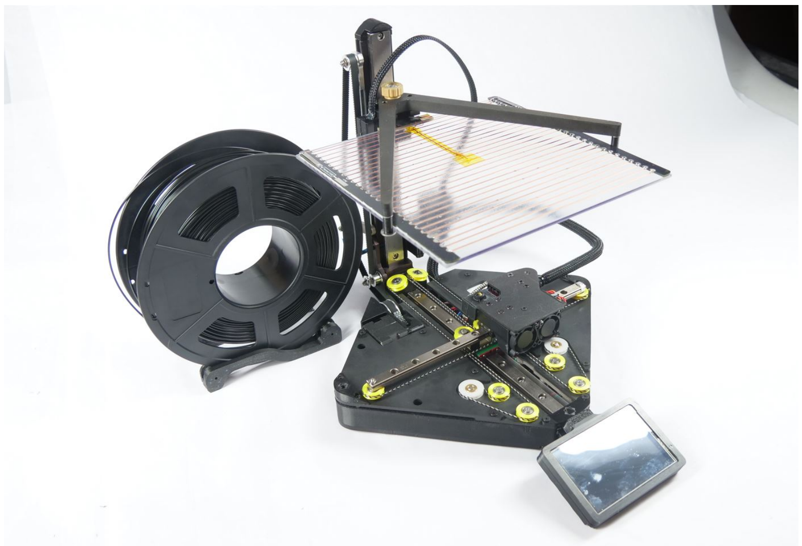
UCF Senior Design I Final Report

Project Title: 3D Portable Printer **(3DPP)**

Group Number: 8

07/24/2023



Note. from (KRALYN3D, 2022)

Group Members:

Zachary Yore	:	Computer Engineer
Jack Qualls	:	Electrical Engineer
Joseph Tuckerman	:	Electrical Engineer
Michael Doyon	:	Electrical Engineer

Review Committee:

Dr. Mike Borowczak
Dr. Zakhia Abichar

Table of Contents

1.0 Executive Summary....1
2.0 Project Description....2
2.1 Motivation / Goals / Objectives....2
2.2 Functions....2
2.3 Optional Features.... 3
2.4 Requirement Specifications.... 3
2.4.1 Software Specifications.... 4
2.4.2 Hardware Specifications....4
2.5.1 Software Block Diagram.... 5
2.5.1.1 Firmware....5
2.5.2 Hardware Block Diagram.... 6
2.5.2.1 Microcontroller / 3D Printer Controller:....6
2.5.2.2 Battery / Renewable Charger....6
2.5.2.3 3D Printer Design / Peripherals....7
2.5.2.4 Stepper Motors/ Sensors.... 7
2.5.2.5 PCB Design.... 7
2.5.2.6 Voltage Regulator.... 7
2.5.2.7 Digital Display....7
2.5.2.8 Hot End....7
2.6 House of Quality Diagram....8
2.7 Project Operational Manual.... 10
3.0 Project Research....11
3.1 Printer Controller Board....11
3.1.1 Printer Controller Board Options.... 11
3.1.1.1 BigTreeTech's SKR Pico.... 12
3.1.1.2 RAMPs Re-ARM.... 12
3.1.1.3 BIGTREETECH SKR 3.... 13
3.1.1.4 Duet3D's Duet 3 Mini 5+....13
3.1.2 Further Controller Board Breakdown.... 14
3.1.2.1 Cost....14
3.1.2.2 Acceptable Voltage Level....15
3.1.2.3 MCU Clock Frequency....16
3.1.2.3 Dimensions.... 16
3.1.3 Printer Controller Board Decision....18
3.2 Firmware....18
3.2.1 Firmware Options.... 18
3.2.1.1 Klipper.... 18
3.2.1.2 Marlin.... 19
3.2.1.3 RepRap.... 20
3.2.1.4 Smoothieware.... 20

3.2.1.5 Firmware Decision......21
3.3 Slicer Software......21
3.3.1 Slicer Software Options......22
3.3.1.1 Cura......22
3.3.1.2 PrusaSlicer......23
3.3.1.3 IdeaMaker......23
3.3.1.4 Simplify3D......23
3.3.1.5 Octoprint......24
3.3.1.6 Astroprint......24
3.3.1.7 Slicer......24
3.3.2 Slicer Decision......24
3.4 Renewable Charger......25
3.4.1 Renewable Charger Options......26
3.4.1.1 Suner Power SP-50W......27
3.4.1.2 Newpowa NPA30S-24I......27
3.4.1.3 Aleko SP20W24V-AP......27
3.4.2 3DPP Solar Panel Selection......28
3.5 Battery Pack......28
3.5.1 Battery Pack Options......30
3.5.1.1 Aegis Energies ALF-024020P......31
3.5.1.2 QZF B07Y35LXYY......31
3.5.1.3 TalentCell PB240B1......31
3.5.2 3DPP Battery Pack Selection......32
3.6 Voltage Regulator......32
3.7 3D Printer Designs......35
3.7.1 Cartesian......35
3.7.2 Delta......36
3.7.2 H-Bot......37
Figure 12: H-Bot 3D Printer Design......37
3.7.3 Polar......38
3.7.4 Belt......38
3.7.5 Printer Design Decision......39
3.8 3D Printer Host Software......39
3.8.1 Octoprint......39
3.8.2 Pronterface......40
3.8.3 Mainsail......41
3.8 3D Printer Host Software Decision......42
4.0 Design Constraints and Standards......48
4.1 Constraints......48
4.1.1 Economic Constraints......49
4.1.2 Time Constraints......49

4.1.3 Environmental Constraints..50
4.1.4 Social Constraints.. 50
4.1.5 Ethical Constraints... 51
4.1.6 Sustainability Constraints.. 51
4.1.7 Manufacturability Constraints... 51
4.1.8 Presentation Constraints...52
4.1.9 Political Constraints...52
4.3 Design Standards.. 53
4.3.1 IEC 62133-2-2017... 53
4.3.2 PEP 8-2016.. 54
4.3.3 IEEE 829-2008 Software and Systems Test Documentation Standard........................ 54
4.3.4 IPC-2221 Standard...56
4.3.5 IEEE 1241-2010 Terminology and Test Methods for Analog-to-Digital Converters Standard.. 56
4.3.6 IEC 61215-1:2021... 57
4.3.7 IEC 61853-1:2011..57
4.3.8 IEC 60034-1:2022... 58
4.3.9 IEEE 1012-2016 System, Software, and Hardware Verification and Validation Standard.. 59
4.3.10 IEEE 829-2008 Software and System Test Documentation Standard........................59
4.3.11 IEEE 802.11n-2008 Standard..60
4.3.12 IEC 61010-1:2010... 60
5.0 Comparison of ChatGPT with Other Similar Platforms...61
5.1.1 ChatGPT Pros/Cons, Limitations.. 61
5.1.2 Chatsonic Pros/Cons, Limitations... 63
5.1.3 Other A.I. Chatbots..63
5.2.1 Positive Uses of A.I. Software in Senior Design..64
5.2.2 Negative Uses of A.I. Software in Senior Design...65
6.0 Hardware Design.. 66
6.0.1 Hardware Key Features..66
6.1 Hardware Design Methodology..67
6.2 Hardware Subsystems... 68
6.2.1 Power System.. 68
6.2.1.1 12V to 5V Voltage Regulator... 69
6.2.1.2 24V to 6V Voltage Divider...70
6.2.1.3 24V Regulator Circuit.. 70
6.2.2 HiLetgo ADS1115 (ADC)..72
6.2.3 Kuman TFT LCD Touch Screen Display.. 73
6.2.4 SKR-Pico... 74
6.2.5 Raspberry Pi 2B...75
6.2.6 CanaKit Wifi Dongle... 76

6.2.7 Nema 17 24V Stepper Motor........ 77
6.2.8 Cooling Fan (Winsinn 30mm Blower Fan)........78
6.2.9 Heat Cartridge (3Dman 24V 50W Heater Cartridge)........78
6.3 Hardware Mounting/Mechanical Components........78
6.4 Overall Hardware Design........ 79
7.0 Software Design........80
7.0.1 Software Key Features........81
7.1 Software Design Methodology........ 82
7.2 Software Tools and Packages........ 83
7.2.1 Integrated Development Environment (IDE)........ 83
7.2.2 Mainsail........84
7.2.3 KlipperScreen........ 85
7.2.4 Moonraker........ 87
7.2.5 Klipper API........ 87
7.2.6 Hostpad........ 88
7.2.7 Adafruit_ADS1x15 Software Library........88
7.2.8 Printer.cfg Programming........89
7.3 Software Flowchart........91
7.4 Software Case Diagram........92
8.0 System Fabrication/Prototype Construction........ 93
8.1 12V to 5V Voltage Regulator........93
8.2 24V to 6V Voltage Divider........95
8.3 24 Volt Regulator Circuit........96
8.4 Gantry Design........ 96
8.4.1 Initial Design Process........ 96
8.4.2 General Part Selection........ 97
8.4.3 Gantry Orientation........97
8.4.4 Specific Part Selection........98
8.4.5 Custom Parts/CAD Assembly........99
8.4.6 Part Fabrication/Construction........101
9.0 System Testing........ 102
9.1 Hardware Testing........102
9.2 Software Testing........109
9.3 Overall System Integration........ 110
9.4 Plan for SD2........110
10.0 Administrative Content........111
10.1 Estimated Bill of Materials........ 112
10.2.1 Senior Design I Milestones........ 113
10.2.2 Senior Design I Work Distribution........114
10.2.3 Senior Design II Milestones........ 117
11.0 Conclusion........118

List of Figures

Figure 1: Initial Design Drawing……………………………………………………………………4
Figure 2: Software Block Diagram……………………………………………………………...5
Figure 3: Hardware Block Diagram…………………………………………………………..5
Figure 4: House of Quality Diagram…………………………………………………………8
Figure 5: Photovoltaic Cell……………………………………………………………………...23
Figure 6: Battery Internal Schematic………………………………………………………27
Figure 7: Linear Voltage Regulator Diagram…………………………………………...31
Figure 8: Switching Voltage Regulator Diagram…………………………………………32
Figure 9: LM317 Programmable Voltage Regulator Circuit…………………………32
Figure 10: Cartesian 3D Printer Design……………………………………………………33
Figure 11: Delta 3D Printer Design……………………………………………………………34
Figure 12: H-Bot 3D Printer Design……………………………………………………………35
Figure 13: OctoPrint Menu…………………………………………………………………...38
Figure 14: Pronterface Menu ……………………………………………………………………..39
Figure 15: Mainsail Menu………………………………………………………………………..40
Figure 16: Screenshot of an example of PEP 8 used in Python Coding Environment……52
Figure 17: Initial Block Diagram for 3DPP Power System……………………………...66
Figure 18: Initial 12V to 5V Programmable Voltage Regulator Schematic…………………..67
Figure 19: 24V to 6V Voltage Divider Circuit……………………………………………...68
Figure 20: HiLetgo ADS1115 PCB……………………………………………………………..70
Figure 21: Kuman TFT LCD Touch Screen Display PCB…………………………………71
Figure 22: SKR PICO Controller…………………………………………………………………72
Figure 23: Raspberry Pi 2B ………………………………………………………………….73
Figure 24: CanaKit Wifi Dongle……………………………………………………………...74
Figure 25: Nema 17 24V Stepper Motor……………………………………………………76
Figure 26: Initial 3DPP Overall Hardware Block Diagram………………………………78
Figure 27: Initial 3DPP Overall Schematic…………………………………………………...79
Figure 28: Software Design Schematic…………………………………………………………80
Figure 29: Battery Monitor Software Schematic………………………………………………81
Figure 30: Agile Development Cycle……………………………………………………………82
Figure 31: KlipperScren Print Profile Status……………………………………………………...85
Figure 32: SKR-Pico Hardware Pinout………………………………………………………….90
Figure 33: Overall Software Flowchart…………………………………………………………….91
Figure 34: Software Case Diagram……………………………………………………………92
Figure 35: Initial 12V to 5V Programmable Voltage Regulator Eagle PCB Design…………93
Figure 36: 24V to 6V Voltage Divider Eagle PCB Design……………………………….…95
Figure 37: Railway for Gantry System…………………………………………………………..97
Figure 38: Gantry Orientation for 2-axis Motion…………………………………………………98
Figure 39: Top View of Gantry Railway…………………………………………………….…98
Figure 40: Belt for Gantry…………………………………………………………………………98
Figure 41: Nema Stepper Motor………………………………………………………..…99
Figure 42: Gantry CAD Base Part 1……………………………………………………………99

Figure 43: Gantry CAD Base Part 2……………………………………………………………99
Figure 44: Gantry CAD Base Part 3……………………………………………………………99
Figure 45: Gantry CAD Base Part 4…………………………………………………………100
Figure 46: Complete Gantry CAD Diagram…………………………………………………100
Figure 47: Fabricated Gantry Parts…………………………………………………………101
Figure 48: Gantry Mounting Parts……………………………………………………………102
Figure 49: Gantry Belt Parts……………………………………………………………….102
Figure 50: Multisim Testing of Voltage Divider……………………………………………103
Figure 51: Voltage Divider Circuit onBreadboard…………………………………………103
Figure 52: DC Power Supply Settings for Voltage Divider………………………………104
Figure 53: Multimeter Measuring Voltage for Voltage Divider………………………….105
Figure 54: 12V to 5V Voltage Regulator on Multisim…………………………………….106
Figure 55: Breadboarding of the 12V to 5V Regulator Circuit………………………….107
Figure 56: Multimeter Measurement for Voltage Regulator Output…………………………108

List of Tables

Table 1: Specification……..3
Table 2: Control Board Cost……..2
Table 3: Control Board Voltage Level……15
Table 4: Control Board Clock Frequency……16
Table 5: Control Board Dimension……17
Table 6: Control Board Inputs……17
Table 7: Monocrystalline Panel…….26
Table 8: Polycrystalline Panel…….26
Table 9: Thin Film Panel……26
Table 10: Group Choice of Solar Panel Type…….26
Table 11: SP-50W Key Parameters…….27
Table 12: NPA30S-24I Key Parameters…….27
Table 13: SP20W24V-AP Key Parameters…….28
Table 14: 3DPP Solar Panel Final Selection…….28
Table 15: NiMH Battery…….30
Table 16: Li-ion Battery…….30
Table 17: ALF-024020P Key Parameters…….31
Table 18: B07Y35LXYY Key Parameters…….31
Table 19: PB240B1 Key Parameters…….32
Table 20: 3DPP Solar Panel Final Selection…….32
Table 21: Filament Types and Printing Temperatures…….47
Table 22: Filament Price for a 1Kg Spool…….48
Table 23: Initial Bill of Materials……112
Table 24: Senior Design 1 Milestones……113
Table 25: Senior Design 2 Milestones……117

1.0 Executive Summary

Our team has identified a pressing need for a versatile tool in the dynamic world of engineering education, where students are constantly on the move, attending classes, engaging in library study sessions, participating in club meetings, and pursuing internships. As a team of dedicated engineering students, we have recognized the significance of finding innovative solutions to help our educational experiences. This report presents our research on a tool that has caught our attention due to its tremendous potential. We are excited to explore the possibilities and benefits a portable 3D printer can offer to engineering students and professionals alike.

The portable 3D printer was designed obviously with portability in mind. The printer will utilize a battery pack hidden within the printer's base to 3D print on the go. Along with this, we will implement our software for the printer so that when the battery voltage does get too low to safely continue printing, a popup window will appear on a touchscreen user interface so that the user will know it's time to recharge. Suppose the printer does entirely run out of battery or shuts off for whatever reason. Do not fear. We have created a software program so that when safely recharged, the printer will continue where it last left off. To recharge our portable 3D printer, users must plug the printer into a standard wall outlet via the charger or a solar panel for even more portability. The printer's overall state-of-the-art physical design will mean that the printer will be able to fold into a smaller portable design, but not only this, the design we will implement allows us to create our 3D print designs at a fast rate. This is extremely important for our team as the printer will have a time limit because of the limited battery life.

This report documents the 3DPP team's design process for creating a portable 3D printer. It will first describe our team's motivation and goals for this project. Next, it will go into the requirements and specifications we set for our project, such as size limits and battery requirements. Following this, we have the bulk of our report in the research section, where our team explores the technologies and products our design will need to implement. Some examples are solar panel technology and picking the best 3D printer control board for our project. Next, the document details the design constraints and standards for a portable 3D printer.

Furthermore, our document will showcase our design compared to OpenAI's ChatGPT response to “create a portable 3D printer”. The paper details the software and hardware designs, such as parts in our plan and software we will use in our team's project. Next, the document goes into system fabrication and system testing, in which we discuss testing our PCB design and our plan of action on testing the printer when we get to Senior Design II. Finally, we get to the administrative content section, where we discuss the schedule that we set for ourselves and an initial bill of materials.

2.0 Project Description

The driving force behind our team's decision to create a portable 3D printer was rooted in our clearly defined motivation, goals, and objectives. These foundational elements served as the compass guiding our project's direction and scope. Once we had a firm grasp of these core factors, we could transition into the next phase of our project: outlining the specifications, requirements, and features that our portable 3D printer would embody. This systematic approach ensured that every aspect of our project was aligned with our initial vision, paving the way for a cohesive and purpose-driven design process.

2.1 Motivation / Goals / Objectives

A 3D printer is indispensable for individuals proficient in 3D CAD design and budding engineers. It offers the advantage of rapid, cost-effective prototyping and solutions to many tasks. However, the portability of these devices often poses a challenge due to their substantial size and dependence on a stationary power supply. We aim to transform this invaluable tool into a portable, user-friendly device, thereby expanding its utility.

While our project has not garnered any sponsorships, our primary stakeholders are our committee of professors. However, we envision a broader customer base that includes engineering students and professionals who could benefit from our project. Our team, composed of three electrical engineers and one computer engineer, is eager to leverage our collective skills to meet this challenge. This endeavor necessitates a deep dive into both the software and hardware aspects of 3D printers, with a particular emphasis on optimization as the cornerstone of our success.

2.2 Functions

This project is designed to 3D print using standard PLA and ABS filaments, delivering high print quality, speed, and detail on par with commercially available 3D printers. However, what sets our printer apart is its portability and lightweight design, making it easy to transport. It also features an intuitive user interface tailored for everyday consumers. Traditional Cartesian FDM (Fused Deposition Modeling) printers are inundated in the current market. We aim to break the norm and create a printer that stands out. The key selling point of our printer is its user-friendly design, making it an ideal choice for consumers on the move. Along with this since the printer will be operating off of a battery the system will come equipped with software to save the positioning of the printer as it prints. This will allow the user to shut off the printer due to to recharge the battery, and when they resume printing again they can begin again right where the printer left off.

2.3 Optional Features

Additional features that depend on the realization of printer constraints include utilizing more complex filaments such as PETG, Polycarbonate, and TPU. Additionally, while we would like to keep the cost reasonable, the project's initial design would likely not be competitive with the affordability of other models on the market. We will always strive to minimize costs without compromising our project's function. Some other possibilities to make the task even more travel-friendly are to stay within TSA guidelines on what power sources can be brought onto commercial flights and stabilize the printer enough to where it can print accurately in a vehicle.

2.4 Requirement Specifications

Table 1: Specification Table

Specifications			
Component(s)	**Component(s)**	**Parameter**	**Specification**
Stepper Motor	Stepper Motor	Low power consumption & precision	<= 1.5Amps
Hot End	Hot End	Temperature and Power Consumption	210°C & <= 40 Watts
Battery Pack	Battery Pack	Battery Life	24V & >=20,000 mAh
3D Printer Controller	3D Printer Controller	Compact & meets our specifications on motor / peripheral ports	It has enough ports for four stepper motors, one raspberry pi, one temperature sensor, one hot end, two coolant fans
Raspberry Pi 2b	Raspberry Pi 2b	Fast processing & compact	1 GHz Processor & <= 65mm x 30mm
Bed Heater	Bed Heater	Low power consumption	50°C & <= 150 Watts/h
Solar Panel	Solar Panel	Compact & able to charge 24V battery	20 Watts & 1 Hour Solar Production = 20 Wh / 0.58 Ah

Interface	Interface	Easy to use/Navigable	Compact yet readable (Size TBD during testing)
Overall Size	Overall Size	Printer/power sources should be easily transportable	The unassembled printer should fit within a 1ft cube

2.4.1 Software Specifications

In defining the software specifications, we aim to optimize the code for efficient communication between the 3D printer driver and its peripherals. Selecting firmware requires a 3D printer controller with a minimum 16 MHz processor speed compatible with the firmware. Ideally, 80 to 100% of the source code should be customizable. The optimization process will also deactivate unnecessary procedures throughout the printing duration, such as turning off the five-volt print bed heaters after completing the first layer. A pivotal software specification includes linking the battery to a Digital-to-Analog Converter (DAC) interfaced with the Raspberry Pi, which is crucial for user monitoring of the battery's voltage. It allows users to timely recharge the battery, ensuring uninterrupted operation when the voltage starts to decline. Additionally, we would like to ensure compatibility with commonly available slicers like Cura and leave room for additional customization of G-Code parameters to optimize the printer's effectiveness in a given battery life.

2.4.2 Hardware Specifications

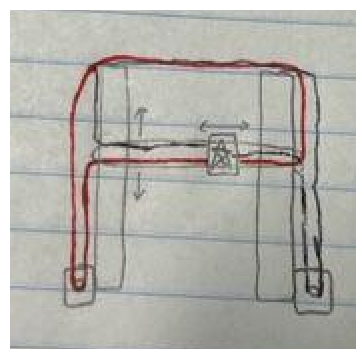


Figure 1: Initial Design Drawing

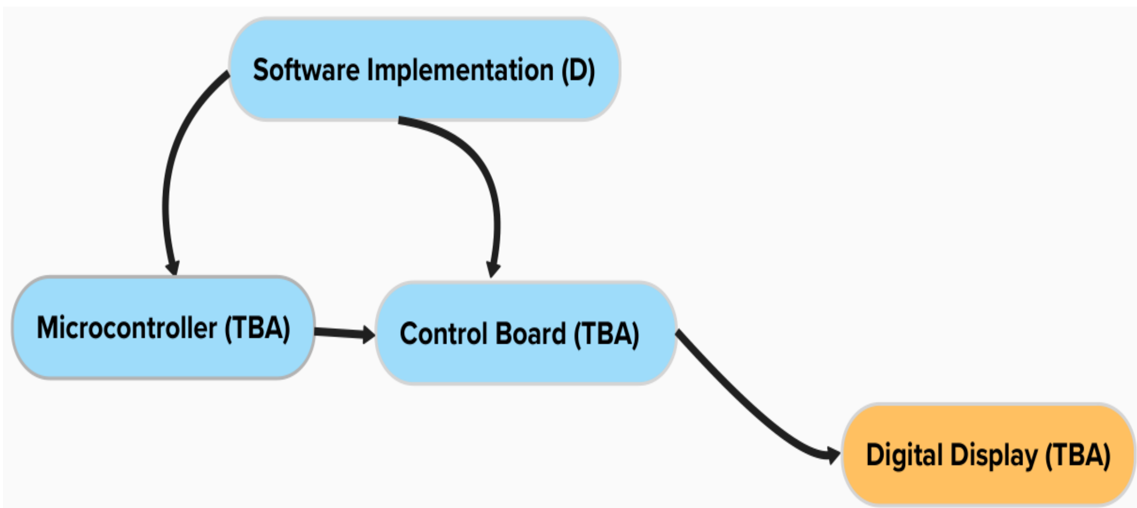
The overall performance of our printer can be measured in a couple of critical categories: print speed, print quality, and print strength. Print speed typically sacrifices print quality. However, we can make a few design choices to minimize this tradeoff. One such of these design decisions is to use an unconventional non-cartesian gantry (gray). The idea behind this gantry design is that the X and Y axis are controlled by two stepper motors (bottom left and right) with belts (red and black) mounted in a fashion shown in the illustration. This design increases the speed at which the printer gantry can move and cuts down on weight and jitter as the print head moves. What kind of timing belt is best for this gantry will be determined. However, the stepper motors used will be Nema stepper motors for their reliability and relatively small size. Since we wish to have the printer in a 1’ cube, the gantry arrangement is subject to change. However, the control scheme will remain the same.

An additional stepper motor will control the z-axis using a threaded rod to adjust the build plate height as the print progresses. As for the build plate, we only wish it to be at least 2” x 2”, though we would prefer larger; maximizing print dimensions is outside this project's scope. The build plate will be heated by a bed heater, which should be able to maintain 50 degrees Celsius at less than 150 Watts/h. The print bed will be leveled manually using a screw and spring system, though this could be automated with touch sensors further in development. The same applies to zeroing the stepper motors on the gantry.

The print head is essential to a functioning printer; therefore, we will begin with a proven design. A stepper motor will feed the filament into the hot end, which will be heated to roughly 210 degrees Celsius, depending on the filament type. The print head should also weigh no more than a pound to reduce the amount of jitter from the printer and ensure good print quality.

2.5.1 Software Block Diagram

Figure 2: Software Block Diagram



2.5.1.1 Firmware

In this project, firmware is a designated software that will be implemented both on the 3D printer control board and the Raspberry P as well as it is injected into the hardware of the 3D printer control board. The firmware is responsible for interpreting the various G-Code commands and performing the correlating actions, such as reading the inputs from a sensor or giving the hardware commands to move the stepper motors. Along with this the firmware will be responsible for handling the users input. This is very important for our project as we intend to have the ability for our users to be able to control the 3D printer via a touchscreen display as well as through a hosting software. Our team has made this an important requirement because of the fact that this will be essential in letting the user know when the battery on our portable 3D printer needs to be recharged. Along with this we will implement a software feature that allows the user to keep the status of their printer even after the printer has been shut off.

Along with this the firmware is responsible for a multitude of factors that include managing temperatures of the system. For a normal 3D printer this system is definitely important as it regulates the heating cartilage for meling the filament. For our custom portable printer, it is even more so because of the fact that our printer will be using a battery it is impaitve that our printer remains at the correct melting temperature. This is because the heating cartridges draw the most heat from the battery out of all the hardware subsystems.

2.5.2 Hardware Block Diagram

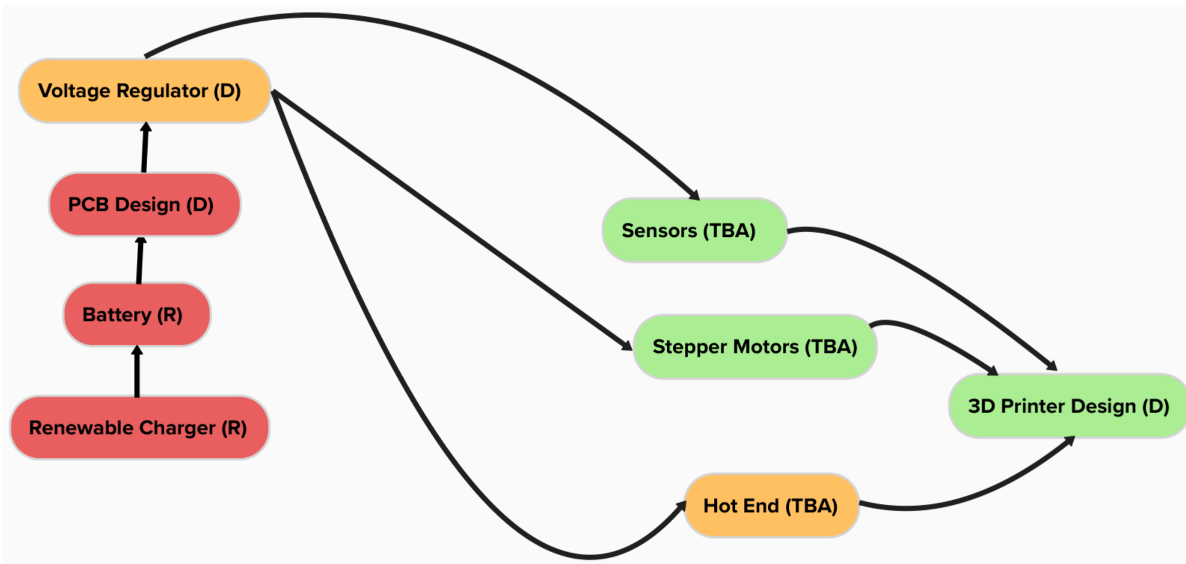


Figure 3: Hardware Block Diagram

TBA - To be acquired
A - Acquired
R - Researching
D - Design
P - Prototype
C - Completed

Zachary
Jack
Joseph
Michael

2.5.2.1 Microcontroller / 3D Printer Controller:

In this project, the microcontroller functions as the brain sends commands to the 3D printer controller, such as G-Code translation commands or for our project voltage levels of the battery. The 3D printer controller, directly connected to the microcontroller, plays two crucial roles. Firstly, it interprets the G-code stored in the microcontroller's SD card. Secondly, it serves as the controls and supplies power for the various peripherals associated with the printer, such as stepper motors, sensors, heating elements, etc.

2.5.2.2 Battery / Renewable Charger

In this project, the battery will function as the heart of the system supplying power to all components. The renewable charger will charge the battery in a portable and eco-friendly fashion since the 3D printer will be untethered and operated. We plan to incorporate solar panels as our system's primary renewable charging source. For the solar power element of our project, we would like to find a lighter and more transportable design so that our users can carry around this vital piece of equipment for our project. With these two components, we plan to provide an affordable yet power-efficient design that will maximize performance and charge life.

2.5.2.3 3D Printer Design / Peripherals

Overall, the printer will resemble other fast printer designs available on the web, as the focus of this project is not necessarily the printer's design itself but rather the power design to allow it to operate untethered. The mechanical setup will only service the speed of the print (since limited power will necessitate a faster print) and the printer's efficiency (meaning no extra energy is used moving heavy print heads, etc.). Our team will also try to maximize portability and accessibility in our 3D printer design. Incorporating some of the cutting-edge printer designs on the market today.

2.5.2.4 Stepper Motors/ Sensors

The stepper motors must be highly power efficient and reliable enough to make adjustments within 1mm to ensure good print quality. The sensors will be thermometers on the print bed and hot end to provide feedback to the control board, properly heat the printer, and touch sensors to automatically zero the gantry and z-axis and level the print bed.

2.5.2.5 PCB Design

The PCB designs will be mainly used for power consumption applications within the 3D printer. We will make the PCB specifically designed to reduce power and energy dissipation. Also, our voltage converters will be implemented in this portion of the system since we will need certain ease of access and configuration ability in the most crucial pieces of the system.

2.5.2.6 Voltage Regulator

Due to the nature of this project, a rechargeable battery will power the printer via solar panels, and a voltage regulator will be required to maintain a fixed voltage supply throughout all operational conditions. The regulator will be custom designed to fit the exact requirements of the printer.

2.5.2.7 Digital Display

This project will use a digital display to monitor several values and conditions of the printer. This includes the print status, the amount of battery life currently available, how much each print material remains, and possibly more factors as this project progresses. The display will be directly connected to the control board.

2.5.2.8 Hot End

The hot end will heat and lay the print material through a nozzle. The hot end is also responsible for maintaining a consistent temperature to ensure successful and quality prints. This printer component consumes more power than other components, so we will conduct thorough research to choose a hot end most applicable to our power and print quality/time constraints.

2.6 House of Quality Diagram

The House of Quality Diagram presented below acts as a holistic tool that encapsulates this project's engineering and marketing needs, specifically focusing on the development of a portable 3D printer. This diagram is a game-changer in our project planning and development process, as it visually maps out how each requirement is linked and impacts the others. It is like a mind map that helps us grasp the intricate relationships between different needs and how a tweak in one area might create a domino effect on others.

This diagram is not just an essential checklist of requirements; it is more like a complex puzzle or a matrix that illustrates the correlation between each need. It is like a social network of requirements, showing whether they are best buddies (mutually supportive) or frenemies (conflicting). Specific symbols represent this correlation, each symbol carrying its unique vibe. Underneath the diagram, users will find a legend that spells out what each symbol stands for. This legend is like a decoder ring, guiding the understanding of the diagram and making it a breeze to interpret the relationships and correlations between the various requirements.

In conclusion, the House of Quality Diagram is not just a tool; it is a visual language that communicates the intricate relationships between different project requirements. It is a testament to the power of visual thinking in project management and a reflection of our commitment to delivering high-quality, innovative solutions in portable 3D printing.

			Engineering Requirements				
			Dimensions	Cost	Print Time	Print Speed	Power Use
			-	-	+	+	-
Marketing Requirements	Low Cost	-	↑↑	↑↑	↓	↓↓	↓
	Portability	+	↑↑	↑			↓↓
	Battery Life	+	↑↑	↓↓	↑↑		↑↑
	Ease of Use	+	↑				↑
	Print Quality	+	↓	↓↓	↓	↓↓	↓↓
	Targets for Engineering Requirements		>1ft cubed	<$600	1 Hour +	70mm/s	50 Watts/h

Figure 4: House of Quality Diagram

Legend

- \+ Positive Polarity
- \- Negative Polarity
- ↑ Positive Correlation
- ↓ Negative Correlation
- ↑↑ Strong Positive Correlation
- ↓↓ Strong Negative Correlation

As indicated in the House of Quality, our current engineering requirements dictate that we must design a printer that is less than 1 foot cubed in size with a power consumption of 50 Watts per hour and a print speed of 70 millimeters per second. This printer must be able to run off of a battery and print for at least 1 hour. The total price of the printer must also be less than $600. Our House of Quality has a strong positive correlation between engineering and marketing requirements for low cost and dimensions, portability and dimensions, battery life and sizes, cost in general, print time and battery life, and power use and battery life. Moving forward with our design process, those relationships are incredibly vital, and we must pay special care to ensure that one does not negatively affect the other.

2.7 Project Operational Manual

The 3DPP portable 3D printing unit is your state-of-the-art battery-powered, wireless, solar-charged 3D printer. This system allows users to be able to print from almost any location. Your mobile device and an internet connection are optional for our device. To use the 3DPP team's portable 3D printer, first, the user has to turn on the printer from an on/off switch located at the base of the printer; it may take a few seconds for everything to properly start up, so you may need to wait 5 to 10 seconds to ensure all systems are operational. Following this, the user will want to upload their G-Code file, which they got from their favorite Slicer. Note: The 3DPP team does recommend using the Slicer Cura, as this is the tested Slicer for our printer. To upload your G-Code file, the user has two options. The first is through the Micro SD card slot on the printer located at the base of the printer.

The user will then drag and drop their G-Code file onto the Micro SD card. Ensure you put the file into the folder /home/pi/Klipper/GCodeFiles/. The second way to get your G-Code file onto the portable 3D printing unit is by connecting to the Raspberry Pi within the 3D printer. To complete this, the user must first ensure that they are in the wifi range of the 3D printer. Secondly, connect to the wifi source labeled 3DPP; the password is 3dpp1234. Following this step, go to your local search engine and type http://3DPP.local/. This will bring you to a webpage called Mainsail. This is where the user can navigate to the G-Code selection page, where you can pick to upload the file from your local device. After you upload your G-Code file, we can select your printing settings. The user can do this in two ways, like uploading your G-Code file. The first is the easiest; you can edit your 3D printer settings using the onboard touchscreen display. Using this display, you can edit print speed and hot end temperature and monitor the print as it is printing.

Similarly, you can do the same via Mainsail, this user interface is different, but you can do the same things with it. As a user, you may have noticed a voltage indicator on the top of both user interfaces. This is the voltage of the 3D printer's internal battery. Once you begin printing, this voltage level will start to go down, but do not worry the average printer settings has a print time of 2 hours. Nevertheless, if this voltage reaches the dropout voltage of 20 volts, a popup display will appear on both user interfaces instructing you to recharge the battery.

Recharging the battery is a simple process; the user has two options. The first option is if you are not in a location where a wall outlet is accessible. You can plug the battery into the solar power charger. Note: This process does take a while. The second and quicker option is plugging the battery into your at-home wall socket, which is faster. You may wonder, "My battery died, so I must start my print again?". The answer to your question is No! You do not need to start your print over. The 3DPP team has designed a state-of-the-art software system that allows its user to start where they have left off, even if the printer has been turned off.

3.0 Project Research

The first step the 3DPP team did to design a top-of-the-line 3D Portable Printer was doing in-depth research on all possible parts and software the printer could utilize. These key features included printer controller boards, firmware, renewable chargers, and PCB designs. We also researched critical technologies that our printer will use, such as solar panels and sensor types.

