|  |  |
| --- | --- |
|  | **Pegasus Producer 3D** |
|  |  |

EEL 4914 – Senior Design II

Spring 2015

Group #2

Giovanny Vasquez

Hector Arenas

Roberto Tang

**Table of Contents**

1 Executive Summary............................................................................................1

2 Project Descriptions............................................................................................2

2.1 Motivation..............................................................................................2

2.2 Significance...........................................................................................3

2.3 Goals and Objectives............................................................................3

2.4 Division of Labor...................................................................................3

2.5 Project Specifications............................................................................4

2.6 Design Contrains and Engineering Standards......................................4

3. Research............................................................................................................6

3.1 Power Supply........................................................................................6

3.1.1 Possible Power Supplies.........................................................6

3.1.1.1 Types of Power Supplies...........................................6

3.1.2 Selecting a Power Supply.......................................................7

3.1.3 LED Strips...............................................................................7

3.1.4 ATX Power Supply Unit...........................................................9

3.1.4.1 Voltage Protection...................................................10

3.1.4.2 What to Look for on an ATX....................................10

3.1.4.3 Efficiency.................................................................11

3.1.4.4 Modifying ATX for General Use...............................11

3.1.5 Switch Mode 12V/240W Power Supply.................................12

3.1.5.1 Architecture Overview.............................................12

3.1.5.2 Switch Mode Power Supply Features.....................13

3.1.5.2.1 Precise Zero Current Detection.................13

3.2 Mechanical Frame...............................................................................14

3.2.1 Axis Purpose and Functions..................................................14

3.3 Microcontroller Introduction and Purpose...........................................15

3.3.1 Advantages and Disadvantages............................................16

3.3.2 Microcontroller Decision........................................................18

3.4 Control Board......................................................................................19

3.4.1 Motor Drivers.........................................................................19

3.4.2 End Stops..............................................................................21

3.5 General Type of Motors......................................................................21

3.5.1 Direct Current Motors............................................................23

3.5.1.1 Motors with Brushless Direct Current......................23

3.5.1.2 Motors with Brushes Direct Current.........................24

3.5.2 Stepper Motors......................................................................25

3.5.3 Servo Motors.........................................................................26

3.6 Printing Materials................................................................................28

3.6.1 Plastic....................................................................................29

3.6.2 Metals....................................................................................30

3.6.3 Organic..................................................................................31

3.6.4 Printing Material Decision......................................................31

3.7 Extruder...............................................................................................35

3.7.1 Nozzle Size and Precision.....................................................36

3.7.2 Hot End.................................................................................37

3.7.3 Cold End................................................................................38

3.7.4 Single and Double Extruders.................................................39

3.8 Software..............................................................................................40

3.8.1 CAD Models..........................................................................41

3.8.2 CAD to G-Code Conversion..................................................41

3.8.3 G-Code Interpretation............................................................42

3.8.4 Open Source Software..........................................................44

3.8.4.1 CAD Modeling.........................................................44

3.8.4.2 CAD to G-Code Converter Software.......................45

3.8.4.2.1 Slic3r..........................................................45

3.8.4.2.2 Cura...........................................................47

3.8.4.2.3 Skeinforge..................................................49

3.8.4.2.4 Tool Selection............................................50

3.9 Heating Platform.................................................................................51

4 Design...............................................................................................................52

4.1 Electrical..............................................................................................52

4.1.1 Microcontroller.......................................................................52

4.1.2 Control Board........................................................................54

4.1.2.1 Power Input.............................................................55

4.1.2.2 Heaters and Fans....................................................56

4.1.2.3 Stepper Motors........................................................57

4.1.2.4 End Stops and Thermistors.....................................58

4.1.2.5 Control Board I2C....................................................59

4.1.3 Motor Selection.....................................................................60

4.1.4 Motor Drivers.........................................................................60

4.1.5 Wiring....................................................................................63

4.2 Power Supply......................................................................................64

4.2.1 AC Circuit..............................................................................64

4.2.1.1 Multiple Winding Transformers................................64

4.2.1.2 Full-Wave Bridge Rectifier Circuit Design...............64

4.2.2 DC Circuit..............................................................................65

4.2.2.1 Capacitor Selection.................................................67

4.3 Software..............................................................................................68

4.3.1 Reading from SD Card..........................................................69

4.3.1.1 Communication........................................................69

4.3.1.1.1 Writing Data...............................................70

4.3.1.1.2 Reading Data.............................................70

4.3.2 Controlling the Motors...........................................................70

4.3.2.1 The Z-axis’ Motors...................................................70

4.3.2.2 The X-axis’ Motor....................................................70

4.3.2.3 The Y-axis’ Motor....................................................71

4.3.2.4 The Extruders’ Motors.............................................71

4.3.3 Controlling the Temperature..................................................71

4.3.3.1 Extruder’s Temperature...........................................71

4.3.4 Configuration Information......................................................72

4.3.4.1 Dimensions..............................................................72

4.3.4.2 Temperature............................................................72

4.3.4.3 Origin.......................................................................73

4.3.4.4 Minimum Step Size for Motors................................73

4.3.5 Arc Function..........................................................................73

4.3.6 G-Code Interpreter................................................................75

4.3.7 Initializer................................................................................75

4.3.8 Terminating Function.............................................................75

4.3.9 Main Function........................................................................76

4.3.10 Software Flow Diagram.......................................................76

4.4 Extruder...............................................................................................77

5 Project Testing and Prototyping........................................................................79

5.1 X and Y Axis Calibration and Testing..................................................79

5.2 Z Axis Calibration and Testing…….....................................................81

5.3 Extruder Calibration............................................................................83

5.4 Project Operation................................................................................84

6 Administrative Content......................................................................................85

6.1 Milestones...........................................................................................85

6.1.1 Senior Design 1.....................................................................86

6.1.2. Senior Design 2…................................................................87

6.2 Budget and Finance............................................................................88

7 Sources................................................................................................................I

8 Copyright Permissions........................................................................................II

9 Appendices........................................................................................................III

**1 Executive Summary**

The idea to build a 3D printer as a senior design project was commissioned to expand on the ever-growing technology and demonstrate the potential for creation and problem solving that a low cost 3D Printer can have. This will be accomplished by designing a 3D printer that can still compete with existing yet expensive 3D printers in the market. A 3D Printer is simply a device that can translate a three dimensional design on a computer and physically print it out to exact measurements dictated by the software. The project, titled Pegasus Producer 3D, revitalizes the idea that an effective, desktop sized, home 3D printing device is capable of being reproduced at a low cost and has a place in everyone’s home. The Pegasus Producer 3D maintains the established three Cartesian directions and method of extruding plastic but innovates in design, component selection, and accessibility.

A 3D Printer has many exciting applications, it is currently used in industry for product development, data visualization, and rapid prototyping. Most of the potential behind 3D printing lies behind the expansion of its technology utilizing more versatile extruders which is the mechanical part that extrudes the material. The 3D printing movement hopes to bring down the cost of metal printing, continue research on organic and food printing, and is even in the process of testing out 3D printed circuits. Big players such as NASA have recognized the versatility for creation that 3D printing brings and has invested millions of dollars to send a 3D printer capable of working in space to the International Space Station. Although the Pegasus Producer 3D will only print in plastic, it is a matter of introducing the idea of a home built 3D printer capable of creation and innovation

The Pegasus Producer 3D will feature wireless technology to communicate with a computer, low power consumption, dual extruders, a high precision nozzle, and the ability to print in the two main types of printing materials. The intention for the Pegasus Producer 3D is to show how easy it is to build a personal 3D printer that can sit in a home office and become an integral part of a household. The design is straightforward by utilizing a widely accessible microprocessor and circuitry, cost will be kept low by utilizing as many parts from used electronics found in many people’s homes such as salvaging a traditional paper printer for motors and spare home improvement parts to build a chassis.

The Pegasus Producer 3D can be broken down into three subsystems, electrical, mechanical and software. All three mentioned subsystems work together seamlessly to 3D print a design precisely and in a timely manner. This team project was challenging and rewarding and allowed for each member to utilize the skills learned in engineering to design, build, and test one of the most exciting up and coming technologies of this decade.

**2 Project Descriptions**

Engineers strive to create tools and solutions that innovate, aid and better the world. In the past 5 years 3D printing has exploded due to its unique ability to create a three dimensional idea into a physical manifestation. Currently 3D printers are being used to print houses, limbs, organs, food, rocket parts, and basic household items. The material used in 3D printing can be molded to create just about anything. The Pegasus Producer will aim to take part in this revolutionary technology.

**2.1 Motivation**

As engineers we strive to create tools and solutions that innovate, aid and better the world. In the past 3 years 3D printing has exploded due to its unique ability to create a three dimensional idea into a physical manifestation. Currently 3D printers are being used to print houses, limbs, organs, food, rocket parts, and basic household items. The recyclable material used in 3D printing can be molded to create just about anything.  In our project, the idea was to design a small desktop sized 3D printer capable of printing complex plastic designs. The low cost 3D Printer utilizes many items that can be found in any hobbyist’s garage or bought at a hardware store and still remains a simple, elegant and effective design.

We believe that 3D Printing will become the future of manufacturing since millions of dollars are spent creating specialized machines that only create one thing. We want to show that almost anything can be 3D Printed cheaply and in a timely manner with material that does not harm the environment. In a sense, we will be designing the chisel and hammer of the future, a tool of creation.

Although the Pegasus Producer 3D will only print in plastic, it is a matter of introducing the idea of a home built 3D printer capable of creation and innovation. This gives regular individuals the possibility to imagine different inventions and providing them the opportunity to prototype them and see them come to life.The main goal is to deliver a fully functional printer that a UCF student would be able to build using the different resources available at the school for a cost close to $250.

**2.2 Significance**

3D Printers have impacted almost every major industry including medical, automotive, space exploration, and engineering prototyping. We want to bring what we consider the most innovative technology of this decade into everyone's home by building a desktop sized, inexpensive, 3D printer that will fit in any home office or garage. By bringing 3D printers to the average people we hope to rekindle the average person's creativity to create almost anything, or download designs from the internet to print things needed in the day by day use. Such as a screw driver, an ice cream spoon, and perhaps one day even food.



**Figure 2.2-1:** *3D printed cast. Permission Granted By Stratasys Direct Manufacturing*

**2.3 Goals and Objectives**

The Pegasus Producer contributes to “the most eye catching development in printing since Gutenberg invented the printed press 600 years ago”8 by becoming an affordable, desktop sized 3D printer. The main goal was to deliver a fully functional printer with a total cost of production of under $350 for a regular UCF student. Even though our total cost of production was close to $500, we have calculated that a student can replicate the printer for around $200, which significantly improves the goal we set to ourselves almost a year ago. This can be achieved by utilizing all the resources available from the university for the students. Such as the laser cutter in the Texas Instrument lab, the 3D printer located in the same lab, and different elements located in labs throughout the university.

**2.4 Division of Labor**

Since the group is composed of three members, the research, design, and implementation of the different aspects of the Pegasus Producer will be divided among them. Table 2.4-1 shows from which aspect of the Pegasus Producer each person in the group will be in charge of.

|  |  |
| --- | --- |
| **Member** | **Area** |
| Robert Tang | Power Supply |
| Giovanny Vasquez | Chipset |
| Hector Arenas | Software |

**Table 2.4-1:** List of Members and corresponding area of labor

**2.5 Project Specifications**

To be able to measure up to the various 3D printers within the market we had to match or exceed some of the specifications that more expensive 3D printers currently have. We wanted to create a desktop sized 3D printer, something an individual would want to keep in an office or work space, and it also has to be able to print standard ABS or PLA plastic, the standard for plastic 3D printing. On the hardware side we had to make sure that we can process G-code as fast as the competition while utilizing a cost effective microcontroller and control board. The power supply will use your standard U.S wall outlet and provide clean power to the motors, heating elements, and microprocessor. Lastly, the software will be built for speed and efficiency while remaining small enough to not require extensive use of flash memory.

The specifications for the Pegasus Produces 3D are listed in Table 2.5-1.

|  |  |
| --- | --- |
| **Area** | **Value** |
| Printing area | 215x215 (mm) |
| Weight | 20 (lbs) |
| Overall size | 600x560x410 (mm) |
| Printing materials | ABS, PLA |
| Build surface | PCB-heated bed |
| Computer interface | USB and SPI |
| Nozzle size | 0.4 (mm) |
| Accuracy | 0.1 (mm) |
| Resolution | 0.02 (mm) |
| Motors | NEMA 17 Stepper Motors |
| Microcontroller | ATmega 644p |
| Power Supply | 275 W |

**Table 2.5-1:** Specifications for Pegasus Producer

**2.5 Design Contraints and engineering Standards**

Throughout the contrstuction of the Pegasus producer 3D we have utilized specific engineering standards that apply in the industry. It is our duty as engineers to provide these standards which can affect economic, environmental, and ethical portions of our society. Specifically for 3D printing, the social issue in which users can print out usable firearms comes into play. On the environmental side, 3D printers are not necessarily less wasteful than traditional manufacturing.

Nevertheless, some of the engineering standards that appied to the Pegasus Producer 3D will be listed below. Due to how many standards can apply to a single project, only the most relavent standards will be shown and discussed.

* **1394-2008 - IEEE Standard for a High-Performance Serial Bus**
* **1005-1998 - IEEE Standard for Definitions, Symbols, and Characterization of Floating Gate Memory Arrays**
* **694-1985 - IEEE Standard for Microprocessor Assembly Language**
* **388-1992 - IEEE Standard for Transformers and Inductors in Electronic Power Conversion Equipment**
* **GMP.ABS+PC.002 - ABS + Polycarbonate - Heat Stabilized**
* **ASTM F1635-11 - Standard Test Method for in Vitro Degradation Testing of Hydrolytically Degradable Polymer Resins**
* **IEC 60122-2-1 Ed. 1.0 b:1991 - uartz crystal units for frequency control and selection - Part 2: Guide to the use of quartz crystal units for frequency control and selection - Section One: Quartz crystal units for microprocessor clock supply**
* **IEC/TS 62098 Ed. 1.0 b:2000 - valuation methods for microprocessor- based instruments1005**
* **IEC 796-1:1990 - Microprocessor system bus - 8-bit and 16-bit data (MULTIBUS I) - Part 1: Functional description with electrical and timing specifications**
* **IEC 60255-26 Ed. 3.0 b:2013 - Measuring relays and protection equipment - Part 26: Electromagnetic compatibility requirements**
* **IEC 60748-2-20 Ed. 2.0 b:2008 - Semiconductor devices - Integrated circuits - Part 2-20: Digital integrated circuits - Family specification - Low voltage integrated circuits**
* **1349-2011 - IEEE Guide for Application of Electric Motors in Class I, Division 2 and Class I, Zone 2 Hazardous (Classified) Locations**
* **113-1985 - IEEE Guide: Test Procedures for Direct-Current Machines**
* **C37.20.1-2002 - IEEE Standard for Metal-Enclosed Low-Voltage Power Circuit Breaker Switchgear-1998 - IEEE Standard for Definitions, Symbols, and Characterization of Floating Gate Memory Arrays**
* **1301.1-1991 - IEEE Standard for a Metric Equipment Practice for Microcomputers**
* **1275.4-1995 - IEEE Standard for Boot (Initialization Configuration) Firmware: Bus Supplement for IEEE 896 (Futurebus+(R))**
* **855-1985 - Standard for Microprocessor Operating Systems Interfaces (MOSI)**

**3 Research**

**3.1 Power Supply**

**3.1.1 Possible power supplies**

Having an understanding on the functionality of a power supply is very important when brainstorming on a power source that’s adequate for the Pegasus Producer 3D printer. There are many options that would fit the specification for the 3D printer and a proper understanding on the functionality and process of the different implementations is very important. As the name implies the primary function of a power supply is to supply power that is in a certain form, and convert it to a specified form, that best fits the design specification, hence why they are called “electric power converters”. In the conversion process every power supply consumes the energy that’s demanded in addition to the power dissipated by the internal components. There are some general classifications of the power supplies;

*Functional:* These are the types that carry functional features integrated such are regulating and maintaining a steady voltage/current even when there’s a change on its input whereas the unregulated power supply functions differently and it fluctuates with the change of conditions.

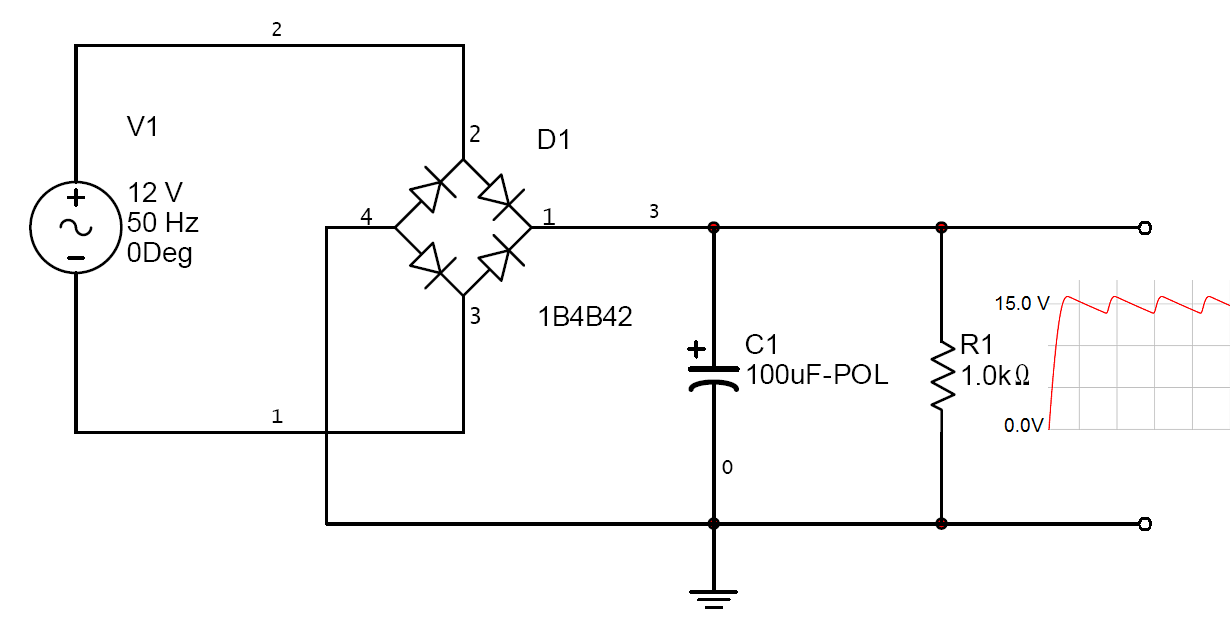
*Mechanical:* A bench source of power supply that tends to be used for purposes such as testing a circuit, this circuitry can be found on mounting bases or even sometimes built in machinery

*Power conversion:* These supplies can be divided into linear and switching types. The more common one linear uses the input power directly along with the transformer and other components such as full wave rectifiers, differently from the switching power supplies; it convers AC to DC in form of pulses before actually processing and these tend to be components that operate predominantly in non-linear modes. Switching coverts are usually more efficiently since components spend less time operating.

**3.1.1.1 Types of power supplies**

*DC power supply:*  The main source of feed for most of the electronics it supplies voltages at a fixed polarity, all the electronic components function utilizing the direct current flow, in which the current flows through the components such as transistor, diodes, MOSFETS resistors etc.

