8. Engrave -> Laser type “cutting” -> set time (mm/s) -> starting finished cut.

5.4. Working Curve Test

When performing the working curve experiments on the custom 3D printer:

1. Remove the build platform as a single layer will be printed of various sizes and it would interfere with the disk growth
2. Pour all the resin into the resin vat
3. Change the resin profile of the printer to ensure that the number of burn layers are set to zero and the number of normal layers are set to one at the desired time for this run of the experiment (initial values set to ten seconds, varies depending on the run)
4. Press print to project the layer into the build platform for the desired time
5. If a stable print was formed, use the plastic scraper to remove the discs from the resin vat and measure their layer thickness at the center using the micrometer (these discs will be compressible, especially at lower times, so do not over tighten the micrometer when measuring)
6. Stir the resin vat well with the plastic scraper to distribute partially cured resin away from the print surface
7. Repeat steps 3-7 until about nine different times produced stable prints that could be measured or the resin vat no longer has enough resin for prints at higher times. For the initial few times increase the print time gradually until a stable print is produced, then increase by about five seconds for three intervals, then ten seconds for three intervals, and then twenty seconds for remaining intervals.
8. Enter all measurements into a spreadsheet to create a working curve. The thickness of the discs measured represents the cure depth of the resin at each time. The time can be multiplied by the power density of the printer to obtain the energy dose provided to the resin. When the ln(dose) was plotted versus the layer height of the disc at each respective time interval, the last two parts of the working curve equation could be obtained from the graph: the depth of penetration and the critical energy of the resin from the slope and intercept respectively.

A

Startup GCODE

This GCODE runs when printer is booted up

```
M211 S1;            Software Endstops Enable
M302 S0;            Disable Cold Extrusion Prevention
M104 S0;            Disable Temperature
M119;               Endstop Status
M114;               Return Current Position
G91;                Relative Positioning
G0 Y-10;            Move tilt axis down (to avoid crashing into endstop)
G90;                Absolute Positioning
M120;               Enable Endstops
G28 Y;              Home tilt-axis
M114;               Return Current Position
M121 Y;             Disable hardware endstops for tilt axis
M206 Y-1;           Set home offset for tilt axis
G0 Y0;              Move tilt axis to position zero
M226 P0 S0;         Wait for pin state of vat holder limit switches, ensures vat is inserted
G28 Z;              Home Z-axis (using BLTouch)
M114;               Return current position
M206 Z+3.7;         Set home offset for Z-axis
G0 Z0;              Move Z-axis to position zero
G91;                Relative positioning
M114;               Return Current Position
G0 Z75;             Move Z-axis up
G28 X;              Home belt axis
M114;               Return current position
M206 X+10;          Set belt axis offset
```