3.1 Printer Controller Board

The portable 3D printing unit will be equipped with a single printer controller board, responsible for handling the multitude of functions integral to the operation of the 3D printer. The control board itself is centered around what is called a primary control chip. Different printer control boards use additional direct control chips with varying speeds of processing and firmware requirements.

Processing speed is directly correlated to a printer's overall printing speed because higher processing means the printer can conduct more g-code commands in less time. Given that a significant constraint for our project is limited battery power which imposes a time limitation, a control chip with a higher processing speed becomes an essential requirement. The controller board uses these commands to manage the various hardware components, such as the stepper motors, hot end, and bed heating. The control board has connections designed for these components to facilitate these operations.

3.1.1 Printer Controller Board Options

A vast array of printer controller board options exists in the market. Initially, the 3DPP team considered using BigTreeTech's SKR Pico, as it met our original software and hardware requirements. However, other viable alternatives included the MKS Robin Nano V3.1, Duet3D's Duet 3 Mini 5+, and RAMPs Re-ARM. These boards were considered due to their high popularity among 3D printing enthusiasts and their customizability. Extensive research compared these boards across varying performance metrics and price points.

3.1.1.1 BigTreeTech's SKR Pico

The SKR Pico, created by BIGTREETECH, drew our interest as the 3DPP team's severe first exploration into printer control boards. Its appeal stemmed from a variety of factors: its compact design, cost-effectiveness, fast CPU, and its compatibility with our chosen firmware and a Raspberry Pi.

What sets the SKR Pico apart is its unique requirement to be connected to a Raspberry Pi to function correctly. This is due to an unusual design decision; rather than flashing the firmware onto the onboard MCU chip as most boards do, it is instead flashed onto the Raspberry Pi. This innovative design approach from BIGTREETECH cleverly utilizes the superior power of the 900MHz quad-core ARM Cortex-A7 CPU found on a Raspberry Pi 2, which outclasses the Arm Cortex-M0+ processor on the SKR Pico.

The Raspberry Pi performs all complex G-code computations in this configuration, leaving the smaller CPU on the SKR Pico free from these heavy-duty tasks. Another upside to using this board is that the firmware the board requires, named Klipper, is entirely open source, programmed in Python, and can be modified easily via the Raspberry Pi. This allows our team to implement the features necessary for a printer that runs off a battery.

3.1.1.2 RAMPs Re-ARM

Another board the 3DPP team considered was the RAMPs Re-ARM 3D printer control board. This is a mighty board used by many 3D printing enthusiasts, with tons of documentation and forums to help if we choose this board. This board is popular because of its low price tag and high-performing onboard processor for the g-code translations.

Now the physical design of the board is similar to the BIGTREETECH SKR Pico as it does require a secondary board, in this case, what is referred to as a "shield," to get the full functionality of the board. The Re-ARM requires a RAMPs 1.4 or 1.5/1.6 Shield, which sits atop the Re-ARM. The shield connects stepper motor drivers, endstops, thermistors, and other components.

According to the Re-ARM specification sheet, the board has a 32-bit LPC1768 ARM processor running at 100MHz with 512KB of flash memory, which is much more powerful than the SKR Picos. It is also supported by almost all major 3D printing firmware, making it easier for our team to pick the best firmware. The downside to this board which ultimately sent the 3DPP team away from using it, is that it is not compatible with a Raspberry Pi, meaning our group could not implement our code to be used by the control board.

3.1.1.3 BIGTREETECH SKR 3

BIGTREETECH SKR 3 is the big brother to the SKR Pico, the 3DPPs initial board of choice. The SKR 3 boasts all-around impressive figures according to the specification sheet. The board can run a unique 17 stepper motors at once, with the default being the traditional 4, and it has 2 USB slots and an SD card slot. It also meets one of the 3DPP teams' most important requirements for a 3D printer control board, as it is compatible with a Raspberry Pi.

One of the core differences between the SKR 3 and its little brother, the SKR Pico, is that the SKR 3 does not require the Raspberry Pi to function or any other board. Instead, it relies on the ARM Cortex-M7 STM32H743VI MCU, which can translate G-Code just as fast if not faster than the RAMPs Re-ARM, known for its fast processing speeds as stated previously though it is compatible with a Raspberry Pi. Instead of the SKR 3 relying on it, it uses the Raspberry Pi as an add-on. Some of these add-on features include controlling the entire 3D printer over Wifi and connecting a live streaming camera to the printer to monitor the progress of the print.

Overall the BIGTREETECHs SKR 3 is a viable option for the 3DPP team. It meets all our requirements when initially looking for a 3D printer control board. This being said, the board does have some downfalls, such as the board's relatively large size of 110*85mm and a higher-than-average cost of $72.99.

3.1.1.4 Duet3D's Duet 3 Mini 5+

The Duet 3 Mini 5+, crafted by Duet3D, emerged as a serious contender during the 3DPP team's examination of printer control boards. Its allure resides in several key attributes, including its compact form factor, swift ARM Cortex-M4F processor, and compatibility with our preferred firmware and a Raspberry Pi.

One feature distinguishing the Duet 3 Mini 5+ is its standalone functionality, meaning it doesn't require tethering to a Raspberry Pi or any other board for operation. This design choice is rooted in its powerful ARM Cortex-M4F processor, a departure from many boards that rely on external computing power. With its exceptional processing speed, this processor can handle G-Code computations efficiently, freeing up resources for other essential tasks.

The Duet 3 Mini 5+, while not dependent on a Raspberry Pi, can certainly take advantage of one. When connected, it opens up avenues for features like WiFi-based printer control and live print monitoring via a connected camera. The firmware of choice, RepRapFirmware, is also open source and can be modified to include custom features necessary for a battery-powered printer. This level of adaptability offers our team the flexibility to tailor the printer's operation to our specific needs. Some downside to this board is its cost, with the average retail price being around $175.28. This board would have worked great for our project, but as our team strives to make an affordable portable 3D printer, this board is no longer considered.

3.1.2 Further Controller Board Breakdown

To further detail our research in deciding which controller board to use for our portable 3D printer, we did a detailed breakdown of some of the essential requirements for our board. These requirements are cost, compatibility with Raspberry Pi, power consumption, MCU performance, and amount of input features.

3.1.2.1 Cost

Cost is one of if not the most prominent factors we considered when choosing the correct control board for our team. We want to create a fast, precise, and long-lasting product on battery. But also keeping the price relatively low, a significant contributor is picking out the best control board. These are how the costs broke down.

Table 2: Control Board Cost Table

Control Board	Unit Price ($)	System Price Increase (%)
SKR Pico	$29.99	3.5%
Re-ARM	$29.99*	3.5%*
SKR 3	$72.99	8.5%
Duet 3 Mini 5+	$175.28	20.4%

* would have to purchase the additional RAMPS 1.4 board ($31.09)

The SKR Pico and the Re-ARM control boards are the cheapest at $29.99. The Duet 3 Mini 5+ has the most expensive price tag, around $175.28. Our team wanted to partially rule out the Duet 3 Mini 5+ because of price, but the board offers significant benefits for our project. Further breakdown is needed to determine precisely what board will be implemented in the final design.

3.1.2.2 Acceptable Voltage Level

Since the 3DPP team has decided to use a battery to make our project as portable as possible, power consumption is a significant factor in choosing practically all of our parts, including the control board. The idea behind this is that we are trying to develop a product that will deliver a long-lasting print time for the consumer meaning that we are looking for a board that can perform as intended on very low power.

Table 3: Control Board Voltage Level Table

Control Board	Input Voltage Range (V)
SKR Pico	12 - 24 V
Re-ARM	24 - 36 V
SKR 3	12 - 24 V
Duet 3 Mini 5+	11 - 25 V

The SKR 3, SKR Pico, and Duet 3 Mini 5+ all share similar input voltage ranges, making them similar options for the 3DPP teams project. However, the Re-ARM 3D printing board stands out from the rest because of its significantly higher input voltage range. This feature of the Re-ARM board presents a bad look for the 3DPP team. Because of its higher input voltage, we would require to obtain a larger battery to power it. This is a drawback for us because, as mentioned previously, we aim to design our project to fit within a 1ft cube. A larger battery would not only increase the size of our design but also be significantly more expensive as our team tries to keep cost at a minimum

Moreover, it's crucial for our project to have a battery that operates within a relatively low voltage range. This is because a lower operating voltage would enable our 3D printer to run for a longer duration. For instance, if a board has a cut-off voltage of 20 volts, we could only run a 24-volt battery down to 20 volts before it stops working. However, with boards like the SKR-Pico, SKR 3, and the Duet Mini 5+, we can run the battery down to around 12 volts. This means that our team and our 3D printer product would have a longer operational time, which is a significant advantage in a school setting where we might need to run the printer for extended periods.

3.1.2.3 MCU Clock Frequency

The internal MCU within the controller must have a fast clock frequency. This is because, as mentioned before, we want our printer to be able to finish the 3D prints as fast as possible due to the time constraints set by using an external battery rather than a continuous power supply. After researching this topic, our team found just how important this is. This is because if a controller board MCU clock speed is slower than the speed at which the board receives g-code translations, the internal software could skip over specific commands resulting in a failed print for the user.

Table 4: Control Board Clock Frequency Table

Control Board	MCU Clock Frequency (MHz)	MCU Name
SKR Pico	133MHz*	RP2040
Re-ARM	100MHz	LPC1768
SKR 3	480MHz	STM32H743VI
Duet 3 Mini 5+	120MHz	ATSAME54P20A

* As mentioned in the SKR Pico description, the board uses the Raspberry Pi MCU for heavy g-code translations. Raspberry Pi Clock Frequency is 900MHz.

After evaluating these findings, we can see that the onboard MCU of the SKR 3 outperforms the rest of the controller boards quite significantly. The SKR Pico uses the Raspberry Pi MCU and its onboard MCU, giving it the overall lead. The 3DPP team was surprised to find that the highest-priced control board, the Duet 3 Mini 5+ clock frequency, was relatively slow compared to the rest of the group.

3.1.2.3 Dimensions

Limited space is one of the biggest problems in designing a portable 3D printer. As a team, we have set the requirement of having the printer fit in a 1ft cube while it is in the “portable state.” We can not afford to waste space as we will need a large battery to power our design. So picking a small 3D printer control board was essential to the overall design of our printer. Here is how the four boards are stacked up against one another.

Table 5: Control Board Dimension Table

Control Board	Dimensions (LxW) mm
SKR Pico	85x56mm
Re-ARM	108x84mm

SKR 3	110x83mm
Duet 3 Mini 5+	123x100mm

As we can see, the SKR Pico has the smallest size relative to the other boards, while the Duet 3 Mini 5+ is the largest. As stated previously, the Duet 3 Mini 5+ is a mighty and well-designed board, but due to the 3DPP team's size constraints, the board needs to be more significant to continue further.

3.1.2.4 Number of Board Inputs

It is very important that the specific 3D printer control board that our team picks out has enough ports for all the hardware features we would like to incorporate into our design. Some of these include having four stepper motor ports, the ability to add an automatic bed leveling system, hot end port, and end-stop sensor compatibility. These key features are one of if not the most important aspects the 3DPP team looked at when trying to find the right board for our project. This being said, the results are as follows.

Table 6: Control Board Inputs Table

Control Board	**# of Stepper Motor Ports**	**Automatic Bed Leveling Feature (yes or No)**	**# of Endstop sensor Ports**	**Support for Bed Heating (Yes or No)**
SKR Pico	**5**	**Yes**	**3**	**Yes**
Re-ARM	**5**	**Yes**	**6**	**Yes**
SKR 3	**5**	**Yes**	**5**	**No**
Duet 3 Mini 5+	**7**	**Yes**	**6**	**Yes**

As shown the boards are relatively similar when it comes to how many stepper motor drivers each one supports, compatibility with automatic bed leveling, number of end-stop sensor connections and also the bed heating compatibility. This being said the Duet 3 Mini 5+ did outperform the rest of the group by having more and being compatible with the previously listed requirements. This being said though for our requirements we only need four stepper motors and three end stop sensors.

3.1.3 Printer Controller Board Decision

After careful consideration, the 3DPP team has decided to go with the SKR Pico as this board will benefit our project the most. The SKR Pico best fits our requirements by being a small compact board with the smallest size of

85x56mm, drawing very little power, which is very important to maintaining long battery life, a low cost of $29.99, and finally, its fast clock frequency as mentioned though it does require a Raspberry Pi for the board to operate correctly which our team found to be a desirable trait the SKR Pico possesses. This makes editing the firmware software much easier because it is flashed onto the Raspberry Pi.

3.2 Firmware

The firmware is the software embedded into the 3D printer control board and is responsible for many things. The firmware on a 3D printer's most crucial job is interpreting the G-Code, a low-level programming language used to instruct 3D printers and CNC machines. The firmware is also responsible for controlling the individual hardware parts such as the X, Y, and Z stepper motors, hot end, and various sensors connected to the 3D printer's control board.

3.2.1 Firmware Options

In this section, our team will break down some of the more popular firmware options for our 3D printer to decide which will work best for our portable printer design. These options will include; Klipper (known for its precision), Marlin (the most popular 3D printing firmware), RepRap (known for being highly module), and smoothieware (known for its multipurpose)

3.2.1.1 Klipper

Klipper, programmed in Python, is one of the newer 3D printing firmware on the block known for its precision and high-speed printing abilities. Some of the key features on why the 3DPP team decided to do research specifically on this firmware is that it is all open source. It is constantly being updated and getting new features by independent developers. It also allows our team to go in and add our own portable 3D printer-specific software to the firmware. For example, we would like to implement a software program that saves the G-Code commands that the firmware has already processed so that if the battery were to die in the middle of a print, the last position the printer was in would be saved. Then when the battery has been recharged, the firmware would know precisely where to start again.

A key feature that sets Klipper apart from most other firmware options is that it can be precise and quick because it separates command computation (interpreting the G-Code) and command execution (receiving/giving commands from the hardware). These commands are generally all performed by the MCU within the 3D printing control board. Instead, Klipper relies on a faster clock frequency external MCU chip to do the heavier command computation allowing the generally slower frequency MCU within the actual 3D printing control board to do the command execution. According to Klipper documentation and from multiple forums, the faster clock frequency external MCU chip generally comes by connecting and flashing Klipper onto a Raspberry Pi that comes with a clock cycle rate of 900MHz. Much faster than your average 3D printer control board.

Overall, Klipper is an excellent option for our team. It meets the requirements of being fast and precise, open source so that we can change and add our software to Klipper, and it is compatible with a Raspberry Pi, meaning that we can use the Raspberry Pi’s GPIO pins to monitor the battery’s voltage while also performing the 3D print.

3.2.1.2 Marlin

Marlin, programmed in C++, is the market's most popular 3D printing firmware. One of the core features of why it's so popular is its support for many different printer configurations, such as Cartesian, CoreXY, Delta, and SCARA 3D printer layouts. This is very important for the 3DPP team as we will be designing a very different 3D printer design due to the portable and foldable printing. Along with this, the Marlin, like Klipper, is open source, meaning that as a team, we can implement our code into the firmware. The only significant difference is that while Marlin programmed most in C++, Klipper is in Python.

Some of the key features that set Marlin apart from other firmware options is that because it is so popular, there is a tremendous amount of community support for this firmware—making it very easy to reach out to community members if the 3DPP team ever had any questions we wanted explicitly answered about the firmware. Along with the wide range of support, the Marlin firmware stands out because of its ease of use. Marlin comes with a very straightforward configuration process, with the user having to modify a few things in the config file. The last feature we will cover that sets Marlin apart from other firmware brands is that since the community is so significant and open source, some very intriguing features have been added to the firmware package, such as automatic bed leveling. This is when the printer can automatically gauge the distance between the hotend and the printer's bed which is very important in 3D printing because it establishes a well-structured first layer.

For the 3DPP team, Marlin stood out because of its extensive community support and ability to be compatible with many different 3D printing design configurations. Since it is programmed in C++, a programming language known for its leveraging between efficiency and control, the firmware runs very smoothly. This firmware would work great for a portable 3D printer.

3.2.1.3 RepRap

The following firmware the 3DPP team considered was RepRap, a C++ based firmware that is known for being highly modular, meaning that the firmware allows the user to use many different types of hardware parts, and similarly to the Marlin firmware, it is compatible with many other 3D printing layouts. The story behind RepRap and its modularity is that the RepRap team wanted to make a firmware product that anyone who has parts lying around could put together and design their 3D printer. This works perfectly for the 3DPP squad because we are building this project on a budget, and this firmware allows us to go cheaper when it comes to getting hardware parts.

Similarly to Klipper and Marlin, RepRap is open source allowing the 3DPP team to make the necessary software changes for a portable 3D Printer. However, it differs from a static configuration file, where users would input their specific 3D printer specifications using that firmware-supported programming language. The RepRap firmware uses G-Code commands to configure the printer. This means you can change many aspects of the printer's configuration without rebooting or recompiling the firmware. In a newer version of RepRap (RepRap 3.0), they introduced conditional G-Code. This allows the user to use conditional programming statements such as if-else and while / until loops in the G-Code. For example, you can create a script entirely in G-Code so that if the hot end gets to a specific temperature, it increases the fan speed.

RepRap is an exciting firmware with many vital features the 3DPP team could use to create our portable printer. We liked the new conditional G-Code as a team but decided to program our printer's necessary code in a more familiar programming language, such as Python. This being said, RepRap, even though it offers a lot, our team will no longer be considering this firmware.

3.2.1.4 Smoothieware

Smoothieware, developed in C++, is a versatile and high-performing firmware. It prides itself on its universal compatibility with multiple machine configurations, making it a perfect fit for the 3DPP team as we create a unique, portable, and foldable 3D printer design. Like Marlin and Klipper, Smoothieware is open source, empowering our team to contribute our coding expertise to the firmware's development and customization. However, it's important to note that unlike Klipper, which uses Python, Smoothieware utilizes C++ as its primary language.

Smoothieware has numerous distinctive features that elevate it among other firmware options. Its substantial user base equates to robust community support, offering a valuable resource for the 3DPP team should we encounter any questions or issues specifically related to the firmware. In addition, Smoothieware is praised for its simplicity and straightforwardness, with configuration as easy as adjusting settings in a single configuration file. A standout feature of Smoothieware is its advanced modular step-smooth-scheduling system, which guarantees exceptional motion control and high-speed performance.

Smoothieware's appeal to the 3DPP team lies in its broad community backing, its adaptability with various machine configurations, and the simplicity of its usage and design. Given its C++ foundation, a language renowned for its balance of efficiency and control. Smoothieware runs incredibly smoothly. This firmware would be highly suitable for a portable 3D printer, promising a seamless user experience and optimal performance.

3.2.1.5 Firmware Decision

After researching the above mentioned firmware, we used the Klipper 3D printing firmware. Klipper, like the Marlin, Smoothieware, and RepRap, is open source, giving the 3DPP team a straightforward way to manipulate and add our software to the firmware. Klipper stands out from the pack more than the other firmware for our team because instead of flashing your usual 3D printer control board, you flash the firmware onto a Raspberry Pi. This gives our team an even easier way to manipulate the firmware and use Python scripts to implement the changes we want. Also, because it is flashed onto the Raspberry Pi and utilizes the higher frequency cycles on the onboard Raspberry Pi MCU, it allows for a much faster and more precise print. This is very important for our team because, as discussed before, we are limited to the amount of battery life, meaning that with firmware with faster printing ability, we can complete more 3D prints before the battery ultimately dies.

3.3 Slicer Software

The role of slicer software in the 3D printing process is indispensable. The critical intermediary translates 3D models into a language the 3D printer can understand. This is achieved by generating G-code, a programming language widely recognized in numerical control. To ensure the seamless operation of our design, the integration of reliable slicer software is a necessity. It will enable our printer to interpret and execute 3D models accurately. In section 3.3.1, we will delve into a comparative analysis of various slicer software options, ultimately selecting the one that best aligns with the needs of our project.

3.3.1 Slicer Software Options

In the rapidly expanding domain of 3D printing, a plethora of slicer software options are available, each with its unique features and capabilities. Among the most renowned are Cura, Simplify3D, PrusaSlicer, ideaMaker, OctoPrint, Astroprint, and Slic3r. Cura, developed by Ultimaker, is an open-source software lauded for its user-friendly interface and extensive customization options. Simplify3D is recognized for its ability to precisely process complex and detailed models, making it a comprehensive professional-grade software compatible with a wide range of 3D printers. PrusaSlicer, initially a derivative of Slic3r, has evolved into a robust standalone slicer software optimized for the Prusa line of 3D printers but also compatible with others. ideaMaker, created by Raise3D, is known for its intuitive interface, advanced slicing capabilities, and efficient support structure generation. OctoPrint and Astroprint offer more than just slicing; they provide complete 3D printer management tools, allowing for remote management and monitoring of 3D printers via a web interface. Lastly, Slic3r, one of the earliest slicer software options, remains popular due to its open-source nature and rich features. These diverse slicer software options cater to a wide range of 3D printing needs, allowing users to choose the one that best aligns with their specific requirements.

3.3.1.1 Cura

Cura is a product of Ultimaker, which has significantly impacted the 3D printing industry. Ultimaker's roots are in open-source projects, and this ethos of accessibility and community contribution is evident in their development of Cura. This slicing software was created to provide a robust, user-friendly tool for the 3D printing community, and it has been offered for free since its inception. Ultimaker continues to support and update Cura, ensuring it remains a leading choice for 3D printing enthusiasts. While the software is free, Ultimaker offers purchase plans for those wishing to access premium features. These additional features enhance functionality and support, making Cura an even more powerful tool for 3D printing projects.

Cura is designed to be accessible to many users, from beginners just starting their 3D printing journey to experts who require advanced features and settings. It's user-friendly interface and comprehensive features make it a versatile tool that can cater to various needs and skill levels. Furthermore, Cura's compatibility with multiple operating systems - Windows, Mac, and Linux - broadens its accessibility. This cross-platform support ensures that regardless of the operating system a user prefers, they can utilize Cura's capabilities to optimize their 3D printing process. This commitment to accessibility and functionality is a testament to Ultimaker's dedication to supporting the 3D printing community.

3.3.1.2 PrusaSlicer

PrusaSlicer was initially a part of the Slic3r project but became a separate entity in 2019. What makes PrusaSlicer a good potential software to use is its open-source nature, allowing users to collaborate to improve upon the software. PrusaSlicer was designed to be incredibly inclusive, with modes for beginners that are new to the world of 3D printers and experts. The beginner mode makes this especially appealing to us, as most of the team is inexperienced with 3D printing and slicer software. PrusaSlicer is most suited for Prusa 3D printers. However, it can be used with virtually any other 3D printer.

Furthermore, PrusaSlicer is available on several popular operating systems, such as Windows, Linux, and Mac. One of the most appealing factors about whether the team would use PrusaSlicer is that it is entirely free. Overall, PrusaSlicer offers a plethora of benefits and is a great contender.

3.3.1.3 IdeaMaker

IdeaMaker was developed by the 3D printer manufacturer Raise3D. This slicer software is most well known for its user-friendly nature, being quite popular for beginners in 3D printing. In addition to being easy to use, it offers many valuable features and doubles as a “3D printer management platform” (Locker, 2023). In other words, while users can use IdeaMaker to prepare their 3D models for printing, they can also use IdeaMaker to track print progress and monitor printing profiles. IdeaMaker is free and offered on Windows, Mac, and Linux. Despite its robust features and user-friendliness, IdeaMaker is less popular than other slicer software and thus has a smaller community base.

3.3.1.4 Simplify3D

Simplify3D has proven to be nearly as popular as Cura and is one of the best choices for many 3D printer users. Simplify3D can be used with most 3D printers and offers an extensive list of features, including a comprehensive toolkit for editing your 3D model, its ability to preview print simulations, and much more. Simplify3D is offered on Windows and Mac and, unfortunately, lacks Linux support, which may be a deal breaker for some users. In addition, Simplify3D is not free and requires an investment of $199 to use the product. Keeping that in mind, the team has decided that Simpify3D, although a great software, will not be used due to our limited budget.

3.3.1.5 Octoprint

OctoPrint, yet another slicer software, offers a unique feature foreign to other slicer software in that it has remote operation capabilities. In other words, OctoPrint allows users to oversee their print progress on different devices via an internet connection. So rather than being forced to go to the printer to view progress, one can be on the go. This feature was especially appealing to the 3DPP team. Octoprint can also be used on Windows, Mac, Linux, and Raspberry

Pi. Furthermore, OctoPrint is free to use. Due to the team's intention to design the 3DPP with a Raspberry Pi and the fact that the software is free and offers many enticing features, OctoPrint is perhaps the most suited slicer software for our project.

3.3.1.6 Astroprint

Astroprint is yet another unique slicer software in that to use it. You need a dedicated device called the Astrobox. The Astrobox, similar to Octoprint and a Raspberry Pi, can monitor print status on any device via an internet connection. While Astroprint does offer a free version, it only has 1 GB of storage space for all of your 3D models, which we, as a team, would feel is a hindrance. Its premium version, however, does not have this storage limit and is $9.90 a month. Astroprint is compatible with an internet browser, a Raspberry Pi, or pcDuino. Overall, the team feels Astroprint is different from the slicer software. This is mainly due to the cost of use and the requirement for an additional dedicated device in our printer design.

3.3.1.7 Slicer

Slicer is a pioneer in open-source 3D printer slicing software, responsible for many other software features today. Slicer has comprehensive features, including micro layering bridge detection, mesh cutting, etc. Unfortunately, because Slicer has such a robust list of features, it can be intimidating for beginners in 3D printing. Slicer is free and available on Windows, Mac, and Linux. Slicer is an overall good option for the team to use.

3.3.2 Slicer Decision

After carefully considering each of the slicer software that the team has researched and discussed above, the team believes that OctoPrint is the best option for our design. As mentioned before, OctoPrint's remote monitoring is a desirable feature. In addition, the team already has a Raspberry Pi on hand that we plan to use, which Octoprint can run on. Finally, Octoprint is free and won't add to the overall project cost. While the team could also use other slicer software such as Cura or Slicer, we still feel like Octoprint best suits our needs.

3.4 Renewable Charger

Our project will incorporate solar panels or photovoltaic cells to harness solar energy from the sun and convert this energy to charge the battery of the 3D printer. This is done through the sun emitting particles called photons and absorbing these photons through the PV cells. Since PV cells are made of semiconductor materials, mostly photovoltaic semiconductors, they can take the electrons created from the photons and convert the moving electrons into an electric current by doping the semiconductor material. These flowing electrons will flow through the wires connected from the PV cells to the battery allowing it to charge the battery.

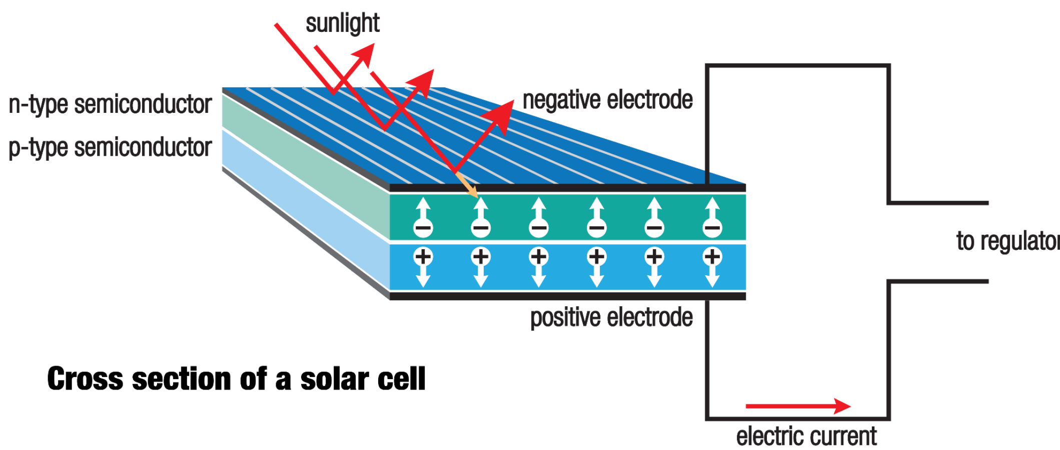


Figure 5: Photovoltaic Cell (from visualcapitalist.com)

When choosing a specific solar panel technology, we had to decide what would best fit our needs for the 3D printer and be cost-effective. Three main types of solar panel technologies are monocrystalline, polycrystalline, and amorphous (thin film). Each of these solar panel types has pros and cons, so as a group, we needed to filter between these three types and determine what would be most efficient for our use case scenario when developing a 3D printer. Monocrystalline panels refer to the type of manufacturing used that uses a single ingot of pure silicon sliced into wafers to create an individual panel. Polycrystalline panels use multiple silicon ingots that are melted, treated, and then cast into a mold of a rectangular panel. Thin film panels are created by applying a photovoltaic substance to the semiconductor substrate.

Table 7: Monocrystalline Panel

Efficiency (%)	**Cost ($)**	**Lifespan**	**Manufacturing Carbon Footprint (g CO2-eq/kWh)**
~24	Most Expensive	Longest	38.1

Table 8: Polycrystalline Panel

Efficiency (%)	Cost ($)	Lifespan	Manufacturing Carbon Footprint (g CO2-eq/kWh)
~20	Moderate	Moderate	27.2

Table 9: Thin Film Panel

Efficiency (%)	Cost ($)	Life Span	Manufacturing Carbon Footprint (g CO2-eq/kWh)
~19	Least Expensive	Shortest	21.4

Table 10: Group Choice of Solar Panel Type

Type of Sonar Panel	Efficiency (%)	Cost ($)	Lifespan	Manufacturing Carbon Footprint (g CO2-eq/kWh)
Monocrystalline	~24	Most Expensive	Longest	38.1
Polycrystalline	~20	Moderate	Moderate	27.2
Thin Film	~19	Least Expensive	Shortest	21.4

3.4.1 Renewable Charger Options

We could go with many options when choosing solar panels, but we needed a solar panel that could provide 20 watts of power or, more incredibly since we are using a 24-volt battery. We also need to look from an ergonomic standpoint at the solar panel, including having a low weight, small dimension size, and portable enough for easy travel and setup. Since, as a group, we have chosen to use the panel, this will allow us to use an efficient yet cost-effective solution to charge a battery for the 3D printer.

We have researched many solar panels, including the three main types, and found three viable options we would like to incorporate into the 3D printer. The specific parts that could be possibly included in our project have the Suner Power SP-50W, which is a 50-watt polycrystalline solar panel, the Newpowa NPA30S-24I, which is a 30-watt monocrystalline solar panel, and finally the Aleko SP20W24V-AP 20 watt monocrystalline solar panel. We extensively researched these three parts and considered efficiency, price, dimensions, and weight parameters.

3.4.1.1 Suner Power SP-50W

Our first option for a solar panel is Suner Power's SP-50W, a 50-watt polycrystalline panel configuration. This extensive solar panel outputs the most power of all three panels at 50 watts. The dimensions of this panel are not appealing when considering we want a portable and lightweight setup for the 3DPP, and also, the high weight of 7.7 lbs reduces the appeal of using this specific panel. Also, the panel for this selection is polycrystalline, which may not be as efficient as the monocrystalline options we have included.

Table 11: SP-50W Key Parameters

Type of Solar Panel	Max Power Rating (W)	Dimensions (inches)	Weight (lbs)	Power Max Current (A)	Operating Temperature (°C)	Cost ($)
Polycrystalline	50	23.4 x 20.5 x 1	7.7	2.78	-40 to 85	89.95

3.4.1.2 Newpowa NPA30S-24I

Our second option for a solar panel is Newpowa's NPA30S-24I, a 30-watt monocrystalline panel configuration. This panel explicitly supplies the second largest power output o96+f all three boards at 30 watts which will be plenty for the entire 3DPP system. The panel for this configuration has a 30-cell total panel that is all monocrystalline, which will give high efficiency, precisely what we need for the 3DPP. This is a strong contender for the particular solar panel we may choose due to its price point and overall quality. The only disadvantage with this solution is that we would need to create our voltage regulator and possibly inverter solution when charging the battery.

Table 12: NPA30S-24I Key Parameters

Type of Solar Panel	Max Power Rating (W)	Dimensions (inches)	Weight (lbs)	Power Max Current (A)	Operating Temperature (°C)	Cost ($)
Monocrystalline	30	19.29 x 14.37 x 1.1	4.78	0.85	-40 to 85	46.99

3.4.1.3 Aleko SP20W24V-AP

Finally, our last option for the solar panel is Aleko's SP20W24V-AP, a 20-watt monocrystalline panel configuration. This panel has the lower power output and the lowest dimensions and weight overall, which would be excellent for portability and setup for our 3DPP system. The only disadvantage we can see is that we would like something slightly over 20 watts of power to generate overhead in case we need to change the circuit configurations if something does not operate correctly. When looking at the other solar panel configurations above, this is also a strong contender for being the panel to charge the 3DPP's battery due to its small size and power generation.

Table 13: SP20W24V-AP Key Parameters

Type of Solar Panel	Max Power Rating (W)	Dimensions (inches)	Weight (lbs)	Power Max Current (A)	Operating Temperature (°C)	Cost ($)
Monocrystalline	20	12.9 x 7.8 x 1.9	4	0.45	-40 to 85	69.00

3.4.2 3DPP Solar Panel Selection

Table 14: 3DPP Solar Panel Final Selection

Manufacturer	Part #	Type of Solar Panel	Max Power Rating (W)	Dimensions (inches)	Weight (lbs)	Power Max Current (A)	Operating Temperature (°C)	Cost ($)
Suner Power	SP-50W	Polycrystalline	50	23.4 x 20.5 x 1	7.7	2.78	-40 to 85	89.95
Newpowa	NPA30S-24I	Monocrystalline	30	19.29 x 14.37 x 1.1	4.78	0.85	-40 to 85	46.99
Aleko	SP20W24V-AP	Monocrystalline	20	12.9 x 7.8 x 1.9	4	0.45	-40 to 85	69.00

After carefully considering all the necessary parameters regarding the solar panel, we have decided to go with the Newpowa NPA30S-24I due to its low cost and moderate dimensions while supplying ample power and being a more efficient type of monocrystalline panel. This panel will best fit our needs without sacrificing expenses, so we can use extra funds to make any necessary modifications or electronic parts, such as an inverter or voltage controller.

3.5 Battery Pack

Our project will incorporate an integrated battery pack as a way to store the energy created from the solar panel and be able to charge the 3DPP system. The process of the battery harnessing the power generated through the solar panel is caused by converting the chemical energy from the electrons from the solar panel into electrical energy. This chemical reaction causes one end of the terminal to become negatively charged, and the other end becomes positively charged. With the above mentioned process, we have a difference in charge called the potential difference, which will allow us to connect a wire to both terminals to harness the created electrical energy.

A battery consists of two metals and an electrolyte, a liquid or dry powder that can conduct electricity. When the two electrodes are integrated into a circuit, the positive ions are attracted to the negative electrode, called the cathode and the negative ions are attracted to the positive electrode, called the anode. With this process, the negative electrons in the circuit's wires are pushed around the entire circuit by the difference in charge of the positive and negative terminals, which transfers energy to all components in the circuit.

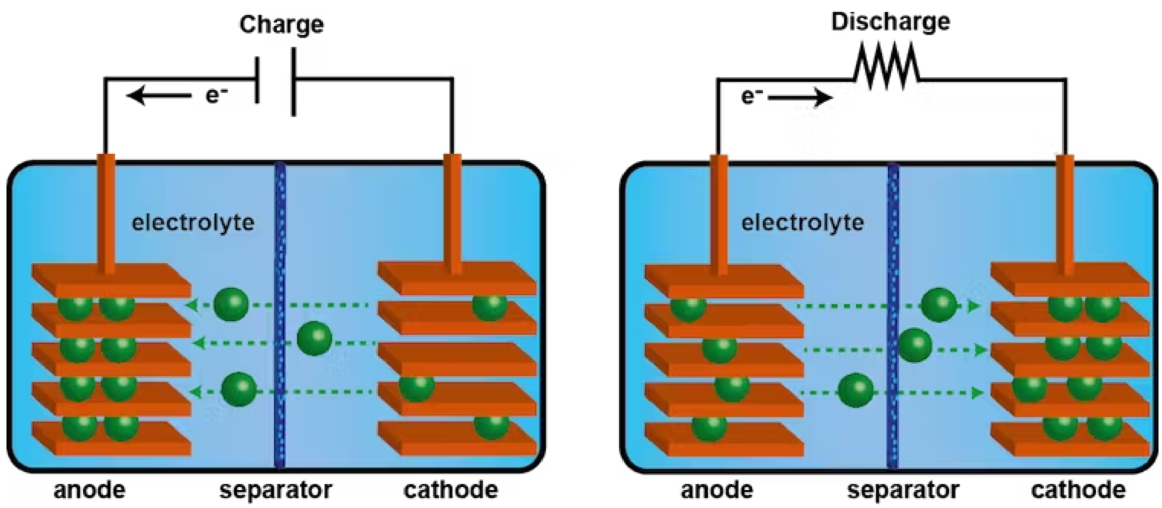


Figure 6: Battery Internal Schematic (from theconversation.com)

The portable 3D printer will have an integrated battery pack with an untethered power supply to allow the feature of easy portability and ease of access from anywhere and no need to worry about a tethered power source. When dealing with batteries in general, three main categories are primarily used depending on the application needed. The three types of batteries include alkaline, nickel metal hydride (NiMH), and lithium-ion. All three types of batteries have pros and cons when considering the application we need specifically for the 3DPP.