*AC to DC supply:* Alternating current is usually the main source of energy for a lot of the electronic equipment’s. In the 3D Printer the main source would be feed from an outlet that supplies 110Vs roughly at a frequency of 60HZ. Such feed would employ the use of a transformer to convert the high voltage to a lower AC voltage usually in the range of 5-20 volts that can be handled by a rectifier that would convert to an unregulated DC voltage. The electronic filter removes most of the voltage variation that are known as ripples/waves that would prevent the printer circuitry components from feeding AC to the stepper motors. *Figure 3.1.1-1* shows a simplified circuit of a rectifier circuit. This circuit would be implemented in our design in order to convert an AC input to DC in order to rectify the output in order to work with this electronic.



**Figure 3.1.1-1:** *Rectifier circuit feed from an AC input*

*(Google open source image)*

*Linear Regulator:* The functionality of a linear voltage regular is basically being able to make a conversion on a DC voltage to a constant lower DC voltage this would be an important component since we are required to step down the voltage from the main rail (12V) to a separate rail that would provide an output of 5V

**3.1.2 Selecting a power supply**

Now that the crucial point in selecting an efficient power supply that will operate within the specified requirements, it is important to keep in mind safety along with the electrical components that operate at different ratings. The proper selection of the power supply is the most essential component of the 3D printer. The main idea of this section is to research enough information to be able to account and deal with any safety concerns that may arise during the building stages of the 3D printer.

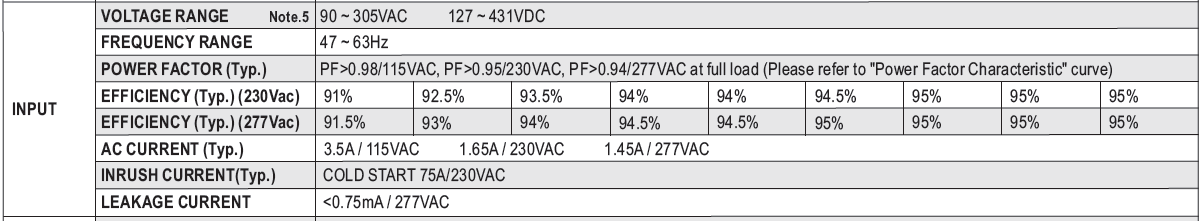
This project’s main purpose is to apply are current electrical knowledge to build from scratch in order to obtain the greatest amount of knowledge. The power supply that we are targeting for this printer ranges between 250 to 350Watts in order to account for the consumption of power throughout the different components, such as the four stepper motors, extruder and the heating bed that require substation dissipation of power in order to operate. Unfortunately for us, “switch-mode” power supplies have what’s considered to be a relatively complex circuitry that is beyond our scope in which it aims to convert alternating current (AC) to direct current (DC) voltages, in which case it is required by the stepper motors and some of the integrated circuits in order to operate. On the other hand linear power supply systems are considered to be completely outdated, which leads to our reassurance that switch-mode is indeed the correct.

**3.1.3 LED strips**

Light-emitting diodes (LED) power supplies are relatively common when it’s strictly designed to provide a fixed DC 12V or even a rail that ranges up to 24V with current that can reach anywhere from 15A to as high as 30A because of the integrated functionality in which the LED collaborate together; due to their small attenuated power consumption it’s able to draw the large currents but intake a small percentage of the total current generated by the power supply. The cost per LED has dropped significantly over the past few years, as the technology has evolved and the idea of more mass production has lowered the cost, allowing LED power supply technology to be considered when making a logistic decision for budget reasons. During the course of the research the attention was drifted towards Powergate LLC which provides different technology along with different power ratings and by targeting their **HLG-320H Series** power supply that provides the design specifications that are required to meet, in order to perform the desired functionality. When selecting a power supply in the Pegasus Producer that would be focused on selecting the one that’s appropriate for the design. *Table 3.1.3-1* shows the input specification that could be tolerated by this power supply, in which it clearly shows that it’s capable of utilizing this system since the input would be provided from a 120VAC outlet, operating at a frequency of 60HZ. *Table 3.1.3-2* shows some of the output settings that are offer by the power supply in which clearly it’s seen that it can be utilized to power the 5 2.5V-0.9A stepper motors individually. One of the main concerns when initiating the building process for this project is to provide safe by all means. *Table 3.1.3-3* shows the protection specifications that are included with the circuitry in order to account for the protection of the circuit in the event of a short. Below would be a discussion on some of the features that are offered by **HLG-320H Series**.

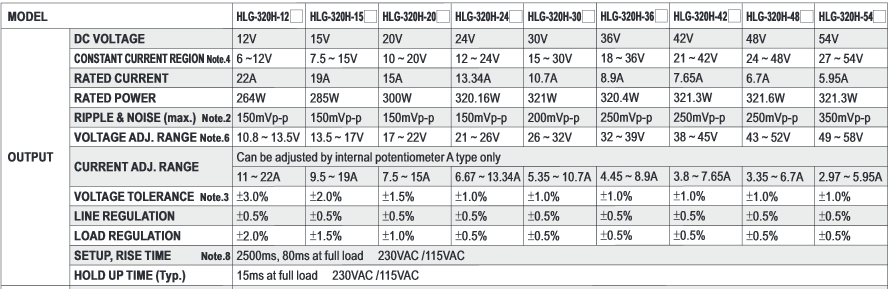
Features

* Ability to have a full AC in ranging up to 305VAC
* Protections: for short circuit, over current, over voltage and temperature
* Compliance to worldwide safety regulations
* Suitable for dry/ damp / wet locations
* Energy efficiency is up to 95%



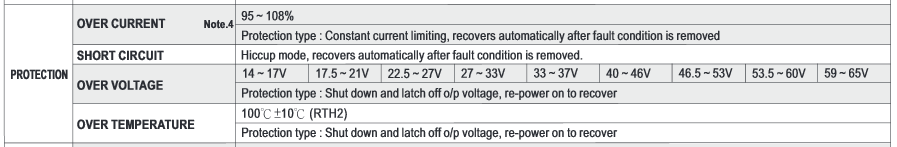
**Table 3.1.3-1:** *Input specification for the* ***HLG-320H series***

*Pending permission from Powergate LLC*



**Table 3.1.3-2:** *Output specification for the* ***HLG-320H series***

*Pending permission from Powergate LLC*



**Table 3.1.3-3:** Protection specification for the **HLG-320H series**

*Pending permission from Powergate LLC*

**3.1.4 ATX Power Supply Unit**

There are certain things that must be taken into account when selecting between power supplies the ATX model has been around for decades and has been produced in mass quantity in the millions so they have it down to a science by this point. The production of these ATX power supplies have also evolved with technology in order to adapt to the forever changing technological advances. The particular power supply that is being targeted in the research is created by Intel. The main reason that an ATX could be chosen over an LED power supply is because of the amount of testing that is carried around the world by hobbyist in which they test reliability on each component. *Table 3.1.4-1* shows nominal voltage values that are provided after preforming lab testing, it’s very clear that the functionality on this power supply can be utilized this power supply since we will be using an outlet that supplies 120V. Now moving to the output stages *Table 3.1.4-2* shows the nominal values for the output stages of the ATX power supply. Stated in *Table 3.1.4-2* these values are adequate to be able to power our four stepper motors along with the heating bed and other components that require the change in voltages.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Minimum** | **Nominal1** | **Maximum** | **Unit** |
| Vin (115 VAC) | 90 | 115 | 135 | VACrms |
| Vin (230VAC) | 180 | 230 | 265 | VACrms |
| Vin Frequency | 47 | - | 63 | Hz |
| Iin (115 VAC) | - | - | 6 | Arms |
| Iin (230VAC) | - | - | 3 | Arms |

**Table 3.1.4-1:** *Shows nominal input voltage values*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **DC Output Voltage Regulation Output** | **Range** | **Min** | **Nom** | **Max** | **Unit** |
| +12V1DC1 | ±5% | +11.40 | +12.00 | +12.60 | V |
| +12V2DC2 | ±5% | +11.40 | +12.00 | +12.60 | V |
| +5VDC | ±5% | +4.75 | +5.00 | +5.25 | V |
| +3.3VDC3 | ±5% | +3.14 | +3.30 | +3.47 | V |
| -12VDC4 | ±10% | -10.80 | -12.00 | -13.20 | V |
| +5VSB | ±5% | +4.75 | +5.00 | +5.25 | V |

**Table 3.1.4-2:** *Shows nominal output voltage values*

**3.1.4.1 Voltage Protection**

The mechanism circuitry that senses the voltage are separate and a different mechanism that are separate from the regulator control and reference. The way the power system is built there should be no fault that’s should disrupt the voltage conditions on any of the output. This system of protection is a crucial tool in order to guarantee proper functionality of our stepper motors without causing any strains due to voltage changes. This supply functionality provides a latch-mode that protects the system. *Table 3.1.4.1-1* shows different over voltage protections for the different rails.

|  |  |  |  |
| --- | --- | --- | --- |
| **Output** | Minimum (V) | **Nominal (V)** | **Maximum (V)** |
| +12 VDC | 13.4 | 15.0 | 15.6 |
| +5VDC | 5.74 | 6.3 | 7.0 |
| +3.3VDC | 3.76 | 4.2 | 4.3 |
| +5VSB1 | 5.74 | 6.3 | 7.0 |

**Table 3.1.4.1-1:**  *Over/Under Voltage Protection*

**3.1.4.2 What to look for on an ATX**

There are many brands and in various lines that could be utilized to build our 3D printer. After doing a good amount of research there seems to be a good amount of desirable features that must be targeted when selecting the ideal power supply

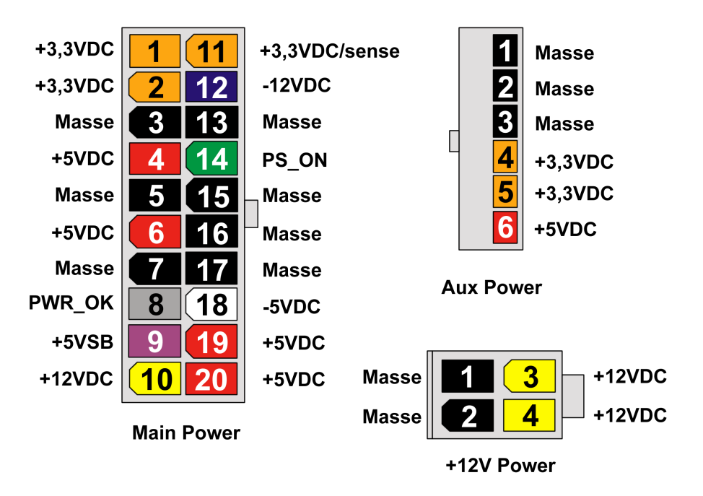
* Power required is within the discrepancy of 30% on the desire 12V rail
* The system must be able to accommodate for an active PFC
* The system must have a cooling systems to dissipate the heat
* We are looking for efficiency labels such as Bronze
* Must be able to toggle between on/off switch.

**3.1.4.3 Efficiency**

When building the Pegasus Producer 3D printer the ATX is equipped with enough efficiency to maximize the delivery of the true power to the system. They were originally designed to provide enough power for up to 250W ratings that are used by the average computers nowadays. Most are designed around roughly 40% greater to the system in order to protect the system against an unusual behavior that the system might experience during normal operation, such as performance degradation on the stepper motors or the heating bed in the 3D printer’s electrical system, due to the fluctuation of the stepper motors in our design. ATX provides roughly 80% efficiency in which the remaining 20% would be dissipated through heat in the components, making ATX a great candidate to provide sufficient power to the printer.

**3.1.4.4 Modifying ATX for general usage**

There are many versions of ATX power supplies that require a minimum power in order to function properly. Some of the power supplies require a minimum of 4.7Ω as a dummy load in order to initiate functionality, otherwise there won’t be any voltage across any of the rails because of the standby state in the PSU due to zero resistance. In order to bypass this functionality there must be a bridge created between the (green wire) and ground. Generally many of the outputs are disregarded in the Pegasus Producer but the black cables, which are all combined to provide a path to ground and the yellow cables are also combined to provide +12V output.



**Figure 3.1.4.4-1:** *Pinouts of an ATX PSU*

*Google open source image*

**3.1.5 Switch Mode 12V/240W Power Supply**

One of the possible designs that are being considered is the switch mode power supply. This power supply should be able to output a steady DC voltage of 12 volts, and generate around 240 watts. The current that it outputs should be around 17 amps to be able to support the heatbed, the extruder, the motors, and the microcontrollers simultaneously.

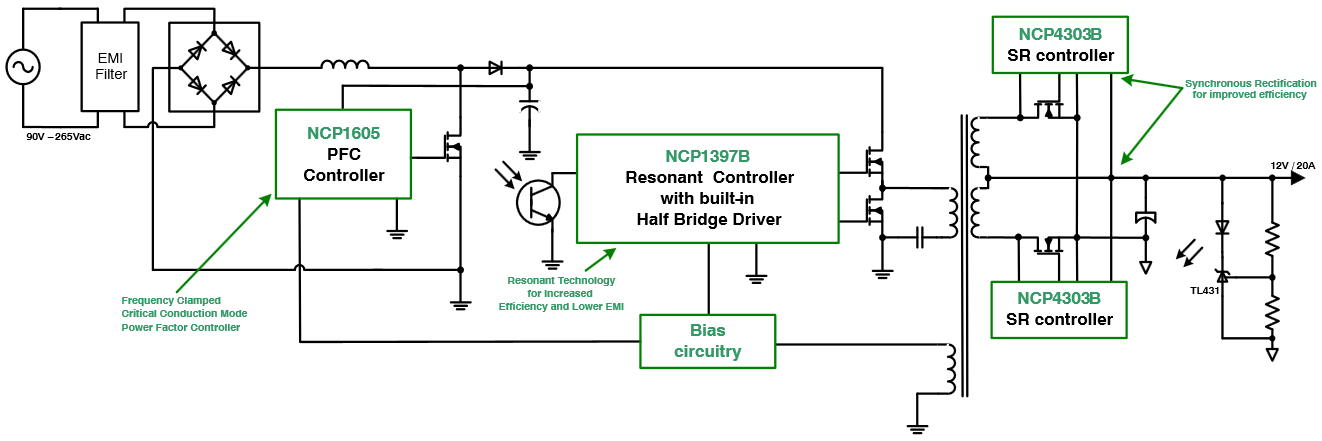
**3.1.5.1 Architecture Overview**

This particular architecture describes a 12V/ 20A output switch mode power supply (SMPS) which would be equivalent to an ATX. The main reason this would be a great choice in selecting a power supply is because It’s less complex than an ATX power supply and it would consist of a double sided printed circuit board with an excellent performance along with minimizing the production cost.

There are many excellent features on this power supply, the circuit uses the chip NCP1605 in order to provide a power factor correction, in which case it allows maximum optimization of the downstream converter. This operation on the NCP1605 is performed by using frequency clamped conduction.

By utilizing a Half Bridge Resonant converter the power also provides more efficiency since it reduces a lot of the Electromagnetic interference and that allows utilizing the transformer more efficiently. The neat thing about this power supply is the way the chips are integrated together to achieve maximum efficiency. Once the circuit clears the half bridge resonant it proceeds over to the synchronous rectifier that is implemented on the secondary side, in which the NCP4303B controller then takes care of the accuracy of the turn-on and turn-off of the MOSFETS.

In summary to the architecture overview this design was mainly selected to maximize efficiency without maximizing the component cost and circuit complexity. Similar to an ATX the power supply would have a 12v rail that would supply power to the RAMP 1.4, from the ramp it would then need to distribute 10A to the heating bed, 5A to the stepper motors and roughly less than an Amp for the ATmega 2560. Efficiency tends to decrease the higher the output current, but the synchronous rectification takes care of these losses on the secondary side. Figure 3.2.1 shows a block diagram and an overview of what the circuitry for the power supply would look like, followed by *Table 3.2.1* that describes the specification for this particular power supply, It is very important to follow these specification in order to achieve the best outcome when designing the power supply for the Pegasus Producer.



**Figure 3.1.5.1-1:** *Block Diagram layout of circuit. Permission from On Semiconductor*

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirements** | **Min** | **Max** | **Unit** |
| Input voltage (AC) | 90 | 265 | V |
| Output voltage (DC) | - | 12 | V |
| Output current | 0 | 20 | A |
| Total output power | 0 | 240 | W |
| Consumption for a 500mW load on STBY | - | 1.7 | W |
| Consumption for a 100mW load on STBY | - | 1.2 | W |
| No load consumption SR operating | - | 870 | mW |
| No load consumption SR turned off | - | 1 | W |
| No load regulation |  | 20 | mV |

**Table 3.1.5.1-1:** *Specification of circuit*

*Permission from On Semiconductor*

**3.1.5.2.1 Precise Zero Current Detection**

This power supply design benefits from a default zero current detection (ZCD) in order to detect if there is a load attached to it. The 100micro Amps attached to the input allows the threshold by utilizing a resistor to maximize the MOSFET conduction in which case it all leads to optimization of the system. The sensing on the secondary side allows proper protection of the transformer from the reverse current and preventing it from flowing back in it. It does this by tuning off the SR MOSFET as fast as possible accounting for the propagation delay of the NCP4303 which is fairly quick sitting at 40nS. One error that could be expected to encounter during the construction of this power supply for the printer is that a high secondary voltage could induce an error on the parasitic inductance of the SR MOSFET package. Parasitic error voltages can impact the circuit by switching the drain to source voltage and cause It to turn off prematurely and causing the efficiency to decrease drastically. The NCP4303 introduces a feature that allows it to compensate for the effect through special input that would offset the ZCD comparator threshold with a special voltage that would a more unique precise detection.

The high voltage of the clock signal pin allows for a straight connection with the drain on the MOSFET, this facilitates the connection by not having to use high resistors, which leads to cheaper manufacturing cost.

Independent minimum on and off featured on the MOSFET allows to overcome false switching due to this parasitic ringing. The NCP4304 Vcc input can be connected directly to the application directly avoiding extra cost on extra components for driving purposes. The chip features a gate clamp driver for either the 12V (NCP4303A) or 6V (NCP4303B).

**3.2 Mechanical Frame**

Despite the fact that the frame for the Pegasus Producer is being purchased, it is important to point out the different factors that led to the decision of the specific chassis model being bought.

**3.2.1 Axis Purpose and Function**

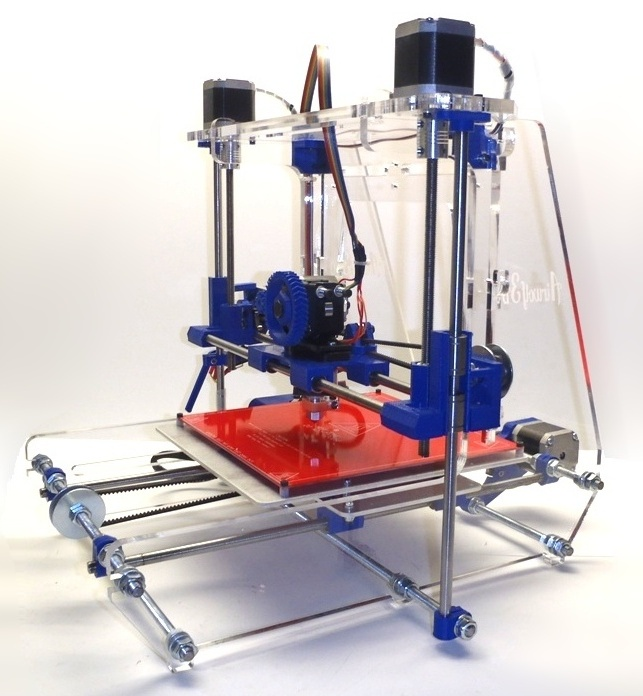
The z-axis has a big impact in the performance of the 3-D printer since it usually involves moving the extruder head or the heatbed. This type of movement usually deals with significant weight, therefore it needs to be decided what type of motors and how many will be used for this specific movement.

Taking into account that the chosen motors for this project are stepper motors, two of them are needed to make sure that weight is supported whether the heatbed or the extruder are being moved.

The first design that was considered consists on a static extruder in the z-axis. The head can move freely in the x axis, but the heatbed moves in both y and z. The main problem with this model is that it complicates the wiring of the printer. The chipset cannot be located under the heatbed for obvious reasons; thus, providing an unaesthetic view with wires laying around.

The second design that was considered was the one where the extruder would move in both x and z axis, while the heatbed would move in the y axis. For this one the extruder would be on a vertical rod with support on both sides. To make sure that the rod stays perpendicular to the heatbed, two motors are utilized, one on each of the supporting rods. For the actual movement along the Z-axis, these rods will have guides encrusted in them in spiral, the motors will rotate them, and the structure that holds the extruders will follow these guides.

The chosen model was the second one, mainly for aesthetic reasons. It can be seen in Figure 3.2.1-1.



**Figure 3.2.1-1:** *RepRap chassis. Content protected by Creative Common Licenses Attribution.*

**3.3 Microcontroller Introduction and Purpose**

The microcontroller is undoubtedly the brains of the entire 3D printer, it processes the software, interfaces with a computer or SD card, and regulates motor control and temperature. Almost any microcontroller can be used with a 3D printer, typically only 8Mhz is needed since all the information is calculated real-time and a small amount of flash memory to hold the software. The microcontroller requires the aid of other electronics to fully control a 3D printer, namely a motor shield. The motor shield is connected to the microcontrollers output and provides the voltage necessary and control parameters to control a stepper motor for example. Almost any mainstream microcontroller can be used including hobbyist development boards such as an Arduino to a full-fledged x86 processor. Since 3D printers do not host an operating system or run any sort of image processing that speed of the microcontroller can remain low speed and low power allowing for a wide variety of budget controllers to choose from.

When buying or building a 3D printer the first thing one would want to decide on is the microcontroller as it’ll give a basis of the extent of work and productivity the 3D printer will achieve, the two options are cost versus capability. Questions such as, will the 3D printer have an LCD screen? Will the 3D printer run an operating system in the background? Must be answered prior to purchasing the microcontroller. Plenty of choices are available on the market, to buy or to create. The most common and accessible 3D printer microcontroller is the Arduino Mega, it is user friendly, cheap, and widely documented. Other choices such as a Freescale Semiconductor FRDM-KL25Z is an ARM microcontroller with a huge amount of inputs and outputs have never before seen features on a development board to implement innovative and creative features to a 3D printer, however the KL25Z is not well widely used by the 3D printing community and would require extensive knowledge of ARM programming and electronics to use one effectively. That raises the question of how much work one is willing to put into building a 3D printer, a smart engineer might be able to get away with building one with an undocumented microcontroller but the general public would require extensive step by step guidelines and user friendly software to build one which makes the Arduino Mega such a popular choice. Another option is implementing Arduino chip, the ATmega2560, onto a custom PCB; this requires basic knowledge of soldering and electronics and would satisfy the requirements of completely building a do it yourself 3D printer. In the next section the different types of microcontrollers that can be used will be outlines and their advantages and disadvantages will be analyzed before making a final decision on the controller that will be used for the Pegasus Producer 3D.

**3.3.1Advantages and Disadvantages**

As mentioned previously, when building a 3D printer the first thing one research is the microcontroller since it dictates the capability of the 3D printer. This section will review some of the widely available microcontrollers and the advantages and disadvantages associated with it. Although speed and power are not a great concern, outputs, PWM control, power outputs, and RAM are huge factors that can make a microcontroller obsolete and unusable with 3D printing technology.

To begin with, the Arduino Mega is undoubtedly the most widely used microcontroller, specifically for the RepRap Pololu Shield that is compatible with it. The Pololu shield fits directly on top the Arduino Mega for easy connection and includes MOSFETS for heaters and fans and terminals to connect a power supply. Not only have that bought since the Arduino is doing all the processing the shield that easily connects to the Arduino has the ability to add SD cards or a control panel for computer interfacing. The Arduino Mega plus the Pololu shield is one of the cheapest and most accessible options there are which makes it a popular choice for 3D printing. Moving on, with the popularity of 3D printing there now exists

entire controller board with the controller, motor drivers, and power supply sockets all in one board. It is a much simpler solution than the Arduino plus shield but really limits the chance to add personal innovation to a 3D printer. The all in one board us known as RUMBA and still contains the Atmega2560 processor. Something like the RUMBA is what we will be shooting to design from the ground up with custom PCB allowing for simplicity, small form factor, and the versatility of the ATmega processor. Moving away from AVR controllers we begin entering the realm of ARM controllers which have the possibility to also allow for the creation of CNC machines and laser cutters. One such ARM controller that is widely used is the Smoothie Board, powered by the powerful ARM Cortex M3 that completely blows away any of the AVR controllers and even allows Ethernet connectivity.

All the mentioned microcontrollers have their advantages and disadvantages and actually all cost around the same. All have had major firmware used for 3D printed applied to the boards and all include the ability to include an SD card and computer interface. The availability of all these controllers is questionable as they are constantly updated and made obsolete within a year or two. As mentioned before, the decision of the microcontroller depends heavily on the applications the 3D printer will utilize, using a well-documented microcontroller will allow for great future innovation since we would not be stuck in designing initial software, wiring, and have the peace of mind that the microcontroller is compatible with 3D printing technology. A quick comparison of some development boards that can support a 3D printer can be seen in table 3.1.1-1.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Board** | **Arduino Uno** | **Arduino Mega** | **Raspberry Pi** | **BeagleBone** | **FRDM-KL25Z**  **Freescale Freedom** |
| **Price** | $29.99 | $39.99 | $39.99 | $89.99 | $12.95 |
| **Processor** | ATmega328 | ATmega2560 | STMicro 32-bit Cortex-M3 | TI AM3358 ARM Cortex A8 | MKL25Z128VLK4 |
| **Processor Speed** | 16MHz | 16MHz | 700MHz | 720MHz | 48MHz |
| **Analog Pins** | 6 | 16 | 22 | 66 | 16 |
| **Digital Pins** | 14 (6PWM) | 54 | 8 | 66 | 54 |
| **Memory** | SRAM 2KB EEPROM 1KB | FLASH 256KB SRAM 8KB EEPROM 4KB | RAM 512MB | RAM 256 MB | RAM16KB  FLASH 128KB |

**Table 3.3.1-1:** *Comparison Development Boards that can Support a 3D Printer*

**3.3.2 Microcontroller Decision**

After much research and discussion the Pegasus Producer team has chosen to go with a custom all-in-one PCB featuring an ATmega 2560 microcontroller and all the necessary inputs for power and outputs for motor control. The board will be custom made and all the components will be soldered, I believe to get the most out of senior design we will have to design our own microcontroller board rather than using a development board, it will be a great learning experience and we have a lot of reference material to fall back on. The controller that will be designed shall be future proof to allow for an addition of a small LCD screen, SD card slot, USB, connectivity to a computer, and if time allows, some form of wireless connectivity.

The ATmega 2560 will have enough speed to control all the motors and it is widely documented that the processor is fully capable of running a 3D printer. Consideration for building the custom PCB include sending the verified EagleCAD design to a third party to manufacture or using copper boards to etch the circuits and add the components by drilling a hole. The latter method may result in a very large board since it is single layered and would not look very professional. Manufacturing the board after the PCB design would perhaps be the most crucial part, following that I would be having correct skills to solder all the components on the board.

A lot can go wrong at that phase so I will order a multiple boards and practice surface mount and IC soldering on junk boards prior to attempting anything. Following that we will begin the debug phase and confirm that all the components are working correctly and that motor control is adequate and up to design specifications. Time management will be very important when it comes to the microcontroller, a mistake with the PCB design towards the end of the project could have catastrophic results. Choosing the well documented ATmega 2560 chip was mainly for security purposes in case something goes wrong because errors and troubleshooting should be widely available on the internet and by other users.

Other close consideration included Texas Instrument’s Beagleboard, the well-known microcontroller is capable of 1GHz speed which allows the use of a full-fledged OS such as Linux and it also includes smaller AVR that can run the 3D printer. The Beagleboard has all the outputs required and is a very competitive choice. The reason the Beagleboard was not chosen is due to its overkill in processor speed, running Linux on the background would add a lot of functionality to any 3D printer but would also complicate the design beyond our level. ARM programming can also be tricky and will be a skill that will have to be learned prior to beginning a design using the Beagleboard.

**3.4 Control Board**

It is well known that an integrated circuit such as at ATmega2560 cannot run motors by itself, it simply cannot provide enough power and it will need extra hardware components to be able to properly control the movement of a motor. Luckily, hardware exists that can be added onto well documented chips like the ATmega2560 to control such motors, such as the RepRap Arduino Mega Pololu Shield (RAMPS), Sanguinololu board, and the R2C2 board. All mentioned bored essentially do the same function such as control the motors, verify temperature, regulate power, and accept expansions such as SD card slots. All these boards however differ in the microcontroller they are wired to use for example, RAMPS is best utilized when there is an AVR ATmega processor as a CPU and so does the Sanguinololu board, the R2C2 goes a level beyond and can utilize a 100MHz 32bits ARM processor with 2GB of internal memory which is often too powerful for everyday use. In addition to the difference in microcontrollers there is a difference in price and the ability to modify the reference design, the RAMPS can be built by creating a custom PCB and soldering all the components yourself or you can buy a premade board with the components on or off for under $20, a higher end board would not have such flexibility.

The RepRap Arduino Mega Pololu Shield (RAMPS) is designed to handle all the electronics needed for a 3D printer in a space saving low cost board. As the name sates it uses the ATmega2560 chip as the brains and due to that has tons of room for expandability to add creative new idea to existing 3D printing technology. RAMPS features input for stepper motor drivers and extruder control, and perhaps most importantly the board is easy to repair or upgrade since all the components are general off the shelf electronics.

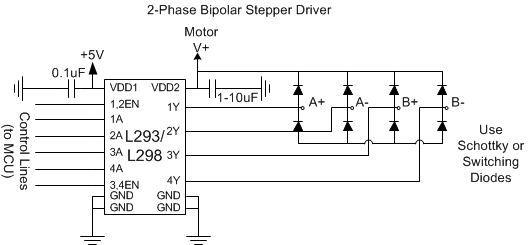
It is decided to use a similar design to RAMPS for the Pegasus Producer 3D’s own controller board. Some advantages for using RAMPS over other boards include all necessary requirements to run a Cartesian robot and extruder, expandable control functions, inputs for fans to properly cool the device, onboard fuses, interface for heat bed, ability to contain 5 stepper motor drivers, I2C and SPI debugging functionality, and ability to add LEDs as indicator for when control board is on. The entire controller board as well as motor drivers and motor themselves can be run off just a 12V 5A (not including power for microcontroller and heated )power in which makes the RAMPS board extremely versatile and an easy pick to integrate into this project.

**3.4.1 Motor Drivers**

No matter how versatile and expandable the RepRap Arduino Mega Pololu Shield is, it will never be able to directly drive the motors unless serious modifications are made. Thankfully it has to input to plug in motor drivers which will add perfect stepper motor control. The purpose of a motor driver is to keep the power that drives the motor away from the power needed by the microcontroller and other electronics. It is well known that most microcontroller simply do not have the power to control a stepper motor so motor drivers were introduced to act as a buffer that also controls how the motor spins. Other benefits of motor drivers is that they allow for the use of functional steps, on example is how stepper motors tend to vibrate at certain RPMs but such vibrations are reduced with the use of fractional steps.

There are 3 different wire configurations in which a stepper motor can be set up. The unipolar configuration is the most common configuration and uses two main coils per phase with each coil containing a center tap within the coils, using such a configuration would allow for the use of only five wires and easier design. Other configurations include a bipolar stepper motor, shown in figure 3.4.1-1, is similar to unipolar but lacks the center tap. With a bipolar motor the wires can be reduced to only four. Lastly there is the eight lead stepper motor which is in between unipolar and bipolar, what this configuration does it allow for high torque and high speed operation at a low current which is exactly what is desired for a 3D printer.

The question that arises is why can’t a stepper controller or controller board be used as a stepper driver? Typically a stepper motor requires a controller to create a 5V direction and step signal and a driver circuit to send the necessary current to the motor that will allow it to move, a motor controller simply lacks the required electronics to run such functions. A stepper motor runs off 3 wires that control “step”, “GND”, and “direction” witch carry the necessary motion information to run the stepper driver, a typical control board is almost purely digital and lacks the power to directly run a motor.



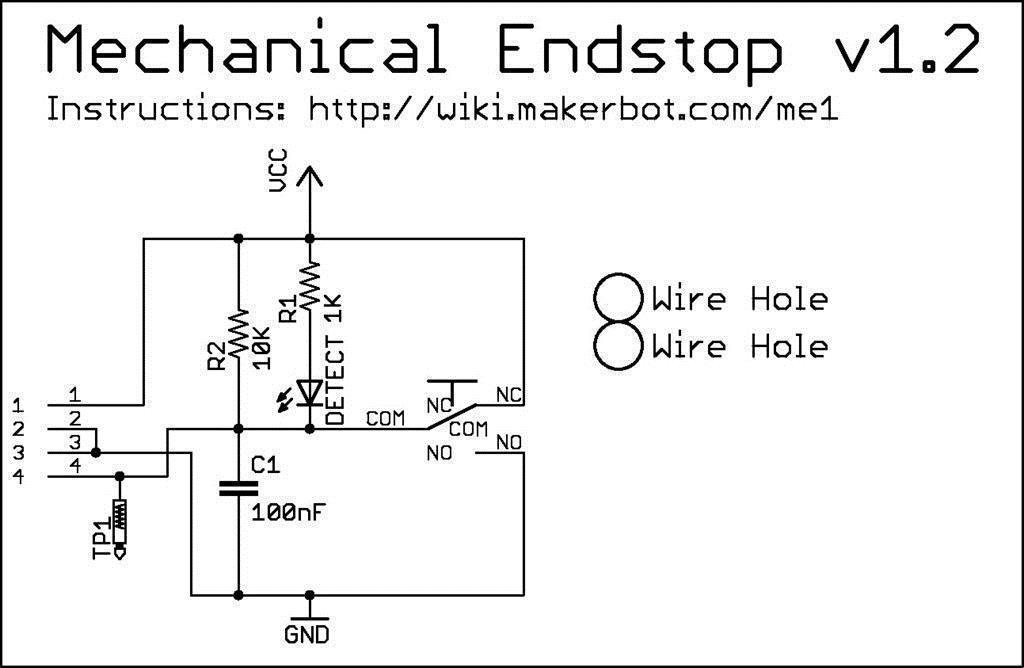
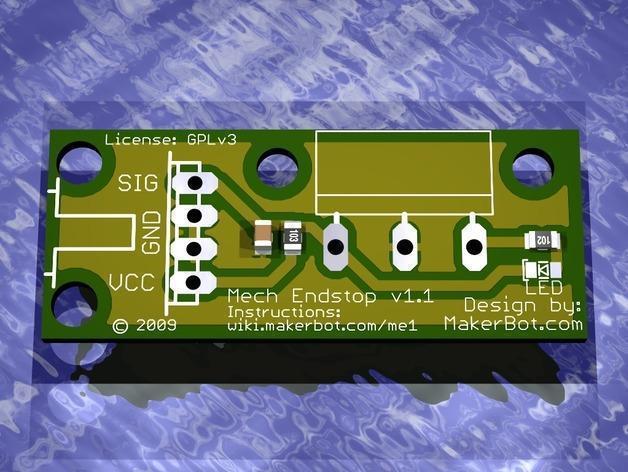
**Figure 3.4.1-1:** *2-Phase Bipolar Stepper Driver*

A stepper motor driver uses Pulse Width Modulation (PWM) to control a motor, typically a driver would chop up supply voltage using an integrated PWM chip. The main advantage about using PWM for the drivers is that they require very little circuitry to use and are relatively cheap, especially when a 3D printer uses about 5 motor drivers. A serious concern with PWM chips and motor drivers is how prone they are to heating up, therefore a thermal shutdown function will be added to every motor driver.

Finding and sourcing the PWM chips and components for a stepper driver that would work with RAMPS is rather difficult, most of the time they are bought pre made. Due to the difficulties in finding the correct components the team has opted to purchase the entire motor driver assembly, the design will be discussed further into this paper.

**3.4.2 End Stops**

End stops bridge the gap between electrical and mechanical functions in a 3D printer, in essence an end stop is just a mechanical switch that sends a signal to the microcontroller when it is pressed at the end of the X and Z axis. Mechanical end stops are much cheaper and practical to implement than an optical sensor because they require a small circuit, seen in figure 3.2.2-1, and just use 2 wires to connect the switch. End stops can be created by the use of 2 pieces of metal, a couple of pull down resistors, and some soldering which makes them one of the easiest systems to implement. End stops are essential for initial calibration, it is much easier to program end stop detection within the microcontroller rather than micromanaging within the firmware since the length of the axis is bound to become misaligned after use.

**Figure 3.2.2-1:** *Mechanical End Stop**(permission pending)*

**3.5 General type of motors**

The motors that it’s to be chosen for this project is one that would meet the specifications that are were implemented. These motors would drive the X, Y and Z coordinates of the modeled 3D printer. There are certain factors that must be taken into consideration when making a selection for the design and should conform to the following:

* Have freedom of movement with no restrictions
* Affordable
* Have the ability to be synchronized
* Take Direct Current as it’s input
* Controllable high speed revolution with torque
* Pulse width modulation speed.

The functionality of the motor would be applied to the project, the device has the ability to convert the electrical power that is fed on its input and convert it to mechanical power. By the use of the induction and a rotor system the motor can make the power conversion. There are the permanent magnet, printed armature, brushed switched reluctance and the coreless motors. The use of commentator is used most of the time to reverse the current direction that is created between the rotor and the circuit. The brushless motors are the ones that are known to have the three phases that use no commutator. The best feature about the stepper motor is the ability to rotate and also hold a position based on the impulse being received by the microcontroller provided by the ATmega 2560. Some of the essential features provided by the different kinds of motors are the ones that have to be taken into account when selecting some of these motors. Ideally, in theory a synchronous motor is a one whose electrical power aligns with the mechanical power. In order to understand the functionality that the stepper motors would play in Pegasus Producer the following equation must be show to understand the relation between mechanical power:

*Pmech= (equation 3.5-1)*

In this equation, the energy represented by k is considered to be kenetic, with that being said this allows for the use of the density factor when it comes to calculating the energy that’s being consumed. And two, kinetic energy can be expressed mostly as the magnitude of tongue. The reason that the above expression was mentioned is because the data that is utilized when evaluating motors is typically give the speed constant Kv and also the revolution per minute, along with the torque constant (Kt). These are the valued that are mostly found when considering certain specification in the building of the 3D printer. Some designed require the stall torque for their design but unfortunately that has to be manually calculated. In most motor design the power loss of the back EMF is to be considered the one force that must be compensated in order to allow the motor to spin. Back EMF depends solely on the construction of the motors in terms of its number of poles, the windings and the flux on each stator along with the velocity how fast the rotor is spinning.