Alkaline batteries are single-use batteries that are non-rechargeable. For this reason, we will not use alkaline batteries because we need a solution that allows charging since we want a non-tethered power solution. Next, the NiMH battery is an older technology that allows for renewable charging, but the only caveat of using this technology is that the batteries take a long time to fully charge and degrade in power production the more they are recharged. Finally, we have lithium-ion batteries, a newer technology primarily used in laptops and phones. Lithium-ion batteries allow for quick charging and consistent power output throughout their entire lifetime, but they are more expensive than NiMH batteries due to these exciting features. So for these reasons, we will only consider options between the NiMH and lithium-ion batteries since we need a chargeable option for the entire 3DPP system to have maximum portability.

Table 15: NiMH Battery

Specific Energy (Wh/kg)	Internal Resistance	Charge Time (hr)	Cell Voltage (V)	Charge Temperature (°C)	Toxicity	Coulombic Efficiency (%)	Cost ($)
60-120	Low	2-4	1.2	0 to 45	Low	~70 Slow Charge ~90 Fast Charge	Moderate

Table 16: Li-ion Battery

Specific Energy (Wh/kg)	Internal Resistance	Charge Time (hr)	Cell Voltage (V)	Charge Temperature (°C)	Toxicity	Coulombic Efficiency (%)	Cost ($)
90-120	Very Low	1-2	3.2-3.3	0 to 45	Low	99	High

3.5.1 Battery Pack Options

When researching options for either a NiMH or Li-ion battery, we have to consider the parameters for the power needed for our system to operate efficiently. Therefore we need a battery that can supply a 24-volt charge and has a battery life of at least 20,000 mAh or greater. We would like the battery to be as portable as possible when factoring in weight, dimensions, and ease of connectivity with the 3DPP system. The dimensions of the actual battery pack will be essential because we would like to create a housing inside the system so that the battery is hidden and provides easy access to connectivity and troubleshooting.

With all the research on specific battery options, we have developed three options with pros and cons depending on how we implement them into the 3DPP system. Our first consideration is the Aegis Energies ALF-024020P, a 24-volt, 20 Ah lithium-ion battery. The second option we decided on was the QZF B07Y35LXYY 24-volt, 10 Ah electric bike lithium-ion battery. Finally, the last option is the TalentCell PB240B1 24-volt, 42.98 Ah lithium-ion battery. We searched extensively for some NiMH products, but the Li-ion batteries are popular today.

3.5.1.1 Aegis Energies ALF-024020P

Our first consideration is the Aegis Energies ALF-024020P, a 24-volt, 20 Ah lithium-ion battery. This is the largest battery of all our selections and also the most expensive. The specific lithium-ion technology this battery uses is lithium iron phosphate which is one of the best configurations for a battery for power delivery and efficiency. Overall, we don't think this will be the heart of the 3DPP due to its high dimensions and weight of 4.8 lbs, making it hard to be portable when factoring in other parts of the 3DPP system.

Table 17: ALF-024020P Key Parameters

Type of Battery	Voltage (V)	Capacitance (Ah)	Dimensions (inches)	Weight (lbs)	Charge Temperature (°C)	Cost ($)
Li-ion	24	20	8.3 x 4.9 x 3.7	4.8	0 to 45	249.99

3.5.1.2 QZF B07Y35LXYY

The second option we decided on was the QZF B07Y35LXYY 24-volt, 10 Ah electric bike lithium-ion battery. With this option, we have a lower capacitance of only 10 Ah, which means that the 3DPP will not be able to operate for the maximum possible time. But with this option comes lower dimensions and reduced weight, giving it an advantage over the first option. This could be a contender for the heart of our system if we can't find another solution and have to sacrifice the runtime of the 3DPP.

Table 18: B07Y35LXYY Key Parameters

Type of Battery	Voltage (V)	Capacitance (Ah)	Dimensions (inches)	Weight (lbs)	Charge Temperature (°C)	Cost ($)
Li-ion	24	10	5.12 x 3.55 x 2.76	4	0 to 45	159.99

3.5.1.3 TalentCell PB240B1

Our last option is the TalentCell PB240B1 24-volt, 42.98 Ah lithium-ion battery. This battery would provide us with about twice the capacitance of the first selection, which would be fantastic for runtime for the 3DPP system. This configuration is ergonomically the smallest out of all options and weighs the least. This would be a tremendous component to use with our system, and we incorporate this battery into the solar panel for complete renewable power. This would be the best of all possible solutions we have researched, and it is also the cheapest of all the selected options, which also helps us overall with the budget.

Table 19: PB240B1 Key Parameters

Type of Battery	Voltage (V)	Capacitance (Ah)	Dimensions (inches)	Weight (lbs)	Charge Temperature (°C)	Cost ($)
Li-ion	24	42.98	1.9 x 3.2 x 6.3	1.8	0 to 45	114.99

3.5.2 3DPP Battery Pack Selection

Table 20: 3DPP Solar Panel Final Selection

Manufacturer	Part #	Type of Battery	Voltage (V)	Capacitance (Ah)	Dimensions (inches)	Weight (lbs)	Charge Temperature (°C)	Cost ($)
Aegis Energies	ALF-024020P	Li-ion	24	20	8.3 x 4.9 x 3.7	4.8	0 to 45	249.99
QZF	B07Y35LXYY	Li-ion	24	10	5.12 x 3.55 x 2.76	4	0 to 45	159.99
TalentCell	PB240B1	Li-ion	24	42.98	1.9 x 3.2 x 6.3	1.8	0 to 45	114.99

After carefully considering all component options with the team, we have decided to go with the TalentCell PB240B1 due to its combination of excellent energy capacitance and relatively small dimensions and weight. Overall, This checks all our constraints and will help drastically with planning to implement this component into the 3DPP. It will also provide excellent power efficiency since it utilizes the Li-ion battery technology, providing much faster overall charging than all other options we have explored.

3.6 Voltage Regulator

The evolution of voltage regulators has played a crucial role in the development and efficiency of electrical systems. In our project, an integrated battery pack is essential for storing the energy generated by the solar panel and facilitating the charging of the 3DPP system. The conversion of chemical energy from the solar panel's electrons into electrical energy is the fundamental process behind the battery's ability to harness power. This chemical reaction induces a charge imbalance, causing one terminal to become negatively charged while the other terminal is positively charged. This disparity in charge, known as the potential difference, enables the connection of a wire to both terminals, allowing us to capture and utilize the generated electrical energy effectively. Voltage regulators will allow us to fully optimize the voltage utilization for certain parts of the 3DPP system since some portions may only need 12V or 5V.

Voltage regulators, vital for maintaining a stable and consistent voltage supply, come in various types, each with unique characteristics. The first type is the linear voltage regulator, which employs a series transistor to regulate the voltage. This type of regulator is known for its simplicity and low cost, making it suitable for a wide range of applications. However, linear voltage regulators are less efficient than other types, as they dissipate excess energy in heat.

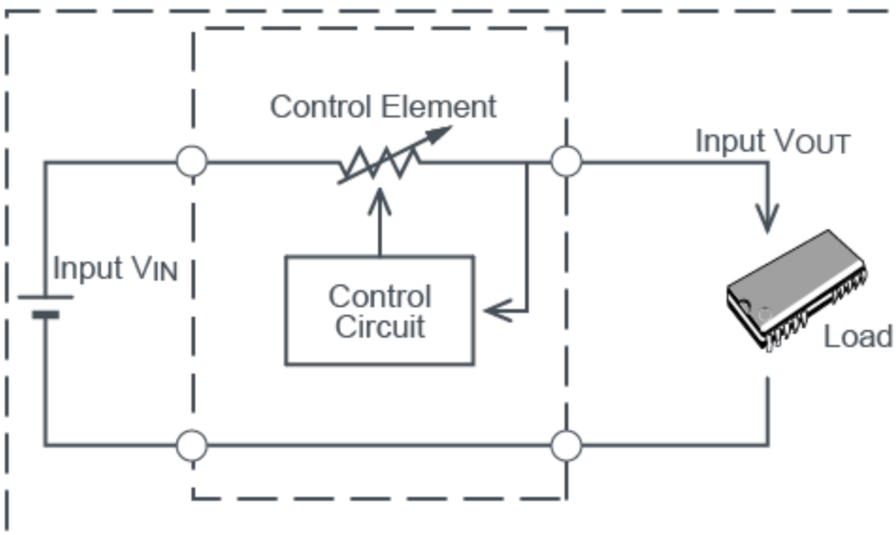


Figure 7: Linear Voltage Regulator Diagram. (rohm.com)

The second type of voltage regulator is the switching regulator, which employs a switching element, typically a transistor or a semiconductor switch, to control the voltage. Switching regulators are renowned for their high efficiency and ability to handle large voltage differentials. They utilize pulse width modulation (PWM) or similar techniques to regulate the output voltage, making them suitable for high-power conversion efficiency applications.

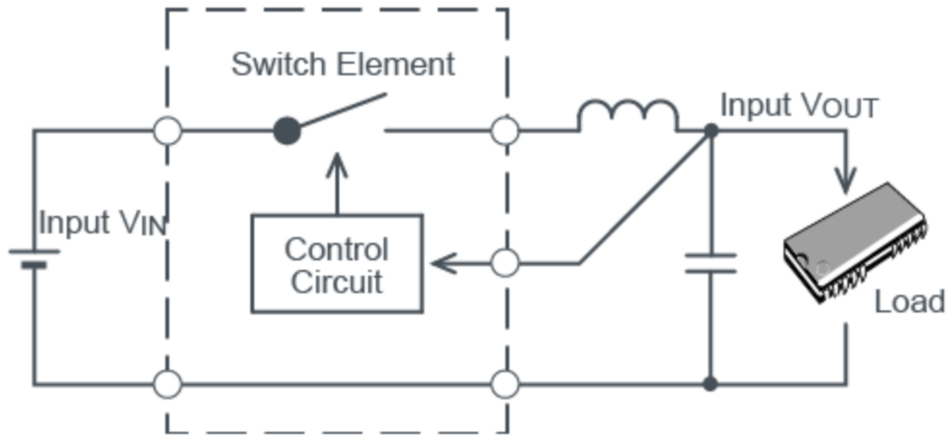


Figure 8: Switching Voltage Regulator Diagram. (rohm.com)

Lastly, we have the programmable voltage regulator, which offers the flexibility of adjustable output voltages. These regulators incorporate digital control techniques, often providing remote sensing and voltage margin features. Programmable voltage regulators are commonly utilized in complex systems requiring precise voltage regulation, such as in data centers and advanced electronic devices.

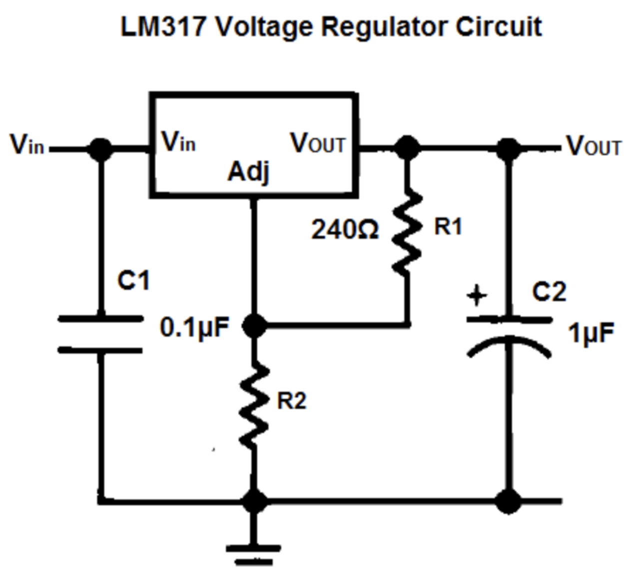


Figure 9: LM317 Programmable Voltage Regulator Circuit (learningaboutelectronics.com)

We want an efficient voltage regulator for our design. In other words, since the batteries supply limited power, we want a voltage regulator that won’t waste energy by dissipating it as heat. Therefore, we likely will not use linear voltage regulators for this design. While switching regulators are more efficient than linear regulators, the team ultimately utilized programmable voltage regulators.

We want the voltage regulation to be as precise as possible, and the switching regulator is less effective than the programmable one for this requirement.

3.7 3D Printer Designs

Researching and ultimately deciding on which physical 3D printer design format we will implement is one of the most critical aspects of our research, as this will influence almost all aspects of our project, such as the overall size, foldability, accuracy, portability, and printing space. Some of the more popular 3D printing design layouts are Cartesian, Delta, Polar, SCARA, and Belt designs. In the following parts of the 3DPP research report, we will discuss how each design works and some pros and cons of each 3D printer design format.

3.7.1 Cartesian

Cartesian design is the most popular design that 3D printers implement, meaning that there is tons of research and forums that we can use when designing our cartesian design. This design was named after the cartesian coordinate system using an X, Y, and Z axis. As the name implies, this is precisely how the printer works. It utilizes three stepper motors in the X, Y, and Z directions to move the printer nozzle around the printing area.

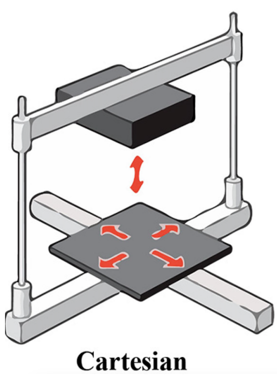


Figure 10: Cartesian 3D Printer Design

Some of the better features the Cartesian design could bring the 3DPP team is that it is the most popular design giving our team tons of research we can utilize on this design. Not only is this design very popular in the 3D printing community, but it also brings stability because of its traditionally rigid structure, which minimizes chances of errors due to vibrations and or wobbling of the printer. This design has some factors that the 3DPP team could have found desirable, such as speed. This design is known to be traditionally slower than others because it requires more significant parts to be moved, such as the entire bed platform. Along with its speed being slower compared to other designs, due to its rigid structure, it could be more compact and able to fold within itself. This is a feature that the 3DPP team finds extremely important as we continue to research 3D printer designs.

The Cartesian printer design offers our team some desirable features, such as its stability and plentiful resources due to its popularity. But our team has decided to move on from this design because of its slower printing speed. As mentioned before, we need to make our printer able to print as fast as possible due to the limited amount of battery life. Along with its slower speed, the design is bulky and rigid. This feature of the design does have its upsides. But because we are designing a printer to be portable, this is ultimately a negative part of the design.

3.7.2 Delta

The delta 3D printing design is an intriguing alternative to the more familiar Cartesian method. This design's printing bed is generally circular compared to the Cartesian printer's square printer bed. It works by utilizing three arms in a triangular formation, hence the name delta, that controls the movement of the print head. The print bed remains stationary in this design, while the three arms can move in all three dimensions.

The Delta design possesses a key design feature that the 3DPP team found desirable: speed and precision. Delta printers are known for their speed. The lighter-weight arms that control the movement of the printer head allow for faster movement than in Cartesian printers, which often move heavier components. The reason why the design is so precise is because of its parallel manipulation. This means the print head is always perfectly level with the print bed and remains that way throughout the print. A feature that the 3DPP team found undesirable in this design is its circular print bed. This factor may limit the types of objects our printer can print, especially if the thing we want to print has a large, square base. Along with the circular printing base, not all slicer software support the delta design method. This could limit the software options available to our team.

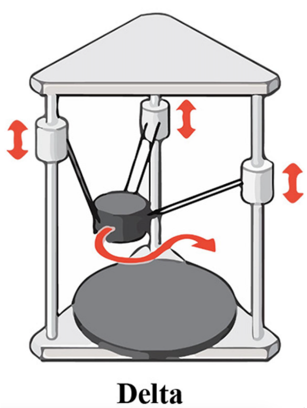


Figure 11: Delta 3D Printer Design

he Delta 3D printer design is indeed intriguing and distinctive, embodying one of our key project requirements - the ability to deliver faster print times. This design, with its three arms moving in harmony to create objects, is not just efficient but also visually appealing, making it a potential crowd-pleaser in our high school engineering showcase.

However, as we've previously discussed, every design comes with its own set of challenges. The circular print bed of the Delta design, while innovative, isn't compatible with some slicer software. This could limit our ability to use certain software tools, potentially impacting the versatility of our project. We might need to spend extra time finding and learning to use compatible software, which could slow down our progress.

3.7.2 H-Bot

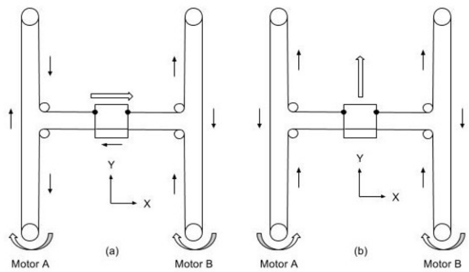


Figure 12: H-Bot 3D Printer Design

The following 3D printer design type the 3DPP team will research is the H-Bot printer design. This design is a spinoff of the already discussed Cartesian design printer but with a unique mechanical structure that separates it from the more traditional Cartesian methods. This design utilizes two parallel axes in the shape of an 'H,' hence the name. These axes are connected by a cross member, creating the 'H' shape. The print head is mounted on this cross member and can move along the X and Y axes. For the Z axes, traditionally, the H-Bot moves the print bed up or down similarly to the cartesian design.

Some pros that the 3DPP team believes would be beneficial for our portable printer design that the H-Bot possesses are its speed, efficiency, and compact design. These are all significant factors the 3DPP team needs in our configuration. This design gets its speed and efficiency from controlling the X and Y axes simultaneously using the same motor. This benefits the more traditional Cartesian method because only one stepper motor can be used instead of two. This is a crucial factor as power consumption is significant for our project. Along with this design being fast and efficient, it is compact. This is because of the design's efficient use of space and unique mechanical components arrangement.

Overall the H-Bot design is a severe contender for the 3DPP team as this design fits many of our design requirements because of this design being fast, efficient, and compact. These are all huge factors the 3DPP team considered when choosing the ensign we want to implement. As a team, we like this design and what it could offer regarding a portable 3D printer design, but it has its downsides. These downsides include limited torque because the X and Y axes only use one motor instead of two. This can lead to accuracy issues if not appropriately maintained. Another negative side of this design is its complexity and few resources, particularly this design. This could be a big issue for the 3DPP team as we want to focus more on making the 3D printer portable rather than having to focus on the actual printer design itself.

3.7.3 Polar

The polar 3D print design differs from the other designs our team has already discussed. The design is known for being a relatively cheaper design as well as having to utilize fewer motors. This design works by rotating on a turntable, and this rotation is used to control the X axes. The print head moves radially in and out from the print bed's center and moves up and down for the Z-axis. This design takes advantage of the rotation of the print bed to move the print head around the object it is printing.

One key feature of the Polar 3D printer design that the 3DPP team found desirable is its efficiency. Similar to the H-Bot design, the Polar design does not need to use the traditional three-stepper motors to control where the print head goes. As discussed previously, this is a significant factor for our team because, with our design, we need to save as much power as possible to keep our battery charged for as long as possible. Also, the polar design is relatively more straightforward to physically design compared to other methods as it only requires one motor for the rotational movement and one motor for the radial motion. One negative attribute of the Polar design is requiring users to use a more complex firmware option. This is because the traditional 3D printing firmware translates the G-Code into X, Y, and Z movements, whereas this design needs to solve the G-Code into polar coordinates.

The 3DPP team enjoyed researching this design as it is a very unique design. Our team will only pursue this design if it requires more advanced firmware options. As stated in the Firmware research part of our report, we have decided to use the Klipper firmware. This firmware does not support translating the cartesian G-Code coordinates into polar coordinates.

3.7.4 Belt

The Belt 3D printer design introduces a novel approach to 3D printing, utilizing a conveyor belt as a print bed and printing at a 45° angle. This unique configuration allows for theoretically infinite Z-axis printing, enabling the creation of objects of any length. Key advantages include printing multiple identical parts in succession without manual intervention and printing overhangs and bridges in the X, Y, and Z axes without additional support.

However, challenges include potential filament compatibility issues and the added complexity of creating models in CAD programs that can leverage the infinite Z-axis. This design increases the learning curve for our team and may only be suitable for some applications.

Despite the Belt 3D printer's intriguing capabilities, the 3DPP team has chosen not to pursue this design due to firmware compatibility concerns. As of 2023, our selected Klipper firmware does not explicitly state compatibility with Belt 3D printers. Therefore, we have explored other design options better aligned with our firmware choice.

3.7.5 Printer Design Decision

When considering all aspects of printer design, it is important to keep in mind the principles of the engineering design process. We began by considering the objectives our project must meet, the first and foremost being that a battery and renewables must power it. This by itself led to many of the other sub-objectives in this project. With limited power comes limited print time, meaning we must prioritize print speed, nozzle flow, and power consumption while sacrificing as little accuracy and other performance metrics as possible. Many of our design decisions, such as deciding to have the printer print upside down, resulted from these objectives. Having a printer print upside down allows the center of mass of the printer to be closer to the ground, which minimizes jitter and allows us to increase our print speed without sacrificing as much accuracy as we otherwise would with a traditional FDM design.

For the most part, we are playing it safe with our design, since powering the printer for a substantial amount of time is already a challenge. Although we may like to design a belt printer, a traditional FDM printer modified to fit our needs is the best bet for completing our objectives for this project.

3.8 3D Printer Host Software

3D printing host software is built for 3D printers so users can control and monitor vital information about their 3D printer while printing. The 3D printer team found this software to be vitally important to our project as it serves as a way for our team to communicate to the user via a local network connection what is happening with their print. Along with this, an essential requirement the 3DPP team has is to display the voltage and battery time to the user; with this software, we can do just that. Our team will discuss several great 3D printer host software in the following sections and explain how we chose the right one.

3.8.1 Octoprint

Octoprint is one of if not the most popular printer host software on the market. This means it has a wide range of community help if our team ever needs help implementing it with our project. With a large community following, Octoprint offers the 3DPP team various features that could help us create a portable 3D printer. Some of these features include; a G-Code viewer, which means the user can see their print in real-time on the web service as the print is being printed. Also, Octoprint has various plugin systems that allow developers to extend the software's basic functionality.

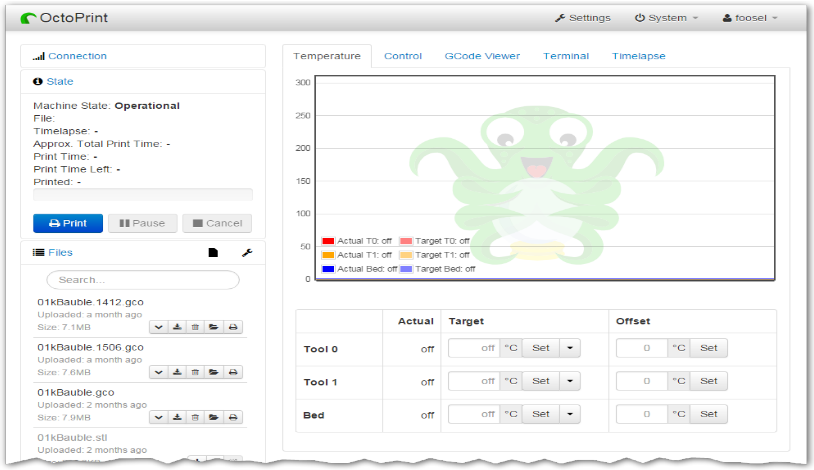


Figure 13: OctoPrint Menu

With all of its popularity and very cool features, the Octoprint printer host software brings to the 3DPP team the most valuable feature it brings is the REST API feature. This feature allows developers to integrate and interact with other software systems or scripts. Using this REST API allows our team to communicate with the API via a Python code located on the Raspberry Pi and display the voltage of our battery system, an essential requirement our team sets when looking for a host software. This API service will also allow us to send various G-Code commands based on the battery voltage level, such as stop or resume printing.

Octoprint gives our team a lot of great features that we will need to fill our requirements, such as the ability to communicate with Python code and the ability to monitor the print from a web application as the print is printing. The one downside of Octoprint is that it is not explicitly designed for the firmware our team picked, which is the Klipper firmware. Even though this software is excellent, we have decided to look at it more to make a final decision.

3.8.2 Pronterface

Pronterface is a free open-source printer host software with a straightforward and easy-to-use user interface. This software is not the most widely used host software for 3D printers, but it offers our team many features similar to Octoprint, which is known to be very popular. Some of these features include having a G-Code terminal which allows for direct input of G-Code commands to the printer. The printer controls will enable the user to control the primary functions of a 3D printer, similar to Octoprint, the G-code viewer.

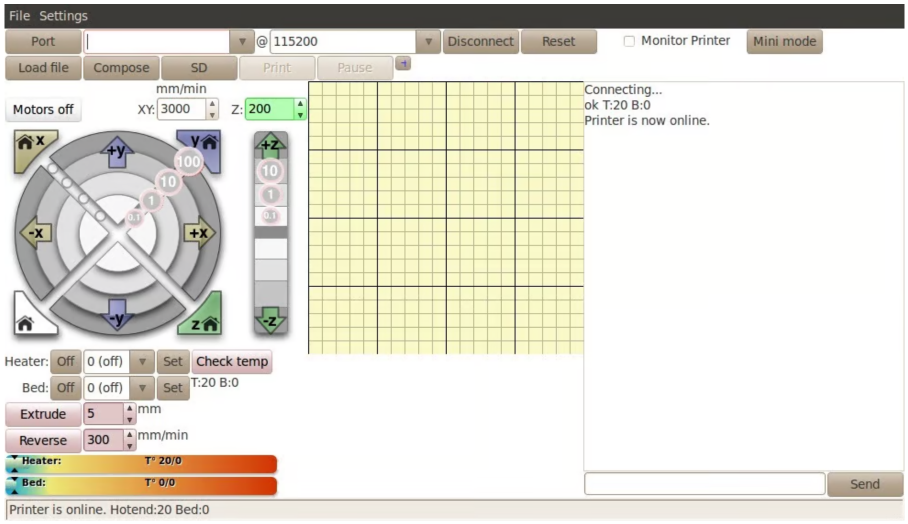


Figure 14: Pronterface Menu

This printing host software has many benefits, such as a straightforward user interface. As mentioned, Pronterface does have many similarities to Octoprint. However, the most prominent negative it does have is that it's not meant to be run off of a Raspberry Pi but instead via a wired connection to a computer with Pronterface installed, for our project where portability is one of if not the most significant requirements for our project this host software will not cut it for the 3DPP team. We are looking for software that fits well with our 3D printer control board, the SKR Pico, and the firmware we have elected to use, which is the Klipper firmware, and unfortunately, this easy-to-use host software will not cut it for our project.

3.8.3 Mainsail

Mainsail, a budding hosting software for 3D printers, has caught our attention due to its promising features and compatibility with our chosen firmware, Klipper. As per the information available on their website, the team behind Mainsail has designed it to offer a responsive, clean, and efficient interface for managing 3D printing operations. This is a crucial aspect for us, the 3DPP team, as we are high school engineering students who value user-friendly and efficient tools that can streamline our project work.

One of the main reasons we decided to delve deeper into Mainsail is its full integration with Klipper, our firmware of choice. This compatibility is a significant advantage as it ensures seamless operation and reduces the likelihood of encountering technical issues. It's like having a pair of puzzle pieces that fit perfectly together, making our project workflow smoother and more efficient.

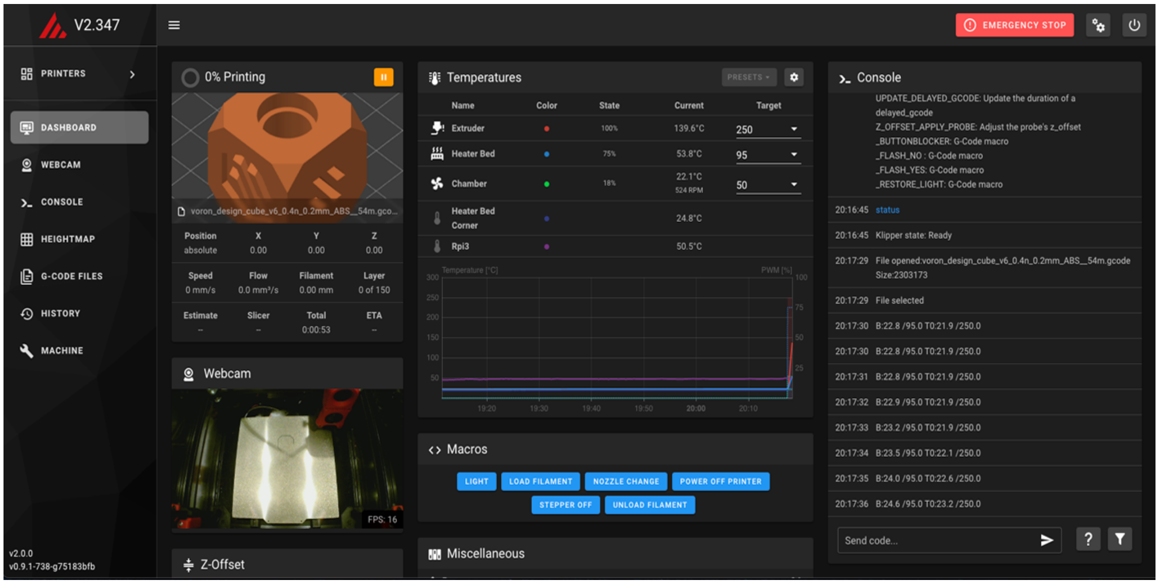


Figure 15: Mainsail Menu

A more in-depth description of some of the features the 3DPP team found desirable is that Mainsail offers touchscreen display capabilities, as stated above. This is a significant requirement for our team because we want to create a product where the user can control the printer as quickly as possible. Also, this allows the user to control the printer directly from the display without requiring a wired connection to a computer. Our team also liked that this software was designed specifically for our firmware of choice. This is helpful for us because while implementing this software for our printer, it could mean fewer road bumps. Finally, the most significant factor we liked about Mainsail is that it is similar to Octoprint. It does not need a wired connection to work. This software can work all over a Wi-Fi connection from the Raspberry Pi.

Overall the 3DPP team liked this software and found its features and compatibility with Klipper to be very helpful toward the goal of a portable 3D printer. Ultimately the team has decided to pick Octoprint or Mainsail, which will be discussed in the following section. The downside to Mainsail is that it is a newer hosting software with a smaller community than Octoprint.

3.8 3D Printer Host Software Decision

After conducting the previous research, the 3DPP team has chosen Mainsail as our printer host software. As mentioned above, we liked how the software was designed explicitly with Klipper in mind and was compatible with an LCD touchscreen display. Something we should have said but is something to take note of about Mainsail is that it is similar to Octoprint. It does have API compatibility. This means that using this API service Mainsail provides, we can communicate to it via the Raspberry Pi and send prompts to the software depending on the battery's voltage level. With all these features in mind, we believe that Mainsail gives our team the best ability to create a portable 3D printer.

3.9 3D Printing Filament

In our research, we have delved into various software and hardware components crucial for the efficient operation of our 3D printer. However, another critical aspect of 3D printing is selecting the appropriate filament type for our portable 3D printer. A thermoplastic filament melts at a specific temperature and is extruded through the printer's nozzle onto the print bed, creating the desired 3D object. Several significant factors will guide our team's filament selection. First, printability is paramount; we aim to choose a user-friendly filament for our project's users. Second, the cost is a significant consideration. As discussed in previous sections, we strive to keep the overall cost of our printer competitive with other small printers on the market today.

Material properties also play a crucial role in our decision-making process. We are particularly interested in the strength of the material and, importantly, its melting temperature. The latter is vital for our portable 3D printing project, as we aim to operate on minimal current. Heating the cartridges to higher temperatures would require a higher current draw, which we aim to avoid. Lastly, we are mindful of the environmental impact of the filament we choose. Given that our printer will extensively use plastics, which are typically not environmentally friendly, we are inclined towards a more eco-friendly option. This comprehensive approach ensures we select the filament that best aligns with our project's objectives and values.

The 3DPP team will be researching various possible filament options which include PLA, ABS, PETG, TPU and Nylon. These filament options are widely used in the 3D printing industry all having their own specific qualities which make them a feasible option for our team's portable 3D printer. In the following sections we will be taking a closer look at each filament to see which will work best for the 3DPP team.

3.9.1.1 PLA

Polylactic Acid, or PLA, is a favorite among the 3D printing community, and for good reason. Its popularity stems from several key factors. Firstly, PLA is relatively easy to print, as it does not require a heated printing bed. This feature is particularly beneficial for our team, as while we plan to incorporate a heated bed in our design, it is a power-intensive component that could significantly reduce the print duration.

Moreover, PLA melts at a lower temperature compared to other filaments, typically between 180 to 220 degrees Celsius. This attribute is crucial for our project, as a lower melting point means less energy consumption, leading to longer print times. Another advantage of PLA is its minimal warping and shrinkage. Warping refers to lifting and curling a 3D-printed object's corners or edges away from the print bed during printing. Shrinkage, on the other hand, is the reduction in the size of a 3D-printed object after it cools down post-printing. These issues are less prevalent with PLA, making it a reliable choice for accurate prints. Finally, PLA's environmental friendliness is a significant draw. As a biodegradable material derived from renewable resources, PLA is one of the most eco-friendly 3D printing materials available. However, it is essential to note that the biodegradation process is slow and requires specific conditions, so it is not a complete solution to plastic waste.

In conclusion, PLA's low melting temperature, ease of use, and biodegradability make it a standout choice for our team. These features align with our project's energy efficiency requirements, user-friendliness, and environmental consciousness.

3.9.1.2 ABS

Acrylonitrile Butadiene Styrene, or ABS, is the second most popular 3D printing filament. This filament boasts various features, including its notable durability and strength, outperforming the more brittle PLA filament. ABS also offers flexibility, making it an ideal choice for 3D-printed parts requiring a small amount of give. Post-processing of ABS prints is relatively straightforward, with users able to easily sand their prints or use Acetone for a smooth finish.

However, despite these advantages, ABS presents challenges that our 3DPP team finds less appealing. One of these is its heat resistance. ABS melts at higher temperatures than PLA, typically between 210-250 degrees Celsius. As previously mentioned, the ability to print at lower temperatures is crucial for our project. Furthermore, ABS is a challenging filament to work with. It tends to warp and shrink as it cools, potentially leading to issues with print adhesion and layer separation. Despite these drawbacks, ABS remains a contender as a possible filament choice for our team.

3.9.1.3 PETG

PETG, short for Polyethylene Terephthalate Glycol, is a favorite among 3D printing enthusiasts. It is celebrated for its durability and strength. Being more flexible and less brittle than PLA, it is ideal for parts that must withstand wear and tear. One unique feature of PETG is its transparency. While it needs to be clarified, it has a higher level of transparency than most other filaments. Additionally, PETG is relatively easy to print with. It does not warp or shrink as much as ABS, which makes it easier to achieve good bed adhesion and prevent layer separation. It is also food-safe, meaning it is safe for food contact. Lastly, PETG is not biodegradable like PLA, but it is recyclable and does not emit harmful fumes during printing.

Our team found several critical aspects of PETG appealing. Its strength could benefit our intended users, who are engineering students. Its hydrophobic nature, meaning it does not absorb water from the air, makes it easy to store and maintain. However, PETG does have some downsides. Like ABS, it has a higher melting point, which is unsuitable for our project. While it is easier to use than ABS, PETG can be prone to stringing and oozing due to its sticky nature, and it requires a well-tuned printer for optimal results.