The way the motors interact with the rotors and the copper plane it’s the main functionality of the brush motors. This plane is very much the storage element that distributes the energy inside. The flow of power is controlled by the flow of power, instead of the signal. In many cases this wouldn’t be a good candidate because the commutator, due to the friction it can usually cause deceleration in the speed of the motor. This feature wouldn’t be all the beneficial because the extruder and the X, Y and Z movement are not very heavy which means that the additional torque wouldn’t be all that necessary. The big issue with brush motors is that if we were to create the design for long term use, or perhaps for constant operation the brushed would have to be realigned in order to have proper alignment of the magnetic field. On the contrary if the brushless motors were to be utilized they have a permanent magnet system that is integrated with the rotor itself, these particular motors have no commutation plane, as a result they can handle larger amount of heat dispersed. The functionality results in less friction, permitting for a longer length of life and a more efficient faster rpm, these motors usually range typically from 10 to 22V depending on the specification and size of the motor.

**3.5.1 Direct Current Motors**

The functionality of the direct current motor is very useful for this project. Most of circuitry involved with precision designs is one that utilizes direct current. The connectivity for the DC motors are very simple to implement, it’s basically tow wires: One end that supplies a positive voltage and the second end that provides connectivity to ground. Most of these motors in the market are used in electronics, ranging in size, ratings, torque and manufacturing design. They can be found with both brushes and brushless.

**3.5.1.1 Motors with Brushless Direct Current**

The lack of brushes design is very much different of that of the brushed motors, where the permanent magnet is resides on the rotor and the windings tend to be on the stator. The feature that reverses this functionality allows for the motor to be brushless and in return it removes a whole lot of disadvantages associated with brush direct currents.

The brushless motor can typically be built with a system of having multiple phase windings and can be made with a single winding that is to be wrapped around the stator core in such a fashion that the interaction between the stator and the rotor’s magnetic field can cause a rotational action. The number of windings, along with the signal input is the main reason for the torque response. For instance if the motors had a lower amount of winding and the input coming from the Ramp 1.4 is supplied at a low current and so these combinations would cause the lower amount of torque.

There are many associated advantages with this type of motor design which includes but are not limited to less electromagnetic interference, the reduction of noise, provides more power, and has a higher efficiency rating. The majority of the efficiency comes from the ability to reduce the loss of energy due to the thermal dissipation the there are no brushes that would create the frictional temperature raise in order to dissipate the energy. That would reduce the cost in components by eliminating any need of additional housing components for the motors to dissipate the power losses.

Now that all the advantages have been pointed out as to why brushless motors would be the ideal selection, now the attention will be drifted to the disadvantages in which cost plays a big factor, the complexity associated with the manufacturing of these motors are passed down to the consumers, the placement of the permanent magnets on the rotor along with the placement of the windings on the stator are the big technological features that jack the prices on these motors.

**3.5.1.2 Motors with Brushes Direct Current**

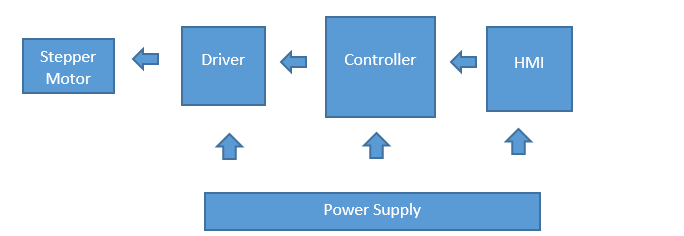
The implementation of direct current motors have been one of the earliest motor designs, the main concept of these motors have been around since the early resistance of the commutated electric motors. These motors function using a coiled wire that stretches around the armature that contains the brushes. The brushes conduct current between the wires in the armature in order to achieve a rotation. When the current is supplied to the coil, the magnetic field around the current carrying conductor (right hand rule) wraps around the conductor causing the armature to rotate by attracting the opposite magnet in the armature. By controlling the commutate signal at appropriate intervals, the armature will continue to steadily rotate until an increase in current is applied causing the magnetic field to expand as a result cause the motor to spin at a higher revolution per minute (rpm). The current could also be reversed so that it would allow the motor to spin opposite. This is an important feature because it allows for flexibility in the design. There are actually some great advantages to utilizing a brushed direct current motor. Some advantages included cheaper cost of manufacturing since the technology has been around for some time and are pretty inexpensive. The motors can be found in different sizes and wide range of voltage ratings to fit the project goals implemented on the design. Brushed direct current motors are mostly used in the servo design because of the proportionality of the amount of current and voltage applied to the motor and it also functions with width modulation that increases precision.

The biggest disadvantage that is associated with using brushed motors is its inability to fully convert the energy from electrical to mechanical. The majority of the lost occurs in the form of thermal noise and this occurs due to the motion of the armature that is spinning inside of the housing of the motor, hence why the housing of this motor is never sealed allowing for the heat to have escape. However, there is a fine line between how much exposure is to be given to the motor because this could also lead to external damages in designing the Pegasus Producer. Another valuable disadvantage when considering this motor is that brushes eventually wear out, from all the friction that is created between the brushes and the coils. All these things need to be accounted for during the selection of the motors because they could potential run extra expenses on the 3D printer and incurring this maintenance over time is a major drawback.

**3.5.2 Stepper Motors**

So far stepper motor is the best potential for the Pegasus Producer. The functionality of a stepper motor has really impacted the market; it has been a great innovative technological advance that has facilitated many breakthroughs on the electrical engineering designs. The stepper motors are used widely across the market ever since its creation. This interesting design initiated from the idea of the open loop control in linear controls options that is more cost affordable than the servo motors. In figure 3.5.2-1, it can be observed how the stepper motors would be implemented into the whole project.

The basic idea behind a stepper motor is it’s functionality of using salient poles along with teeth for degree step, usually around 1.8 degree step. The stator carries the electromagnet while the rotor consists of a permanent magnet. The majority of the stepper motors include the same idea of utilizing a two-phase that is being driven by the quadrature phase input. The movement of the stepper can range from many small increment steps to a full continuous rotation, this is achieve when the poles are powered up one at a time, like a chain effect. When one pole is depowered, the subsequent pole to its front or back is powered and this process is done repetitively by causing the magnet to line up with the next energized pole.

**Figure 3.5.2-1:** *Block diagram for a Stepper Motor*

Rule of thumb is that by increasing the number of poles would allow the rotor to move in sync with the stator. For the Pegasus Producer a stepper motor that provides a rotation of up to 1.8 per step is sufficient for the precision on the extruder. The drawbacks is that due to the lack of feedback in the stepper motor design, the alignment of the rotor and stator could synchronize and allow the stator to run some power through the stator even though the motor is not moving.

Advantages

* No feedback loop
* No mechanical damage from heavy load
* High reliability
* Low Cost
* High torque at low speed

Disadvantages

* Low Frequency
* Great deal of power irrespective of the load
* Torque falls rapidly with velocity
* Undesirable for high speed applications
* Missed steps
* Low torque-to-inertia ration
* High heat/Performance
* Low power output
* Noisy operation
* Large errors and oscillations

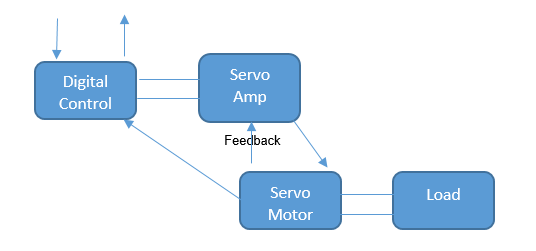
Some of the disadvantages mentioned above don’t really apply to the Pegasus Producers, since the design specifications for the 3D printer doesn’t really require high speed application on the contrary; precision is desired over speed. The damping factor in the stepper motors is very low and that results on being under damped when controlling the main functionality of the motor. This can make stabilizing the more much more difficult when compare to the other motors. After all it is an open loop functionality which implies its inability to be feedback controlled. *Figure 3.5.1-1* shows an illustration on the overall connectivity

**3.5.3 Servo Motor**

Unlike the stepper motor, a servo motors is composed of three wires that carry AC/DC, a potentiometer, an output shaft bearing and an integrated circuit. The servo is available in both AC and DC input. The functionality of a servo can be utilized in many forms. Servo can be found in applications that are radio controlled. This wouldn’t apply to the Pegasus Producers since it doesn’t include any radio controlled equipment on the schematics.

Looking closely what sticks out of the motor casing is one for the power, one for the ground, and one is a controlling input line. The neat thing about servo motors is that the actual shaft can be positioned to specific angular positions by sending a small coded signal. As long as the input line for the angular position isn’t touch then the motor will maintain its angle position on the shaft.

The setup for a servo motor on the Pegasus would be fed from a DC power supply along with other components. The servo operates with the gear reduction unit, which tends to be a circuit of some sort, and then mounted a surface that would angle it to a position vector. These three components work in collaboration to allow the motor to receive constant feedback in order to operate efficiently, the best part about the feedback is that it’s constantly making effort with the necessary corrections to stabilize the system. The corrections could either by increasing or decreasing the speed of the motor which would be the X and Y movement for the platform, and the Z movement of the extruder. In the case of the servo the angle is controlled by pulses that are sent in order to modulate. The servo is programmed to expect a pulsation every for every 20ms intervals. Figure 3.5.3-1 shows how the servo motor works using a feedback.



**Figure 3.5.3-1:** *Block diagram for a Stepper Motor*

A great feature about the Servo motors is that they will attempt to hold their position even against an external force applied to it. The maximum of external force amount that the Servo motor can take is rated for the maximum torque. A major disadvantage to a servo motor would include the need to utilize a mechanical stop feature integrated, which it is there to limit the shaft of the motor from possibly rotating more than 180°. The position sensor is the one that determines the degree at which the servo motor’s shaft moves. If by discrepancy should the angle of the shaft be different from what the circuit states, than the servo motor would find ways to adjust. The three wires attached to a servo motor are:

* Red: Power Supply
* Black: Ground
* White: Control wire

The white wire is connected back to a feedback controller such as a balancing PID controller, which allows for the adjustment of the revolutions per minute, with proper calibration this would produce accurate results. Unlike the stepper motor; the servo cannot operate on an open loop control, if the white pin weren’t to be connected then the motor will not operate. An average servo motor requires roughly between 4 to 6 volts and can maximize their torque for their size. Servo motors tend to stay cool when in constant operation, the heat dissipate among the motors are much less compared to those others. The greatest disadvantage with the Servo motor would be the behavior if the circuit were to break, it would continue to spin until stopped.

**3.6 Printing Materials**

Just like 3D printing technology, the materials used for 3D printing have come a long way. The spectrum for 3D printing material has increased tremendously including powder, pellets, granules, resin, and of course filament. When materials require special specifications to use there is often only one platform that can use it which is a major drawback, fortunately extruders are becoming more versatile and are able to print more than just one materials. Materials such as Ceramics have recently joined the group of materials that can be 3D printed, the problem with brining new materials into 3D printing is the accommodation which has to be given to each material, for example, ceramic would still need firing and glazing which is a process any ceramic part has to go through for product; a 3D printer would have to accommodate for that.

Plastic has been material that most people think when 3D printing is mentioned. Typically for plastic, Nylon or Polyamide is used in filament form which is broken down using the FDM process. Nylon or Polyamide is used due to its strong, flexible and durable properties that has proved invaluable for 3D printing. Continuing with plastic, ABS is another commonly used plastic that fits the criteria for 3D printers, especially for home printers. The color carrying ABS plastic is strong, cheap, and non-proprietary which is what makes it the number one choice for 3D printers. Moving on, PLA is a bio-degradable plastic material that is becoming popular due to its “green” background. PLA is versatile because it can be used in resin format or as filament and is one of the only materials that can print in transparent color. Unfortunately for PLA it is not very durable and not as flexible as ABS.

Metals have been widely used for industrial 3D printing and normally consist of cobalt or aluminum. There is no argument that metal would be the strongest 3D printing material, stainless steel is a commonly printed material that starts off as powder form and through the use of different process gets molded into a design. Metals such as gold and silver have been added to the 3D printing catalog, such metals have a lot of applications for the jewelry market. Industrial 3D printers are capable of using titanium, the strongest metal any 3D printer can print. It is only a matter of time before smaller, home 3D printers will have the same capabilities. Interesting printing materials such as paper are also becoming more popular. Mcor technologies developed a method to bring cheap, easily obtainable printing material that can be bought anywhere. By introducing paper as a printable medium, Mcor technology has also ushered “green printing” since what is printed is also environmentally friendly and easily recyclable.

Perhaps the material that is being researched the most is bio materials, 3D printing bio materials would allow for organs, skin, and muscle tissue to simply be printed out. The possible implications of bio materials for 3D printing are tremendous. If muscle tissue can be printed so can foods such as meat which can be used to feed humans and livestock. Extruders to process food and bio materials has been experimented with a lot for the last couple of years. A 3D printer with the capabilities to print food would revolutionize the appliances everyone keeps in the kitchen.

**3.6.1 Plastic**

When it comes to 3D Printing plastic ABS and PLA are the most popular and most versatile options there are. Acrylonitrile Butadiene Styrene or ABS is a lightweight thermoplastic that is designed to be injection molded and extruded. Overall, ABS needs less force to extrude when compared to PLA due to its lower coefficient of friction as well as being a lot less brittle. ABS is ideal for printing small parts due to its easier extrusion process but requires a temperature of 105 degrees Celsius to be extruded. When plastic is being melted fumes tend to be released, ABS created tolerable fumes which can still be harmful if inhaled for a long time by people or pets. ABS has a tendency to warp when printing large parts, printing ABS often requires a heating platform to compensate for the warping. Typically ABS can be found rather easily and usually sells for $20/pound which is a lot of material for a home 3D printer.

PLA on the other hand is known as Polyactic acid and is a bio-degradable plastic polymer that is manufactured from lactic acid which are fermented from organic materials. Due to its organ properties, PLA can be manufactured almost anywhere including poverty stricken areas and this world countries. PLA is harder and ABS and also melts at a temperature of 200 degrees Celsius. Unlike ABS, PLA is more often to cause an extruder jam since it has a high friction coefficient. PLA is a tremendously good printing material for small, home 3D printers which is being built. PLA is dimensionally stable which removes the need for a heated bed and is inexpensive and easily bought. Overall, one can get a better quality surface finish using ABS but at the cost of PLA better mechanical properties offering a stronger, studier 3D product. The increased friction from PLA can be compensated by its lower viscosity when molten which causes there to be less pressure in the nozzle or melt chamber and a lower driving force when extruded. A major drawback of PLA is that it can absorb moisture from the air which can turn into steam bubbles when it is being extruded. The main cause of the unwanted bubbles can be compensated by using a special extruder head made to work with PLA. To stop PLA from absorbing moisture from the air tricks such as microwaving the plastic or leaving in the oven have been tried with some success, a spool of PLA plastic can be seen in figure 3.6.1-1.



**Figure 3.6.1-1:** *Spool of ABS plastic*

Other plastics that are starting to gain traction are polycarbonates, it is nowhere as popular as ABD or PLA but it offers interesting capabilities. Polycarbonate, much like its rivals, is a strong thermoplastic with somewhat high melting temperature. It differs from PLA due to its quick transition temperature which allows for the lower temperature on the hot end which counteracts its high melting temperature but causes for slow melting. Polycarbonate is useful when paired with other plastics in a dual extruder due to its fast transition temperature. At its current manufacturing process, polycarbonate being extruded can release some harmful fumes especially with high temperatures so special consideration on the extruder must be taken. Polycarbonate is slowly being accepted as a 3D printing material but more research and a wider range of extruders must be implemented before polycarbonate filaments can take off.

**3.6.2 Metals**

3D printing metals has been a big topic in recent years, huge companies such as SpaceX and BAE systems have just started implementing 3D printed metal parts into their aircrafts. Just merely thinking about how printed metal can revolutionize any industry is mind boggling. 3D printers that are capable of printing metal are priced accordingly, 3D systems, a big player in 3D printing has introduced a method known as Direct Metal Printing or DMP. DMP allows industrial 3D printers to create concept models that are precise and can be used as functional prototypes, such 3D printers range in the hundreds of thousands of dollars. As the technology progresses metal 3D printing continues to come down, an Australian Kickstarter recently unveiled a $4000 3D printer capable of printing metal with a thickness at about 50 microns.

A drawback with metal printing is how the filament is supplied, rather than plastic coiled up, metal for 3D printers comes in powder form and is usually very expensive. Typically the metal 3D printing powder allows for a high packing density while still allow for good flow properties. There is hope that future technologies will bring the cost of metal 3D printers down so a general public could own a household metal printer. Two-Photon laser curing is a method being researched to do such a thing, the two photon laser essentially focuses on the material to cure the metal while still leaving the end product very dense. Two photon laser curing is similar to how a piece of glass is etched in the center. Two photon laser and other metal printing techniques such as femtosecond lasers are still a decade or so away but it is good to know that metal 3D printing is being worked on. If it wasn’t for the high cost associated with building a 3D printer as well as the high cost for the powder material a metal 3D printer would have been our first choice for the senior design project.

**3.6.3 Organic**

The prospect of organic 3D printing is very real and just across the horizon. University of Pennsylvania created a technique for printing blood vessels using a sugar casting from which live cells can grow around. Food is another area that is currently being experimented on with food such as chocolate and sugar being the product. 3D printing material in the organic are is under heavy research and currently in the experimental stages, it will be years before something is commercially available so information on food or organic 3D printing is very limited. In terms of the senior design project, organic or food 3D printing is completely out of the scope.

**3.6.4 Printing Material Decision**

The use of our printing material really comes down to one choice, plastic. From plastic we have another two choices, ABD and PLA. Both types of plastic hold advantages and disadvantages and both will be covered researched extensively prior to making a choice of printing material. I have mentioned almost all the materials being researched for 3D printing but ABS and PLA are overwhelmingly used for home 3D printers. ABS and PLA are both thermoplastics which means they are able to be shaped when hot but hard and sturdy when cooled and this can occur multiple times. Thermoplastics is also the plastic used most currently for sort of applications such as packaging and casting.

For a material to be practical for 3D printing it has to be able to be extruded as plastic filament, must be able to be trace-binded during the printing process, and have an end application. By meeting all 3D printing material requirements. the plastic will be able to go from a raw material and fed into an extruder without problem to form an accurate representation of the digital model being printed and also meet the requirements in strength and durability for a long lasting print. Many plastic are able to pass one test but fail at others such as polycarbonate, a widely used plastic for other applications but fails in 3D printing due to its temperature resistance. There is a lot of science and engineering behind the production of 3D printable plastic and ABS and PLA currently hold the top two spots. Before choosing one of the two most popular thermoplastics for the 3D printer, each plastic will be tested for storage, smell, accuracy, and chemical properties.