In conclusion, our team is keeping our options open. We appreciate many of PETG's features and prefer it over ABS. However, its downsides, such as the higher required melting temperature, are noteworthy. We will continue to evaluate our filament options as we progress with our project.

3.9.1.4 TPU

Thermoplastic Polyurethane, commonly known as TPU, is a unique type of 3D printing filament renowned for its flexibility and durability. Unlike the more rigid structure of PLA, ABS, and PETG, TPU offers a flexible and adaptable quality. Intriguingly, users can control the degree of flexibility in their 3D prints by adjusting the printing speed, temperature, and layer height, but most importantly the infill. Adjusting these features allow for an incredible amount of customization with this filament. In addition to its flexibility, TPU is highly resistant to abrasion, wear and tear, and impact, resulting in durable and long-lasting prints. These features make it ideal for parts that will experience significant stress and wear and tear.

Regarding the printing process, TPU shares many characteristics with PETG, including minimal warping or shrinking. However, it can be prone to stringing and oozing. It also requires a much higher printing temperature, which isn’t ideal for prolonging our print as long as possible, since the increased heat demand increases the draw on our power supply.

Despite its advantages, TPU presents some challenges that make it difficult to use for our project. Like ABS and PETG, TPU requires a higher melting temperature, which is unsuitable for our 3D printer. Despite this, as far as more advanced filaments go, TPU may be next to be integrated due to its versatility. Additionally despite its high power demand, it will not draw as much power as some of the other more advanced filaments discussed in this section.

3.9.1.5 PolyCarbonate/Continuous Carbon Fiber

Polycarbonate is a short-carbon fiber based filament. Short carbon fibers are small pieces of carbon fiber no longer than 6mm, though in filaments it can be in the micrometers. This is opposed to continuous carbon fiber, which is a continuous strand of fiber. Short carbon fiber filaments are thermoplastics with small pieces of carbon fiber mixed into it, while continuous carbon fiber filaments are large lengths covered in a thermoplastic. Due to the nature of the somewhat random spread of short carbon fibers throughout polycarbonate filament, the fractile toughness can be inconsistent throughout a given length of filament. Conversely, continuous carbon fiber offers the same properties throughout the entire roll of filament.

There are many advantages to these filaments, including their increased strength, fracture toughness, and heat resistance relative to traditional thermoplastics like PLA and ABS. This makes it ideal for high stress parts. Continuous carbon fiber is some of the strongest additively manufactured materials that can be used in a FDM printer. For example, one of its possible uses is for the nose cone of rockets, as when continuous carbon fiber is printed inside ceramic filament, it makes up for ceramics' relatively poor fracture toughness, which combined with the ceramics makes for an incredibly heat resistant yet strong material.

While we would love to integrate the ability to use these filaments in our printer, there are unfortunately many limitations to its use. First and foremost, we are hoping to limit the power consumption from our nozzle, which means operating it at lower temperatures. Carbon fiber based filaments have a much higher melting point than traditional thermoplastics. This is not ideal for maximizing our print time, and would only serve to increase the quality of our prints, which is outside the scope of this project. Additionally, carbon fiber filaments are heavier than traditional thermoplastics, which is not ideal for the portability of our printer. These filaments are also prone to warping and require a much hotter build plate in order to adhere to the print bed, which increases its power consumption further. There currently only exist two commercially available printers for printing continuous carbon fiber. The reason for this being that, unlike other filaments and even polycarbonate, continuous carbon fiber has to be cut ever time the print head jogs from one part of the build to the other, since filament extrusion would have to stop, and residual filament is not wanted. In order to cut the carbon fiber, some sort of blade is needed to be mounted on the print head, with slim margins that allow it to slide between the nozzle and the build. This is incredibly difficult to do even for large additive manufacturing companies, so it is most definitely out of the question for us.

Despite its advantages, we determined it is outside the scope of this project to integrate the capability of printing carbon fiber based filaments, and that the demand on our design outweighs any possible benefits to adding this capability to our design.

3.9.2 3D Printing Filament Decision

As mentioned previously, the 3DPP team was looking for a filament that met the following requirements, is cost effective, easy to print with, environmentally friendly and the overall strength of the filament.

Table 21: Flament Types and Printing Temperatures

Filament	**Printing Temperature**
PLA	180-220 degrees Celsius
ABS	220-250 degrees Celsius
PETG	220-250 degrees Celsius
TPU	200-220 degrees Celsius
PolyCarbonate	260-310 degrees Celsius

The table above provides a comprehensive comparison of various filaments that we, the 3DPP team, have researched for our project. One of the key takeaways from this comparison is that PLA (Polylactic Acid) filament requires the lowest printing temperature among all the options. This characteristic of PLA is a significant advantage for our project, and here's why.

One of our project's primary requirements is to achieve longer print times. Lower printing temperatures mean less energy consumption, which directly translates into longer print times. With PLA's lower temperature requirement, our 3D printer can operate for extended periods without draining the battery quickly. This is particularly crucial for our project as we aim to create a portable, efficient, and long-lasting 3D printer.

Table 22: Filament Price for a 1Kg Spool

Filament	Cost For a 1Kg Spool
PLA	$20 - $30
ABS	$20 - $40
PETG	$25 - $50
TPU	$30 - $60
PolyCarbonate	$50 - $80

As shown in the table above again PLA beats out the competition with the lowest average cost for a 1Kg spool of filament. This is important for our project as we try to keep the overall price down for our users as well as for us.

Overall, all of the mentioned 3D printing filaments had some really good features that went with them. This being said though the 3DPP team has decided to go with PLA for our portable 3D printer. This is because of a multitude of factors such as those mentioned above, having a low cost, being easy to print with due to the fact that it does not tend to warp or shrink, which is especially good for new users. Along with this PLA is more environmentally safe then the other options our team considered. But as mentioned previously it does take a long time to biodegrade as well as needing to be recycled via industrial composting facility, so it's not a complete solution to plastic waste. Finally and most importantly as mentioned before it has a low melting temperature allowing our team to save battery life and to be able to print longer.

4.0 Design Constraints and Standards

This section discusses design constraints, national standards our portable 3D printer must adhere to, and how the overall design is impacted. This section plays a vital role in our portable 3D printer's design process due to the constraints and standards involved. Specifically, they will be responsible for narrowing potential options that we will have for the design process, such as software that we can use and components that we may wish to select.

4.1 Constraints

The 3DPP project hosts several engineering requirements that affect the development and implementation of the printer. The constraints that will be identified in detail are the following: Economic, Time, Manufacturability, Environmental, Social, Political, Ethical, Sustainability, Safety, and Presentation constraints. By adhering to the constraints mentioned in this section, we will hold ourselves accountable and design a final product that we, as a team, feel proud to present.

4.1.1 Economic Constraints

The total cost of materials constrains the development of our portable printer. To specify, each component of our design that we do not already own will add to the total cost of our project. The price of these components is subject to the overall economy and availability of materials. For example, the recent global semiconductor shortage has increased costs for several markets, including vehicles, graphics cards, and other printed circuit boards. Our printer utilizes several printed circuit boards, some of which we must design and purchase from a manufacturer that may or may not increase the purchase cost. As a result of this economic constraint, our printed circuit boards will be limited to components that will save as much capital as possible, mainly because the overall target for our total cost of this project, which is mentioned in the House of Quality, is approximately $600. The economic constraint is especially vital to the design process of our project because our team has yet to receive funding or sponsorships from any company. This means that each team member must contribute towards the project's overall price, hence the need to select components that will not only fulfill specific requirements for the printer but also save capital. The economic constraint will also affect the team’s ability to prototype the printer and test several components. To specify, the team must be cautious when installing components to prevent unintentional disassembly, which would cause obtaining replacement components that add to the total cost. The team must also thoroughly research each part to determine the ones that best fulfill our needs before purchasing, rather than obtaining several types of the same component and testing them for suitability afterward, contributing to the total project cost.

4.1.2 Time Constraints

In addition to the economic constraints this project is subjected to, is time. We have limited time to design, prototype, and present the portable 3D printer. Furthermore, each part of the development process of our project is subjected to a timeline for which they must be completed. The overall design process began in May 2023, and our final prototype will be presented in December 2023. From May 2023 to the beginning of August 2023, our team must have all parts identified and purchased, including developing our custom-printed circuit boards and the firmware and software in which our printer will run. This timeline will ensure that there is ample time for prototyping and presentation. That period marks the duration of Senior Design 1. The latter half of August 2023 to the beginning of December 2023 will keep the prototyping of our printer to identify potential issues and improve efficiency, as well as the final design presentation. That second timeline is Senior Design 2. This paper delves deeper into when each portion of our project must be completed in the Administrative Content section before the Appendices.

4.1.3 Environmental Constraints

Recently, renewable energy has been pushed to meet our energy demands. The overall energy policy for several major countries, including the United States, is to become carbon neutral in the next few decades. In addition, several companies have also pushed for renewable energy, such as Adobe, HP, General Motors, Semens, and more. Even individuals on social media sites such as TikTok have been pushing for environmentally friendly practices, participating in trends such as eco-bricking, in which they strive to reduce and sequester plastics out of the biosphere. In short, the world is moving towards clean, environmentally friendly energy. Our team is no different; we consider renewable energy our top environmental constraint. Henceforth, we are designing our 3D portable printer to run on rechargeable batteries recharged by solar energy. Due to this constraint, our design must be able to print for at least 1 hour running off of this battery power for a specified duration. This limits many options and forces the team to be creative in selecting solutions to meet this engineering requirement and environmental constraint. Furthermore, one of our project’s other engineering requirements is to be portable. Since the printer must run on rechargeable batteries that will take up space, other printer components must be smaller to meet that demand.

4.1.4 Social Constraints

Our team is creating this design with engineers and hobbyists in mind, in that they can 3D print something on the go and just want a smaller size, shorter in duration print. As a result, we have the social constraints of creating this printer to be affordable, fast at printing, and as compact as possible. Since the entire idea of creating this project was to make a 3D printer that people like us would want, these social constraints are amongst our most important.

An additional social constraint for our project is the design's ease of use, which is mentioned in the House of Quality diagram at the beginning of this paper. We want this printer to be easy to use in all aspects for anyone who may use our design. In other words, we intend to design the printer so that it is simple to recharge the batteries, refill the print material, and easy to navigate the LCD to monitor print progress, battery life, and start or stop prints. We will adhere to this constraint by programming the display so that it looks simple and the controls aren't complex, as well as having instructions for using other features on the printer.

4.1.5 Ethical Constraints

The market for 3D printing technology has continued to expand since the inception of the 3D printer and will likely continue in upcoming years. Because there are many examples of ready-to-purchase 3D printers, our team has an ethical constraint to ensure we do not infringe upon any existing patents. This means that our design will be solely our own, and any software and firmware we use will be open-sourced, with credit given to the contributors.

4.1.6 Sustainability Constraints

Although our primary engineering requirement for this project is for the printer to be portable, affordable, and fast at printing, our intention and goal is to design a printer that will last. We would like to build something that will last for many uses without the user having to replace the printer. Furthermore, due to the nature of the printer in that, we will have a component that regularly heats to high temperatures. We must design the printer so that it can function in such an operating environment for many uses. Since our design will also contain fragile electrical components, we intend to build sturdy housing around them so that the internal components will survive if the printer experiences an unintentional displacement.

4.1.7 Manufacturability Constraints

During the design process of our 3D printer, one manufacturing constraint to note is the components' availability. In other words, for each element we plan on using, we must consider how much each part is available for purchase. As mentioned in the economic and time constraints, a semiconductor shortage recently affected several significant markets, including the printed circuit board market. As such, many components we need may not be available, or there may be only a limited amount. This is especially important to take note of during the design process of our printed circuit boards. If we use Eagle AutoCAD to design the panels and choose board components using their software, we need to be observant of the availability of each element. Eagle explicitly states the quantity of each part available so that we don't choose a component only to find out that it is out of stock later. If our design ever goes out to market, we must carefully consider which manufacturers we purchase components from to avoid future supply chain issues. Choosing pieces on Eagle or any other AutoCAD software that includes component information will help in this endeavor, as we can select components on the software that have stocks in the thousands rather than those that have supplies in the hundreds or less.

4.1.8 Presentation Constraints

Due to the nature of our project, in that it is a 3D printer that will take some time to print an object, we have a presentation constraint. In other words, to fulfill our team's requirements for the final project presentation, we must find ways to make our presentation more interesting and demonstrate our printer's successful functioning. We will likely record several videos of our printer printing something to stay within this constraint. We will then edit the videos to be time-lapses and thus a short video. This is a potential solution for us because one of our design requirements for the printer is to be able to print for at least an hour, and the overall presentation will need to be longer to demonstrate a print of such duration. Instead, in addition to the edited videos, we will also create an AutoCAD file that will be small enough so that we can print it during the presentation, and it will finish promptly.

4.1.9 Political Constraints

During the design process of our project, we conducted extensive research about 3D printers, some of which were political. Our study found some political controversy regarding the use of 3D printers. Specifically, we found several news articles about how people use printers to create ghost guns. Ghost guns are guns that are not registered to the proper authorities. People also use these printers to print guns because they are plastic and easy to hide. Felons who are legally not allowed to purchase or own a firearm and those who can but want to hide it from authorities would find this especially appealing. However, despite this controversy, our team concluded that it isn't relevant to the design of our portable 3D printers, as it isn't our intention to design a printer for people to print guns on, but instead, it is our intention for people such as hobbyists or engineers to print a quick project on the go. The process of designing a gun on Autocad to print on our printer is entirely unassociated with the team.

4.2.0 Safety Constraints

Our design utilizes many components, some of which have proven to pose a potential safety risk. The elements that pose the most trouble are the batteries and the hot end. Our team will use lithium-ion batteries, known to swell with age and potentially explode if punctured. Such an event has been widely documented across several products that utilize lithium-ion batteries. One product that was notorious for defective lithium-ion batteries destroying the phone, damage to property, and human injury, was the Samsung Galaxy Note 7. We strive to avoid such an issue with our design and consider this safety constraint of grave importance. We plan on considering this constraint by adhering to the IEC 62133-2-2017 standard discussed in the section. This international standard, considered de facto, or required, by the international community, specifies guidelines for using lithium-ion batteries. These guidelines are precisely how to document the batteries and test them safely and correctly.

An additional safety constraint the team identified during research would be the printer's hot end. This component heats up and maintains high temperatures during use to melt the materials used to print. Contact with the hot end during operation can result in severe burns. We plan to avoid such injuries by designing the printer to make the hot end inaccessible. If we find that we can’t do so, we will, at the very least, include documentation that identifies that the hot end is a potential hazard.

4.3 Design Standards

In addition to constraints that our design must adhere to, there are also many sets of standards and regulations set forth by governments and regulatory bodies that we must follow. Some of the leading entities that produce these standards are the Institute of Electrical and Electronics Engineers (IEEE), the International Electrotechnical Commission (IEC), and the Python Enhancement Proposals (PEP). Many more entities produce standards and guidelines that most companies and people adhere to, but the ones mentioned above are most relevant to our design. The following criteria are discussed in this paper: IEC 62133-2-2017, PEP 8-2016, IEEE 829-2008, IPC-2221.

4.3.1 IEC 62133-2-2017

The IEC 62133-2-2017 standard is an international standard “for the safety of rechargeable lithium-ion batteries, which are commonly used in a wide range of consumer electronics and other applications”(Intertek, 2023). This standard, widely regarded as de facto internationally, describes requirements and tests for testing lithium-ion batteries that must be adhered to. By complying with this standard, which many manufacturers and regulators do, the safety and reliability of these batteries are more assured. In addition to setting requirements and tests for the batteries, according to intertek.com, the standard also requires thorough labeling and documentation of any lithium-ion battery used. The IEC 62133-2-2017 standard differs from its counterpart, IEC 62133-1-2017, in that it deals specifically with lithium-ion batteries instead of nickel batteries. Our team will strictly adhere to IEC 62133-2-2017 because of the use of lithium-ion batteries in our design. Specifically, our portable 3D printer will be powered by the lithium-ion batteries specified in this paper's materials list and research sections. By completing the requirements for compliance with this standard, our design will be deemed safer to use, which thus helps us stay within our safety constraints.

4.3.2 PEP 8-2016

PEP 8-2016 is a set of guidelines for writing Python code best. The PEP 8 standard is not considered de facto but is a widely used Python coding convention. PEP 8, which stands for Python Enhancement Proposals 8, is widely considered one of the most readable coding styles, hence why so many programmers prefer to adhere to them for their projects. PEP 8 features indentation, blank lines, comments, encoding, and proper naming conventions to ensure readability (educative.io, 2023). The following figure depicts the correct PEP 8 coding style use:

```python
name="User" # I am initialising the name variable with string User.
# I am adding three blank lines



if name=="User":
    print("User can enter") # if condition is easier to read
                            # with indentation.

# this is a comment.
# this program compares the variable name with a predefined
# value User to check validation.

# One can add all kinds of comments. Helps one understand your
# approach, logic and what you're trying to achieve.

def Print_square(number):
    print(number * number)
    # Always try adding spaces between operator

```

Figure 16: Screenshot of an example of PEP 8 used in a Python Coding Environment. From educative.io

Notice that each line of code is easy to read because of proper naming conventions, white spacing, indentation, and comments. PEP 8 is relevant to our project design because the printed circuit board and open-source software we will use are coded in Python. Utilizing PEP 8 will make our code easy for the programmer and anyone reading the code.

4.3.3 IEEE 829-2008 Software and Systems Test Documentation Standard

The IEEE 829-2008 Standard provides the guidelines for the stages of software testing. To specify, the standard describes all the steps engineers must go through when testing software to ensure that the software functions correctly. The introduction of the standard directly states, “The purpose of software and software-based systems testing is to help the development organization build quality into the software and system during the life cycle processes and to validate that the quality was achieved”(IEEE, 829-2008). Furthermore, the standard goes into depth regarding the software's supply, development, operation, maintenance, and testing the software. This standard is essential and widely accepted when building and testing software, and according to IEEE, it applies to all software-based systems. This standard also applies to our portable 3D printer since it will utilize open-source software to run the printer. We must test the LCD to ensure that each command performs its corresponding task correctly. Such duties include selecting a file to print, starting the print, pausing a print, or even aborting a print. We will also test the display to ensure it displays print progress and battery life. Chapter 5 of the standard describes the test processes that software should undergo. This process contains several other functions defined in another standard, IEEE/EIA 12207.0-1996: Management, Acquisition, Supply, Development, Operation, and Maintenance. To ensure successful testing, the standard describes in detail each of the above five processes, which are summarized below:

Management - To fulfill the management process, one must monitor the execution of the testing plan, analyze any anomalies during the procedure, report on all progress, assess the test results to ensure Standard conformance, and determine whether the project is completed. (IEEE, 829-2008)

Acquisition - One must formulate a plan to acquire the system that needs to be tested. In other words, the software itself. This is done by analyzing project demands, selecting potential software suppliers, and choosing the source of the software.

Supply - Completion of this process would mean that after the supplier's acquisition, a contract is signed so that the acquirer of the software is provided to software or service by the supplier. For the case of our portable printer project, since the software we will be using is open-source, we have already completed this process and will only need to credit the creators.

Supply - Test Planning - This process delves into creating a plan for testing the acquired software. To complete this process, one must identify the plan's scope, metrics, and integrity level described by the IEEE 829-2008 standard.

Development - This process includes all software development activities, including analysis design, coding, component integration, testing, installation, and more. (IEEE 829, 2008) To complete this process, our team should customize the open-source software for use in our portable 3D printer and implement it into the printer.

Operation - This process describes the operation of the software being used. After implementing the software in our project, we used the LCD of the printer running the software. To specify, we used the display and tested it to determine if the printer prints and displays all required metrics.

Maintenance - The final process to ensure conformance with the IEEE 829-2008 Standard. Completing this process to meet conformance means that our team has already run the software, tested it, and then made final alterations to the software to suit our needs. This could mean we improved something by eliminating a bug, changing or adding a metric, or more. This process can be considered the final edits to the testing plan defined by IEEE 829-2008.

4.3.4 IPC-2221 Standard

The IPC-2221 standard is commonly called the generic standard on printed board design. This standard provides rules and guidelines for designing printed circuit boards. IPC-2221 specifically specifies rules on spacing, line width, clearances, thickness of materials, types of materials for PCBs, and more. Furthermore, IPC-2221 provides guidelines for placing components on a PCB to ensure an optimized circuit layout. IPC-2221 also describes procedures for thermal management on the PCB. Overall, this standard is very generalized when designing PCBs, and other standards, such as IPC-6012 or IPC-6018, go into depth on specific types of PCB designs. Due to the nature of this standard in that it is very generalized, the team will use this standard as a reference to observe when designing the printed circuit boards required for the 3DPP project.

4.3.5 IEEE 1241-2010 Terminology and Test Methods for Analog-to-Digital Converters Standard

The IEEE 1241-2010 standard, titled "IEEE Standard for Terminology and Test Methods for Analog-to-Digital Converters," provides a comprehensive framework for evaluating the performance and characteristics of analog-to-digital converters (ADCs). This standard establishes a common set of terminology and definitions related to ADCs, ensuring consistent communication and understanding across industries and applications. It outlines rigorous test methods and procedures for assessing various parameters such as accuracy, resolution, linearity, and noise, which are critical for determining the reliability and quality of ADCs. By following the guidelines set forth in IEEE 1241-2010, manufacturers, engineers, and researchers can confidently assess and compare the performance of different ADCs, leading to more informed decisions during the selection and implementation of these essential components in electronic systems and measurement instruments.

Moreover, the IEEE 1241-2010 standard aims to promote interoperability and compatibility among ADCs from various manufacturers, making it easier for end-users to integrate these components into their applications seamlessly. Additionally, the standard plays a crucial role in providing a foundation for subsequent ADC-related standards, facilitating further advancements in ADC technology and ensuring its continued adherence to rigorous performance and quality standards. As technology evolves, IEEE 1241-2010 remains a key reference for the industry, supporting innovation and standardization in the design and evaluation of analog-to-digital converters for a wide range of applications.

4.3.6 IEC 61215-1:2021

IEC 61215 is an internationally recognized standard developed by the International Electrotechnical Commission (IEC) that focuses on the design and qualification requirements for crystalline silicon terrestrial photovoltaic (PV) modules, commonly known as solar panels. Published in several parts, this standard outlines the testing procedures and performance criteria for PV modules to ensure their reliability, safety, and overall quality. It covers various aspects of PV module design, including electrical characteristics, mechanical integrity, environmental performance, and durability, offering a comprehensive set of guidelines for manufacturers and users to evaluate the performance and suitability of solar panels for terrestrial applications.

The IEC 61215 standard provides a rigorous and standardized approach for the testing and certification of crystalline silicon PV modules, enabling consistent and reliable performance assessment across different manufacturers and installations. By adhering to IEC 61215, manufacturers can demonstrate compliance with stringent international standards, instilling confidence in consumers and industry stakeholders regarding the reliability and long-term performance of their solar panel products. Moreover, end-users can use the standard as a benchmark when selecting PV modules, ensuring they invest in solar panels that meet high-quality standards and are capable of withstanding various environmental conditions for many years of efficient electricity generation.

4.3.7 IEC 61853-1:2011

The IEC 61853 standard is a significant guideline developed by the International Electrotechnical Commission (IEC) that pertains to the performance testing and power rating of photovoltaic (PV) modules, including solar panels. This standard provides detailed procedures for accurately measuring and characterizing the electrical output characteristics of PV modules under specific operating conditions. By following IEC 61853, manufacturers can determine the power output of their solar panels with a high degree of precision, helping end-users make informed decisions based on reliable performance data. Additionally, this standard aids in ensuring fair comparisons between different PV modules in the market, facilitating transparent and standardized information for consumers and industry stakeholders.

IEC 61853 encompasses different types of PV modules and considers factors such as temperature, irradiance, and spectral content when conducting performance tests. The standard's comprehensive approach helps address various real-world scenarios and operating conditions, making it a valuable resource for assessing the efficiency and performance of solar panels in different geographic locations and climates. By promoting accurate power rating and performance characterization, the IEC 61853 standard plays a crucial role in enhancing the credibility and reliability of solar PV technology, ultimately contributing to the wider adoption of renewable energy sources worldwide.

4.3.8 IEC 60034-1:2022

IEC 60034 is a significant international standard developed by the International Electrotechnical Commission (IEC) that pertains to rotating electrical machines. Published in multiple parts, this standard provides comprehensive specifications for various types of electric motors and generators, including induction motors, synchronous motors, and DC motors. IEC 60034 establishes common requirements, test methods, and performance characteristics for these rotating electrical machines, ensuring consistency and reliability across different manufacturers and applications.

The IEC 60034 standard covers a wide range of aspects related to electric motors, including design, dimensions, performance, and testing procedures. It provides guidelines for determining parameters such as efficiency, temperature rise, and noise levels, which are critical for assessing the quality and suitability of rotating electrical machines in different operating conditions. By following IEC 60034, manufacturers can design and produce electric motors that meet stringent international standards, instilling confidence in consumers and industry stakeholders regarding the performance and safety of these essential components used in various industries, from manufacturing to transportation. Additionally, this standard aids in promoting energy efficiency and environmental sustainability, as it defines efficiency classes for electric motors to encourage the adoption of more energy-efficient designs, thus contributing to global efforts to reduce energy consumption and greenhouse gas emissions.

4.3.9 IEEE 1012-2016 System, Software, and Hardware Verification and Validation Standard

IEEE 1012, titled "IEEE Standard for System and Software Verification and Validation," is a significant standard developed by the Institute of Electrical and Electronics Engineers (IEEE). This standard provides guidelines and best practices for the verification and validation (V&V) processes in software and system development. It outlines the methods and techniques for assessing whether software products and related processes comply with specified requirements and whether the software is fit for its intended purpose. The primary goal of IEEE 1012 is to improve the quality, reliability, and safety of software and system products by ensuring rigorous V&V activities throughout the development life cycle.

IEEE 1012 covers a wide range of V&V processes, including planning, verification, validation, and reporting. It emphasizes the importance of systematic and disciplined approaches to V&V, promoting the use of appropriate verification and validation techniques, such as inspections, testing, and analysis. By following the guidelines set forth in IEEE 1012, software and system developers can enhance their confidence in the correctness and dependability of their products. Additionally, this standard provides a valuable reference for project managers and quality assurance professionals, helping them establish effective V&V plans and procedures that align with industry best practices and ensure compliance with regulatory requirements. Ultimately, IEEE 1012 contributes to the delivery of high-quality and reliable software and system products that meet user needs and expectations.

4.3.10 IEEE 829-2008 Software and System Test Documentation Standard

IEEE 829, titled "IEEE Standard for Software and System Test Documentation," is an important standard developed by the Institute of Electrical and Electronics

Engineers (IEEE). This standard outlines the format and content of software test documentation, providing a consistent and structured approach to documenting testing activities throughout the software development life cycle. IEEE 829 covers various types of test documentation, including test plans, test designs, test cases, test procedures, and test reports. By following the guidelines set forth in IEEE 829, software testers and quality assurance professionals can effectively communicate testing information, results, and progress to stakeholders, ensuring transparency and traceability in the testing process.

The standard emphasizes the importance of clear and comprehensive test documentation, which aids in understanding the testing objectives, scope, and strategies. It provides a template for organizing test documentation, ensuring that critical information is readily accessible to project teams and other stakeholders. By adhering to IEEE 829, software development projects can foster effective collaboration, facilitate decision-making, and improve the overall quality and reliability of software products. Moreover, the standard serves as a valuable resource for project managers, allowing them to plan, track, and assess the testing efforts efficiently, leading to successful and timely software deliveries.

4.3.11 IEEE 802.11n-2008 Standard

The IEEE 802.11n standard, introduced in 2009, is a significant advancement in wireless local area networking (WLAN) technology. Also known as Wi-Fi 4, it is an amendment to the 802.11 standard that brings significant improvements in data rates, range, and overall performance compared to its predecessors. One of the key features of 802.11n is Multiple Input Multiple Output (MIMO) technology, which utilizes multiple antennas to transmit and receive data simultaneously. This technology enables higher throughput and better coverage, allowing users to experience faster and more reliable wireless connections.

With support for both the 2.4 GHz and 5 GHz frequency bands, IEEE 802.11n can achieve data rates of up to 600 Mbps. This increased speed and bandwidth make it ideal for multimedia streaming, online gaming, and other high-bandwidth applications. Additionally, 802.11n is backward compatible with older Wi-Fi standards, enabling seamless connectivity with legacy devices. As one of the most widely adopted Wi-Fi standards, IEEE 802.11n has significantly contributed to the widespread use of wireless technology in homes, businesses, and public spaces, revolutionizing the way people access the internet and communicate wirelessly.

4.3.12 IEC 61010-1:2010

The IEC 61010-1 standard, titled "Safety Requirements for Electrical Equipment for Measurement, Control, and Laboratory Use - Part 1: General Requirements," is an important safety standard developed by the International Electrotechnical Commission (IEC). Published in several editions, the latest being IEC 61010-1:2010, this standard applies to a wide range of electrical equipment used for measurement, control, and laboratory applications. The primary objective of IEC 61010-1 is to ensure the safety of operators, technicians, and users of electrical equipment in various industrial, laboratory, and research settings.

IEC 61010-1 outlines safety requirements related to electrical, mechanical, and environmental hazards that may be associated with measurement and control equipment. It covers aspects such as electrical insulation, mechanical strength, protection against electric shock, and temperature limits to prevent potential risks and hazards during operation. The standard classifies equipment into different categories based on their intended use and application, with specific requirements tailored to each category. Compliance with IEC 61010-1 is essential for manufacturers, suppliers, and users of electrical equipment to ensure the safe and reliable operation of instruments and devices used in diverse industries and scientific settings.

5.0 Comparison of ChatGPT with Other Similar Platforms

This section will analyze the impact of artificial intelligence (AI) software like ChatGPT and other similar platforms based on OpenAI on the learning outcomes in senior design. We will examine not only the impact of artificial intelligence on modern learning but also how its quick rise in recent months has had an even more adverse effect on students, teachers, and other staff. The concerns of cheating and plagiarism due to this new technology have risen significantly since the beginning of the year, and due to the fast pace at which this technology improves, software detecting the usage of AI has a difficult time keeping up with new models and algorithms. This provides an exciting challenge for instructors, who now have limited options for ensuring the validity of their student's work.

This isn't to say that the rise of AI programs is all doom and gloom. After all, it is a tool. As engineers, it is essential to use the tools at our disposal to generate the best solution to a problem, and as researchers, it is often hard to synthesize all the literature available online without dumping hundreds of hours into complex research. ChatGPT allows students to use AI as a springboard for idea inspiration, research synthesis, and simply as a tool for answering questions. If students use this new technology responsibly, it can increase the efficiency and even the quality of engineering projects and research, while still allowing them to grow and learn in a meaningful way.

5.1.1 ChatGPT Pros/Cons, Limitations

ChatGPT is the most popular application of the recent AI boom. It's name recognition alone makes it synonymous with AI and all implications that come with that term. As for chat GPT itself, it uses online resources to generate responses to user prompts, which makes it a powerful tool for almost anyone.

For an example of a personal anecdote, last spring semester, Jack spent the last few weeks attempting to make an HC-05 Bluetooth module work with an Arduino in an attempt to control a competition robot over Bluetooth with a computer. His team and had spent weeks researching the correct wiring, code, and instructions for making this work, but had yet to find it. Weeks into research, one of his students recommended asking ChatGPT about this issue. Surprisingly, ChatGPT provided almost every method previously attempted had over the last couple of weeks of the semester in seconds. Even though none of the processes directly solved the issue, the time that could have been saved using ChatGPT as a starting point was astronomical.

This is just one of the many benefits that ChatGPT offers. Since it acts as an information aggregator, it is a powerful tool for getting information quickly from various sources. Hours of research can get cut down into seconds. ChatGPT is also an excellent tool for synthesizing studies and ideas. This is where the main concern about plagiarism comes into play. However, if what ChatGPT says is used to guide independent research, the lines become less blurred. This also includes using ChatGPT to generate ideas for this class. The group prompted ChatGPT for idea proposals and it generated on we liked about a wearable exoskeleton. It even outlined many of the details and specifications of the project. Idea inspiration is something many students struggle with, so this tool is beneficial for moving the creative process along.

Finally, ChatGPT is excellent for answering questions. Since it draws from a large pool of sources, a consensus can often be formed about a specific question, and answers reported to the user. This is excellent for informing researchers where to seek answers to a question. As engineers, this is also helpful for technical questions, such as how to wire certain electrical parts together correctly or find power requirements. Again, this saves valuable time spent elsewhere in the design process. However, especially in engineering, this comes at a cost.

The main downside of ChatGPT is the need for more reliability of its answers. Since ChatGPT draws from sources all around the internet, it is not only possible but shared that this information needs to be corrected, inefficient, or will not be applicable. We mentioned above how that happened with the HC-05 module in my project. This is common, especially in engineering, since the principles discussed and the technology used often need a large pool of online resources. There are entire forums outlining how to solve tiny technical issues in software and hardware, and even those forums often contain more “solutions” that don’t work than ones that do. The simple reality is that the more niche and complex an engineering field is, the more challenging time ChatGPT and AI as a whole will have to deliver accurate information on the topic. This is a problem for obvious reasons for engineers, as we often aren’t turning to AI to answer simple questions. In our experience, we have found that code from ChatGPT is especially susceptible to this issue, though it does act as great inspiration for how a coding problem could be approached.

5.1.2 Chatsonic Pros/Cons, Limitations

Chatsonic is another OpenAI-powered chatbot. Its newer features specifically address many of ChatGPT’s weaknesses. ChatGPT uses GPT-4 from OpenAI, making its power comparable to ChatGPTs. However, the developers have designed it to exceed its rival in some key areas. It uses Google Knowledge Graph to import relevant data from google analytics. Chatsonic is also more affordable than ChatGPT, with its long-form premium plan costing just below $13 a month, while ChatGPT Plus is $20 monthly. Chatsonic is better suited for writing specific text types, such as essays, articles, or other formatted text, while ChatGPT is better for providing simple text-based responses.

Chatsonic is overall very similar to ChatGPT, albeit with arguably more reliable responses and increased customization. That said, ChatGPT was trained off of a more extensive and diverse data source. Ultimately we don't believe it offers enough to warrant usage over ChatGPT, but it does pose some exciting solutions to ChatGPTs shortcomings.

5.1.3 Other A.I. Chatbots

Many other media and information companies have begun creating competitors to chat GPT, such as Google's Bard and Snapchats' My AI. While most are in the earlier stages of development, there has been an effort to distinguish their software from those already available. Snapchat, in particular, has marketed and designed it around being a companion, even listing it at the top of the user's chats list like it was one of their real-life friends. This offers a different type of usage from ChatGPT, which is much more geared toward information gathering and answering prompts. AI chatbots like Snapchats MyAI has slightly less relevance to the engineering practice due to its specialized nature, but it is able to more or less serve the same purpose by answering user questions.

The specialization of AI software does raise an interesting question about the future of AI in the engineering space. We can say from personal experience that when it comes to answering difficult engineering questions, the quality of the answers significantly decline with the questions complexity. This is likely due to the limited literature on some burgeoning technologies, and the deep specialization of the engineering field. AI, much like any technical questions forums, will provide many solutions that may work for one scenario but not a users specific scenario. However as specialization in AI software grows, it is possible that we will see Ai specifically designed to answer more complex engineering questions and have features such as advanced coding capabilities, and advanced mathematical capabilities.

This would introduce another multitude of new ethical concerns, as the limited capabilities of AI and its questionable accuracy, force students and professionals to be confident in their answer and diagnosis of a response from the AI. This fear ultimately leads to the need to understand the question at hand, as to not rely too heavily on the AI response, at the risk of it being inaccurate.

5.2.1 Positive Uses of A.I. Software in Senior Design

When initially coming up with ideas for our senior design project, we were permitted and encouraged to use ChatGPT and other AI chatbots to develop ideas. As previously mentioned, when our team used this brainstorming method, it yielded detailed ideas that we even considered using. The ideas generated were relatively unique, doable in the given timespan for our project, and geared towards the ECE majors. Although we ended up going with a human-generated idea, the responses from ChatGPT gave us a good idea of the appropriate scope, goals, and requirements for a senior design project. This could have been accomplished by viewing old senior design projects, though that would require additional research, that the use of ChatGPT circumvented. There also would be a risk of inspiration taken from those ideas to be considered plagiarism. The ChatGPT ideas, while likely influenced by older projects, could be used and modified to create a unique project for this class.

We also used ChatGPT to identify some requirements for what would eventually become our project. This helped ensure everything was fine with the project idea, just in case some requirement had slipped past us before we committed to this idea for the following two semesters. It also helped us check the project's validity, ensuring it was doable in the given time frame. 3D printers are complex machines with many specifications, so it was essential to know that any proposed solutions are feasible given the power demand. It also helped us identify similar projects or experiments conducted on the topic, which would be helpful information for us to use in our design process.

ChatGPT is an incredibly useful tool for swiftly addressing queries. It's like having an intelligent assistant on standby, ready to provide answers and guidance. This will be particularly beneficial in the design and development of our project's electrical, mechanical, and software systems.

For instance, consider a scenario where we've downloaded a CAD assembly for a component of our project, but we're unsure about the process of importing it into our custom CAD assembly. Traditionally, we might have to sift through numerous YouTube tutorials to find the information we need. However, with ChatGPT, we can simply pose our question and receive a step-by-step guide, saving us time and effort.

5.2.2 Negative Uses of A.I. Software in Senior Design

While A.I. is an incredible tool to increase productivity, numerous concerns arise with its increased use. Especially in a field like engineering, accuracy, and competency is vital to a student's learning. The lack of citations and a large pool of sources means it can be hard to verify the credibility of information received from AI software, which can lead to incorrect information being used by the user.

Furthermore, the reliance on A.I. for generating educational content raises ethical considerations regarding the ownership and attribution of intellectual property. With vast amounts of data being processed and generated by AI algorithms, it becomes challenging to discern the original creators or sources of information. This ambiguity can lead to inadvertent plagiarism and copyright infringement, as students may unknowingly use content without proper attribution or permission from the original authors. In an educational setting that prioritizes academic integrity, the lack of transparency in A.I.-generated content can undermine the principles of proper citation and intellectual honesty, potentially impacting the credibility and reputation of both students and educational institutions.

Another significant concern relates to the potential for A.I. bias in educational content. Since AI algorithms learn from existing data, they may inadvertently perpetuate inherent biases present in the training datasets. In engineering and other fields, biased information can lead to misconceptions and reinforce stereotypes, hindering students' ability to develop a well-rounded and inclusive understanding of complex topics. Addressing and mitigating bias in AI-generated content requires continuous monitoring, diversity in dataset curation, and algorithm refinement to ensure a fair and unbiased representation of information. Educational institutions must actively evaluate and implement measures to promote diversity, equity, and inclusion in the content delivered through AI-based platforms.

Despite these challenges, there are opportunities for A.I. to enhance the learning experience in engineering and other disciplines. A carefully designed and well-implemented AI system can complement traditional education by providing personalized and adaptive learning pathways, catering to individual student needs and learning styles. AI-powered educational tools can analyze student performance, identify areas of weakness, and recommend tailored study materials and exercises to address specific learning gaps. Moreover, A.I. can facilitate interactive simulations and virtual labs, offering students practical hands-on experiences that are otherwise challenging to replicate in a traditional classroom setting. To fully harness the potential of A.I. in education, educators and institutions must strike a delicate balance, leveraging the benefits while actively addressing the challenges to ensure a comprehensive, equitable, and ethically responsible learning environment.

6.0 Hardware Design

The 3DPP team will explore the critical hardware components integral to our portable 3D printer in the following sections. We'll shed light on their operational principles, roles, and the specific features that align with our project requirements. A significant focus will be on our Printed Circuit Board (PCB), the heart of our printer, the central hub where all electronic components connect and communicate. The intricate planning and precise execution of its design will be discussed, considering factors such as component placement, electrical connectivity, and space optimization. Furthermore, we'll detail the integration of this PCB into our project, encompassing not just its physical installation but also the electrical connections to other components, the driving firmware, and the testing and troubleshooting processes. This comprehensive overview aims to provide a holistic understanding of our project's hardware aspect, highlighting the synergistic interplay between various components that culminate in a fully functional, portable 3D printer.

6.0.1 Hardware Key Features

The 3DPP offers numerous impressive hardware features that redefine convenience and accessibility in 3D printing. One of its standout features is its compact and lightweight design. Crafted with portability in mind, this printer can easily fit into a backpack or a carrying case, making it perfect for creators on the move. Its reduced footprint doesn't compromise performance; precise printing capabilities rival larger, stationary 3D printers. This feature is particularly beneficial for professionals attending workshops, students in educational settings, or hobbyists who want to showcase their creations at events.

In addition to its portability, the 3DPP incorporates a user-friendly interface. Equipped with an intuitive touchscreen display, it allows for effortless navigation and control over the printing process. Users can conveniently adjust printing settings, monitor progress, and initiate printing jobs directly from the interface, eliminating the need for complex external controls or software installations. This user-centric design ensures that beginners and experienced 3D printing enthusiasts can easily engage with the printer, unleashing their creativity without technical barriers.

Another critical hardware feature is its compatibility with various printing materials. Despite its portable size, this printer supports various filaments, including PLA, ABS, PETG, and more. The versatile extruder and heated print bed enable smooth and reliable printing with different materials, accommodating diverse project requirements. Whether users are crafting intricate prototypes, functional parts, or artistic designs, the printer's capability to work with various materials offers a level of adaptability that genuinely sets it apart in the 3D printing market.

6.1 Hardware Design Methodology

The hardware design methodology of the 3DPP is a meticulous process that revolves around achieving a perfect balance between portability, performance, and user-friendliness. The primary goal is to create a compact, lightweight printer that can be easily carried and used in various environments. The design team starts by carefully selecting materials that are not only durable but also lightweight, ensuring that the printer's overall weight is minimized without compromising its structural integrity. They conduct thorough research to identify the most efficient and space-saving arrangements of components within the printer's chassis, optimizing every inch of available space.

Power efficiency is critical in the hardware design to ensure true portability. Low-power consumption components and intelligent power management systems extend the printer's battery life. The team focuses on minimizing energy wastage, allowing the portable printer to operate effectively while being powered by a rechargeable battery. This emphasis on power efficiency enables users to print wirelessly without needing an external power source, making the printer portable and suitable for on-the-go 3D printing tasks.

User experience is at the heart of the hardware design methodology. The printer's interface is designed to be intuitive and user-friendly, even for those new to 3D printing. A touchscreen display allows for easy navigation through the printer's settings, file selection, and monitoring of the printing process. The design team aims to simplify the setup and printing procedures to eliminate technical barriers, ensuring users can quickly and efficiently bring their ideas to life. Emphasis is also placed on creating a hassle-free maintenance process, with easy-to-access components that users can replace or upgrade, reducing the need for professional assistance and ensuring a longer product lifespan.

The hardware design methodology also incorporates rigorous testing and quality control procedures. Prototypes are extensively tested under various conditions to identify potential issues and areas for improvement. The printer is subjected to reliability and performance tests to ensure it meets the highest quality standards before reaching the market. This commitment to quality and continuous design refinement ensures that the 3DPP delivers exceptional performance and a seamless user experience, solidifying its position as a reliable and innovative tool for 3D printing enthusiasts and professionals.

6.2 Hardware Subsystems

This section of the paper identifies and discusses each subsystem of the 3DPP. It specifically covers the power system, which supplies power to all the major components. The SKR Pico system covers the controller and all components connected to it, and the Raspberry Pi system comprises the Raspberry Pi and all hardware connected to it.

6.2.1 Power System

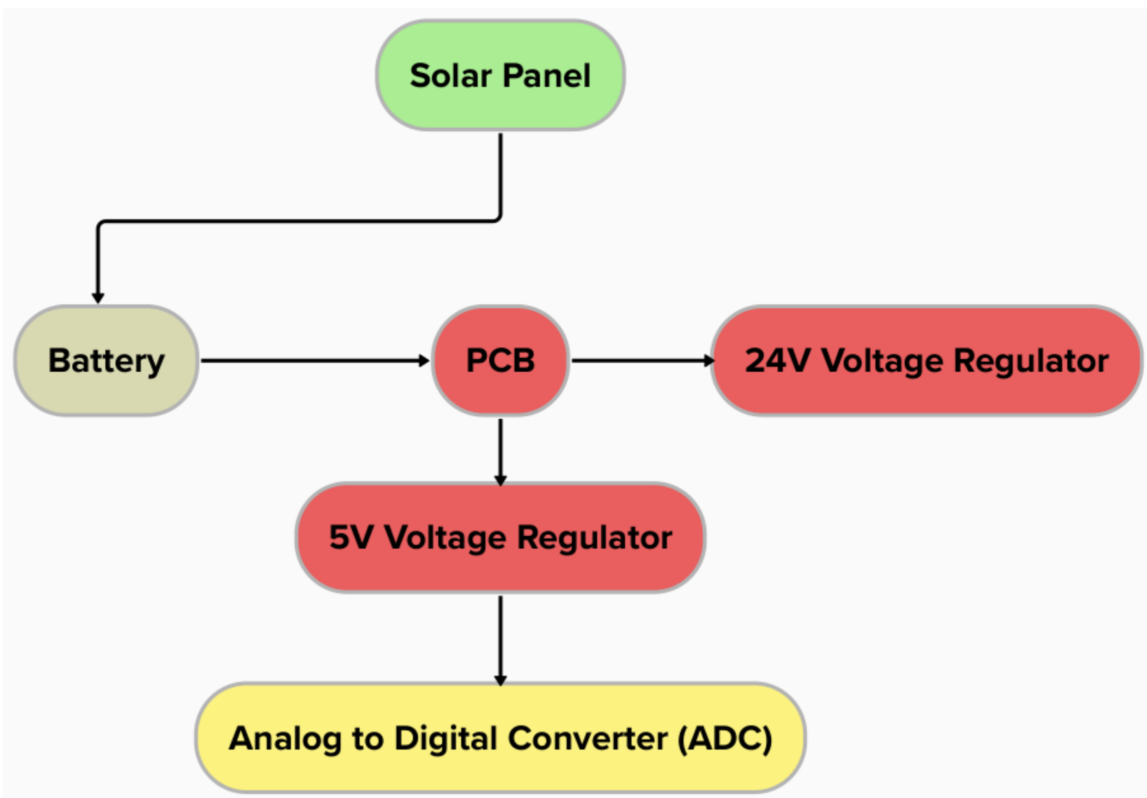


Figure 17: Initial block diagram for 3DPP power system

The first subsystem of the 3DPP will contain all portions of the system that require electrical power. The components highlighted in red in the above figure will all be created on our unique PCB design that we will develop and revise, and we plan to add more functionality later when more testing is done. The solar panel is a vital portion of this subsystem because it is the only component that will charge our battery. The battery we have chosen for the 3DPP is a 24V lithium-ion battery that will be the system's heart. Once the power is generated from the solar panel and transferred into the battery, the power will be filtered into the PCB design to be converted into specific voltage levels using voltage regulators. As we get further into Senior Design 2, we plan to add more components once we have adequately researched and tested voltage profiles needed for features like sensors or networking capabilities for the 3DPP.

6.2.1.1 12V to 5V Voltage Regulator

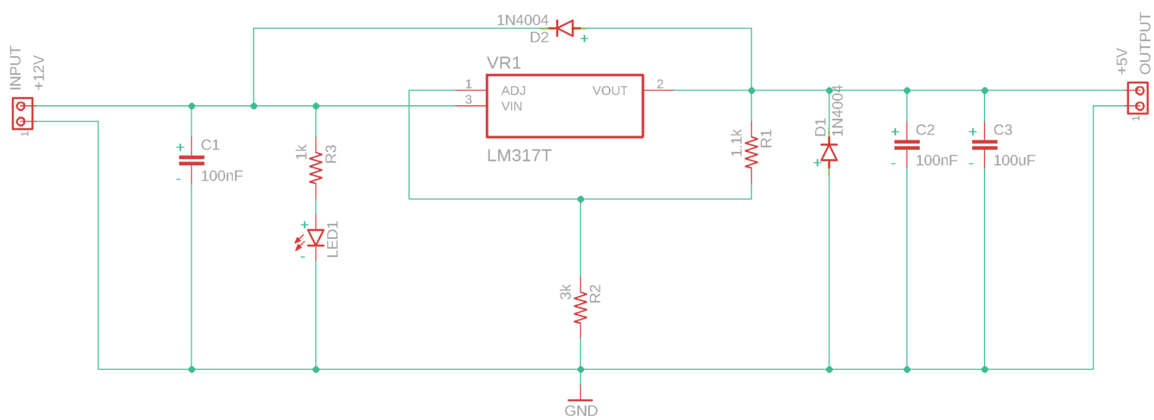


Figure 18: Initial 12V to 5V programmable voltage regulator schematic

The voltage regulator used to operate the 3DPP system will be a vital portion of the electrical design and must receive high priority and care so we don't damage the Raspberry Pi in our system. The 5 volts that are produced in this circuit will be used to power an analog-to-digital converter (ADC) so we can feedback the output voltage of the Raspberry Pi so we can see if there is a stable voltage and current since the battery will degrade over time when powering all components of the 3DPP. We plan to create a software program to take this information and display the voltage and current on the LCD, alert the user if the battery is close to being depleted, and save the current progress of the 3D print.

Another function of this regulator is to power the Raspberry Pi with its proprietary voltage source so we can quickly troubleshoot and calculate the power needs of the entire system. The whole 3DPP team considered this since this will provide the easiest and most efficient way to manage our system's energy requirements conservatively. Since we decided to go with this route, we will have two outputs from the 5V output terminal to power both the Raspberry Pi and the ADC with a constant 5V for easy measuring. Overall, we are happy with the simplicity and stock these components will bring to the 3DPP system to perform effectively.

6.2.1.2 24V to 6V Voltage Divider

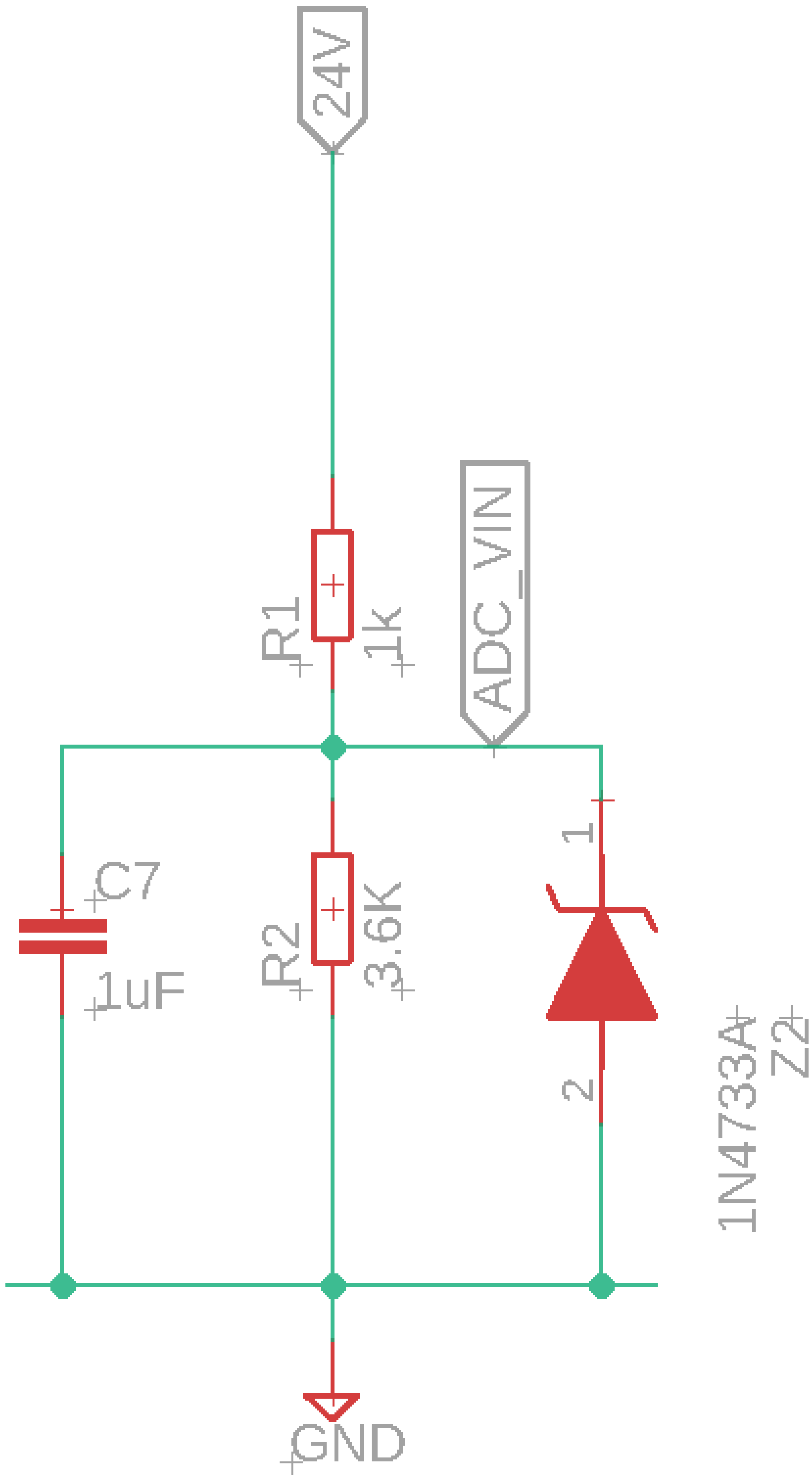


Figure 19: 24V to 5V Voltage Divider Circuit

The 24V to 6V Voltage Divider circuit connects directly to an output of the SKR Pico regulator circuit. The voltage divider then connects to the HiLetgo ADS1115 Analog-to-Digital Converter input. In addition to the 12V to 5V Regulator circuit discussed in 6.2.1.1, the voltage divider ensures that the ADC receives the proper information to power the Raspberry Pi sufficiently. This circuit is comprised of three components. The first component is resistor 1, 1k Ohms, and the second component, resistor two, is 3.6k Ohms. This 1:3.6 ratio ensures that the input voltage of 24V steps down to 5V. The final component is the three-terminal pin header. This pin header contains the ports in which the input, output, and ground will be soldered during the assembly of the 3DPP.

6.2.1.3 24V Regulator Circuit

A sufficient regulator circuit must be chosen to power the SKR Pico Controller and all components connected to it. The SKR Pico can run from an input voltage ranging from 12 volts to 24 volts. The stepper motors, fan, and heating cartridge controlled by the SKR Pico can also run on a 24-volt input. With this in mind, regulated information of 24 volts can be applied to this subsystem. Remember that while the battery selected to power the 3DPP has a 24-volt output, it is not regulated like its 5-volt output. Thus, it can be subjected to voltage fluctuations that exceed the values for which the SKR Pico is rated.

Hence the need for a regulator circuit. All of the stepper motors connected to the controller require 1.5 amps of current each as well, and in addition to that, the hot end with the 50-watt heater cartridge requires approximately two amps of current, and the SKR Pico itself requires roughly an amp of current. This subsystem needs at least 9.5 amps of current for all components to function correctly. With that in mind, this is a high-powered circuit. The original plan for meeting these power requirements was for the team to design a custom PCB to fulfill the needs of the subsystem. However, the circuit we need is rather complex. We realized that finding a regulator capable of supplying such a high current demand was somewhat challenging, as most mainstream regulators like the lm317 or the lm7824 have maximum output current ratings that don’t meet our design demands. We also realized that a custom design would entail having a heat sink capable of handling all of the power dissipation from this high-power subsystem, which would aggravate the overall cost of the PCB. Finally, we took into account our safety constraints. If we make a mistake when designing this circuit, it could pose a safety risk for the team. In addition, it would prove quite costly as it would fail all components involved in the subsystem. As a result of these conclusions, the group decided that we would utilize a pre-built regulator capable of meeting our design requirements safely and efficiently to prevent any setbacks to our timeline. However, the team still intends to design our regulator for this sub-system, and we will continue to research and explore potential designs that we could use instead of the pre-built circuit.

The prebuilt circuit that the team decided on using, for now, is the Drok DC to DC Step Down Power Supply Module circuit. This circuit is suitable for our needs because its output can range from 0 volts to 60 volts and from 0 amps to 12 amps, in which our power requirements fall within the range. The circuit also contains a dedicated cooling system to handle power dissipation. Furthermore, while the circuit can be purchased from Amazon for $29.99, it boasts a typical efficiency of 95% and a response time of less than 50 milliseconds. This circuit can be programmed via UART to meet our requirements of 24 volts and ~9.5 amps output. A figure of this circuit will be provided in this section of the document pending approval by the circuit manufacturer, Dorking.

6.2.2 HiLetgo ADS1115 (ADC)

This component will save the progress of a 3D print when the voltage from the battery becomes too low to continue printing. With this piece of hardware, we can alert the user that the battery needs charging through the LCD and save the exact progress of the 3D print, which will be very helpful so no work is lost due to the discharge of the battery.

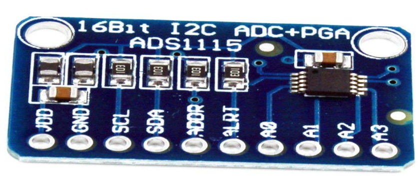


Figure 20: HiLetgo ADS1115 PCB

This ADC module supports I2C communication, a master-slave architecture allowing serial communication between the hardware connections we decide on. This ADC also provides 16-bit precision at 3300 samples/second when using the I2C communication protocol, giving very accurate readings of the percentage of the total 3D print operation. It also comes with an input range of 2.0-5.5V, which is convenient for our system, so we can have a capable piece of hardware that is low power and relatively low cost. Even though this ADC would be a great addition to the 3DPP, we would like to create our own ADC that will be placed on the custom-made PCB design that we will complete later. With that said, this ADC will be a significant test component since this is a functional component that will give us a benchmark on how the ADC should convert the progress of the

3DPP so we can store the memory of the 3D print in the internal memory of the Raspberry Pi.

Designing and implementing our custom ADC for the 3DPP offers several advantages beyond the performance benchmarking. By developing our own ADC, we gain full control over its specifications, ensuring it is tailor-made to meet the precise requirements of our unique application. This level of customization allows us to optimize the ADC's performance, accuracy, and speed specifically for monitoring the 3D print operation, resulting in more reliable and consistent readings. Additionally, having an in-house designed ADC grants us the freedom to incorporate features that enhance the 3DPP's overall efficiency and user experience, such as real-time data visualization, adaptive sampling rates, and streamlined communication with other system components.

In conclusion, while the existing ADC module serves as an excellent starting point for testing and validation, the decision to develop our custom ADC reflects our commitment to innovation, precision, and optimizing the 3DPP's performance. The combination of benchmarking with the off-the-shelf ADC and the subsequent development of our custom solution allows us to strike the right balance between immediate functionality and long-term adaptability. As we progress with our custom-made PCB design and the 3DPP's overall implementation, the custom ADC will stand as a testament to the team's technical expertise and dedication to creating a cutting-edge and transformative 3D printing system.

6.2.3 Kuman TFT LCD Touch Screen Display

This piece of hardware for our project is the primary way our users get information from the printer and, most notably, the voltage of the battery, letting the user know how much time they have left for their print. We will pair our project's Kuman TFT LCD Touchscreen with our Raspberry Pi 2B. This will allow us to display information gathered by the Pi to the display. A hardware feature that the display does have that the 3DPP team found undesirable is the fact that it does require to connect to ten of the forty GPIO pins on the Raspberry Pi. This was of concern to us because if the display took too many of the pins our team would not be able to also include the ADC on the GPIO pins. Fortanty for us though the display uses a five volt, SPI connection whereas the ADC uses a 3.3V I2C connection.

This specific display interfaces to the Raspberry Pi via a Serial Peripheral Interface (SPI communication. This type of connection works using a master-slave architecture, where the primary device initiates the data frame. SPI uses four lines - SCLK, MOSI, MISO, and SS. The master generates a clock pulse SCLK, sends data to the slave through the MOSI line, and receives data from the slave through the MISO line. The SS line selects the slave device with which communication is intended.

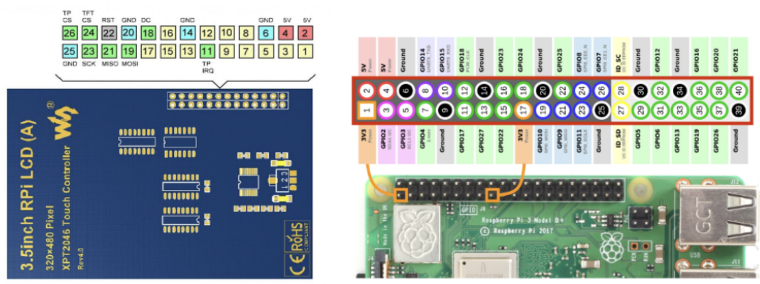


Figure 21: Kuman TFT LCD Touch Screen Display PCB

Along with using the SPI interface, our Kuman display will also be utilizing the ground and five-volt pins on the Raspberry Pi, which did cause us to have to change our way of connecting the SKR Pico and the Raspberry Pi. Initially, the 3DPP team believed we could get enough current to the ADC, the touchscreen display from the SKR Picos connection to the Raspberry Pi via five GPIO pins which gave the Raspberry Pi two arms. After conducting some research, our team discovered that the display draws too much current for our team to make this connection, and more than two amps were needed. Ultimately we have decided that because of the LCD, we will now power the Raspberry Pi via a USB connection connected to the PCB, which gives the Pi a five-volt and three-amp connection, giving us more than enough power to power the display.

6.2.4 SKR-Pico

As mentioned previously, one of the critical features for why our team decided to use the SKR Pico is that it did not do many software functions and instead relied on the faster Raspberry Pi MCU. This being said, where the SKR-Pico lacks in the Software department because of its slower MCU, it makes it up in the hardware department, being one of if not the essential pieces to our project. First, the SKR-Pico allows the 3DPP team to connect some of the critical hardware parts to our project, such as stepper motors, cooling fans, endstop sensors, and heating cartridges via the onboard input board pins. Along with this, the SKR-Pico is connected to the Raspberry Pi via a USB C connection which plugs into the USB port on the Raspberry Pi. This connection is vital as it is the only way the printer controller and Pi communicate.

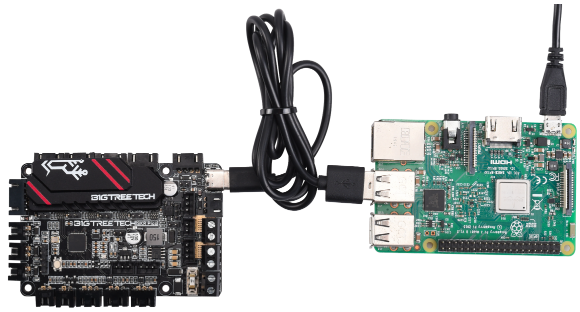


Figure 22: SKR-PICO Controller

When designing our PCB board, the SKR-Pico was the most crucial feature we had to figure out how to power via a battery since it does have a 24 V to 12 V voltage tolerance, which is a pretty extensive range. This being said, it does require a lot of current being supplied for the board and its peripherals to operate correctly, for the four stepper motors our team will be using need 4.8 A, the fan will draw around 0.1 A, the hot end heating cartridge will draw 3.0 A and finally, the SKR-Pico itself will draw 0.5 A. This all being said, we need to ensure that for our PCB design, the Pico and its peripherals will need a current supply of around 10A to stay safe. If the board does not get enough current, this could lead to the board not working correctly and the other hardware parts not working properly, leading to failed prints and inaccuracy.

A more detailed explanation of how the SKR-Pico works is that the SKR Pico is a compact 3D printer controller board by BigTreeTech. It uses a 32-bit ARM Cortex-M0+ processor to control the printer's movements precisely. The board has four integrated TMC2209 stepper motor drivers for controlling the stepper motors and ports for connecting other components like a heated bed, hotend, thermistors, end stops, and a display. The SKR Pico can be programmed with the firmware Klipper, and it regulates power from the printer's supply to each connected component.

6.2.5 Raspberry Pi 2B

The Raspberry Pi 2B will essentially work as the brain of the 3D printer, doing most of the software-heavy work, such as conducting G-Code translations, checking the batteries' voltage using the ADC, and hosting a web application. The Raspberry Pi is a crucial software system for the 3DPP team and an essential hardware item for our project. This subsystem allows us to connect to and control various other hardware units such as the CanaKit Wifi Dongle, connecting to it via a USB connection, the HiLetgo ADS1115 ADC; using an I2C connection, and finally, the TFT LCD Touch Screen Display, which is connected to the Pi via an SPI connection.

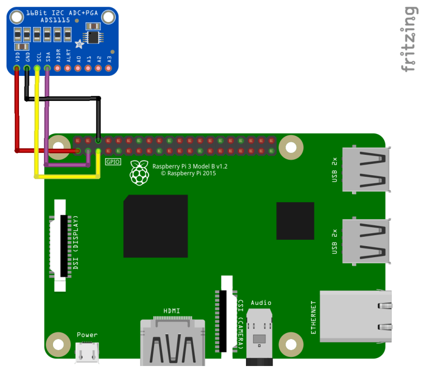


Figure 23: Raspberry Pi 2B

Since our team is using the Raspberry Pi to power these various other essential hardware and software features, our team must use the proper type of regulator so that the Pi has enough power. Our team has figured out from the Pi datasheet that it must always be powered by a five-volt (+ or - 5%) power supply. If it drops out of this range, the board will not work. On the other hand, if it goes above this range, it could start damaging components. Along with the Pi being powered by five volts, our team has discovered that with the various hardware parts plugged into the Pi, it must always be getting at least a current of 2.0 A. Our team has found that the wifi dongle draws a current of 0.2 A, the TFT Touchscreen display draws approximately 0.2 A, the ADC draws 150μA, and the Raspberry Pi itself draws 0.8A. As you can see, even with all of the connections and the Raspberry Pi, the current draw does not exceed 2.0 A. Our team wanted to stay on the safe side, though, because if we do not have enough current for the Pi, this could result in significant issues, such as connected peripherals not working, Undervoltage warnings, and unexpected behaviors.

A detailed explanation of the hardware behind the Raspberry Pi is as follows. The Raspberry Pi 2 Model B is a compact, credit-card-sized computer with a versatile platform for various computing tasks. It features a Broadcom BCM2836 system-on-a-chip, including a quad-core ARM Cortex-A7 CPU and a VideoCore IV GPU, which is responsible for executing the operating system and software applications. The Pi 2B has 1GB of RAM for temporary data storage and uses a microSD card slot for the operating system and file storage. It offers a range of connectivity options, including four USB ports, an Ethernet port, and a GPIO header for interfacing with various external hardware devices, sensors, and circuits. The device also includes an HDMI port for video output, a 3.5mm audio jack, and a Camera Serial Interface for connecting a Raspberry Pi camera. Power is supplied via a micro-USB port. Despite its small size, the Raspberry Pi 2B is a fully functional computer that can run various operating systems, most commonly a version of Linux such as Raspberry Pi OS.

6.2.6 CanaKit Wifi Dongle

The CanaKit Wifi Dongle is an essential part of the hardware design of our portable 3D printer. This simple piece of hardware connects to our Raspberry Pi 2B via a USB port and allows our board to connect to wifi and or create its wifi source, as mentioned in the previous section. This piece of hardware is essential to our project because it allows our users to connect to our 3D printer and access our local host software Mainsail which allows our users to control and monitor various parts of our project from their local device. Our team will run software on the Raspberry Pi that will allow the dongle to act as a hot spot rather than an internet card, which will be essential in creating the most portable 3D printing system. Instead of the Raspberry Pi and your local mobile device, the user must connect the 3DPP wifi source from the CanaKit Wifi Dongle.

Figure 24: CanaKit Wifi Dongle

This wifi dongle works by communicating with your wireless router or access point. When you plug the dongle into a USB port on the Raspberry Pi, it acts as a network interface card, receiving and transmitting data over WiFi. The dongle contains a radio transmitter and receiver (transceiver) that operates on a specific WiFi standard 802.11n. When you want to access the internet, your device sends the data to the dongle, which transmits it wirelessly to the router. The router sends the data to the internet over a wired connection. When data from the internet is returned, the process is reversed. The router receives the data and sends it wirelessly to the dongle, which transmits it to your device.

6.2.7 Nema 17 24V Stepper Motor

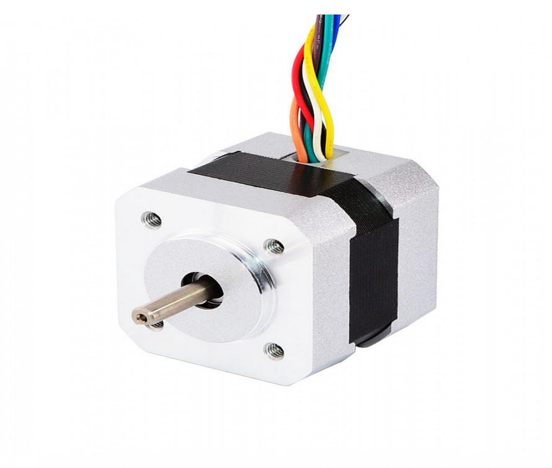


Figure 25: Nema 17 24V Stepper Motor

The stepper motor will play a vital role in the positioning of the extruder when the 3DPP is operating and physically printing the G-Code file. This is achieved by the principle of electromagnetic coils to attain precise and controlled motion. The input voltage for this particular stepper motor is 24V and requires 1A of current for operation, and this motor also comes with two separate phases for power control. Also, this stepper motor has a step angle of 1.8 degrees, meaning that to complete a full revolution of 360 degrees, the motor would need to have 200 total steps.

When the stepper motor is correctly electrically connected to the battery and the SKR PICO, we should be able to control the extruder for the filament inside the print head. Then we should be able to print multiple layers of PLA filament and have 360 degrees of movement depending on the 3D model we are publishing. For this specific stepper motor, the only axis that will be controlled is the X/Y axis, and we will order another specific engine to control the Z axis later on. Incorporating our software to connect the 3D model to the G-Code and finally control the entire gantry system, this stepper motor will provide a sound movement system when printing.

Integrating this stepper motor into the 3D printer system ensures a smooth and reliable printing experience, ultimately leading to high-quality prints that meet the desired specifications. Its electromagnetic principles, paired with the 24V voltage, 1A current requirement, and 1.8-degree step angle, make it a dependable and efficient choice for the precise motion control demands of 3D printing.

6.2.8 Cooling Fan (Winsinn 30mm Blower Fan)

Cooling fans play an essential role in improving the performance and durability of a 3D printer. They’re mainly used to regulate and dissipate heat generated during printing, preventing issues such as warping, layer shifting, and overheating electronic components and ensuring the first layer adheres to the print bed. Cooling fans cool the extruder during printing which helps to prevent clogging and maintain a steady stream of filament. For the cooling fans on our build, we are looking for compact fans that can deliver the airflow needed to cool off intense electrical components like voltage converters. However, we are looking for budget-friendly alternatives for earlier prototypes. One of these is the Winsinn 30mm blower fan, also featured on the Positron line of printers. These cooling fans feature a ball bearing for seamless rotation, fit within a 1 ¼” cubed space, draw only 0.06 amps of current, and can connect to the raspberry pi.

6.2.9 Heat Cartridge (3Dman 24V 50W Heater Cartridge)

A heating cartridge is the core of how a FDM printer works. A heating cartridge is used to heat the nozzle within the extruder, which then melts the thermoplastic filament. Heating cartridges are made of heating wire, usually with a ceramic core, due to the ceramic's incredibly high melting point. Given our constraint on power consumption, finding the correct heating cartridge for us is significant. We hope to control our heating cartridge precisely enough to heat the nozzle to the desired temperature and then use insulation to help maintain that temperature.

We chose to use the 3Dman Heater Cartridge due to its uses in similar designs, great power specs, and compatibility with our current electrical setup. This cartridge also features a built-in thermistor, enabling the printer control board to regulate the nozzle's temperature accurately.

6.3 Hardware Mounting/Mechanical Components

With so many electrical components, it is essential that our mechanical design houses the electronics safely, while still keeping our design portable and user-friendly. The mechanical design must also maintain proper airflow for the electronics and have the infrastructure for mounting cooling fans to create said airflow. As for mounting the electronics themselves, the design will feature standoffs that limit contact with the hull of the printer itself. Mounting holes must also be considered so that components can be laid out in the proper orientation.

We also would like to avoid any conductive and flammable materials, which could not only interfere with the printer's functioning, but could also lead to severe part damage. The current prototype has room for electronics to be mounted beneath the gantry and has a base plate made of MDF, a wood composite. This is not ideal for heated components, but for a gantry prototype will serve well and with proper precautions and standoffs will maintain enough distance from any hot electronics while also offering no conductivity to interfere with their function.