Many people recall having to throw out ink cartridges for a regular printer because they eventually dry out and become useless, a 3D printing plastic that can be stored for a long time is big factor for choosing a printing material. Due to plastic properties it is recommended to seal off ABD and PLA from the atmosphere for long term storage since humid environments can have a negative effect on the plastic. When exposed to moisture ABS will bubble and trickle from the nozzle when printing as the air bubbles inside the filaments pop when heated up. Such an issue will cause defects on the part being printed and clog the entire extruder if the humidity the ABS was exposed to be severe. In Florida, humidity can be an issue and can cause the above problems to prints but a solution is to dry the plastic with a source of heat that won’t melt the plastic. PLA on the other hand responds differently to moisture, not only does it bubble and spurt from the nozzle but it also changes color. When PLA is in a humid environment and then heated up by the extruder then entire plastic undergoes de-polymerization. Unlike ABS, PLA can also be heated to remove the moisture but at the cost of altering its melting temperature and chemical properties however, a 3D printer with a variable temperature on the extruder can overcome such problems. Overall when it comes to storage, ABS is marginally the better choice.

When being heated up all plastic release a certain smell, some are highly toxic and even considered carcinogenic. For home use a 3D printing material must be absolutely safe to inhale in small quantities. The smell of melted plastics greatly depend on the manufacturing process and the temperature it is melting at. Two of the same plastics can smell differently when melting and actually an outlandish way to measure the quality of the plastic. ABS is free from all contaminants and only made of pure plastic material, like any plastic it always releases a smell. The smell from melting ABS plastic is almost unnoticeable and varies from person to person. A well ventilated room and a good extruder design will prevent almost any smell from the melting ABS plastic. Moving on, PLA is somewhat organic because it is a byproduct of sugar. Due to its sugar properties, when melting PLA gives off a sweet smell which is often considered a benefit since it is better than smelling hot, melting plastic.

The biggest factor when it comes to choosing a 3D printing material is the accuracy with which the material can print in. No or too little accuracy has no value when it comes to 3D printing because the goal is to print exactly what was designed on a computer and any inaccuracies will defeat the purpose of 3D printing and prototyping. It is well noted that both ABS and PLA can print accurately but both thermoplastics have advantages and disadvantages.

As mentioned before, ABS plastic requires a heating platform as the bottom surface of the 3D printer otherwise the ABD will begin to curl upwards causing major loss of accuracy. Adding extra chemicals to stop the upwards curling from happening is a common solution, many users add acetone or a small dash of an aerosol. ABS has trouble configuring to sharp corners found when printing out a perfect cube and will often print slightly rounded. To aid in the printing of sharp corners a fan is introduced for some cooling around the nozzle which can also lead to detrimental defect such as cracking if too much cooling is used. ABS require a careful balance of this party heating, chemicals, and cooling to print accurately. PLA proves to be much more resistant to warping than PLA. Firstly, PLA does not require a heat bed on small parts which reduces costs of the entire 3D printer, a heat bed may still be required when printing large parts. PLA has the advantages within its thermal properties because when heated it is a liquid with low viscosity. With some cooling from a fan, very smooth sharp corners can be made and there is less risk for cracking or warping. More fluid plastic not only reduces the chance of the extruder clogging but also strengthens the binding layers leading to an overall studier print. In terms of accuracy both ABD and PLA can achieve the same level off accuracy, however PLA requires much less work to do so.

Material properties play a big part in thermoplastics, a plastic that is hard to produce, melt, alter, or recycle would be undesirable and the same applies to 3D printing materials. ABS comes in many forms but in general it is durable plastic with some flexibility. Prior to any colorants ABS appears light white and its flexibility allows for the creation of interlocking design or separate prints that are supposed to connect together. ABS is soluble in Acetone and Acetone is often use to “glue” to pieces of ABS plastic together through dissolving. ABS can have its small printed defects that are always present by being dipped in acetone for a short time, this process can also be used to create a high gloss on the plastic itself. The recyclable thermoplastic ABS excels in strength and flexibility and can be used for real world applications due to its high heat resistance. PLA is composed from the byproduct of corn, sugar, and potatoes and is praised for how easy it is to manufacture and its eco-friendly production when compared to ABS which require oil to manufacture. In a mechanical sense, PLA is much more rigid than ABS so it is difficult to work with a final printed product. The final printed product made PLA plastic will have a glossier look but PLA altered to have different degrees of color and opacity. Overall PLA is much harder to work when it comes to sanding down and the lower melting temperature can cause PLA printed designs to melt and deform when exposed to high ambient temperatures.

In summary, ABS has good strength, it is flexible, can be altered with the use of chemicals, and has a high resistance to heat which makes it an ideal candidate for prototyping. ABS can be recycles but like most plastic it is made out of oil and has a similar burning plastic smell when going through the extrusion process To properly handle ABS printing a heat bed is required and a lot of chemical balancing is needed to prevent defect when printing out a design.

PLA on the other hand is much more household and novice 3D printing friendly due to its wide variety in colors and opacity. The plant based PLA plastic smells sweet when extruded and requires much less attention than ABS due to its more viscous state when being extruded. PLA has the upper hand in high accuracy with low warping and defects as well as printing speed. A comparison of PLA and ABS can be seen in table 3.6.4-1.

|  |  |  |
| --- | --- | --- |
|  | **ABS** | **PLA** |
| **Temperature to Extrude** | ~225C | ~180-200C |
| **Requirements to Print** | Needs heated bed and passive cooling | Heated bed optional |
| **Passive Cooling** | Cooling required for accurate prints | Benefits from cooling |
| **Possible Defects when Printing** | Prone to cracking and warping | Upward curling of corner |
| **Flexibility** | Very Flexible | Brittle |
| **Adhesiveness** | Can be bonded together with household solvents | Requires glue to bond |
| **Smell** | Melting plastic smell | Sweet “sugar" smell |
| **Manufacture Process** | Requires oil | Plant based |
| **Color** | Limited range of colors | Wide range of colors and opacity |
| **Availability** | Widely available | Widely available |
| **Packaging** | Coil or spool | Coil or spool |
| **Diameter (Filament)** | 1.75mm and 3mm | 1.75mm and 3mm |
| **Recycling** | Partially recyclable | Fully recyclable |
| **Price Per Kilogram**  **(Natural- No color)** | $18.96 | $19.19 |

**Table 3.6.4-1:** *ABS and PLA Comparison*

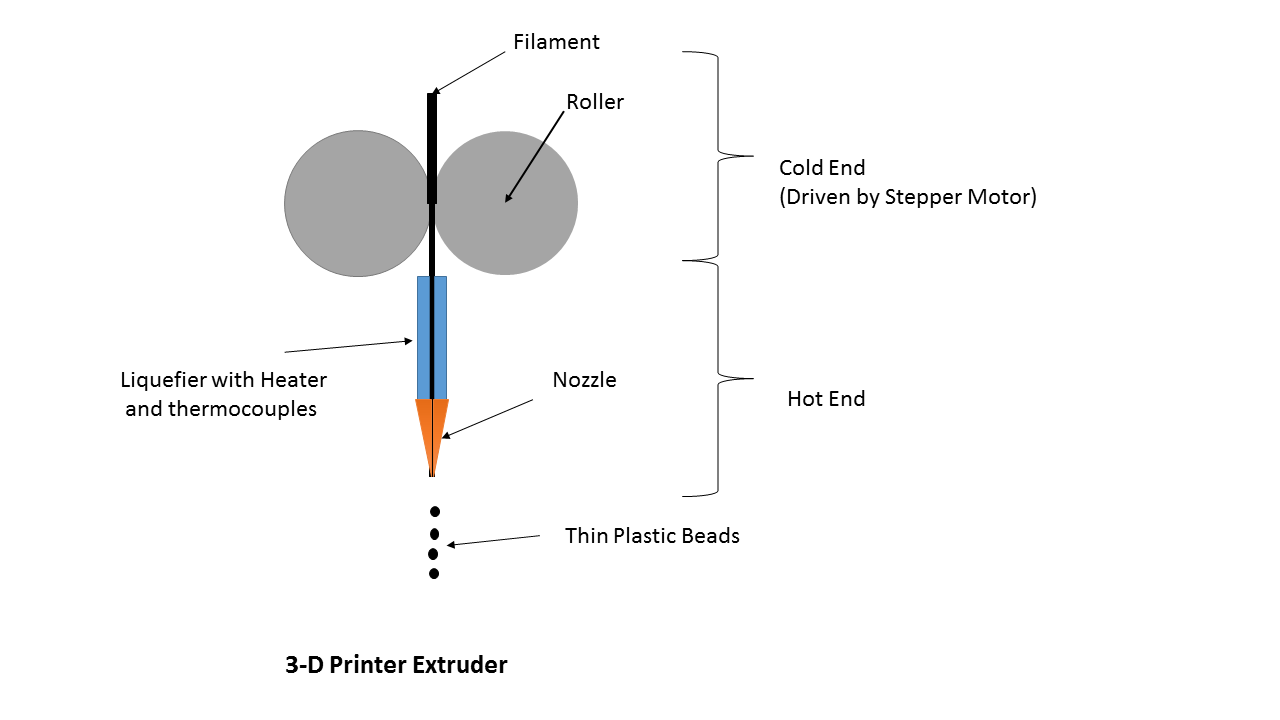
For a project to build a 3D printer it is decided to support both PLA and ABS, both plastics have advantages and disadvantages that make it very valuable. By passively supporting ABS and its need for a heating platform and active cooling it is also allowing for PLA to be used as a printing material. The major hurdles in supporting the two main types of thermoplastics is the temperature and extruder required to support both of them. The extruder will have to be designed to support ABS which needs a temperature of 225C to extrude, allowing ABS also brings the possibility of the plastic jamming inside so proper precaution have to be taken. The extra cost brought by the heating platform should be minimal but the real problems comes from adding another heating element that is not the hot end of the extruder, such additions require extra research and tougher design specifications. Lastly, by introducing ABS the 3D printer will require passive cooling which will benefit PLA. As an engineer, allowing ABS would bring tremendous benefits if it is ever decided to prototype with high quality prints for practical purposes other than household prints which PLA is best known for creating.

**3.7 Extruder**

The 3D printer extruder is the perhaps the most important part of the 3D printer and would be the equivalent of an engine for a car. The extruder is the what takes in the acrylic plastic and deposits melted plastic from a small metal nozzle that is heated it up. The acrylic plastic is pushed in to the hot nozzle by a motor and extruded out as melted plastic at a constant rate. There are many different types of extruders such as dual extruders and single extruders, every current 3D printer requires an extruder regardless of price. Most budget 3D printers use a single extruder that only extrudes one plastic at a time, however more high end 3D printers adopt dual extruder technology which is able to print more complex, and hollow prints that single extruders would have a hard time completing, one example is the 3D printer at UCF's innovation lab where the $50,000 3D printer was making a prosthetic arm for a little girl and required the second extruder to fill in the hollow area with chemical soluble plastic material.

The nozzle is important because it determines to what precision an object should be printed. The drawbacks is that the smaller the holes the more precise it is but also the longer it takes for the print to complete, a wide nozzle would result in the opposite. It’s rare to see a 3D printer with exchangeable nozzles because the 3D printer software would have to be calibrated to take into consideration the new size. The extruder is also the link to one of the X, Y, or Z axis. Depending on the 3D printer the extruder is usually attached to one of the axis and moves in that direction, one way to think about it is those arcade cranes where you put in money to try to win a prize, the extruder which in this case is the arm/hook is attached to the x axis and only moves left or right, the y axis moves the entire extruder and x axis moves forward or backward. To move the extruder on the x axis it would require some type of motor control, for compatibility reasons some extruders have a mounted motor specifically for that function, other extruders simply have the ability to attached to a belt that is moved by a motor at the end of one of the axis. There however always needs to be a motor on the extruder, this is because a rather powerful motor is require to push the plastic inside the hot filaments, the filaments then fills with melted plastic and is released via the tiny hole on the filament. If the plastic pushing motor is too weak it will seize and no plastic will be extruded, typically a motor spins a cog or wheel that pushed the plastic in at a steady rate, the motor would have to compensate for how much plastic it can push and how much plastic is extruded which is calibrated via the software and microcontroller.

The hot end works a lot like the solder iron and is heated the same way. The hot end lies just above the filament and is used to melt the plastic, the hot end is connected directly to the power supply to heat it up. The hot end needs to be heated up to a correct temperature that can melt the plastic but not destroy it and is usually regulated by the microcontroller. Like any device that uses a heating element the 3D printer has to heat up before it can began printing similar to how an oven has to pre heat to a certain temperature before it can begin to cook food. Other things found on the extruder is accommodation for placement on the axis, most 3D printers run on an oiled, thin, metal cylinder to provide low friction movement and stability. The entire assembly of an extruder can be seen in figure 3.9-1.



**Figure 3.7-1:** *Diagram of a 3D Printer Extruder*

**3.7.1 Nozzle Size and Precision**

As mentioned in the introduction paragraph the filament refers to the very precise diameter plastic that exits the extruder and is the technique most homemade and professional 3D printers use to build plastic models. The act of forcing melted material such as plastic, wax, or metal to be deposited alongside another material through heat or adhesion is known as Fused Filament Fabrication (FFF). A typical nozzle would be between 1mm and .3mm with .5mm being the most common configuration for users. Problems associated with filament size com in the form of die swelling and stretching. Die swelling and stretching occurs when the plastic deforms as it exits the hole and morphs to a size and final diameter that is larger or smaller than the hole where it was extruded from. Factors for die swelling include plastic used, temperature, and hole diameter. Different types of plastics used for printing have different properties that can prevent such printing deformation. An example of die swelling is if the printer heads moves to fast with a small filament diameter, the resulting melted plastic will be applied in a very thin way giving unwanted results.

Through research the 3D printing community has created principles to relate printing speed to filament size to create desired and quality prints. A filament size of .77mm will usually print twice as fast when compared to a .5mm filament but both will not nearly have as much detail as a .3mm filament. The limits for going into such small filaments can be game breaking for example, if you extrude slow but move the print head fast the filament will stretch and break beyond a certain threshold producing an undesirable result. On the other hand, a fast extruder whose head moves slow will cause the filament to bulge producing blobs and air bubbles. The 3D printing community has come up with a chart to fine tune the extruder filament size to prevent die swell and stretching that take into consideration material being used, filament size, and maximum and minimum ranges. A table showing the required nozzle diameters to accommodate the two major types of filaments can be seen in table 3.7.1-1.

|  |  |  |  |
| --- | --- | --- | --- |
| **Material** | **Nozzle Diameter** | **Minimum Range** | **Maximum Range** |
| [ABS](http://reprap.org/wiki/ABS) | 0.5mm | 0.3mm | 0.5mm |
| [ABS](http://reprap.org/wiki/ABS) | 0.3mm | 0.25mm | 0.4mm |
| [PLA](http://reprap.org/wiki/PLA) | 0.4mm | 0.3mm | 0.4mm (\*) |

**Table 3.7.1-1:** *Nozzle Size Requirement vs. Printer Material*

**3.7.2 Hot End**

The hot end is perhaps one of the most complex parts of a 3D printer, it takes into consideration temperature, and material properties of the plastic being melted and is also where all the melting and extruding happens. Generally the hot end is simple a metal case with a resistor that heats up and melts the plastic at around 200C and a thermistor which measures temperature. Interestingly enough, the micro-controller measures temperature and adjusts accordingly by altering the power supplied in the form of pulse width modulation.

Understanding a hot end requires some knowledge of thermoplastics and complex topics such as glass transition temperature which is when plastics go from a solid consistency to a rubbery form and expands as temperature rises. Following the glass transition temperature, the melting temperature is the next important factor to understand, this is when the plastic becomes a liquid and can be extruded. Finally, there is the critical transition phase and arguably the most critical part of the extrusion process. The critical transition phase lies between the glass transition temperature and the melting temperature in a rubbery/liquid state.

Not correctly mitigating the critical transition phase can cause problems because the plastic will expand and grip the inside of the hot end which will slow down the extrusion process and likely causing a jam. To avoid such problems precautions must be taken such as reducing the area that rubbery plastic can grip by reducing the transition zone, and reducing the friction between rubbery plastic within the interior walls of the hot end. A collection of various hot end is listed below in figure 3.9.1-1.



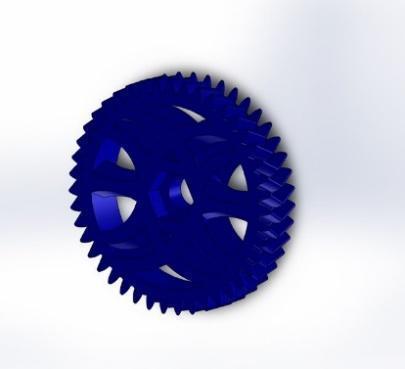
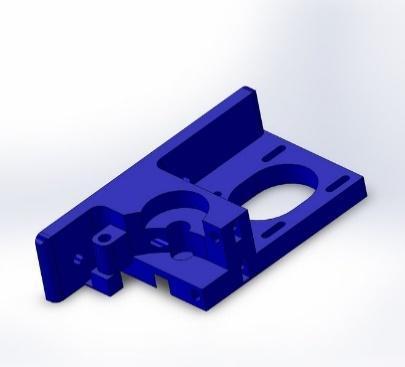
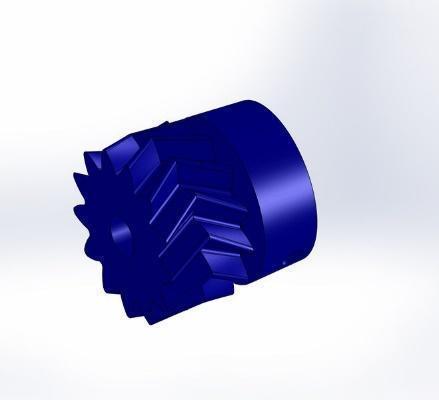
**Figure 3.7.2-1:** *A depiction of the variety nozzles used in plastic 3D printing**(permission from author)*

**3.7.3 Cold End**

Unlike the hot end the cold end is simply the upper half of the extrusion mechanism and mechanically feeds material to the hot end which melts it as explained above. A cold end would host the motor that would push the plastic inside and many different varieties exist since it’s the only part of the extruder that can be 3D printed by another printer. A cold end would typically consist of bearings, bolts, nuts and washers to hold everything in place including the motor and hot end which is extremely modifiable to allow for different motors and hot end attachments. As the bulk of the extruder, the cold end is often the carriage on one axis and supports the rest of the parts.

The cold end can be split up into two different parts; the part that is in charge of moving the filament and the part that is connected to the carriage. The motor that pushed the filament towards the hot end is known as the driver and it is usually a motor that rotates a hobbed, knurled, or toothed pinch wheel against a pressure plate or another bearing is the filament in-between is forced and pushed through. Due to the proximity to the hot end and a motor the cold end often needs passive cooling such as a heat sink or a fan blowing on it to keep it cool.

The most vital part of the cold end is its connection to the hot end, it requires an insulator tube to reliably pass filament from one side to the other and prevent as much heat transfer from the hot end as possible, the cold end is usually made of plastic and the improper insulation can cause catastrophic effects to the entire extruder. A popular choice for the insulator is the tried and tested Bowden cable remanufactured as a tube, its great mechanical properties and resistance to temperature makes it an ideal insulator for 3D printing although other materials such as Teflon is becoming popular. STL files for three major parts of a cold end can be seen in figure 3.9.2-1. The gear (left) is rotated by the stepper motor and pushes the plastic filament toward the hot end. Extruder Body (middle) is the support for the entire assembly and holds together the motor, hot end, and cold end. Small gear (right) is placed on the stepper motor itself and precisely turns the large gear to push plastic inside.