6.4 Overall Hardware Design

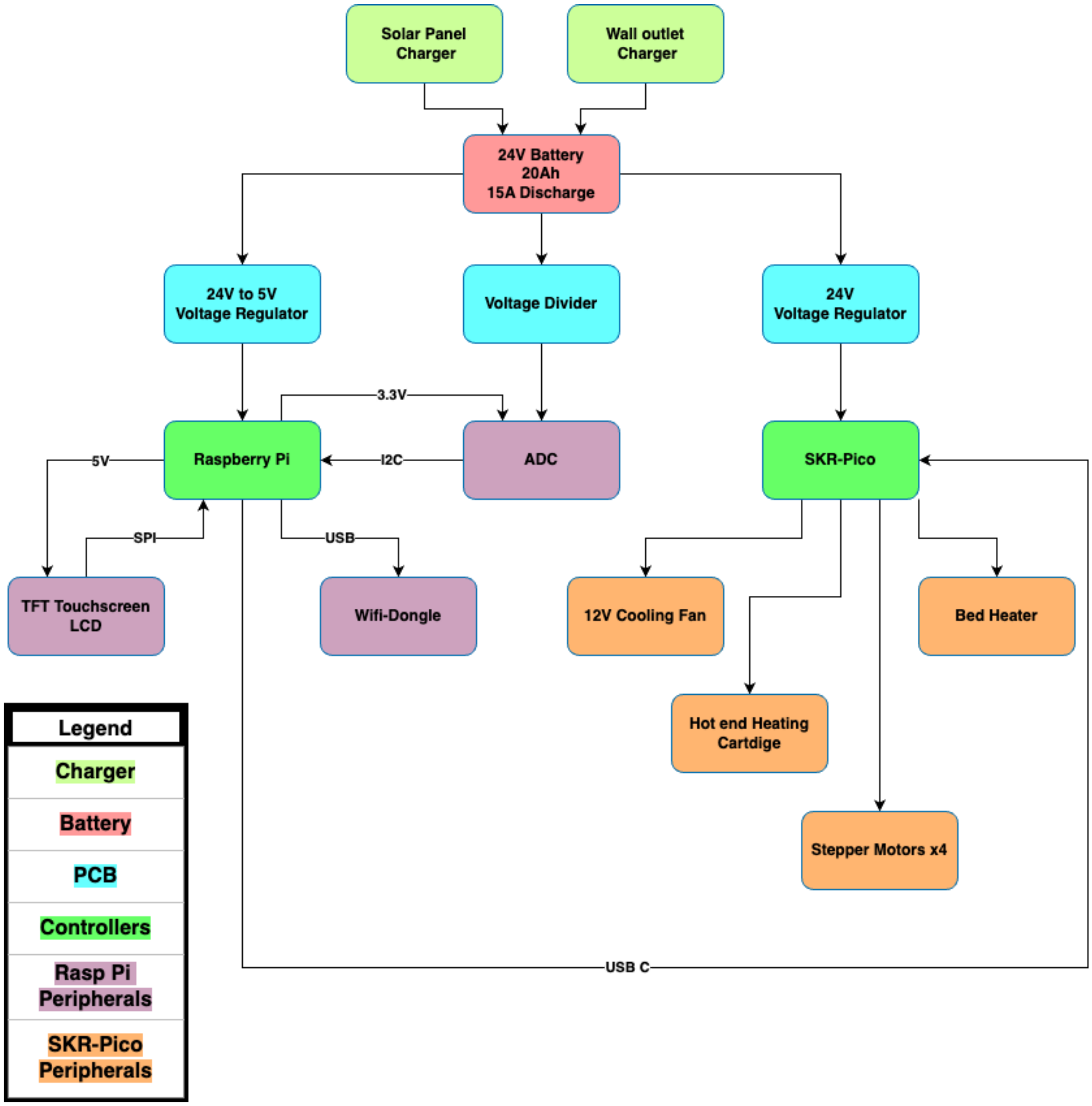


Figure 26: Initial 3DPP overall hardware block diagram

The above block diagram represents the electrical connections of the entire 3DPP system. With all the attached components, the system will draw considerable power that we must consider when building the system. Our main objective here is to eliminate redundant connections and make the system's logic as easy as possible to follow for references and testing purposes so we can visualize any problems with the 3DPP. If we can achieve this practical build, we can create the best version of the 3DPP while being as cost and time efficient as possible.

Another main point of this system that the 3DPP's design team has noticed is that the SKR PICO and all components attached to that specific ecosystem will likely have the highest power consumption in the entire system. The bed heater and the stepper motors, in particular, will draw the most power and generate the most heat. We anticipate the bed heater to take up about 150W or less, a considerable amount of power consumed. Since we are using four stepper motors rated at about 1A, we should expect the power draw for the motors to be about 96W which we will need to optimize the battery power system to handle this process. This portion of the 3DPP will be a great challenge to optimize running the overall 3D printer operations, but we are confident in our research to apply our newfound knowledge to develop a fantastic system.

The 3DPP team must consider the mechanical design when incorporating all electronic and power components. This will include different housing locations on the AutoCAD design that will house components for ease of access and protection from the elements. The mechanical setup will also include the gantry system responsible for moving the X/Y/Z axis of the printer head, which needs to be as precise as possible and not have any kind of mechanical jerking so it can have a smooth placement of the PLA filament. The current prototype of our gantry will be able to fill our printer head movement needs so we can prototype further designs that will incorporate these gantry designs.

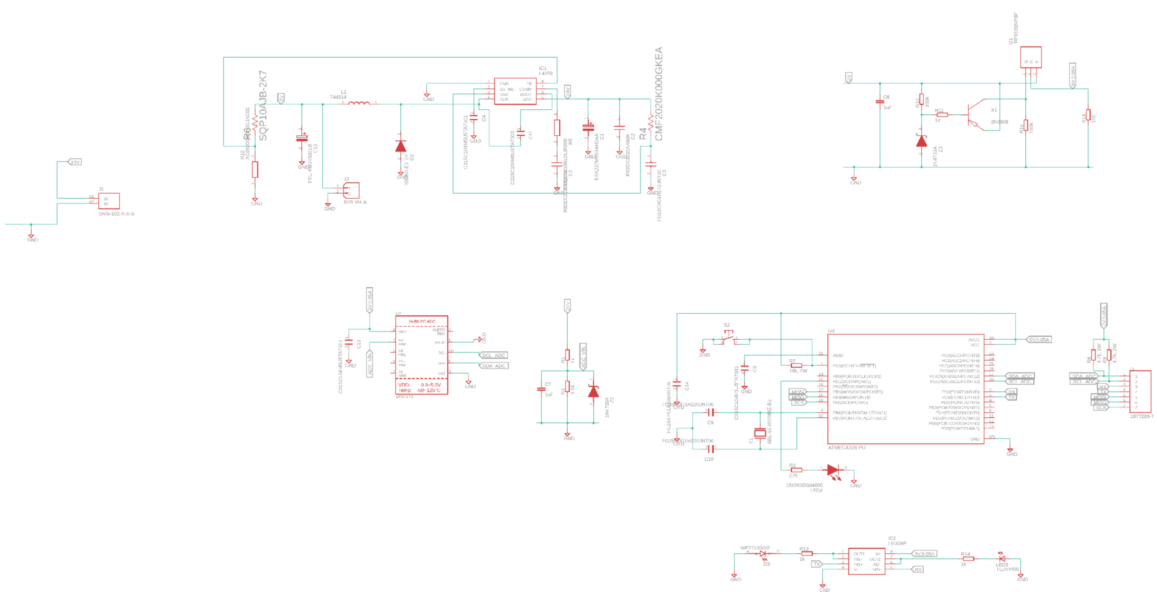


Figure 28: 3DPP Overall Schematic

7.0 Software Design

The software design for the 3DPP team uses diverse software programs and programming languages, all of which must be capable of interfacing with various hardware components to construct a portable 3D printer. The team will employ Python to program various critical functions on the Raspberry Pi, and we will also need to gain a deep understanding of G-Code, the standard language for 3D printing control.

G-Code is important in our project for facilitating communication between the 3D printer controller and the various hardware features, enabling the precise and quick controls necessary for 3D printing. Also, it can issue specific commands to the printer, allowing the team to create custom software commands.

In the following sections, our team will take a deeper look into the details of our design process, shedding light on the different software tools we will utilize and explaining how these tools will be used to achieve our different objectives and requirements. This comprehensive approach ensures a robust and efficient design that meets the challenges of creating a portable 3D printing.

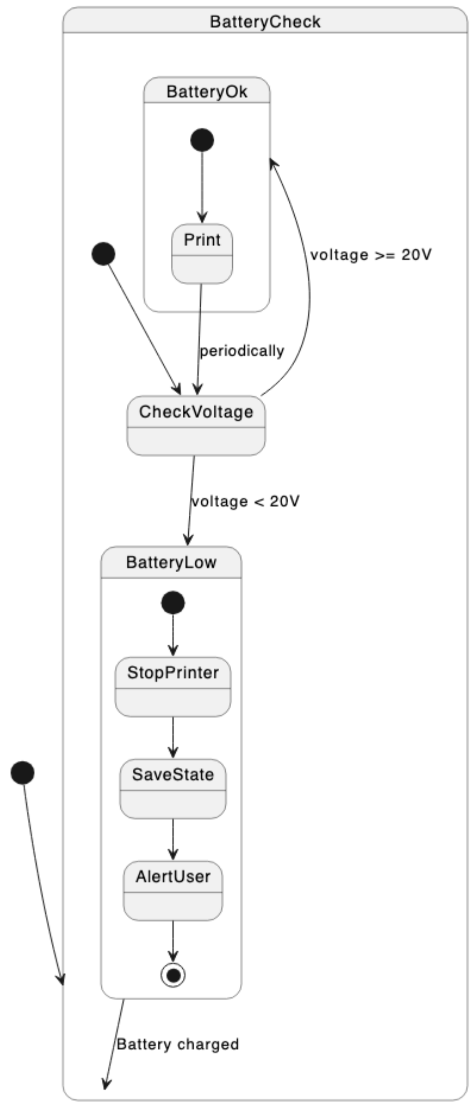


Figure 29: Battery Monitor Software Schematic

7.0.1 Software Key Features

The 3DPP team will be using various software tools and programs to fulfill the software requirements for our project. One of these critical aspects of our project is the ability to monitor the voltage supplied by the battery to the 3D printer. This is an important factor for our project as it ensures the printer receives the correct amount of power for optimal operation. We will be using an analog-to-digital converter (ADC) with our Raspberry Pi to achieve this. This ADC will continuously monitor the voltage level and feed this data to a Python script located on the Raspberry Pi.

The voltage level will be displayed on two interfaces. The first being the 3D printer's touchscreen located on the 3D printer, the second is utilizing a local host network called Mainsail, which will allow remote monitoring of the voltage level. In addition to voltage monitoring, these interfaces will also display other important information about the 3D printer's operation such as the printer's current operating status, the progress of the current print job, and any possible error messages. This information displayed will allow the users to quickly identify and address any issues, ensuring a smooth and efficient operation of the 3D printer.

The vast majority of the software for these features will be developed using Python. Along with this our team will also be using G-Code, the standard language for 3D printing control. This language will also play a important role in our project. G-Code commands guide the 3D printer's movements, precisely controlling each motion's timing, speed, and nature. This ensures that the printer can accurately recreate the 3D models' input by the user. In the following sections, we will delve deeper into the software components of our project, providing further details about how we will implement these features and the benefits they will bring to our 3D printer system.

7.1 Software Design Methodology

The 3DPP team has chosen to implement the Agile software development methodology for our project's software. Agile is an iterative and incremental approach to software development that emphasizes flexibility, collaboration, and customer satisfaction. This approach allows us to adapt to changes and continuously improve our software product based on the outcomes of each phase or 'Sprint.'

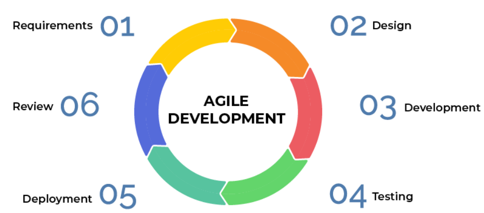


Figure 30: Agile Development Cycle

For our team's project, the Agile methodology is particularly beneficial due to the team's varying levels of familiarity with the technologies that we will be using throughout the creation of 3DPP. For many team members, this project will be their first experience with several of the software and hardware technologies that will be necessary in our team design. Agile allows us to learn and adapt as we progress through the project, adjusting our approach based on the insights we gather throughout the design process. For instance, if we encounter an issue during the 'Testing' phase, such as discovering that a critical piece of software is incompatible with our Raspberry Pi microcontroller, we can quickly revert to the 'Design' phase to explore alternative solutions. This flexibility is a crucial strength of the Agile methodology and will be invaluable in our project.

We plan to use GitHub for our team's software development to facilitate our Agile approach. GitHub is a platform with version control and source code management functionality. It allows us to store, manage, and track changes to our code while preserving previous software versions. This capability is crucial for maintaining an organized and efficient development process, particularly in an Agile environment where changes and revisions are common. Each Sprint will involve planning, designing, coding, and testing stages to produce a working product increment. By storing these Sprints in GitHub, we can effectively document our progress, track changes, and continuously refine our process based on the outcomes of each Sprint. This approach aligns perfectly with the principles of Agile, promoting continuous improvement and adaptation in response to project demands and challenges.

7.2 Software Tools and Packages

As previously mentioned, our 3DPP software team will leverage a diverse array of software tools and packages to fulfill the requirements of our portable 3D printer project. The following section will dive deeper into greater detail about the specific resources necessary to achieve our requirements for our portable 3D printer. These resources encompass APIs, IDEs, programming languages, and hardware interfaces, each playing a crucial role in the successful execution of our project.

7.2.1 Integrated Development Environment (IDE)

An Integrated Development Environment (IDE) is a software application providing a consolidated software development environment. It has essential tools such as a debugger, code editor, and code builder. These tools will be invaluable in our software development process, allowing us to test and ensure that all features of our code are working without syntax errors before transferring the code onto the Raspberry Pi. Given that most of the 3DPP team's code will be developed in Python, we've used PyCharm, a popular Python IDE.

However, we anticipate a challenge in debugging and testing our Python code directly on the Raspberry Pi OS. Instead, we'll have to try it on a separate computer and, if successful, transfer the Python code file to the Raspberry Pi. To create the most efficient 3D printing product, we've selected a lightweight operating system for our Raspberry Pi, the Raspberry Pi OS Lite (32-bit). This OS retains all the features of a full-sized Raspberry Pi operating system but lacks a graphical user interface. This absence makes direct debugging of our code with an IDE on the Raspberry Pi unfeasible.

While Python will be used for most of the logic, including voltage measurement, procedure execution based on specific voltage values, and user information display, we'll also need to use the G-Code programming language. G-Code will allow us to instruct the SKR-Pico, the 3D printer control board, about the connected hardware devices and their intended movements/functions. For debugging and testing our G-Code, we'll use the Mainsail software. Although Mainsail is not strictly an IDE, it's a powerful software that allows us to control our 3D printer and is one of the methods we'll use to display the battery's voltage to the user. Mainsail will be a vital tool throughout our project, capable of identifying errors in our code when we upload our specific 3D printer configuration file.

7.2.2 Raspberry Pi OS Lite (32-bit)

The 3DPP team has chosen to utilize Raspberry Pi OS Lite as the operating system for our portable 3D printer project, specifically on our Raspberry Pi 2 Model B. Raspberry Pi OS Lite is a 32-bit operating system required for our chosen hardware. This operating system is a port of Debian Bullseye, meaning it is a version of Debian Bullseye that has been adapted to work on the Raspberry Pi platform. One of the defining characteristics of Raspberry Pi OS Lite is that it does not come with a desktop environment. This means it lacks a graphical user interface (GUI), and all interactions with the system are done via a command-line interface, typically through an SSH connection. While this may seem less user-friendly, it offers several advantages for our project. Firstly, Raspberry Pi OS Lite is lightweight, running fewer system resources than a full-fledged desktop operating system like Raspbian. This is crucial for our project as it allows the Raspberry Pi to dedicate more of its processing power to handling the G-Code translations required by the Klipper firmware, which is integral to the operation of our 3D printer.

Secondly, the absence of a GUI makes Raspberry Pi OS Lite ideal for "headless" operation, where the Raspberry Pi is controlled remotely without needing a monitor, keyboard, or mouse. This aligns well with the portable nature of our 3D printer project. Lastly, despite its lightweight nature, Raspberry Pi OS Lite still provides full access to the extensive software repositories of Raspberry Pi OS. This means we can easily install any additional software needed for our project. In summary, the decision to use Raspberry Pi OS Lite was driven by its compatibility with our hardware, its lightweight nature, and the flexibility it offers for our specific needs. It provides an efficient, streamlined platform for running the Klipper firmware and controlling our portable 3D printer.

7.2.2 Mainsail

As discussed previously, not only will our team be using Mainsail to test and debug our G-Code, but it will also have many other vital functions for our project to work correctly. First, Mainsail is a web interface for managing and monitoring 3D printer jobs. It's designed to work with Klipper firmware which is the firmware that the 3DPP team decided to use for the SKR-Pico.

Mainsail will be used in our project as an online user interface that allows the user to connect to the 3D printer via a wifi connection and monitor the print job. This will enable the user to gain critical insights into real-time print monitoring, file management, and the ability to control various movements of the printer, such as the X, Y, and Z axis. Now the essential feature Mainsail offers the 3DPP team is the ability to add custom features to the display as well as the ability to call upon Python scripts that are within the Raspberry Pi file system to conduct tasks that the SKR-Pico can not do, such as read the input voltage of a DC battery. Our team will create software that uses an ADC converter to send the voltage of our project's battery to a Python code on the Raspberry Pi. The Python code will read these different voltages and send the data to Mainsail via the Klipper API, where the voltage will then be displayed for the user to see on Mainsail's user interface. Our team will display the battery's voltage within the Console window, along with other information, such as the predicted operational time for the battery, and tell the user it is time to recharge it.

As previously stated we will also be using mainsail for file management. Meaning that the user can store their G-Code files straight on the platform so they are able to print on the go without the need of hooking up their laptop to the printer. Furthermore, Mainsail provides real-time monitoring of the 3D printer's status, offering valuable information such as the current temperature of the extruder and bed, the progress of ongoing print jobs, and the printer's precise position.

7.2.3 KlipperScreen

As mentioned before, not only will our portable 3D printer be able to display messages on Mainsail, a web application meaning that the user will have access to a mobile phone or computer to access it, but also through the Kuman TFT LCD touchscreen display. To do this, the 3DPP team has chosen the software KlipperScreen. This open-source software interfaces the Klipper firmware to a touchscreen display. Some key aspects this will allow the user to utilize are real-time control of the 3D printer, print monitoring, and

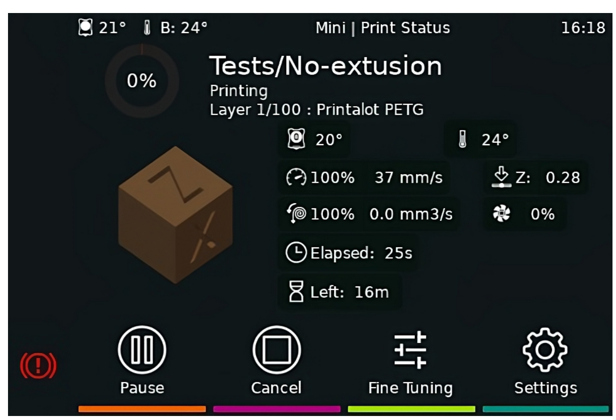


Figure 31: KlipperScreen Print Profile Status

G-Code viewer. KlipperScreen communicates directly with the Klipper firmware running on the same Raspberry Pi. To send commands, KlipperScreen utilizes the web service API Moonraker which allows KlipperScreen to send an HTTP request to Moonraker, which then communicates with Klipper to carry out the command. Similarly, Moonraker regularly sends status updates to KlipperScreen via HTTP.

The most crucial feature that KlipperScreen has for the 3DPP team is that KlipperScreen is open source. This allows us to add our components to the display, such as the voltage level/percentage of battery life, unlike Mainsail, which has a built-in function where a user can send whatever information they want from the Raspberry Pi(in our case, the battery's voltage) with KlipperScreen, our software team will need to go into the open source KlipperScreen code located on Git-Hub and add the ability to display the voltage. Since KlipperScreen is predominantly programmed in Python and our members have a strong background in the language, we will have the necessary tools to complete this task. Unlike the Python code for Mainsail, the Python code for klipperScreen will need us to read values from the battery using the ADC and the Raspberry Pi and write these values to a file within the Raspberry Pi. Next, we will modify the Klipperscreen code to read this file and display the results on the touch screen.

As mentioned, KlipperScreens code is all open source on git-hub, with most of the code written in Python. As a team, we wanted to illustrate better how we plan to edit the KlipperScreen software to fit our requirements better. First, we have to figure out the layout of the KlipperScreens software team's code, meaning that we must figure out what each file does and which files we need to update and change. We also need to know what the code does because we do not want to start adding our code to their software, and it doesn't work or even breaks theirs. Following these guidelines, our team inspected the KlipperScreen GitHub files and found two files our team will need to change. First, we must add some CSS coding so the user interface knows where and what we want to display on the touchscreen. Secondly, there is a file labeled “panels” in the KlipperScreen GitHub, where our team will add our Python code.

Our Python code for editing the KlipperScreen software will look slightly different than the Mainsail code. As mentioned, Mainsail has a built-in feature so users can send whatever information they want to display through the Klipper API. While using KlipperScreen, instead of being able to use an API to communicate with the application, our team will need to write to a .txt file the information we would like to be displayed. We want KlipperScreen to display the battery's voltage, so that is the information we will send to the .txt file. To recap, our team will need to add some CSS code to the KlipperScreen code so that the software knows where and what we want to display. Secondly, we need to create a code in

Python that reads voltages from our battery and writes it to a .txt file. Finally, we will add some code to the KlipperScreen code under the panel's folder that reads the .txt file that displays the voltage and displays it on the touchscreen display.

7.2.4 Moonraker

Moonraker is an API service software installed on Raspberry Pi. It leverages the Klipper firmware to facilitate communication with various other programs via HTTP requests. The 3DPP team will employ this API service to establish communication between the Klipper firmware on the Raspberry Pi and user interface peripherals such as KlipperScreen and Mainsail. These software packages come with built-in modules that convey additional information over the HTTP request, such as print time and status updates. However, by utilizing the Raspberry Pi, we can transmit our own Python scripts and other necessary information for our requirements over this protocol, making it an integral part of our product. While our team will not directly incorporate our code into the open-source Moonraker code, we will extensively utilize this software throughout our design process.

7.2.5 Klipper API

The Klipper API, a crucial component of the Klipper 3D printer firmware, facilitates interaction between external applications and the Klipper firmware. It serves as a conduit for these applications to transmit commands to Klipper and receive status updates, lessening the gap between Klipper and any external interfaces or applications. This API will play a pivotal role in the execution of our team's Python scripts running on the Raspberry Pi, enabling us to communicate with peripherals like Mainsail and relay information such as the battery's voltage. In addition to transmitting voltage information to Mainsail via the Klipper API, we can dispatch other data that aligns with our team's requirements. For instance, we can issue G-Code commands to the printer through the Klipper API using logical statements in Python. One such command could be to halt the 3D printer if the battery voltage drops below 20 volts and to resume operation once the voltage reaches or exceeds 24 volts.

While developing the Python code that utilizes the Klipper API, our team must adhere to its specific syntax format for receiving and transmitting information, as depicted in Figure XXXX. The information must be structured within a JSON dictionary and comply with the following stipulations outlined in the Klipper API documentation: it "must contain a 'method' parameter that is the string name of an available Klipper 'endpoint,''," it "may contain a 'params' parameter which must be of a dictionary type," and "The request dictionary may contain an 'id' parameter which may be of any JSON type." These guidelines ensure smooth and efficient communication between our Python scripts and the Klipper firmware.

The Klipper API works as follows, when a command is issued from a user interface or script, the Klipper API receives and processes it, validating it, and converting it into a format the Klipper firmware can interpret as mentioned previously. The processed command is then sent to the Klipper firmware on the Raspberry Pi, which executes the command and controls the printer's hardware accordingly, which in our case is sending a message to the KlipperScreen and Mainsail. The firmware then sends feedback, such as command execution status or sensor data, back to the Klipper API, which can relay this information to the user interface or script. The Klipper API's flexibility and extensibility allow for the development of custom user interfaces and scripts, and its compatibility with a wide range of 3D printer hardware makes it a versatile choice for many 3D printing projects.

7.2.6 Hostpad

The primary objective for the 3DPP team in creating our portable 3D printer is ensuring portability. Stable internet connectivity is only sometimes guaranteed, especially when the user's mobile device and the Raspberry Pi must connect to the same network to access essential applications like Mainsail. This is where Hostapd comes into play. Hostapd is software installed on the Raspberry Pi that creates an automatic hotspot named "3DPP" when the user operates their 3D printer without an internet connection. The user can connect to the printer and access the various features offered by Mainsail by connecting to the 3DPP Wi-Fi network, launching their web browser, and navigating to http://3dpp.local/. This software automatically turns off the hotspot when the Raspberry Pi detects a familiar Wi-Fi source, such as a home network, or when an Ethernet cable is plugged into the Pi. While our team won't be adding any additional software to Hostapd, it remains a crucial tool in our arsenal to maximize the portability of our printer.

7.2.7 Adafruit_ADS1x15 Software Library

As previously discussed, our team will employ the Adafruit ADS1115 analog-to-digital converter (ADC) to monitor the incoming voltage from our printer and battery. Along with this we will be implementing the Adafruit's software package, ADS1x15, which is made for Python programming on the Raspberry Pi. This software package provides a robust interface for interacting with the ADC via the I2C control pins on the Raspberry Pi, thereby simplifying the process of voltage monitoring.

The ADS1x15 software package offers several key features that makes it easier to get the correct voltage readings. It allows our team to adjust the onboard gain amplifier. It also enables the creation of multiple channels for the ADC, allowing us to monitor numerous voltage sources at the same time. Additionally, the package includes a comparator mode that alerts the users when a specific voltage value is detected, which is very important for our project as. We aim to not to allow the user to continue to operate the printer at low values as this can result in poor print jobs.

Programming the ADC and using the ADS1x15 software package to read the DC voltage coming from the battery is pretty straightforward. After reviewing the documentation on the Adafruit website, we identified the following steps to accomplish this:

1. Initialize the I2C bus.
2. Create the ADC object using the I2C bus.
3. Create a single-ended input on channel 0.
4. Print out the voltage value.

As demonstrated, this software package allows our team to read the battery voltage in just four steps! significantly simplifying the process. Using this software package will be crucial in developing our project, allowing us to focus on more complex aspects while ensuring accurate and efficient voltage monitoring.

7.2.8 Printer.cfg Programming

G-Code, a low-level programming language, is the backbone of our custom 3D printer project. It provides essential instructions for 3D printers and CNC machines, dictating movement, speed, and path. In our project, G-Code will be used in several pivotal ways throughout our project.

The first and most important application of G-Code is creating our printers configuration file for our 3D printer. This file communicates vital information to the SKR-Pico about the printer's structure and enabling the stepper motors to move precisely and efficiently. This is crucial for the printer head to navigate the print space accurately. This file will be saved within the Raspberry Pi under the Klipper folder. This is important because we can edit this file from our hosting software Mainsail. Beyond movement, G-Code also plays a significant role in controlling the extruder, managing the hot end, and interacting with the various sensors connected to the printer. Each hardware component must be assigned to a specific pin on the SKR-Pico, as illustrated in Figure 30. For example in the figure below, the X stepper motor pin is labeled as 2B, 1B, 1A, 2A; using this notation, we will program the config file so that the board knows specifically what hardware parts are hooked up to it.

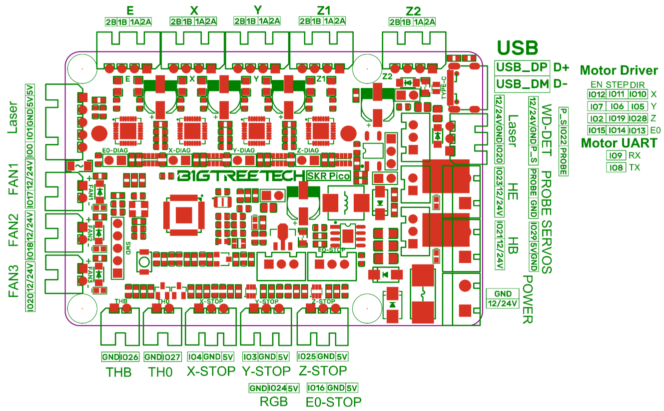


Figure 32: SKR-Pico Hardware Layout

To successfully implement this, the 3DPP team must understand the gantry's operations, which is the mechanical control for the X and Y movements of the 3D printer. Additionally, we must have a good understanding of what the Klipper and SKR-Pico software require such as entering the name of the Raspberry Pi in the config file as mentioned in the Klipper specification sheet also each hardware component will be connected to its respective pin assignment on the SKR-Pico. For example, the X stepper motor will be connected to the X pin assignment on the board.

Next, we will begin adding code to the configuration file. This file must be formatted according to the specifications provided on the Klipper website. Mainsail, with its user-friendly graphical interface, will aid in adding code to the file. The final step in this process is calibrating the printer and the software. This involves moving the printer by millimeters to teach the printer the precise movements needed to print an accurate object. This calibration process is not just a step in the setup; it is a critical component to ensure the accuracy and efficiency of our custom 3D printer. Through this meticulous process, we can ensure our 3D printer will produce high-quality, precise prints.

7.2.9 Programming In Linux without GUI

As mentioned the Raspberry Pi operating system our team has decided upon Raspian OS lite (32-bit) does not have a graphic user interface. Meaning that when we write our Python scripts as well as any other code necessary for our project. We will not be able to use an IDE to test or debug this code. Our team will be conducting our program via an SSH connection to the Raspberry PI. Instead we will be using porgram like nano or vim to be creating our actual Python code. It's a more hands-on, less visually oriented way of working, but it's also more efficient in terms of system resources, which is crucial for a lightweight device like the Raspberry Pi.

7.3 Software Flowchart

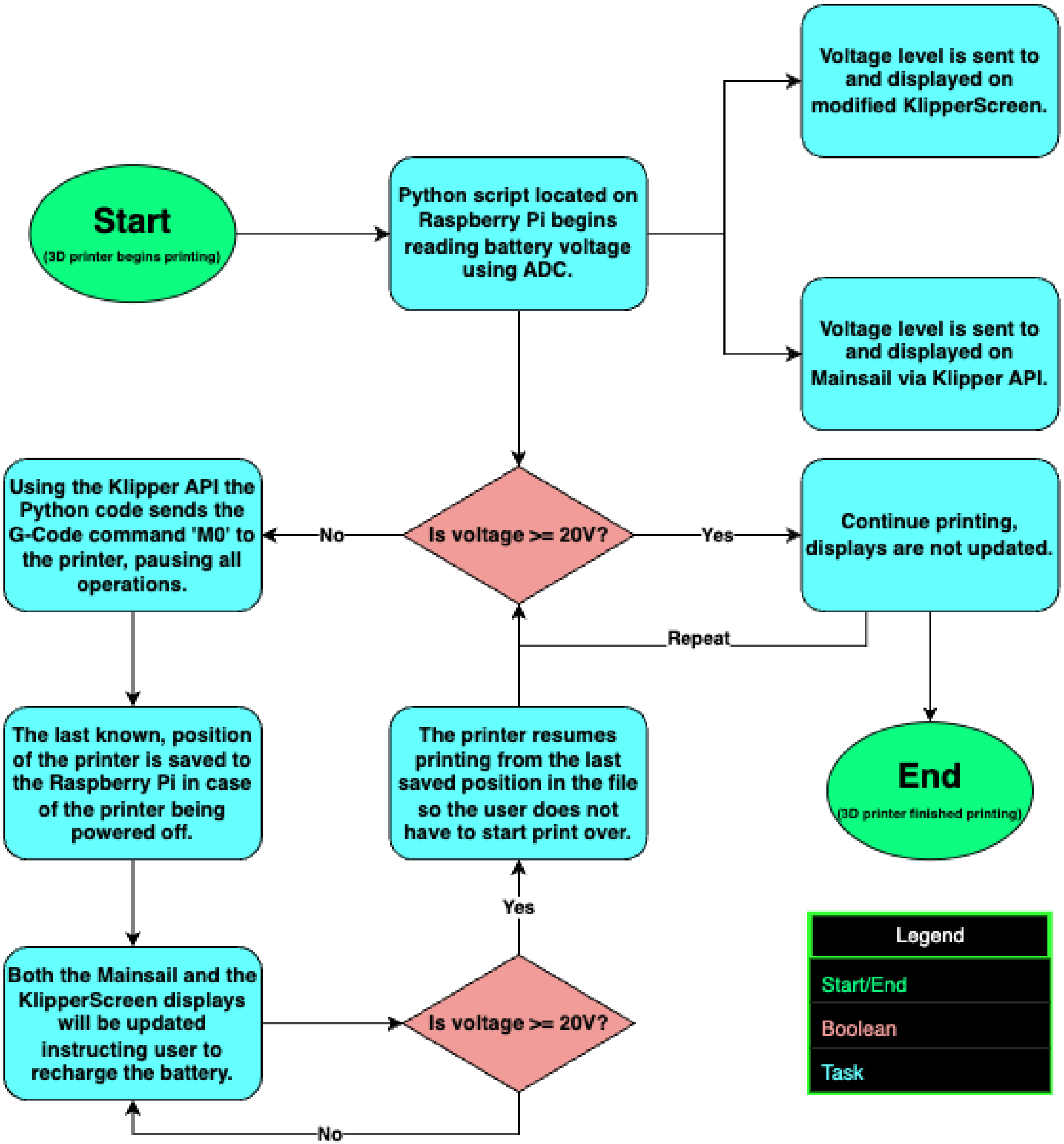


Figure 33: Overall Software Flowchart

The software logic for our project, as illustrated in Figure XXX, is activated as soon as a print job is initiated. The voltage detection script on the Raspberry Pi runs continuously throughout the printing process. This script's primary functions are to read voltage values from the ADC using the Adafruit_ADS1x15 Software Library and assess whether the battery has enough charge to continue printing.

The script triggers a series of actions when the battery voltage falls below the set threshold of 20 volts, the threshold necessary for the printer to continue printing accurately.

Firstly, the Python sends the 'M0' G-Code command to pause the print job using the Klipper API. To send this command as mentioned under the Klipper API section it has to be formatted as a JSON dictionary. It will then update the user interfaces on the Mainsail graphic and the edited KlipperScreen software on the touchscreen display, alerting the user to recharge the battery. Updating the KlipperScreen with this information though is different then displaying it on Mainsail, as mentioned to do this our team will have to write out the commands to a file rather than use the KlipperAPI for this step. Simultaneously, the script records the last known position of the 3D printer, a crucial feature for resuming the print job from the point it was paused if the battery completely drains or the printer is switched off. Once the battery is recharged, the script verifies the voltage level. If the battery is sufficiently charged, the script sends the start-up G-Code commands to the printer, allowing the print job to resume. Upon completion, the user can retrieve their newly 3D-printed object.

7.4 Software Case Diagram

The use case diagram illustrates the interactions within your portable 3D printer system, which includes the User, Raspberry Pi, 3D Printer, and Battery. The User operates the 3D printer, monitors the battery voltage, and views the voltage display on Mainsail and KlipperScreen. The Raspberry Pi, acting as the system's microcontroller, manages these operations, including stopping the 3D printer when the battery voltage is too low and alerting the user.

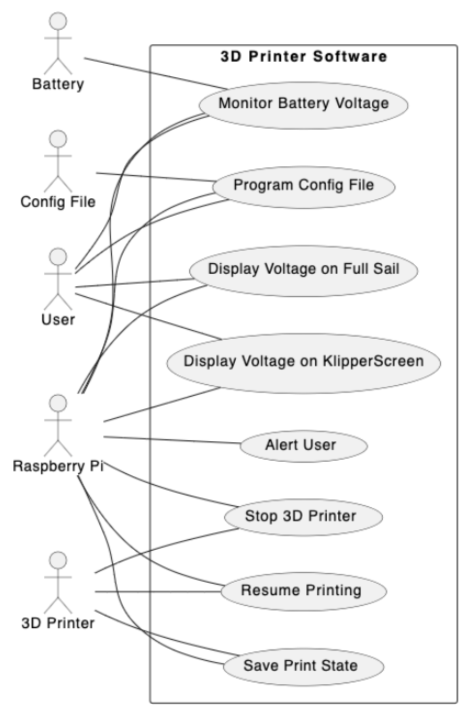


Figure 33: Overall Software Flowchart

The 3D Printer, controlled by the Raspberry Pi, can be stopped and resumed based on the battery voltage. As the power source, the Battery's voltage is monitored by the Raspberry Pi. The use cases, such as "Monitor Battery Voltage," "Display Voltage on Mainsail and KlipperScreen," "Stop 3D Printer", "Alert User," "Save Print State," and "Resume Printing," represent the system's functionalities. These use cases provide a high-level overview of the system's operations and the interactions between its components.

8.0 System Fabrication/Prototype Construction

This section discusses the initial PCB prototype designs that the team produced. Specifically, it will delve into the designs and why specific components are used to fulfill design requirements. Chapter 8 will also describe the design process for the gantry of the 3DPP, such as the use of AutoCAD software and why confident design choices were made.

8.1 12V to 5V Voltage Regulator

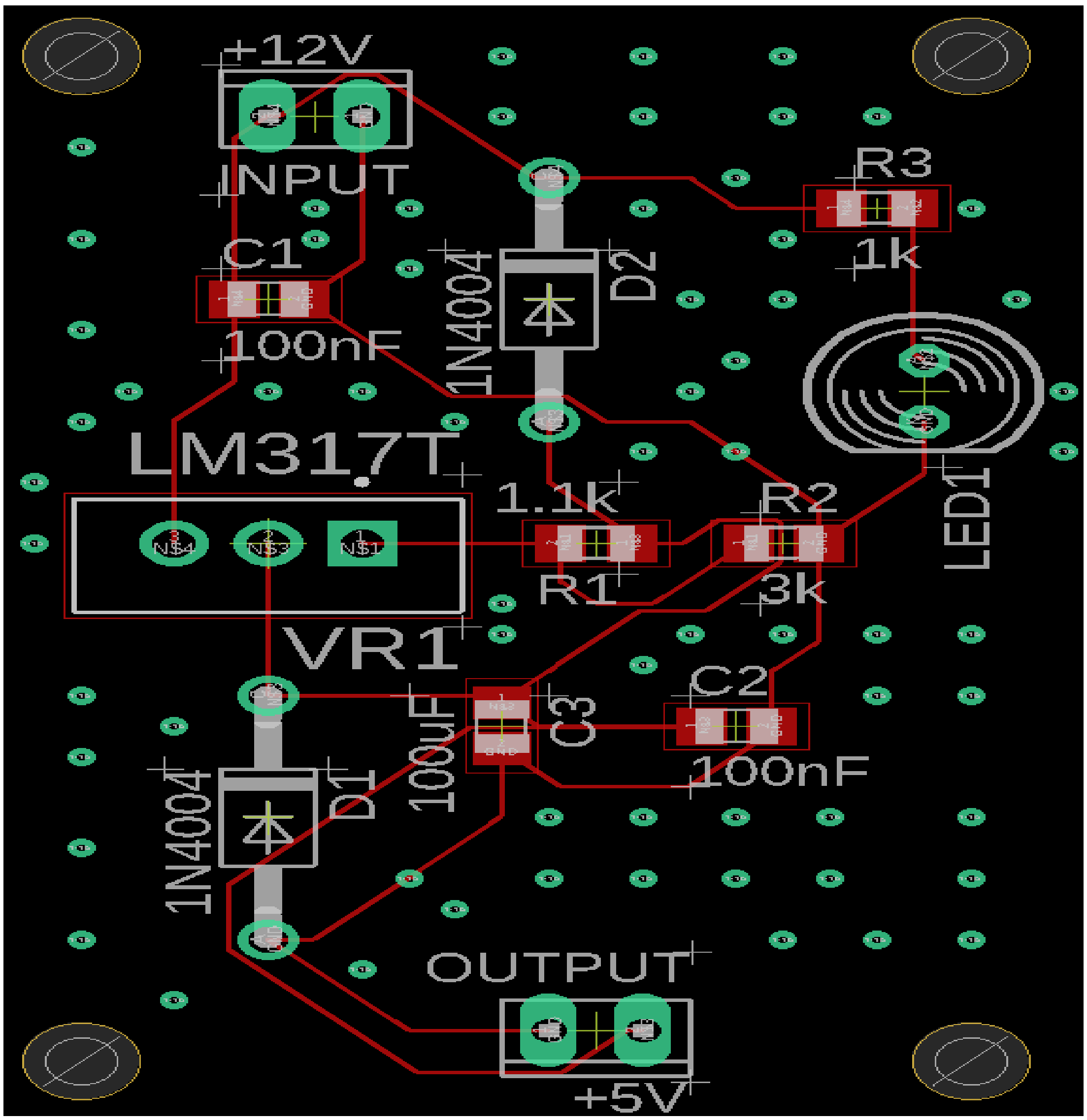


Figure 34: Initial 12V to 5V programmable voltage regulator Eagle PCB design

We designed this specific PCB to have a simple 12V to 5V conversion that will mainly power the Raspberry Pi so that it has a clean, reliable power connection since the Pi will have control over saving the progress of the ongoing print so the user does not lose any progress if the battery becomes depleted. This design was created in Autodesk Eagle, and the dimensions for this particular board are 6.299 x 3.937. We would like to drastically make the dimensions smaller since we want to incorporate all portions of the PCB on one board for aesthetics and easy troubleshooting. The large size is attributed to using diodes and the LED indicator, which is something we can perfect after multiple revisions of this specific PCB design. Also, this design connects the ground plane between the top and bottom layers since we will only be using the top layers for component placing and internal wire connections.

The design and prototype we're working on are in their early stages, and as such, we anticipate that modifications will be necessary as we progress, particularly in relation to the circuit's physical outputs concerning voltage and current. The Raspberry Pi, a key component in our setup, operates most efficiently when it receives a current of 2.0A. This includes adjustments such as setting the LCD touchscreen display to the lowest brightness setting and configuring the ADC for low power consumption. The Raspberry Pi's operating system, being a "Lite" version, is already optimized for low power consumption.

Our design employs the LM317T voltage regulator, chosen specifically for its ability to meet the power requirements of the Raspberry Pi subsystem. The LM317T, according to its datasheet, delivers outputs of 5 volts and 1.5 amps. Given that the Raspberry Pi requires a 2.0A current and the LCD display, which is connected to the Raspberry Pi, adds to the total current demand, the LM317T is capable of providing the necessary power.

It's important to note the inclusion of the 1N4004 Diodes and capacitors in our design. The diodes serve a crucial role in preventing current from flowing into the regulator in the wrong direction, while the capacitors help to reduce ripple voltage and noise, thereby stabilizing the output.

8.2 24V to 6V Voltage Divider

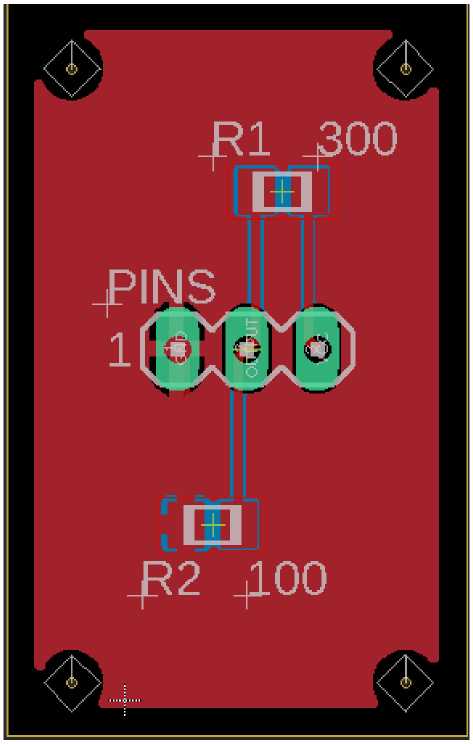


Figure 35: 24V to 6V Voltage Divider Eagle PCB Design

This PCB was designed to be as simple as a voltage divider circuit. The PCB only contains three main components: the 100 Ohm resistor, the 300 Ohm resistor, and the three terminal pin headers located in the center of the board. The central pin header contains the primary ground connection, the input connection point, and the output connection point. This voltage divider circuit was designed to aid in powering the HiLetgo Analog-to-Digital Converter to ensure that the Raspberry Pi and, subsequently, the touch screen LCD receive sufficient input for operation. This PCB was designed using Autodesk Eagle.

Due to the simplicity of this PCB, the width is 0.655 inches, and the height is 1.043 inches. The overall area of the board is 0.68 inches squared. The board has two layers: the top layer in red and the bottom layer in blue. Both layers act as grounds. As the development of the 3DPP is still in the early stages and has not yet reached the point of prototyping, this simple voltage divider PCB may be altered in the future or even discarded for a more efficient circuit as we test components and other potential designs.

8.3 24 Volt Regulator Circuit

While the team has decided on using the Drok DC to DC Step Down Power Supply circuit to power the SKR Pico subsystem for the time being to avoid setbacks to the timeline, designing our PCB that can achieve the same results is still our goal. It has been established that the SKR PICO subsystem will require 24 volts and approximately 9 amps of current to power the controller and all components connected. These power demands are high, so the team must take extra steps and make more considerations than that of the 12V to a 5V voltage regulator with much lower power demand. Designing a PCB to fulfill these requirements will require a regulator rated to have a high enough output voltage and current and a sufficient cooling system because the power dissipation will be too great to run the circuit without one. Furthermore, a transformer will be required to aid in producing our desired output. Unfortunately, most regulators cannot produce such an output, such as the LM317 used in the 12V to 5V course or the LM7824, which, while delivering a sufficient output voltage, is far from capable of producing the required current. The team must adhere to safety constraints, so a potential circuit would also need to incorporate a fuse in the event of a fluctuation that could damage components. These factors alone make the PCB the team would have to design more complicated and expensive.

8.4 Gantry Design

This section presents a detailed overview of the design, fabrication, and prototyping of the gantry system of the 3DPP. To specify, it describes the initial design process and what that entails, ranging from identifying constraints, selection of parts, and overall orientation of the gantry as far as how the linear rails will be oriented and controlled. Then it delves into the CAD assembly of the gantry and then, finally, part fabrication from the CAD files and construction.

8.4.1 Initial Design Process

For the design of the mechanical systems in this project, we wanted to prioritize the speed and accuracy of our gantry. Due to having the constraint of powering our printer using only a battery, the print speed, and consequently, the gantry speed, was essential to the success of this project. Like most engineering decisions, there is a trade-off to maximizing one aspect of a design. In the case of print speed, this often means lower accuracy. This is due to many factors but primarily because the print head is not slowing down enough to print fine detail and jitter. Unlike the first issue, jitter can be mitigated. This is accomplished by moving the center of mass as close to the printer's base as possible. With further research and inspiration from other custom printers like the Positron series, we decided that the best course of action was to print upside down. This design means heavier parts like stepper motors and linear guide rails are at the printer's base, and the much lighter build plate gradually moves upward as the print continues, minimizing the jitter in the print head.

8.4.2 General Part Selection

We decided to use linear and sliding guide rails since they are simple, standard in many 3D printer designs, and engineered to mitigate most friction that could otherwise disrupt precise movements from a belt and pulley system. Initially, we hoped to use synchronous cables to control our gantry due to their ability to change the drive plane, meaning it would be easy to change the elevation of the cable for a multi-axis gantry. With a standard timing belt, this is quite hard to
Do, as the belts are not designed to twist and change the drive plain. Unfortunately, we cannot get any synchronous cable for this project, and as a result, we are using standard timing belts and pinions to drive the motion of the gantry. This decision significantly impacted our initial prototype, though we may switch to a synchronous timing cable design for later iterations.

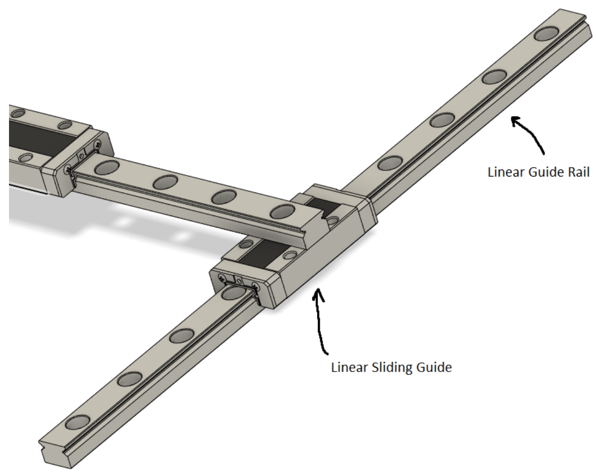


Figure 37: Railway for Gantry System

8.4.3 Gantry Orientation

With the initial design decisions, it was time to decide how the linear rails would be oriented and controlled. A standard gantry comprises a linear sliding rail attached at either side by more linear sliding rails, allowing the print head to move freely in 2-axis and easily controlled by stepper motors in cartesian directions. Usually, this design requires three stepper motors, two for the lower guide rails to drive the guide rail mounted to them on the y-axis and one mounted to the mounted guide rail to move the sliding guide along the x-axis. Removing one of the stepper motors on the y-axis could lead to the gantry getting stuck since the torque on the motor-mounted rail is much less than that on the non-motor-mounted rail. Through our research into 3D printer gantry design, we found that it is possible to drive a 2-axis gantry with only two stepper motors.

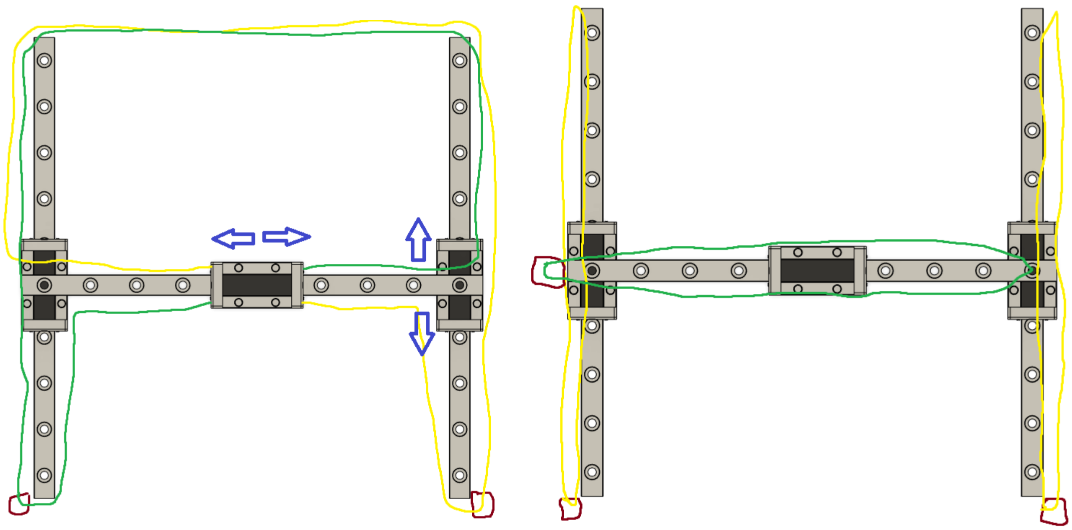


Figure 38: Gantry Orientation for 2-axis Motion

As shown above (stepper motors in red, belt orientation in yellow and green), the same 2-axis motion can be accomplished using a unique belt orientation but with one less stepper motor. The non-cartesian belt orientation (first picture) works because the forces on the linear guide cancel out if both stepper motors move at the same speed, which only causes motion in the y-axis. If only one motor moves, the linear guide will be pulled along the x-axis, and if the motors move at varying speeds, both axes can be controlled at one. This design also saves power due to the decrease in stepper motors and allows the x-axis to be controlled only using one motor. As mentioned earlier, the inability to use synchronous timing cables made it so we designed the prototype around the above gantry rather than the more space-friendly design featured on the Positron series of printers shown to the right.

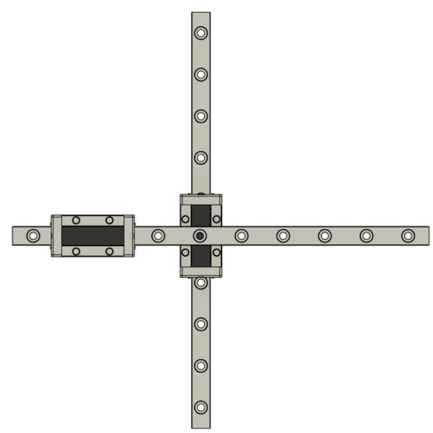


Figure 39: Top View of Gantry Railway

8.4.4 Specific Part Selection

For our initial prototype build, many of the parts that will be used were either parts I already owned or recommended by buyers or other engineers. This helped minimize the cost of prototyping. I have Nema 17 stepper motors, which I have experience using before. A Zeelo GT2 timing belt with matching 20-tooth 5mm bore pinions was used for the timing belts and pinions. These pinions are the perfect size for our build and were affordable. They were also recommended for use on CNCs and 3D printers. 200mm

Figure 40: Belt for Gantry

MGN12H mini linear rails were chosen for the linear guide rails since their size was perfect for ensuring we met our build plate size goal while keeping the printer relatively small. I also have some standard ball bearings used to stabilize the pinions. All additional parts were custom-made.

Figure 41: Nema Stepper Motor

8.4.5 Custom Parts/CAD Assembly

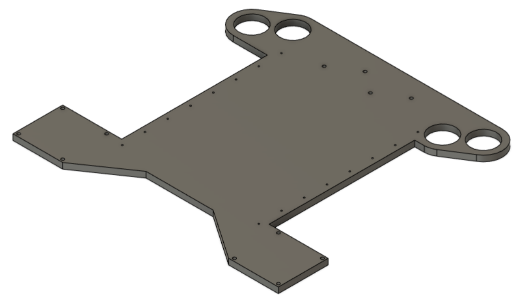


Figure 42: Gantry CAD Base Part 1

The frame of the printer had to support and mount the gantry while also being easy to fabricate. We designed a base that positioned the stepper motors at the ends of the gantry and positioned holes for the ball bearings. The positions of the ball bearings would determine the functions of the pinions, which had to be appropriately spaced and at different heights so the belts would not interfere with one another. We also left pilot holes to make the assembly process more straightforward. Additional parts were made to secure the stepper motors and pinions, each with custom standoffs to ensure proper space for the pieces to fit.

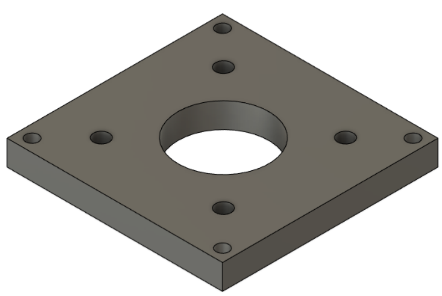


Figure 43: Gantry CAD Base Part 2

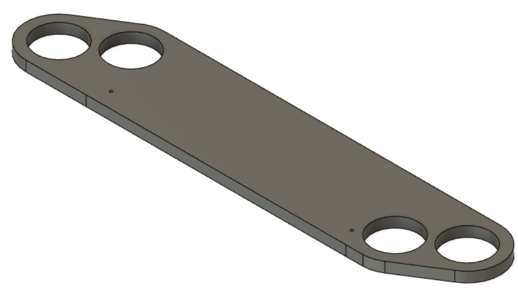


Figure 44: Gantry CAD Base Part 3

Each part was designed with the fabrication and assembly process in mind, so pilot holes were designed based on the fastener that would be used to mount the related components.

An additional part was also needed to mount the y-axis rail guides to the x-axis rail guide. So a custom gantry adapter was designed. Not only does this part serve as the connection point between the gantry stages, but it also serves as part of the timing belt routing and features a mounting hole for a pinion that will turn the belt 90 degrees and run it along the x-axis of the gantry.

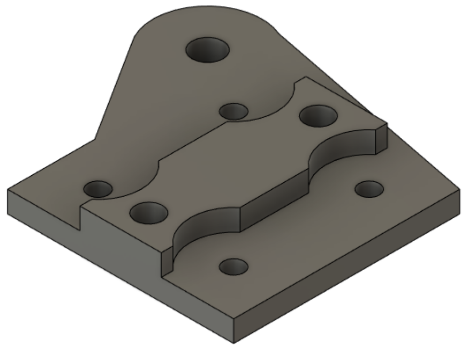


Figure 45: Gantry CAD Base Part 4

After all the parts were created, CAD models of existing and custom features were made into an assembly, serving as a reference for the actual build.

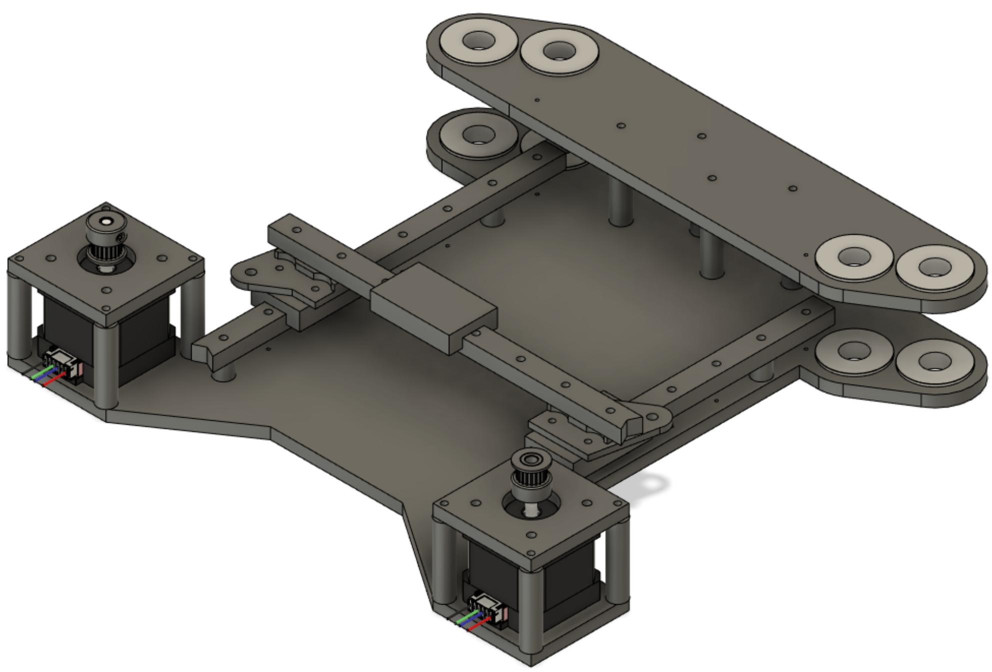


Figure 46: Complete Gantry CAD Diagram

8.4.6 Part Fabrication/Construction

All custom parts were designed to be easily fabricated. The plates used to secure the stepper motors and pinions and the base plate were laser cut in the UCF idea lab. This was done due to the quick fabrication time. These parts were cut from

MDF, a wood composite, which is more than appropriate for a prototype of this nature. For the final design, a flammable material like wood would not be suitable for this printer since the extruder and print bed need to be heated to operate. All the standoffs were manufactured using 3D printing. This allowed them to be quickly fabricated and made them accurate up to 0.5mm.

We have yet to completely finish the prototype assembly due to the time constraint on this paper. However, we have tested the tolerances and concluded that all manufactured parts will serve their purpose well. Many of them are shown below.

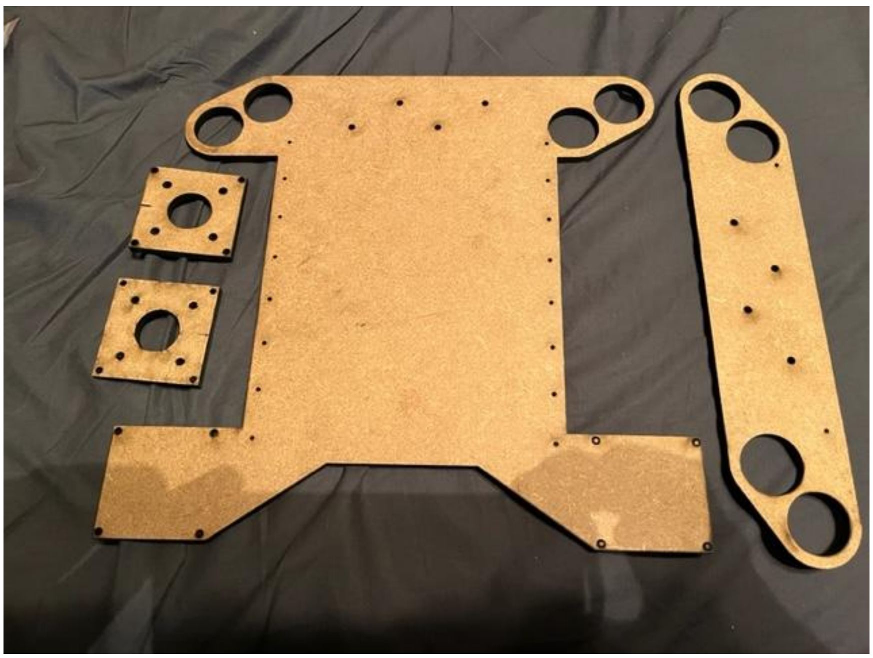


Figure 47: Fabricated Gantry Parts

Figure 48: Gantry Mounting Parts

Figure 49: Gantry Belt Parts

9.0 System Testing

Before entering the prototyping phase of the 3DPP, system testing of all hardware and software components must be conducted. This is a crucial element to the production of this project, given the constraints of a limited budget and the use of high-powered features. Proceeding with the prototyping phase without thorough testing could result in catastrophic system failures that incur unnecessary costs and setbacks in the project timeline.

9.1 Hardware Testing

This section discusses testing the hardware that will be incorporated into the 3DPP. The PCB designs were the first components of the overall design that the team tested. Testing these PCB circuits is vital to ensure they function as intended before ordering them to be manufactured. Testing these circuits entailed both software simulations as well as on hands circuit breadboarding. The voltage divider circuit was the first circuit to be tested due to its simplicity. We tested this circuit using NI Multisim 14.3. The circuit simulated utilizing this software is depicted in the figure below, along with a virtual multimeter measuring the output voltage:

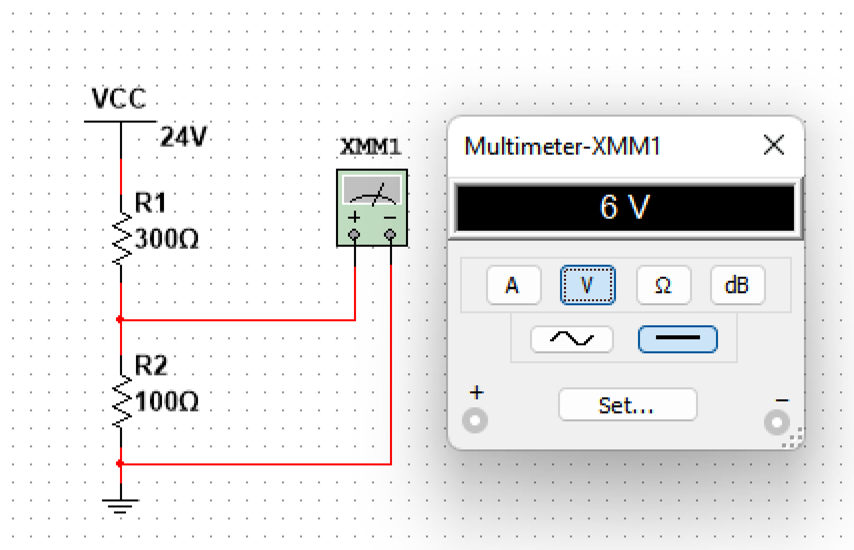


Figure 50: Multisim Testing of Voltage Divider

As seen in the figure above, the voltage divider design is correct, as it takes a 24-volt input and, in turn, produces a 6-volt output due to the 3:1 ratio of R1 to R2. Next, before ordering the PCB to be manufactured, the team breadboarded the circuit to ensure that the actual values of the course will be equivalent to the ideal values produced by the Multisim software. This entailed using one breadboard, a DC Power Supply Unit, two resistors, and a handheld multimeter.

The following figure depicts the voltage divider in its simplest form:

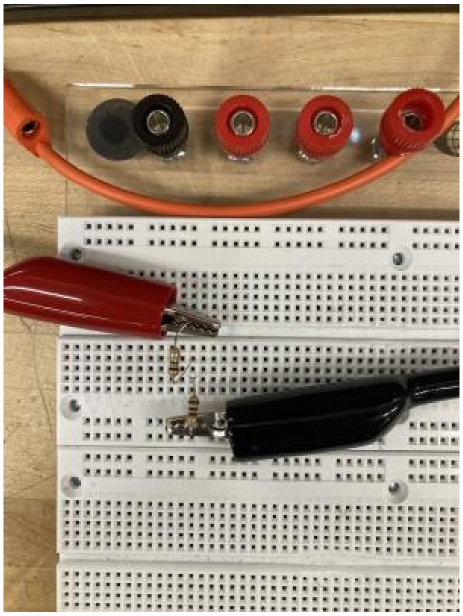


Figure 51: Voltage Divider Circuit on Breadboard

The red alligator clip is the DC voltage source while the black is the ground. Red is connected to R1 and black is connected to R2. As seen in this next figure, the circuit was supplied with 24 volts from a DC power supply to simulate the battery:

Figure 52: DC Power Supply Settings for Voltage Divider

With the entire circuit connected, the team then measured the voltage running through the voltage divider circuit, specifically probing the common node where the two resistors connect and the ground. By doing so, the team determined that the voltage divider circuit does as intended, as seen below:

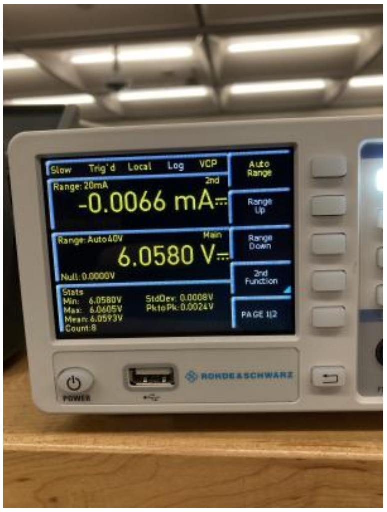


Figure 53: Multimeter Measuring Voltage for Voltage Divider

An important thing that the team took note of while testing the voltage divider is that the resistors became rather hot to the touch. The resistors used were supplied by the senior design lab in room 456 of the Engineering 1 building.

We believe this to be a result of the nature of a voltage divider circuit. Stepping down the voltage using resistors results in power dissipation in the form of heat through the resistors. For the next iteration of this circuit design, the team is considering incorporating a heatsink or finding resistors rated for higher temperatures just to be safe.

The team has also considered using a different type of circuit for supplying power to the Analog-to-Digital converter. During the testing phase of this circuit, we had begun to explore the possibility of using an op-amp voltage buffer with voltage divider circuit. The reason behind this thinking is that the voltage divider circuit on its own could have noise in its output. Using an op-amp with a voltage buffer in addition to the voltage divider will further regulate the output, ensuring that the ADC is receiving the exact input demands it requires.

The team then tested the 12V to 5V regulator circuit using Multisim. The circuit was built using multi-sim online instead of NI Multisim 14.3 because the team needed help finding the specific LM317 on that software version. The circuit is depicted in the figure below:

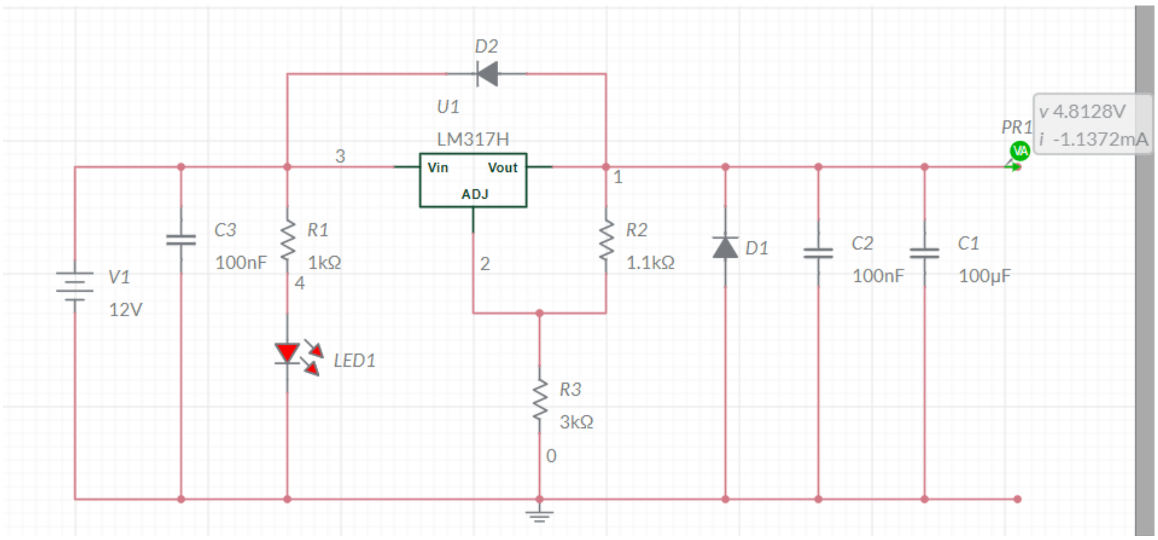


Figure 54: 12V to 5V Voltage Regulator on Multisim

The circuit designed in MultiSim incorporates components that closely match the actual values we intend to use in our final design. This includes the resistors and capacitors, which have the same values as those in our planned circuit. The diodes used in the simulation are a different model but have similar characteristics to those we plan to use. An LED is also included in the circuit design, serving as a power indicator. When the LED emits a red light, it signifies that the circuit is powered and driving a load.

The output voltage of this simulated circuit was measured at 4.8128V, which is 96.256% of the desired 5V output. This voltage level is sufficient to power the Raspberry Pi, which is a key component of our design.

Before we proceed with ordering the PCB, we took an additional step to construct the circuit on a breadboard. This allowed us to compare the actual values and behaviors with those observed during the MultiSim testing. The image below provides a visual representation of how the circuit was configured on the breadboard. This hands-on approach gives us a more tangible understanding of the circuit's operation and helps us identify any potential issues before committing to the final PCB design.

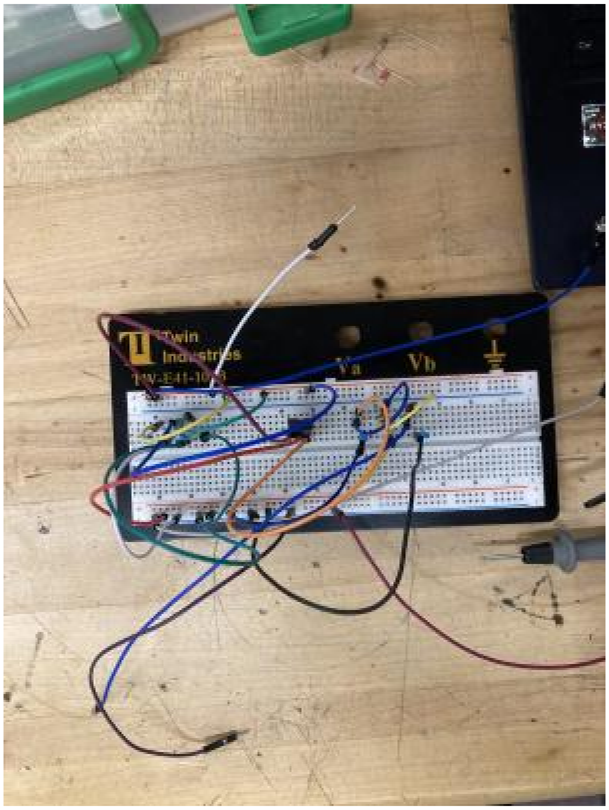


Figure 55: Breadboarding of the 12V to 5V Regulator Circuit

During testing, the team made use of a LM7805 instead of the LM317 used in the circuit. This was because at the time, the team did not have an LM317 on hand. The team made necessary alterations to the circuit to comply with the LM7805 datasheet. The purpose of testing this circuit was to ensure that the team can effectively demonstrate a voltage regulator circuit that correctly regulates a 12 volt input to a 5 volt output. The final product will not use the LM7805.