**Figure 3.7.3-1:** *STL Files for Three Major Parts of a Cold End (GNU Free License)*

**3.7.4 Single and Double Extruders**

Extruders are typically the most expensive part of the printer and adding a dual extruder can make a low end 3D printer double in cost, nevertheless the benefits of a dual extruder can outweigh the cost depending on the 3D Printers use. A common misconception is that a 3d Printer with dual extruders can print twice as fast, which is false. A dual extruder will speed up the process but only because of continuous printing, the real benefits comes from being able to use two different filaments of different color, consistency, and plastic. A typical scenario for a home printer with dual extruders is if one runs out of material in the middle of a long print, the 3D printer can simply switch which extruder head it is using and continue the print.

The biggest limitations of dual extruders is that the printing material needs to print from the same nozzle since each extruder is locked and cannot move independently. Until extruders can move independently the printing process will not speed up dramatically, the cost of implementing such technology would require some major rework of the established 3D Printer design.

The major advantage of dual extruders comes from the use of two different filaments, for example using low quality filament as structural support that can be discarded and high quality filament to build the actual model. In cases such as the UCF 3D printer in the Texas Instruments Innovation Lab, a dual extruder was printing soluble support filament that can be dipped in a solution to easily and precisely remove the low quality material and keep the rest, such a process was used to print a prosthetic arm and has countless applications.

Lastly, the last advantage of a dual extruder would simply be the ability to have a dual colored model, such a design would be very useful for prototyping and general aesthetics. Dual extruders that print multicolored cannot blend two materials to make a different color, such properties will not exist until the chemistry of the filament is engineered to allow for blending at a certain temperature.

For the typical home 3D printer a single extruder is ideal, however as the technology matures a dual extruder with limitless capabilities is not too far beyond the horizon. Just recently, a software engineer created a filament that is conductive allowing for circuits to be printed. A dual extruder that can print conductive material and plastic would lead to a revolution, imagine printing out a circuit board for a broken TV or for cheap prototyping. Providing that manufacturer’s print designs are available along with the pace of 3D printing technology I would not be surprised if a 3D printed smart phone will exist within the next decade.

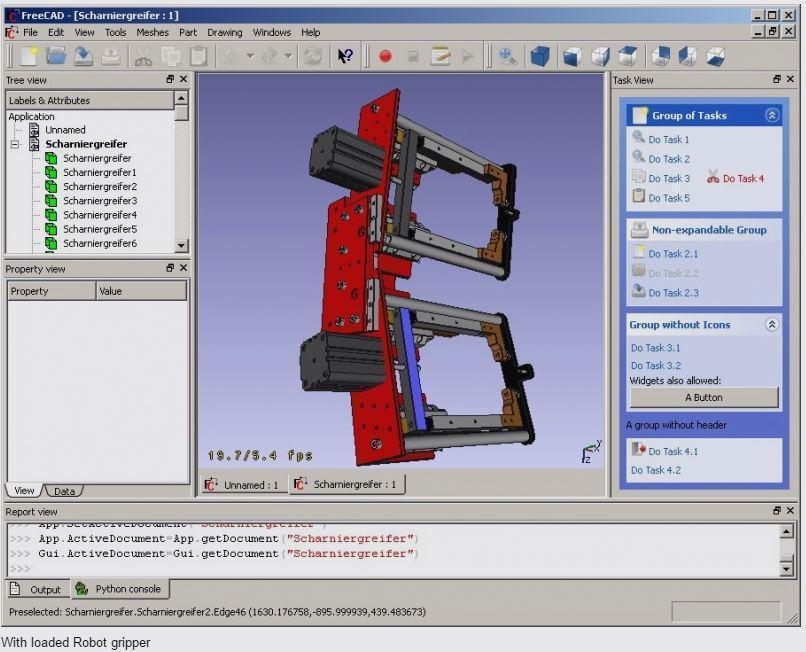
**3.8 Software**

It can be argued that the software is one of the most important parts in the 3D printers, since it is what, with the help of the microcontrollers and the power supply, tells the motors exactly what to do. However, because it is very unlikely that two pieces have exactly the same dimensions, and when moving the extruder miniscule distances are used, it is very important to calibrate the printer. Confirming that it is doing what it should be doing.

**3.8.1 CAD Models**

Computer-aided design (CAD) are models that have been schemed with the help of a computer software. It is important to point out that CAD models are both 2D and 3D. They have been used long by architects, engineers, and to build models with computer numerical control (CNC). However, since the irruption of the 3D printers they have been used to design models that would later be interpreted and printed, not before a translation occurred so that the printers could ‘understand’ what they are supposed to do.

There are different tools that can be used to design these models. Ranging from free open source software, such as FreeCAD, to paid tools varying in prices. Being DesignCAD ($99.99), Solidworks, and AutoCAD ($4195) some of the most known softwares among the latter category. Figure 3.8.1-1 shows an example of a CAD model being designed using the FreeCAD software.

**Figure 3.8.1-1:** *CAD model being designed using FreeCAD (Content protected by Creative Common Licenses Attribution 3.0)*

**3.8.2 CAD to G-Code conversion**

As discussed before, in order to print the CAD model there has to be a conversion of the file into a language that the machine can understand. This language, which is called G Code, will be discussed later on.

There is a considerable amount of open source tools that do this type of conversion including software such as Slic3r, Cura, or Skeinforge. What they do is slice the design into several 2D layers, so that the machine can move in the X-Y plane, when it is done printing one layer, it moves up to the next layer. As aforementioned, some machines use two extruder heads in order to print a filling material -which will be later removed by submerging the final model in a chemical- to deal with hollows design. Therefore, this code, not only controls the movement of the printer in the Cartesian plane but also controls how fast it is extruding material and from which head (if applicable) the material is being printed. The commands themselves will be further discussed later on, but it is important to mention that the G-Code, takes the extruder as a fourth dimension, and controls it using Cartesian coordinates.

**3.8.3 G-Code interpretation**

The easiest way to design an element is using a CAD tool and converting the file into G-Code.To some degree it is also possible to write the G-Code itself, or use a low level library like mecode -which provides a convenient, human-readable layer just above GCode, according to its developer. It is important to point out that mecode does not slice the CAD files, and therefore cannot be used to translate them into G-Code. In other words, it is also possible to design an object by telling the printer what to do. Whether it is using a language that the printer will directly understand, or using a language that can be easily translated into G-Code and is easier to understand by humans.

In order to provide a better understanding of what is going on at the low level, a few commands of G-Code will be discussed and how they are implemented into 3D printers such as the RepRap family.

For the most part, the nnn in each instruction present in the table 3.8.3-1 corresponds to the number of instruction on that specific instruction. For example in the G instructions G0 corresponds to rapid move, and G1 corresponds to controlled move.

However, specifically for the case of N instructions, the nnn correspond to the number of line that is being executed, in case it has been corrupted. It is a way to keep track of the execution in case a line has to be resent. It is also important to point out that the character ‘;’ denotes comment in this code. It is considered a good practice to remove the comments before they are sent, as the printer will completely ignore them.

As a basic general rule of thumb, the G commands mean standard G-Code movement commands. This means that his commands will solely related to movement. Either by telling the printer to make linear movements, arc movements, or small precise movements, or by telling the printer that it will start to use relative positioning or absolute positioning from now on. This last couple of commands are extremely important, because if they are not implemented in the firmware, and the G-Code generates them, the printer will probably go to the wrong location to keep printing.

The second set of commands that are used very often in the 3D printers industry are the M commands. These are associated more with all other aspects of the printer that do not directly involve movements. Such as heating on or off the extruder and the extruder bed, turning on or off a cooling fan, or showing some data into and LCD screen.

The rest of commands are rarely generated by Cura, Slic3r and the other G-Code generators, and therefore are not be implemented in the Pegasus Producer.

|  |  |
| --- | --- |
| Letter | Meaning |
| Gnnn | Standard GCode command, such as move to a point |
| Mnnn | RepRap-defined command, such as turn on a cooling fan |
| Tnnn | Select tool nnn. In RepRap, tools are extruders |
| Snnn | Command parameter, such as the voltage to send to a motor |
| Pnnn | Command parameter, such as a time in milliseconds |
| Xnnn | An X coordinate, usually to move to |
| Ynnn | A Y coordinate, usually to move to |
| Znnn | A Z coordinate, usually to move to |
| Innn | Parameter - not currently used |
| Jnnn | Parameter - not currently used |
| Fnnn | Feedrate in mm per minute. (Speed of print head movement) |
| Rnnn | Parameter - used for temperatures |
| Qnnn | Parameter - not currently used |
| Ennn | Length of extrudate in mm. This is exactly like X, Y and Z, but for the length of filament to extrude.Skeinforge 40 and up interprets this as the absolute length of input filament to consume, rather than the length of the extruded output. |
| Nnnn | Line number. Used to request repeat transmission in the case of communications errors. |
| \*nnn | Checksum. Used to check for communications errors. |

**Table 3.8.3-1*:*** *G-code commands with small explanation of what they do. Content protected by the Creative Common Licenses Attribution.*

An example of a very simple G-Code:

*N0 G90 ; Sets the printer to absolute positioning*

*;(relative to origin)*

*N1 G0 X12 ;Moves the extruder in a straight line to the position X=12 in the XY*

*;plane. The extruder speed will be the one defined in the*

*; configuration, or the latest one specified*

*N2 G91 ;Only used in Teacup firmware. Sets the printer to relative positioning*

*;(relative to current point)*

*N3 G1 X2.5 Y 3.4 E22.4 ;Moves the extruder from current point to X=2.5, Y=3.4*

*;extruding a total length of material equal to 22.4 mm*

To select different tools the command T is used, which will make it possible select what extruder will be used. However, when an extruder is not being used it has to be at a standby temperature; likewise, for it to operate, it has to be at a previously specified operating temperature. Therefore, this specific command follows a specific sequence in terms of movements, and changes in temperature in different parts of the printer.

The command *G10 P2 X17.8 Y-19.3 Z0.0 R140 S205* is one of the many that are used in combination of the T command in order to change elements. In this example, the P2 means that the parameter for the extruder 2 is being setted, the X, Y and Z values specify the offset with respect to the main extruder (note that all of the extruders should be at the same height for practicality). Finally the R value specifies the standby temperature in oC, while the S value specifies the operating value. Even though this command is not supported in some of the most popular firmwares of the RepRap printer, the combination of the T and G10 can be replaced by a different combination depending on the firmware that is being used.

**3.8.4 Open Source Software**

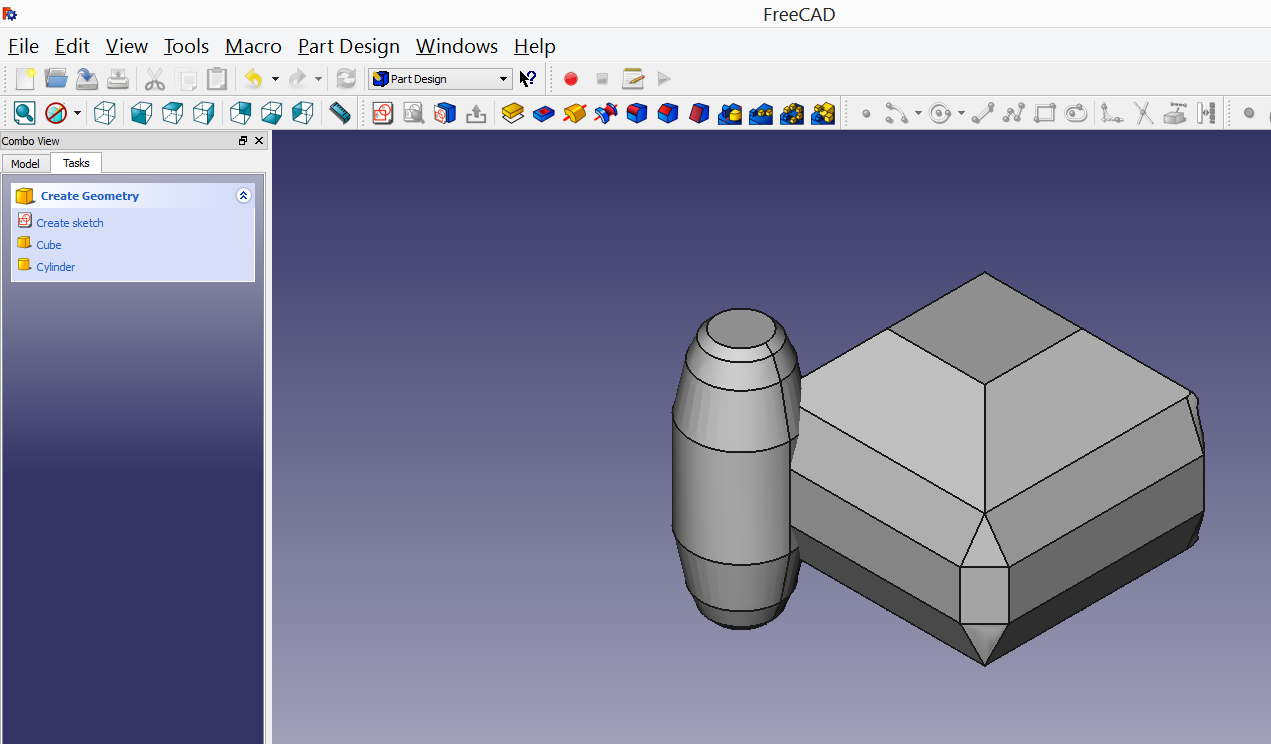
As it has been shown, there plenty of open source software with different kind of licenses. It is important to take advantage of this, in order to correctly guide the development of the software in the right direction.

**3.8.4.1 CAD Modeling**

For the CAD modeling the open source software FreeCAD was be used, as mentioned earlier. This software provides modeling tools to create any CAD object (.obj file) that the user wants.

With multiple tutorials available on their website, CAD modeling does not take long to learn the basics and how to use it. CAD provides similar tools to those found on commercial parametric modeling software. Figure 3.8.4-1, shows an example of an object that is being designed by one of the group members.

It is important to mention that even though it is an open source software (this might be the reason why) there are plenty of online tutorials that teach how to model different objects using FreeCAD. Sometimes it is a little slow and still has a couple of bugs here and there, but it was be good enough for what the members of the group wanted to do.



**Figure 3.8.4.1:** *Example of CAD object being modeled*

**3.8.4.2 CAD to G-Code Converter Software**

As mentioned before, there is a variety of open source software that can be used for this project. There are multiple advantages for using each of the software available. The main advantages will be analyzed next.

For Pegasus Producer the code will be written in a Windows computer, therefore the advantages and disadvantages will be analyzed when converting from CAD to G-Code in a Windows operating system.

**3.8.4.2.1 Slic3r**

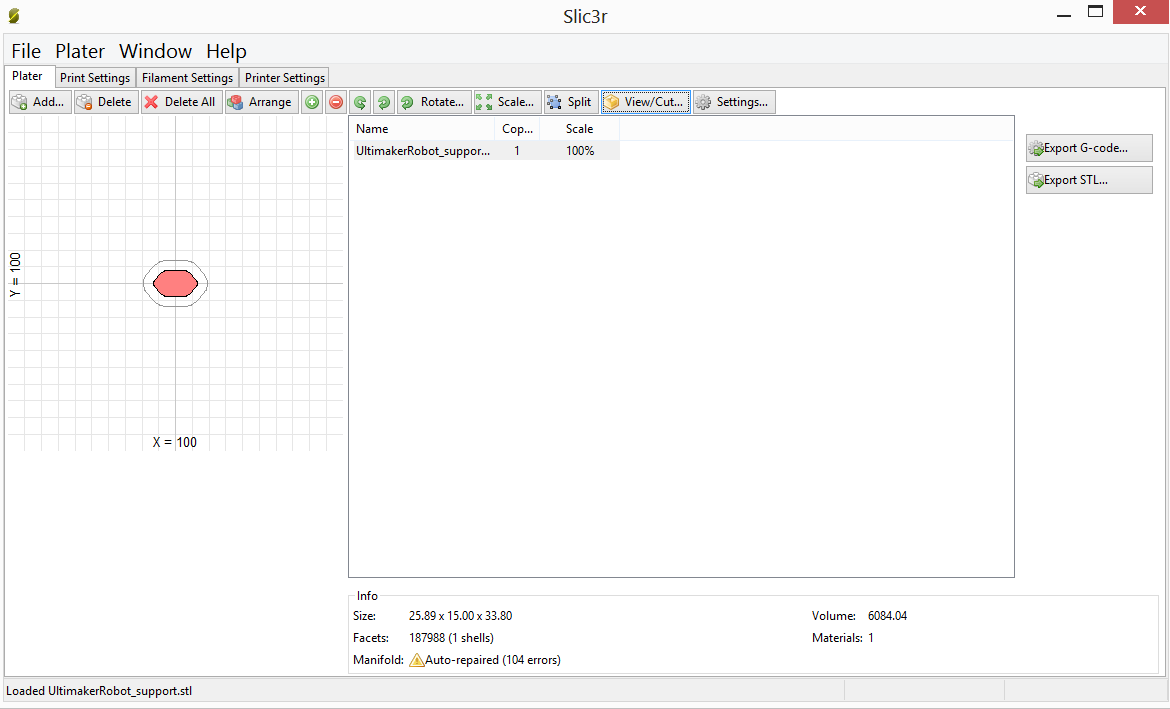
What stands out the most about Slic3r is that it can be controlled through the command line. Even though there is a graphical user interface (GUI) included with Slic3r, it provides an option to run it through the command line giving the user the opportunity to get more familiar with what is going on in the CAD to G-Code conversion.

When first installing the software, it will ask for a series of configuration settings that depend on what 3D printer is being used, the dimensions of it, and the temperature of both the heat bed and the nozzle when extruding.

Because the Pegasus Producer have the frame of a RepRap 3D printer, the dimensions were first extracted from the website, and this option will be selected.

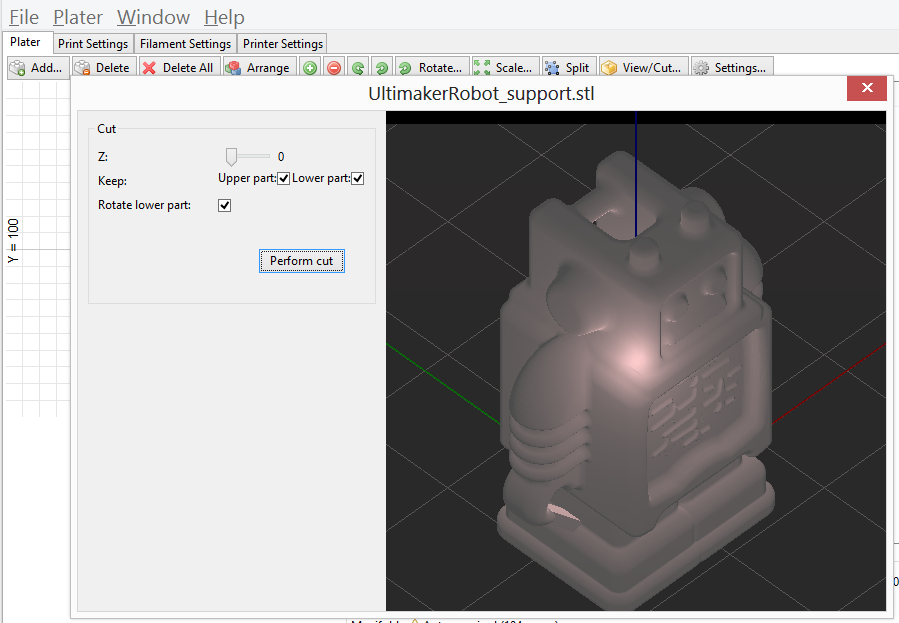
Figure 3.8.4.2.1-1 shows the software being used. As aforementioned, this will generate G-Code commands, from a CAD object previously designed.

It was first believed that after the G-Code is created some small changes will be need to be performed to it, in order for the printer to interpret it. However, the printer was able to handle this commands without any previous modification.



**Figure 3.8.4.2.1-1:** *Main view of Slic3r. Cross section plane of the CAD object*

When using the GUI for Slic3r -seen in figure 3.8.4.2.1-1 there are two things that must be mentioned. The first one is that the view of the CAD object is not the whole object, but the cross section at the Z- axis. Secondly, there are multiple tabs to change the configuration, so that printer can interpret the g-code, if needed. Other than that, it provides an easy way to construct the g-code where the “Export G-code” button must be pressed.



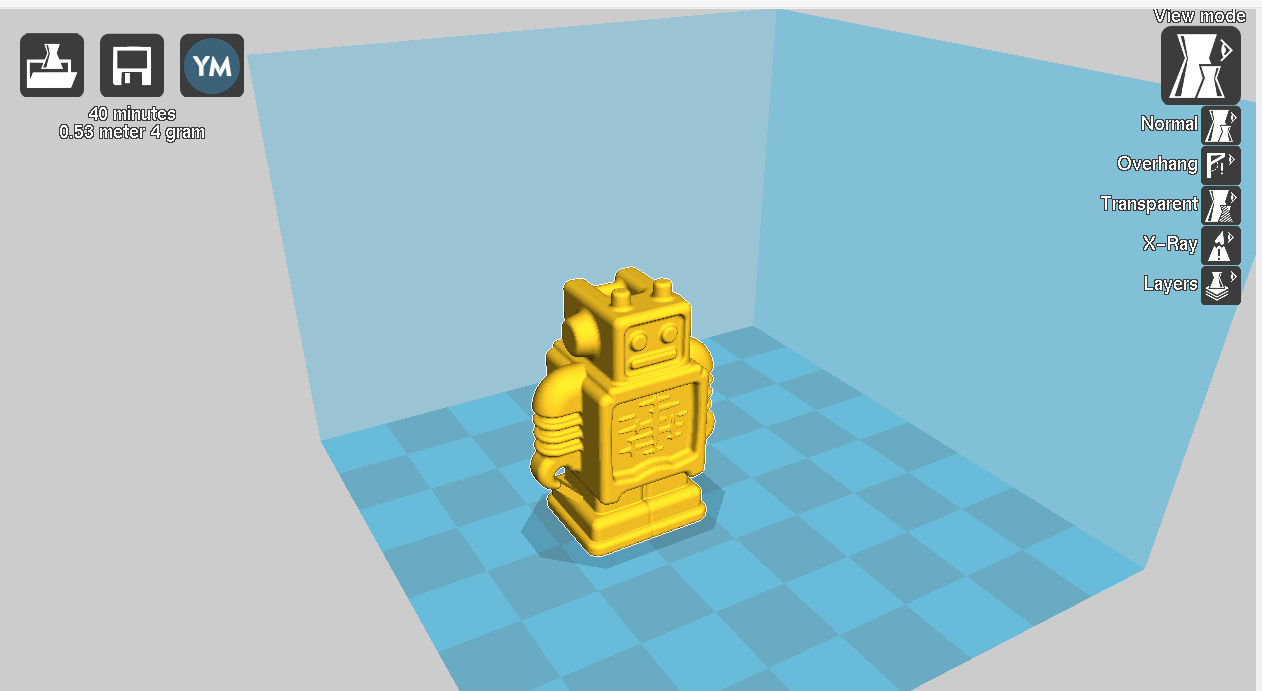
**Figure 3.8.4.2.1-2:** *View of the CAD object in Slic3r*

If it is desired to see the object that will be printed or to change the view to a different cross section, the “View/Cut” button must be pressed. This will bring up a new panel, where, as observed in Figure 3.8.4.2.1-2, the whole object can be seen. To change the view to a cross section of a different point in the z-axis there is a bar that can be moved, the number next to it represents the actual height in millimeters.

The advantages of using Slic3r are the opportunity to use it from command line, where a configuration file must be created if it is not desired to write the commands to configure the translation every time a new g-code is being produced. The option of using it from the GUI, where it provides a very simple form of producing the code and changing configurations if needed, but it does not do a good job in showing and interacting with the CAD object that will be printed by the Pegasus Producer. In spite of this, the prints that came after using Slic3r for generating the G-Code had a significantly higher quality than any other generated by a similar software.

**3.8.4.2.2 Cura**

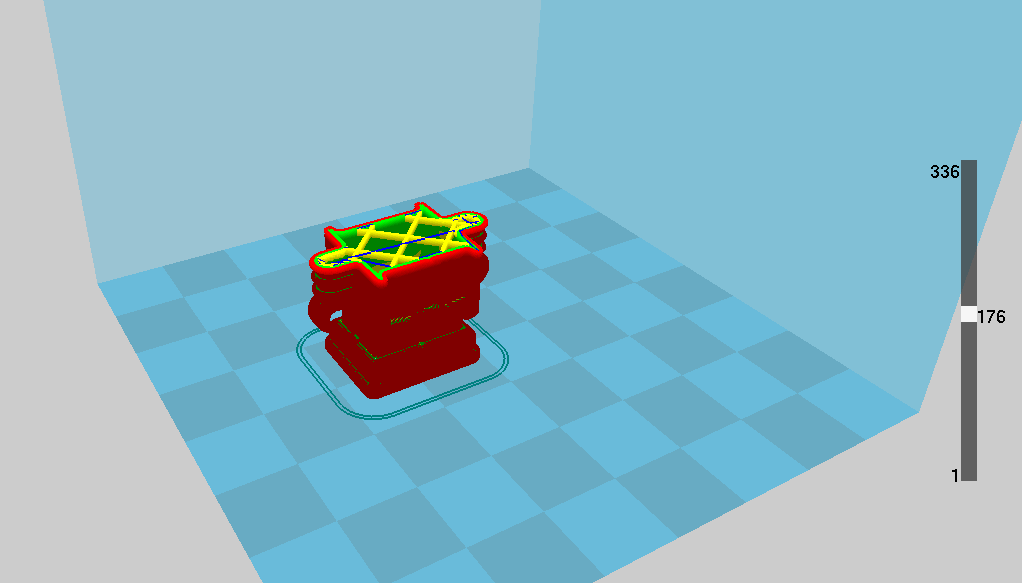
Cura is a open source tool provided by Ultimaker, a company that creates and sells 3D printers. The fact that this software was created, and distributed by a company that sells 3D printers is appreciated in the lack of command line usability and GUI design.



**Figure 3.8.4.2.2-1:** *View options of the CAD object in Cura*

What stands out the most of Cura is the fact that the object can be seen in the main view of the application. Figure 3.8.4.2.2-1 shows how the object -in this case a robot- is displayed in a 3D box to provide the depth perception and how the user can select different ways of displaying it. The two that stands out the most are the cross section view and the transparent view.

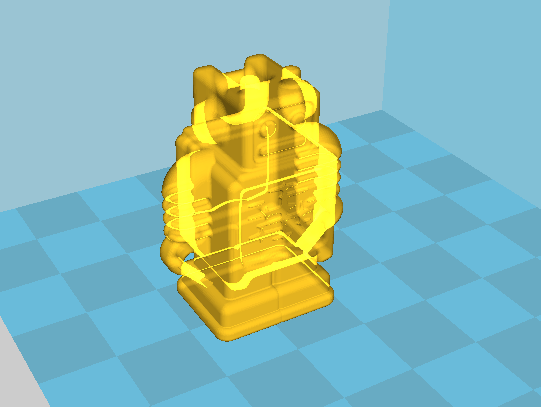
Like in Slic3r, the cross section desired can be selected by adjusting the value at a bar where the numbers represent the height in mm. The difference resides in that it does not show the 2-D plane cross section only, but a view of the object up until that height, as it can be observed in Figure 3.8.4.2.2-2. Again, the 3D box is still present to provide the depth perception.



**Figure 3.8.4.2.2-2:** *Cross-section view* *the CAD object in Cura*

On the other hand, the transparent view, shown in Figure 3.8.4.2.2-3, shows the whole object with the front layer being transparent displaying what is happening both inside the object (which is less transparent), and the back layer. Even though this is feature could be very helpful, the implementation is not as good as it should be since it can be unclear where in the x-y plane is located a specific segment of the object.

One of the best features that this software offers is the ability to perform some small modifications to the positioning of the object for stability purposes. As seen in the Figure 3.11.4.2.2-4 there is a rotation tool available. This tool will allow the user to rotate the object in any of the 3 axis, and the user is responsible for finding the position that gives the most stability to the object. Along this there is also the option for scaling, and mirroring, or inverting, the object with respect to a certain axis.



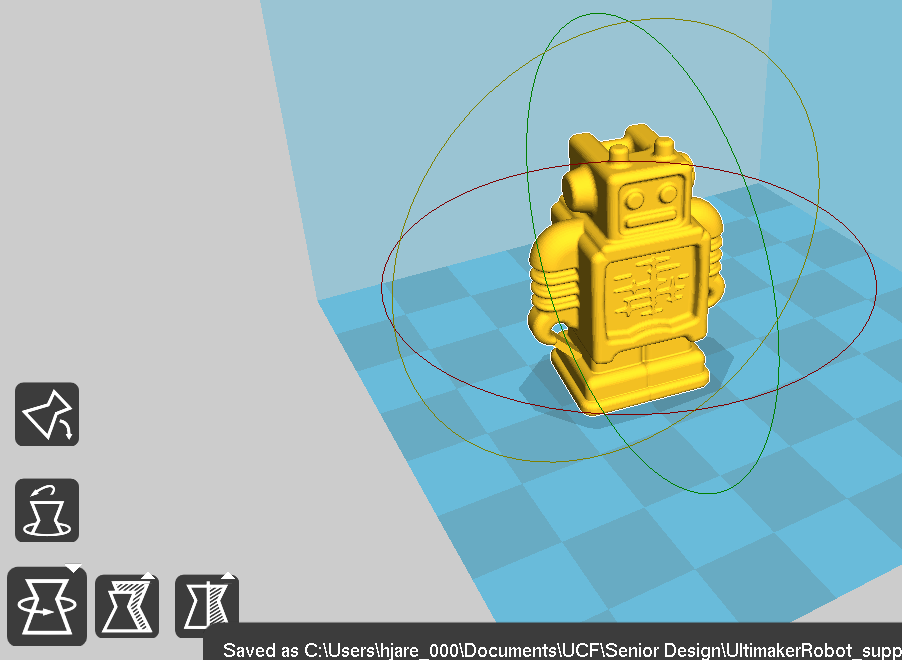
**Figure 3.8.4.2.2-3:** *Transparent of* *view ofthe CAD object in Cura*

In addition to the mentioned features, Cura performs a setting configuration when first opening the software. It does let the user to change this configuration later on, just as in Slic3r, but it is not as easy to get to the configuration window.

Cura also allows the user to open multiple CAD files simultaneously. Which combined with the ability to move, scale and rotate objects within the program it allows user to be able to print up to however many elements fit on the printer, simultaneously. This is also possible due to the fact that Cura lets the user know if an object is ‘out of bounds’ or if it is overlaying with another object.

The main advantages of this software is that the GUI is notably better designed and allows small changes to the CAD object for optimization purposes, such as providing design stability. The user, in general, should have experience a better interaction with this software than what they will perceive using any of the other software for this purpose that are discussed in the paper.

One important disadvantage of Cura is that it does not provide a way to be ran from command line. Therefore, to perform any change in the customization of the software it is necessary to navigate through the different setting windows until the desired option is found. This also deprives the more advanced user from having a closer interaction with the creation of the g-code. Yet, the most important disadvantage is that the quality of the print after using the G-Code generated by Cura is not good. It does not generates extra layers to polish the print, and it looks like the print was stopped in the middle of the process.



**Figure 3.8.4.2.2-4:** *Rotational tools in Cura*

**3.8.4.2.3 Skeinforge**

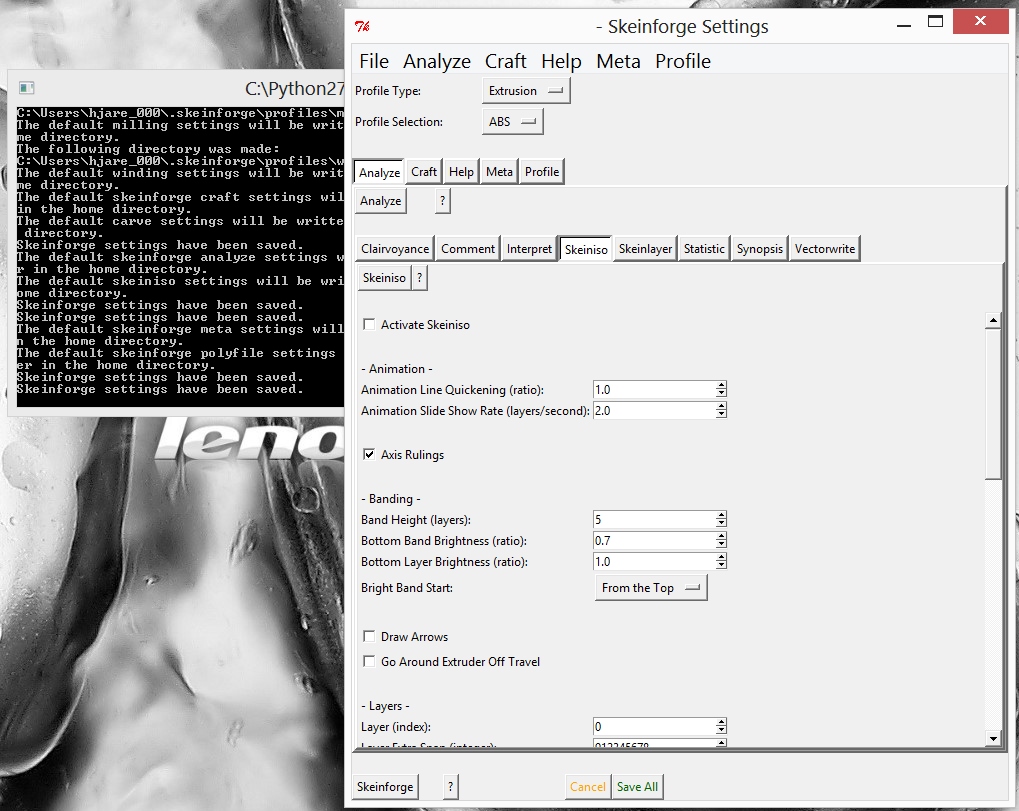
Skeinforge is an open source software composed of a series of Python scripts that converts 3D models into G-Code. It was developed to be used specifically for Rep Rap printers.

To be able to run this software it is necessary to first download and install Python 2.x. According to the Python Software Foundation, Python is “an interpreted, object-oriented, high-level programming language with dynamic semantics”18. It is considered a scripting language.

Then, Psyco can be downloaded. This is optional, as this software is meant to speed up the running of skeinforge. According to the Psyco website, it is a “Python extension module which can greatly speed up the execution of any Python code”. Nonetheless in the latest news, it mentions that the software is currently unmaintained, and it redirects to different compilers for Python. Since the software is unmaintained it was decided that it would not be installed.

Finally, the Skeinforge needs to be downloaded, and unzipped in the root folder of the computer. To open it, it is necessary to locate the file skeinforge.py and double click it.

As it is shown in the Figure 3.8.4.2.3-1, and unlike the softwares previously discussed, this tool provides a significant number of configuration options. This is because, it models the CAD object, plane by plane, and then converts it to g-code. Even though this tool does provide a GUI to visualize the CAD element, it is hard to find.



**Figure 3.8.4.2.3-1:** *Example of Skeinforge being used*

In general, the tool itself is complex to use, it is no longer maintained -the last version came out in 2012-, and it deals with elements that are not necessary for the Pegasus Producer, such as keeping track of statistics like the cost per hour of operation.

**3.8.4.2.4 Tool selection**

Even though each of the option discussed provide advantages and disadvantages, the software that was used to convert the .obj files (CAD objects) into G-Code was Cura mainly due to the easiness of use. This decision means that the G-Code interpretation, which is the software that will be developed and installed in the printer, must understand the instructions that Cura produces; this is an important factor to take into account in the designing process.

However, after not being satisfied with the quality of the prints that were performed, Slic3r was used, and it was obvious that the prints were now significantly more precise, and with better finishing touches. Therefore, the choice drifted in spite of Cura’s more user friendly interface.

**3.9 Heating Platform**

A heated platform, seen in figure 3.9-1, is the platform a 3D printer uses to layout the material, the purpose of the heated platform is to improve printing quality by preventing any warping from happening. Warping occurs when plastic cools off and shrinks due to the change in temperature, when shrinking does not happen throughout a printed part evenly, the bottom part of the printed material will warp. A warped print can be identified by looking at the corners, if the corners are lifted up from the platform the print is flawed. The heating platform allows for the printed part to stay warm as printing occurs. The use of a heating platform will yield higher quality prints with thermoplastics such as ABS and PLA, as discussed in the material research portion, a heated platform is a requirement when working with ABS plastic.

Heated platforms can be constructed from many materials, some materials hold advantages over other but generally any material that can stay consistently warm and not melt can be used. Regardless of material a heated platform needs to have proper insulation, safety precautions need to be taken to take care of electronics or any printer components that may lay underneath the platform. Typically a piece of cotton cloth, cardboard, or wool can be used as an insulator, this will prevent damage to thinks such as the Z axis bearings which would be connected to the heating platform and moving it up or down accordingly. To provide heat to the heated platform 6 Amps of current need to be provided, this current should be ran over 20 gauge wire for proper insulation, the wire can be bought or stripped from other components such as speaker wire. Durable yet flexible wire is required for the platform, the wire will be constantly flexed as the platform moves and will be exposed to temperatures over 100 degrees Celsius. Unsuitable wire can damage electronics, cause a fire, and produce an electric shock.

Special consideration has to be taken for the surface of the heated platform for the melted plastic to adhere to the surface. Many cheap materials can be used and even high end expensive 3D printers opt for the use of such surface materials. One material is Kapton Tape, with careful application of the tape allowing for no air bubbles it will provide for good surface that melted plastic adheres to and finished parts can easily be peeled off from without any damage. Another option is blue painter’s tape, it is cheaper than Kapton tape and the waxy surface of the tape allows for hot extruded plastic to adhere. The downside of blue painter’s tape as a surface material is that it easily peels and rips when removing an finished part, due to its price availability it is not hard to find and replace. Other usable surface materials include coating the heated bed with a glue stick such as Elmer’s glue to keep melted plastic from sticking onto a glass heated bed or moving when being printed. It is not recommended to use a bare surfaces, metal surfaces such as aluminum, and copper and bare glass do not provide enough adhesion with the melted plastic resulting in a print that will be moved around.