During testing, the team also began to research the total power dissipation for this circuit as well, and determined that because the circuit will be drawing approximately 1.5 amps in addition to its voltage, it would be a good idea to incorporate a heatsink into this circuit just to be safe.

As seen in the figure below, when measuring the output of the regulator circuit with a multimeter, we found our output to be approximately equal to the intended output of 5 volts:

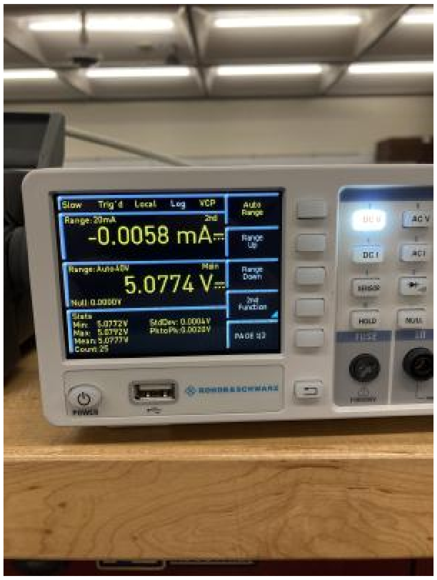


Figure 56: Multimeter Measurement for Voltage Regulator Output

After testing our PCBs on the breadboard and verifying that they work, the team has plans to test each of the prebuilt pieces of hardware. The team will start with making the proper connections for the Raspberry Pi subsystem. This involves connecting the LCD to the Raspberry Pi, the wifi adapter, and the Analog to Digital Converter. We will then connect the 12V to 5V regulator circuit and voltage divider circuit outputs to the Analog to Digital Converter and, finally, connect the battery outputs to the inputs of the breadboard circuits. Once this is complete, the entire subsystem will be connected like the prototype, and each component will be ready to be tested. We will then gradually feed power into the features and use a multimeter to measure the voltage and current running throughout the connections to verify that all values are what they should be. Assuming no issues are present, the team will feed the total 24-volt input into the subsystem and measure the voltage and current at all components one last time. If the team encounters any issues regarding insufficient input values to power the raspberry pi and display, then the team will explore other circuit designs for the power subsystem.

The team will also test the SKR Pico subsystem in the same manner as the Raspberry Pi subsystem. However, testing of this subsystem will only come after the Raspberry Pi subsystem has been found to function correctly. The team will take this precaution because the SKR Pico subsystem has a far more excellent power dissipation and total current load, so in the event of an error, the subsystem can experience catastrophic failures. It is better to test the lower load subsystem first.

Once all subsystems are tested individually and are functioning as intended, the team can test the entire system of the 3DPP. This entails connecting the power, Raspberry Pi, and SKR Pico subsystems. This can be done using the breadboard circuits or the finished PCBs as long as either function correctly. Once the team feeds battery power to the entire system, we will take measurements of all components and analyze the data to determine if this iteration of the 3DPP design can meet our design requirements. After testing, if the team finds that the plan meets all requirements and constraints, it will streamline the design and make it more efficient. However, if the team considers the results unsatisfactory, the team will go back to designing a second iteration of the design while utilizing all data collected during this one.

9.2 Software Testing

The 3DPP team has devised a meticulous and comprehensive approach to software testing for our portable 3D printer project. Our strategy commences with unit testing, where we scrutinize each software component individually to verify its functionality. For instance, we will create a Python script that reads the incoming voltage from the ADC. This script will be tested against a voltmeter to confirm the accuracy of the voltage reading, ensuring that our software correctly interprets the ADC's output.

Additionally, we will test the functionality of Mainsail, our chosen web interface for managing 3D prints. We will ensure that Mainsail is operational on our local network and that we can interact with its user interface correctly. This includes testing all Mainsail's features, such as starting and stopping prints, adjusting printer settings, and monitoring print progress. We will also test the adaptability of the KlipperScreen code, our chosen touchscreen interface for the printer. We will verify that we can modify the KlipperScreen code to display the information we want, such as the current battery voltage.

Following unit testing, we proceed to integration testing. This phase involves merging software components and assessing their interoperability. A critical integration test will be to display the voltage read by the Python code on Mainsail correctly using the Klipper API. We will also combine this code to display the voltage using KlipperScreen. This will ensure that our software components can communicate with each other effectively and display the correct information to the user.

Next, we conduct functional testing to evaluate the behavior of software subsystems. We will simulate a 3D print operation and manipulate the battery voltage to fall outside the acceptable range. The success criterion for this test is the Python code pausing the print and displaying the message "Please Recharge Battery." This test will verify that our software can correctly respond to changes in battery voltage and take appropriate action to prevent damage to the printer or the print job.

Finally, we undertake end-to-end testing, which involves testing the entire software system in a real-world scenario to ensure all components work together as intended. This consists in running print as a whole job from start to finish, using both Mainsail and KlipperScreen to control the printer, and monitoring the battery voltage throughout the process. This comprehensive testing strategy ensures the reliability and readiness of our software for deployment.

9.3 Overall System Integration

The 3DPP has many component system integrations when considering both hardware and software aspects. Firstly, we have a battery voltage checker that will constantly check to see if the voltage produced by the battery is sufficient enough to power the entire system. Our team used an analog-to-digital converter connected to the Raspberry Pi 2B to do this. This is being checked through a Python program using the ADC library to check the analog voltages that will be converted into digital values for our boolean condition statements in the code.

9.4 Plan for SD2

The 3DPP team still has quite a long journey to build the 3DPP system, but since all of the work we have put in during SD1, we should have a smooth transition from prototyping to actual development when SD2 begins. Some of the most critical aspects that need to be emphasized for the 3DPP are the mechanical designs and electrical power systems since these will probably be the project's most complicated and problem-inducing portions. Even though we have initial design prototypes for mechanical and electrical designs, it would be beneficial to have multiple sessions of stress and failure testing so we know our absolute minimum and maximum test cases when we have to demo the 3DPP so it can perform adequately. We also need to fine-tune our software program to correctly interface with our battery life and energy aspects of the 3DPP's power system.

Our plan for the coming weeks is to complete the prototype of the gantry and integrate it with the electronics, particularly the 3D printer control board. That way we will be a step ahead in our software/hardware integration come SD2. We plan to meet in the time between semesters to continue to prototype and troubleshoot various aspects of our design. In preparation for next semester we hope to have the CAD model of the entire build completed during that break, as well as do additional research on crucial components like the extruder, hot end, and heating cartridge, as experimentation on those components was only done theoretically through software like Autodesk Eagle and Multisim. Once we find the appropriate parts that fit our goals and budget, we will order them and use our development kit provided to us at the beginning of this semester to troubleshoot any issues at home. Much like our gantry prototype from this semester, we plan to prototype the z-axis in a similar way: fabrication with additive manufacturing and laser cutting.

Another goal for SD2 is to reduce our total budget for the 3DPP. Our budget is about $890, which we would like to reduce with price comparisons of parts and electrical components drastically. We are also thinking about changing some features if they don’t meet our total power requirements, like the battery or solar panel, but overall we are confident with the selections that were initially made.

10.0 Administrative Content

This section of the paper discusses the administrative content relevant to our project's design process. Three points are described. One is the estimated Bill of Materials which contains the price of each component and the total project cost. Another point is our timeline for this project. The timeline is broken down into Senior Design 1 Milestones and Senior Design 2 Milestones. The Senior Design 1 Milestones include weekly team meetings and critical due dates for the design process. Finally, work distribution is described.

10.1 Estimated Bill of Materials

Table 23: Initial Bill of Materials

Item	Price	Quantity
SKR Pico V1.0 Controller	$29.99	1
Raspberry PI Zero	$40.00	1
Nema 17 20mm (X / Y Motors)	$29.99	2
Nema 14 28mm (Z Motors)	$21.65	1
Nema 14 Round (Extruder)	$20.99	1
3.5in TFT GPIO RPI	$22.79	1
50W Heater Cartridge	$9.99	1
IR Bed Probe	$18.99	1
2510mm Fan 24V	$17.99	3
Kapton Film Heater 200mm	$12.00	1
24V 3010 Fan Blower	$10.00	2
CanaKit Wifi Dongle	$12.99	1
PCB Design	$120.00	1
ADS1115 ADC	$6.99	1
24V Battery Pack	$73.83	2
Droking DC to DC Step Down Circuit	$29.99	1

Estimated Total: $627.98		

The Bill of Materials current estimates the total cost to be around $888.37 depending on the website for each specific item. This exceeds our target budget requirement of less than $600 as outlined in our teams House of Quality diagram. It is important to note though that this figure is not final and is likely to change as we progress with the project. We are currently in the early stages of development of our project and we are committed to taking the necessary financial steps to ensure we meet our engineering budget requirement.

In terms of financing our teams project, our primary goal at this step in Senior Design is to secure sponsorship to cover the costs. If we're unable to secure a sponsorship deal, we have a contingency plan in place. The cost of the project will be divided equally amongst all of the 3DPP team members. If we manage to keep the project cost under $600, each team member would contribute $150 or less. However, based on the current Bill of Materials, each team member would need to contribute approximately $214.60. We're optimistic about finding ways to reduce costs and meet our budget target as we continue to refine our design and sourcing strategies.

10.2.1 Senior Design I Milestones

Table 14: Senior Design I Milestones

Week #	Due Date	Activity
1	05/19/2023	Project Idea Meeting
3	06/02/2023	Initial Divide and Conquer
4	06/05/2023	DC Meeting
5	06/12/2023	Weekly Meeting
6	06/19/2023	Weekly Meeting, Initial Complete CAD Model
7	06/26/2023	Weekly Meeting, Complete Hardware/Software Block Diagram Design
7	06/30/2023	60 Page Draft, Begin PCB Design
8	07/03/2023	Weekly Meeting, Buy Materials
9	07/10/2023	Weekly Meeting, Final Complete CAD Model
10	07/17/2023	Weekly Meeting, Finalize PCB Design
11	07/25/2023	Final Document, Buy PCB

During the first week of Senior Design 1, we conducted a meeting to discuss project ideas. During that meeting, we discussed two potential projects to design. One was a delivery drone, and the other was the project that we decided on-the portable 3D printer.

In our second week of Senior Design 1, we had little progress, as we were still discussing projects to work on and potential ideas and concepts for each project.

In the third week, our team finally decided on the printer project, and we worked on the Initial Divide and Conquer 10-page paper. We held two virtual meetings on Discord to finish the paper.

In the fourth week, we needed more progress. However, we did conduct our first DC meeting to discuss the Divide and Conquer paper. We received feedback about our initial block diagrams and made the proper edits.

In the fifth week, we held a team meeting about ideas each member researched for the project. We began working on the 60-page paper as well.

We set our website up in the sixth week and held an in-person and virtual meeting over Discord. We made good progress on the 60-page paper.

The seventh week was very productive for the team. We finished the 60-page paper and submitted it for review for discussion during our second DC meeting in the eighth week.

During the eighth week of Senior Design 1, the team purchased several circuit components and held a weekly meeting to discuss PCB design.

In the ninth week, the team had another meeting and made more headway on the PCB design. The design for the gantry system was started as well. Finally, the group began working towards the 120-page final paper requirement.

In the tenth week, we have entailed more pcb and Autocad designing and further progress on the final paper. The team held a routine meeting to discuss progress on the project.

Finally, on the eleventh and last week of our Senior Design 1 timeline, we finished our initial PCB and gantry designs, although they may be subject to change during Senior Design 2 prototyping. The team also finished the final 120 papers and conducted several tests on components, and recorded some video demonstrations of major design subsystems.

10.2.2 Senior Design I Work Distribution

This subsection describes the total work distribution for Senior Design 1.

<u>Zachary Yore - Computer Engineer</u>

As the computer engineer of the group, Zachary handled everything on the software side of the project. He ensured that all hardware was compatible with all of the software systems that are necessary for our portable 3D printer. He specifically made sure that using the Mainsail software, Klipper firmware, Printer: cfg Programming, and much more was compatible without an overall system.

As a computer engineer he has programmed in the coding language Python extensively throughout his time in University as well as through personal projects. Which include machine programming language which is generally programmed in Python as well. This has helped the 3DPP team extensively as our team will be writing the majority of our software in Python.

The most extensive of which will be decoding the KlipperScreen softwares' open source code, so that our team will be able to create our own individual functions for the touchscreen used on our 3D printer. Zachary also wrote several significant components of this paper, including chapter seven, discussing software and various software and hardware research elements throughout the document. Along with this Zachary was vital in deciding in which hardware parts we needed for our design. Some examples of this was picking out the correct microcontroller, where he decided to go with the Raspberry Pi 2B along with this he picked out the peripherals that went with this board, such as the CanaKit Wifi Dongle and the ADS1115. He also picked out using the SKR-Pico because of its small stature and ability to run his firmware of choice Klipper.

Jack Qualls - Electrical Engineer

Jack, an electrical engineer, was responsible for all the CAD design of the gantry system. He has a lot of experience with mechanical design through his involvement in the American Society of Mechanical Engineers (ASME) UCF chapter. This made him the natural choice to handle the mechanical aspects of the project, and allowed him to introduce that aspect of engineering design to the team.

He has an interest in power systems, and though circuitry is not his strong suit, he hopes to get practical experience through his involvement in this project. Jack also worked in the Intelligent Systems Laboratory at UCF under Dr. Dazhong Wu, and worked with him to prototype a continuous carbon fiber ceramic 3D printer. His interest in 3D printers has continued through the years, as he has accrued a lot of knowledge through his research. That interest led to the inception of this project.

Overall, he handled the mechanical aspects of the project design. Jack wrote a significant part of chapter 8 about the gantry system, as well as writing most sections on this project's mechanical aspects, including: 3D printer design, filament choices, various parts, z-axis, mechanical capabilities, and more. He also heavily contributed to the research sections for hardware and software, and the use of ChatGPT in this class.

Jack hopes that this project will give him a better understanding of professional electronic design, particularly in product development. He hopes to further develop his skills in multisim and Autodesk EAGLE. Finally, further developed his teamwork and leadership skills, and hopes that his team members see him as a reliable and intelligent peer.

Joseph Tuckerman - Electrical Engineer

Joseph, aspiring electrical engineer, has prior experience with the Eagle AutoCAD software and has a solid foundation and understanding of circuit design thanks to his experience in the Electrical Engineering major at UCF's College of Computer Science and Engineering. Naturally, due to his knowledge in circuit design, Joseph was tasked with designing and researching parts of the printed circuit boards for the power subsystem of the 3D portable printer.

Joseph is responsible for the voltage divider printed circuit board and the research on the Droking DC to DC Step Down Circuit. He was instrumental in all aspects of electronic design, working closely with Mike to ensure that all circuits and custom PCBs were safe and reliable. Additionally he worked with Zach to ensure seamless software to hardware integration. Joseph wrote the chapter on constraints and international standards and, like Jack, contributed to parts of the research sections, such as chapter eight's hardware testing and chapter three's Slicer Software options.

Joseph hopes to break into the power systems industry at a company like Duke Energy or Florida Power and Light. He plans to do this by taking classes on power systems such as Fundamental of Power Systems and Electrical Machinery.

Michael Doyon - Electrical Engineer

Michael, the third electrical engineer of the group, worked on the other parts of the printed circuit board research and design. He has a lot of experience in circuit design and electronics through courses taken at UCF. Also through his involvement with IEEE he has project experience with leading and designing electrical engineering projects. His passions are in the digital logic and computer architecture realm where one day he would like to make this a career for a lifetime. He also has research experience in this domain with Dr. Rickard Ewetz working with computing paradigms with VLSI systems.

Michael is responsible for the 12V to 5V voltage regulator printed circuit board and the overall power system including the solar panel and battery. Michael contributed to many of the significant components of the paper, such as Chapters three, four, six, and eight. He also contributed to the overall hardware design along with Zachary and Joseph to make a seamless hardware/software system. Michael also plans to assist Zachary with the Python programming for the 3DPP system to gain more experience in programming and so Zachary doesn't get overloaded with coding.

Michael hopes to use the experience gained in this project to further develop his knowledge in the field of computer hardware and software integration. He gained much experience with software such as Autodesk EAGLE and Multisim in an effort to strengthen his knowledge and abilities in the field of circuitry. He hopes to one day put these skills to the test in the real world by breaking into the computer hardware industry at companies such as NVIDIA and AMD.

10.2.3 Senior Design II Milestones

Table 25: Senior Design II Milestones

Week #	Due Date	Activity
1	08/21/2023	Weekly Meeting, Review all designs and compatibility checks
2	08/28/2023	Weekly Meeting, Begin incorporation of components and software
3	09/04/2023	Weekly Meeting, Continue incorporation of components and software
4	09/11/2023	Weekly Meeting, Finish incorporation of components and software
5	09/19/2023	Begin hardware and software unit testing all individual parts.
6	09/26/2023	Begin hardware and software integrational testing parts.
7	10/03/2023	Begin hardware and software functional testing parts.
8	10/10/2023	Begin hardware and software end-to-end testing parts.
9	10/17/2023	Ensure that all hardware and software features work as intended.
10	10/24/2023	Ensure that all hardware and software features work as intended.
11	10/31/2023	Ensure that all hardware and software features work as intended.

12	11/07/23	Assembly of all hardware systems, flash Raspberry Pi with any additional software.
13	11/14/23	Assembly of all hardware systems, flash Raspberry Pi with any additional software.
14	11/21/23	Prepare for final presentation. Ensure system works as intended.
15	12/08/2023	Final Presentation

11.0 Conclusion

This section will summarize the goals, achievements, progress, and plan for senior design two. Any conclusion drawn from our work done this semester will be elaborated upon.

Our project began as a simple idea: 3D printing on the go. As engineers we are constantly on the move, and must make important decisions and fixes in precarious situations. Designing a 3D printer that can operate untethered and is portable enough to take almost anywhere is not only a great solution to a practical problem, but also a great challenge for this project. This project incorporates mechanical design and software challenges, with a primarily electrical engineering based core that is more than appropriate for an ECE project. We determined that the scope of this project is appropriate for senior design in its rigor, technical challenges, and utility, while also remaining a feasible challenge for our current capabilities. This project also provides a number of scalable goals and presents many opportunities for design improvement beyond our initial prototype. For example: the mechanical design of the printer currently incorporates a single removable solar panel, though given the opportunity our team would like to design a mechanism to deploy multiple solar panels and integrate smart technology to determine when additional power is required. The mechanical design can also be optimized to have a more ergonomic and consumer friendly shape. Though the current design would meet our goals and be satisfactory for this class, we believe we chose a project that could continue to be developed beyond that point and even turn into some sort of business venture.

Through our design process we learned and documented a substantial amount of research on 3D printers and their electrical components. Our research guided our design decisions in a substantial way, and allowed us to set realistic goals and expectations, and gave us a great start point for our design. Our first substantial breakthrough was discovering the Positron series of printers from a designer named Kralyn3D. The goal of his design was to have the printer fit within a standard filament box. To accomplish this, he incorporated many cost and energy saving features, as well as optimized the printer's performance so that it was comparable, and even better, than other printers on the market. Many of the features on the Positron printers translated incredibly well to our design, including the upside down extruder, non-traditional gantry, and many of the electrical components like the 3D printer control board, raspberry pi and heating cartridge.
One of Kralyn3D’s previous project was designing a portable power supply capable of powering his printers, so we knew that this design also had the potential to operate untethered. This discovery, along with our additional research led to our initial design decisions regarding not only the type of printer we would build, but also many of the components we would use.

Our first decision was between using stereolithography or a traditional fused deposition modeling printer. We decided due to the increased literature on FDM printers to move forward with that type of printer. As previously discussed, many of the mechanical features such as the upside down extruder, were decided to be most beneficial to achieving our goals. As for the electrical components, we needed parts that would not only be compatible with our design, but also would fit within our power constraints. We addressed this by creating a block diagram that was useful for organizing our requirements and breaking them down into subsystems. This methodology allowed us to pick important components like the 24V battery, and then design a PCB or find a part that would allow our power source to provide the appropriate current and voltage to components. After identifying key components and power sources, initial circuit work was done in multisim and Autodesk Eagle to check the validity of the system. Additional parts like the HiLetgo ADS1115 ADC converter and custom PCBs were ordered, so that they could be tested in a lab environment for any issues such as heat dissipation.

One valuable lesson we learned from the research and development process was the prevalence of multiple solutions to the same issue, and how to troubleshoot an issue to find the solution that best fits our needs. There were many times as well where it seemed that our requirements for different parts and components would change as a result of slight shifts in the configuration or requirements of a different component. This made the circuitry and PCB design quite difficult for this project, as we not only had quite demanding power requirements that required dramatic shifts in voltage and current, but we were sometimes unable to solve these issues with just a PCB converter. One such example of this is the use of a buck converter to step down the voltage from 24V to 5V, which requires significant heat dissipation, and if done improperly could not only result in injury to other components but also to us during testing.

After completion of the mechanical and electrical design, components were ordered, fabricated and tested. One of our goals was to have an initial prototype of the gantry up and running by the end of senior design one. This required that the mechanical parts be fabricated in a quick and easy manner. Many of the mechanical parts for the prototype such as the Nema 17 stepper motors and ball bearings were already obtained by one of our team members, which left the linear guide rails, fasteners, and belt system to be ordered, and custom parts like chassis and standoffs to be fabricated. In an effort to quickly prototype these parts, we fabricated them using 3D printing and laser cutting. Though we do not have the gantry running as required, we are in a good position to do so before the start of the fall semester.

As for the electrical components, they were recently benchmarked in the senior design lab to ensure the proper input/output voltage, current, and power was observed for each part. Additionally the custom PCBs were constructed on breadboard and tested for their effectiveness. This testing process may be the most important takeaway from this class, as it gives us direct experience with an essential part of the engineering process, and skills that will certainly be used when we enter the workforce.

Reflecting now on all the work done, our team feels like we are in a great position heading into senior design two. We have all our components for the prototype, have begun significant troubleshooting, and have met all the requirements for this course. We plan to use the offseason between semesters to complete our prototype of the gantry and have it up and running, controlled by the 3D printer control board, and possibly even powered by battery, given we are satisfied with the testing on the custom PCBs. We have a comprehensive plan laid out for senior design two, that will provide us ample time to reach our project goals, the class goals, graduation requirements, and time to improve the aesthetics of our design should we choose to do so.

All the skills learned in this class were crucial to developing us into great candidates for engineering positions in our respective industries, and we are better off having experienced a taste of what working on a professional project team is like. We would like to thank all of our reviewers and coordinators for their support and all the time they've spent preparing us for our futures.

Appendix A - References

How to connect an adjustable voltage regulator. (n.d.). http://www.learningaboutelectronics.com/Articles/How-to-connect-an-adjustable-voltage-regulator

"IEC 62133-2 Safety Testing for Lithium Ion Batteries." IEC 62133: Safety Testing for lithium-ion batteries. (n.d.). https://www.intertek.com/energy-storage/battery-safety/iec-62133/#:~:text=IEC%

2062133%20is%20an%20international,consumer%20electronics%20and%20other%20applications.

"IEEE Standard for Software and System Test Documentation," in *IEEE Std 829-2008*, vol., no., pp.1-150, 18 July 2008, doi: 10.1109/IEEESTD.2008.4578383.

"ISO/IEC/IEEE International Standard - Systems and software engineering -- Life cycle processes -- Requirements engineering," in *ISO/IEC/IEEE 29148:2018(E)*, vol., no., pp.1-104, 30 Nov. 2018, doi: 10.1109/IEEESTD.2018.8559686.

KRALYN3D. (n.d.). *Positron V3 by KRALYN3D*. Youtube. Retrieved June 1, 2023, from https://www.youtube.com/c/kralyn3d. The design can also be found on the creator's website, https://positron3d.com/, and their youtube channel.

Linear Regulator and switching regulator＜what is the difference between linear and switching regulators?＞. ROHM. (n.d.). https://www.rohm.com/electronics-basics/dc-dc-converters/linear-vs-switching-regulators

Locker, A. (2023, June 13). *Top 20: Best 3D printer Slicer Software (most are free)*. All3DP. https://all3dp.com/1/best-3d-slicer-software-3d-printer/

What is Pep 8? Educative. (n.d.). https://www.educative.io/answers/what-is-pep-8

(2022). *SKR-Pico Schematic* [Photograph]. https://www.google.com/url?sa=i&url=https%3A%2F%2Fdocs.vorondesign.com%2Fbuild%2Felectrical%2Fv0_skr_pico_wiring.html&psig=AOvVaw2b3G9nv-aZfgevqyK50g3M&ust=1690336154602000&source=images&cd=vfe&ved=0CBAQjRxqFwoTCOiS-YHfqIADFQAAAAAdAAAAABAD

KlipperScreen (n.d.). *KlipperScreen*. Retrieved July 4, 2023, from https://klipperscreen.readthedocs.io/en/latest/

BIGTREETECH (n.d.). *Skr-Pico*. SKR-Pico Github. Retrieved July 5, 2023, from https://github.com/bigtreetech/SKR-Pico

(2018). *KlipperScreen User-Interface* [Photograph]. https://www.google.com/imgres?imgurl=https%3A%2F%2Fwww.obico.io%2Fassets%2Fimages%2Fklipperscreen-thimbnails-4d83a7c54af614a2463d8dc7bedfa97f.jpeg&tbnid=g6rW7WaprNCePM&vet=12ahUKEwiVzP7z4KiAAxVagoQIHQHsB7wQMygHegQIARB8..i&imgrefurl=https%3A%2F%2Fwww.obico.io%2Fblog%2Fkilpper-screen%2F&docid=RCGEBk4ajSICHM&w=960&h=640&q=klipperscreen%20userinterface&ved=2ahUKEwiVzP7z4KiAAxVagoQIHQHsB7wQMygHegQIARB8

Klipper (n.d.). *Information on Klipper API*. Retrieved July 14, 2023, from https://www.klipper3d.org/API_Server.html

Brush, K. (n.d.). *Agile software development*. Techtarget. Retrieved July 14, 2023, from https://www.techtarget.com/searchsoftwarequality/definition/agile-software-development?Offer=abt_pubpro_AI-Insider

(2023). *Agile Software Development* [Photograph]. https://www.google.com/imgres?imgurl=https%3A%2F%2Fduws858oznvmq.cloudfront.net%2Ffgj_7c73da06d6_0f77201237.jpg&tbnid=zUiL-6J5cd6UFM&vet=12ahUKEwjGteWs4qiAAxUWjoQIHW4RCUcQMygregUIARDMAg..i&imgrefurl=https%3A%2F%2Fwww.appsierra.com%2Fblog%2Fwhat-is-the-agile-methodology&docid=kj4Yf6pKrICNiM&w=1000&h=640&q=agile%20software%20development&ved=2ahUKEwjGteWs4qiAAxUWjoQIHW4RCUcQMygregUIARDMAg

Klipper (n.d.). *Moonraker*. Github. Retrieved July 5, 2023, from https://github.com/Arksine/moonraker

CanaKit (2023, July 3). *CanaKit Wifi Dongle*. Retrieved July 5, 2023, from https://www.canakit.com/raspberry-pi-wifi.html

pimyliffeup (2023, July 3). *Hostapd*. Pimylifeup. Retrieved July 3, 2023, from https://pimylifeup.com/raspberry-pi-wireless-access-point/#:~:text=hostapd%20is%20the%20package%20that,through%20our%20Raspberry%20Pi%20itself.

(2023, July 3). *Raspberry Pi Analog to Digital Converters*. Adafruit. Retrieved July 3, 2023, from https://learn.adafruit.com/raspberry-pi-analog-to-digital-converters/ads1015-slash-ads1115

(2023, July 3). *Mainsail*. Mainsail. Retrieved July 3, 2023, from https://docs.mainsail.xyz/

Raspberry Pi (n.d.). *Raspberry Pi 2 Model B*. Raspberrypi. Retrieved July 3, 2023, from https://www.raspberrypi.com/products/raspberry-pi-2-model-b/

(2022). *CanaKit Wifi Dongle* [Photograph]. https://www.google.com/imgres?imgurl=https%3A%2F%2Fm.media-amazon.com%2Fimages%2FI%2F61whBA2i9QL._AC_UF894%2C1000_QL80_.jpg&tbnid=Bso35GhhkVzJEM&vet=12ahUKEwjftez_5KiAAxXOYjABHZ3HCCkQMygBegUIARCFAg..i&imgrefurl=https%3A%2F%2Fwww.amazon.com%2FCanaKit-Raspberry-Wireless-Adapter-Dongle%2Fdp%2FB00GFAN498&docid=krDS3xReHvlLyM&w=894&h=817&q=canakit%20wifi%20dongle&ved=2ahUKEwjftez_5KiAAxXOYjABHZ3HCCkQMygBegUIARCFAg

(2022). *Adafruit ADS1115* [Photograph]. https://www.google.com/imgres?imgurl=https%3A%2F%2Fcdn-learn.adafruit.com%2Fassets%2Fassets%2F000%2F112%2F710%2Fmedium640%2Farduino_c ompatibles_sensors_raspi_ads1015_i2c_bb.png%3F1656449834&tbnid=iw-pN1jPIbVaQM&vet=12ahUKEwjL1Yq75aiAAxWeZjABHZCFAU4QMygHegUIARDiAQ..i&imgrefurl=https%3A%2F%2Flearn.adafruit.com%2Fadafruit-4-channel-adc-breakouts%2Fpython-circuitpython&docid=xvm5dBQDjevnZM&w=640&h=480&q=raspberry%20pi%20ads1115%20wiring&ved=2ahUKEwjL1Yq75aiAAxWeZjABHZCFAU4QMygHegUIARDiAQ

(2022). *SKR-PICO Raspberry PI* [Photograph]. https://www.google.com/imgres?imgurl=https%3A%2F%2F2779286893-files.gitbook.io%2F~%2Ffiles%2Fv0%2Fb%2Fgitbook-x-prod.appspot.com%2Fo%2Fspaces%252F-MTqLw6KI5FruaRNEMZD-887967055%252Fuploads%252F4Wh5WM62fo52GBg6eXjq%252Fimage.png%3Falt%3Dmedia%26token%3Db6d2c2fc-6915-405e-8ff8-a00a31ef9ce0&tbnid=pVY2wa8MM3UO7M&vet=12ahUKEwiy7fWC5qiAAxWwXTABHTw7CboQMygTegUIARD6AQ..i&imgrefurl=https%3A%2F%2F3dwork.io%2Fen%2Fskr-pico-analysis-and-complete-guide-of-this-great-little-controller%2F&docid=yqu8ywjIXToikM&w=3893&h=2096&q=skrpico%20raspberry%20pi%20usb%20conncection&ved=2ahUKEwiy7fWC5qiAAxWwXTABHTw7CboQMygTegUIARD6AQ

Positron (n.d.). *Positron Github*. Github. Retrieved June 25, 2023, from https://github.com/KRALYN/PositronV3

(2022). *ADS1115* [Photograph]. https://www.google.com/imgres?imgurl=https%3A%2F%2Fwww.wellpcb.com%2Fwp-content%2Fuploads%2F2022%2F04%2F0-3.jpg&tbnid=jVXuF23kn--QSM&vet=12ahUKEwjzjazs5qiAAxUFg4QIHcjXDc8QMygVegUIARCVAw..i&imgrefurl=https%3A%2F%2Fwww.wellpcb.com%2FADS1115.html&docid=6heRRYiTeMaKBM&w=400&h=265&q=ADS115&ved=2ahUKEwjzjazs5qiAAxUFg4QIHcjXDc8QMygVegUIARCVAw

(2022). *Mainsail* [Photograph]. https://www.google.com/url?sa=i&url=https%3A%2F%2Fgithub.com%2Fmainsail-crew%2Fmainsail&psig=AOvVaw3O0qfnUXAvfUDDLzWr9cyT&ust=1690338391750000&source=images&cd=vfe&opi=89978449&ved=0CBAQjRxqFwoTCIDa5qznqIADFQAAAAAdAAAAABAE

(2022). *Pronterface* [Photograph]. https://www.google.com/imgres?imgurl=https%3A%2F%2Fi.all3dp.com%2Fworkers%2Fimages%2Ffit%3Dscale-down%2Cw%3D1200%2Cgravity%3D0.5x0.5%2Cformat%3Dauto%2Fwp-content%2Fuploads%2F2021%2F12%2F22121159%2Fthe-pronterface-dashboard-reprap-wiki-211102_download-1.jpg&tbnid=TCsZ71yoyqrCxM&vet=12ahUKEwjLtoLY56iAAxXGr4QIHRI-B6gQMygBegUIARC6AQ..i&imgrefurl=https%3A%2F%2Fall3dp.com%2F2%2Fpronterface-how-to-download-install-and-set-it-up%2F&docid=pYRSYsgWLz4RzM&w=981&h=568&q=Pronterface%20user%20interface&ved=2ahUKEwjLtoLY56iAAxXGr4QIHRI-B6gQMygBegUIARC6AQ

(2022). *Octoprint* [Photograph]. https://www.google.com/imgres?imgurl=https%3A%2F%2Fcamo.githubusercontent.com%2Fed05aae82d94896f0bd5071eccfbd34b748b3073ecdefa46f1f10efae41f5d46%2F68747470733a2f2f6f63746f7072696e742e6f72672f6173736574732f696d672f73637265656e73686f742d726561646d652e706e67&tbnid=qGmAusUhQKWsVM&vet=12ahUKEwj59bHz56iAAxW0TjABHe06DaUQMygEegUIARDcAQ..i&imgrefurl=https%3A%2F%2Fgithub.com%2FOctoPrint%2FOctoPrint&docid=2MNZXhFIR4bWfM&w=996&h=740&q=octoprint%20user%20interface&ved=2ahUKEwj59bHz56iAAxW0TjABHe06DaUQMygEegUIARDcAQ

(2022). *HBot 3D printer Design* [Photograph]. https://www.google.com/imgres?imgurl=http%3A%2F%2Fwww.buildlog.net%2Fblog%2Fwp-content%2Fuploads%2F2017%2F10%2FH-Bot_diagram_600x339.jpg&tbnid=4mOKDmo-YcNikM&vet=12ahUKEwjStJWa6KiAAxWpgoQIHch_CBgQMyhXegUIARCnAQ..i&imgrefurl=https%3A%2F%2Fwww.buildlog.net%2Fblog%2F2017%2F10%2Fthe-midtbot-a-new-flavor-of-h-bot%2F&docid=Mi_yshQXGIDP2M&w=600&h=339&q=Hbot%203d%20printer%20design&ved=2ahUKEwjStJWa6KiAAxWpgoQIHch_CBgQMyhXegUIARCnAQ

(2022). *Cartesian 3D printer Design* [Photograph]. https://www.google.com/search?q=cartesian+3d+printer+design&tbm=isch&ved=2ahUKEwj59bHz56iAAxW0TjABHe06DaUQ2-cCegQIABAA&oq=cart&gs_lcp=CgNpbWcQARgAMgQIIxAnMgQIIxAnMgcIABCKBRBDMgcIABCKBRBDMgoIABCKBRCxAxBDMggIABCABBCxAzIICAAQgAQQsQMyCAgAEIAEELEDMggIABCABBCxAzIICAAQgAQQsQM6BQgAEIAEOggIABCxAxCDAToECAAQAzoFCAAQsQNQwARYvgxgphVoAHAAeACAAWeIAccDkgEDNC4xmAEAoAEBqgELZ3dzLXdpei1pbWfAAQE&sclient=img&ei=bDO_ZPmPJ7SdwbkP7fW0qAo&bih=833&biw=1512

(2022). *Delta 3D printer Design* [Photograph]. https://www.google.com/imgres?imgurl=https%3A%2F%2Fwww.norwegiancreations.com%2Fwp-content%2Fuploads%2F2017%2F01%2Fcartesian_vs_delta.jpg&tbnid=XNDKoDhNvr0RgM&vet=12ahUKEwif8f7u6KiAAxWbuIQIHRrOA5wQMygBegUIARDOAQ..i&imgrefurl=https%3A%2F%2Fwww.norwegiancreations.com%2F2017%2F01%2Fbuilding-a-large-kossel-delta-printer-pt-1-parts-and-planning%2F&docid=9R1Af8pYmrjkgM&w=800&h=546&q=delta%203d%20printer%20design%20figure&ved=2ahUKEwif8f7u6KiAAxWbuIQIHRrOA5wQMygBegUIARDOAQ