The platform itself comes in many different materials and a lot of research is being done to find out what material provides the best plastic quality for low cost 3D Printers. Glass is a popular material for heated beds, cut glass can be found at any hardware store and only 3mm thickness is needed. Glass has low thermal conductivity so sporadic temperature spikes are rare but if heated unevenly it can fracture and special tools are required to drill holes on the glass for mounting on to an axis, it is also rather heavy for 2 stepper motors to move. A flat sheet of metal can also be utilized as a build surface, its high thermal conductance will ensure that heat will spread evenly. The downside of metal is how much it expands when heated, this expansion can caused the printed material to warp and become distorted. To do the actual heating nichrome wire, or resistors are placed underneath the heated platform and connected to the power supply.



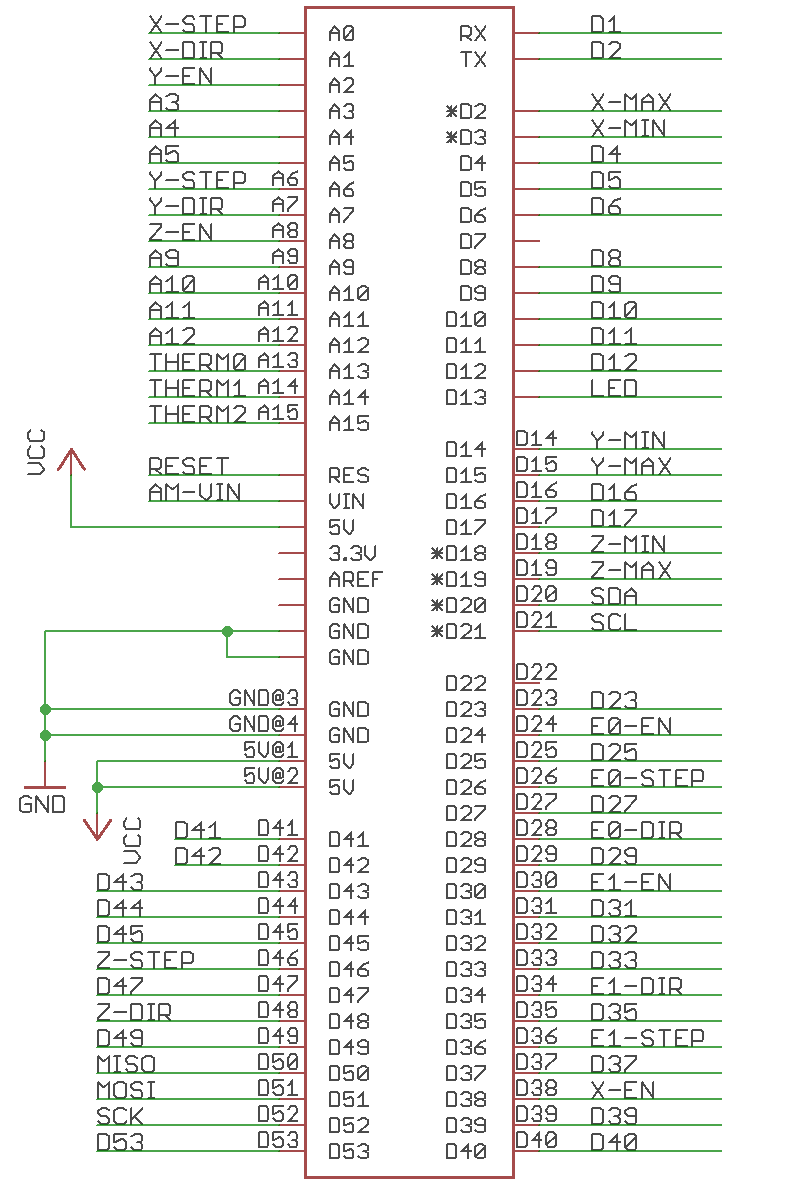
**Figure 3.9-1:** *Heated Platform*

**4 Design**

**4.1 Electrical**

**4.1.1 Microcontroller**

The choice of microcontroller has been the ATmega644p for its price, speed, digital and analog output, and documentation available. The ATmega644p features an advanced 8-bit RISC architecture, 8 KB of FLASH memory, standard debug protocols such as I2C, 86 programmable I/O lines, and low power consumption. Achieving a throughput of 1MIPS per MHz it is very easy to optimize power consumption versus processing speed. 256K bytes of In-System programmable flash with read write capabilities can accommodate all the hardware needed to process g-code. For the Pegasus Producer 3D a single ATmega644p will be used which will control the motors and process all the g code data. The ATmega644p will not be producing any of the PWM signals for the micro-stepping, it will simply send the digital signals to the control boards which will allocate movement accordingly. At first the electronics and system will be tested with an Arduino MEGA, following successful signal and software processing a custom PCB will be made to host the ATmega644p chip and all of its pin outs. The custom microcontroller has the ability to be preprogrammed with firmware and can be tested with existing firmware that is currently in use with other 3D printers. The custom PCB is designed using eagle and made by a third party, soldering of components will be done by the group. On the custom PCB almost everything remain that is on the Arduino MEGA but the major changes will include the ability to be powered on by a ATX power supply and not through the USB. Programming for the ATmega644p will be done through its in-system programmable flash using one of the available debug protocols and the AVRISP MKII In-System programmer. The in system programmer uses serial peripheral interface (SPI) to upload code onto the microcontroller. The AVRISP MKII is a free IDE distributed by Atmel to program Atmel microcontrollers. Once the final working firmware is loaded onto the microcontroller it will not be altered which is why it is so important to first experiment with Arduino Mega development board prior to uploading buggy firmware onto the ATmega644p microcontroller. Most of the communication from the microcontroller to the motor control board will happen through SPI, on the PCB all the outputs have to be traced and placed appropriately for the motor control shield to work appropriately. The custom PCB will have pin outs for all the inputs detailed on figure 4.2.0-1, as one can see almost of the ATmega644p’s outputs will be used for the purpose of 3D printing.



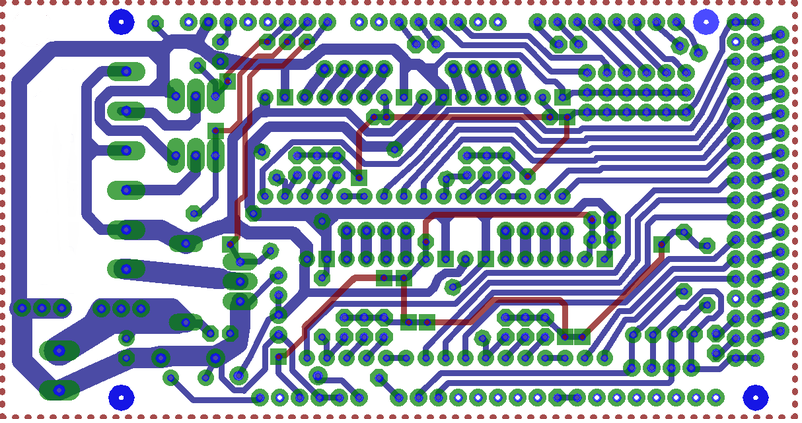
**Figure 4.1.1-1:** *ATmega644p Pin Layout*

**4.1.2 Control Board**

A control board for 3D printer is typically an all in one circuit board that has connectors for motors, end-stops, thermistors, heaters, and power connectors. The Pegasus producer uses a control board based off RAMPS or RepRap Arduino Mega Pololu Shield. RAMPS supports an input voltage of 12V and runs six mosfets for controlling 2 heaters, one heat bed, and passive cooling fans. RAMPS reference design supports up to five stepper motors using a that require five separate stepper motor drivers. Motor control is achieved using a digital potentiometer, one each for the X-axis, Y-axis, the extruder, and two Z-axis. RAMPS as well as the design includes functionality to be able to run the printer untethered to a computer allowing the logic to be powered directly from an internal power supply. RAMPS supports expansion by allowing the addition of an SD card reader, LCD display, and any I2C controlled device. RAMPS and the Pegasus producer custom control board design both utilize the Marlin firmware for 3D printing and due to the use of the ATmega644p chip used in an Arduino Mega, the Arduino IDE is also be used for firmware uploading.

The custom control board for the Pegasus producer consists of one or two boards. One design idea is to integrate both the ATmega644p and all the functions of RAMPS into a single PCB, the motor drivers are still to be separate. This idea will allow for a cleaner more aesthetic look with less wires, the design however will be much more complicated requiring very precise soldering and high room for error. The second idea is to separate the microcontroller from the control board, the ATmega644p will have its own custom PCB with all the pins laid out with external connectors which the custom PCB of the control board can interface with. This idea is much simpler to manufacture and design but will take up much more physical space that may compromise the size specifications of the Pegasus Producer.

With the design we chose, the control board will have the same connectors for the motors, heaters, heat bed, and cooling fans. In general, the control board layout will look like figure 4.2.1-1.



**Figure 4.1.2-1:** *Control Board PCB layout*

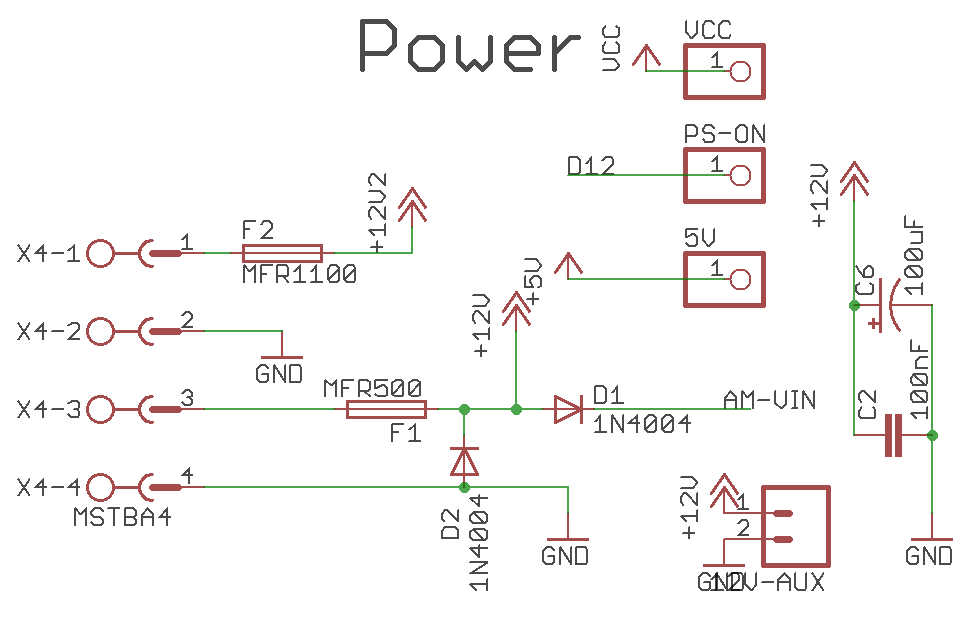
The following discussion will go in depth with the design and components of each section of the control board laid out in Figure [6]. Discussion will begin with power and move on into, heaters and fans, stepper drivers, end stops, thermistors, and I2C design. It is important to note that the Pegasus Producer’s entire control assembly will be ran from this one board even if the microprocessor is not present on the control board.

**4.1.2.1 Power Input**

Power for the control board is supplied from an internal power supply delivering 12 volts and minimum 5 amps to run the motors and 11 amps to run the heat bed and other heated components. The ATmega644p is unable to power the control board or vice versa, both boards will use different power rails regardless of whether they are on the same board or not.

Figure 4.1.2.1-1 shows how power will be delivered into the control board, a green power terminal will be soldered onto the PCB and power from the power supply will be distributed to the other components, specifically into the motor drivers. Power for the control board has been designed for the use of an ATX power supply, the PS-ON pin is there to short the required wires to turn the power supply on similar to a computer, the microcontroller manages this via the D12 pin.

Consideration for powering the ATmega644p microcontroller has been left in the control board design, diode D1 (1N4004 diode) is there to connect the control board input voltage to the microcontroller and power it from there removing the need for a USB or discrete PSU rail to the power the microcontroller, the D1 diode is still under experimentation and may be removed. For safety of all electronics within the control board, 35 volt electrolytic capacitors are being used along with the MF-R500 5 amp PTC fuse rated at 30V and the MF-R1100 11 amp PTC fuse rated at 16V. In this design we are using components with a high overvoltage protection even though only have of the rated maximum voltage will be used, this is important to properly protect the stepper motor drivers from damage.

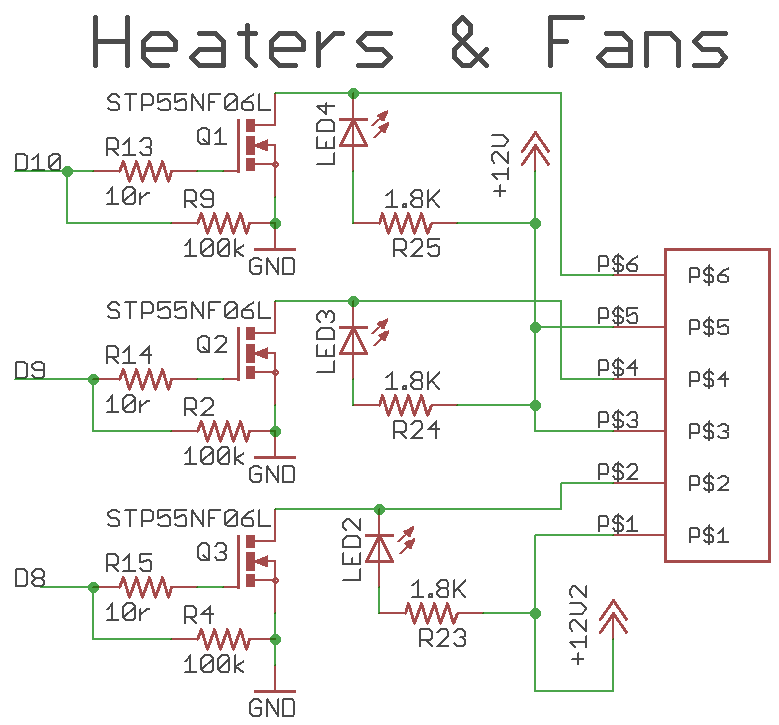


**Figure 4.1.2.1-1:** *Power Input and Output for Pegasus Producer Control Board*

**4.1.2.2 Heaters and Fans**

The heaters, fans, and heat-bed is all controlled from mosfets terminals and look similar to the power terminal. The purpose of the heater and fan design is to deliver the required power and amperage to produce enough heat to melt the plastic within the hot ends and to maintain a steady stream of power to keep the heat bed hot. Without proper heating 3D printing will not be possible and inadequate heating will lead to disastrous results if the plastic does not reach the melting transition point, due to that precise heating is very important. To properly heat the hot end and the heat bed 11 amps or more of power will be delivered via the power connectors are redirected into the mosfets, the mosfets used are three N-Channel STP55NF06L Mosfets.

Microcontroller pins D8, D9, and D10 deliver the required signals to control the Mosfets and deliver adequate heating, the power and amperage is then distributed evenly into the mosfet terminal depicted on the right of the schematic seen in figure 4.1.2.2-1. The mosfet outputs is where the connectors go to that heat up the hot end of the extruder and the heat bed if one is required. Ideally, only one mosfet and its corresponding output would be needed to run a 3D printer with a single extruder but design decision on the Pegasus producer called for the possibility of expansion into dual extruders and a heat bed for the use of ABS thermoplastic printing material. The fan output s uses the same outputs as the heating elements, fan is just there to have the option to add passive cooling to cool either the stepper motor drivers or the extruded plastic.

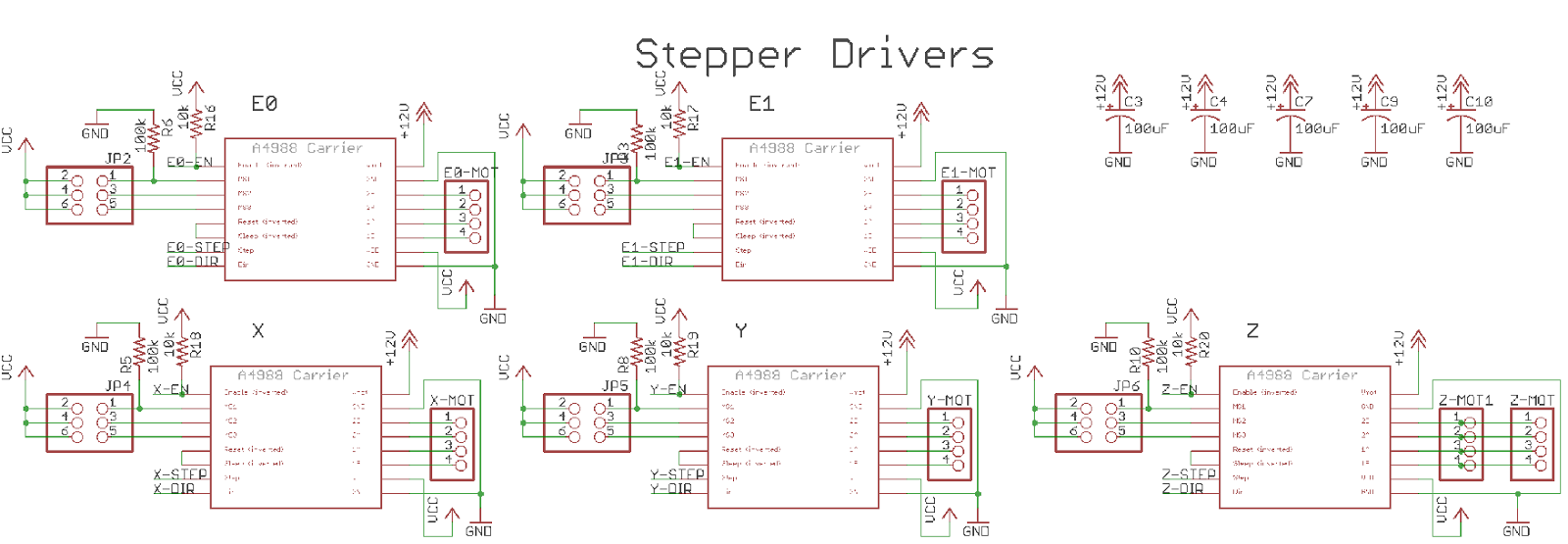


**Figure 4.1.2.2-1:** *Heater and Fan Configuration of the Control Board*

**4.1.2.3 Stepper Motors**

The control board for the Pegasus Producer’s main purpose is to control the stepper motors, without this 3D printing would be impossible as there would be no movement of the X,W, and Z axis and no plastic being pushed into the extruder. The idea is for the control board to be the interface between the microcontroller and the stepper motors, the control board is supposed to translate the signals from the microcontroller and convert them into direction, speed, and distance that a motor driver can translate into movement of the actual stepper motor. Stepper motors are not directly wired onto the control board, the stepper motors are connected to motor drivers which is then placed on output pins on the control board. With current design specifications the Pegasus Producer is able to support four stepper motors, one motor for the Z and Y axis, and two motors for the Z axis and extruder. The two Z axis extruders move in conjunction so only a single output is required, this is better depicted in figure 4.1.2.3-1.

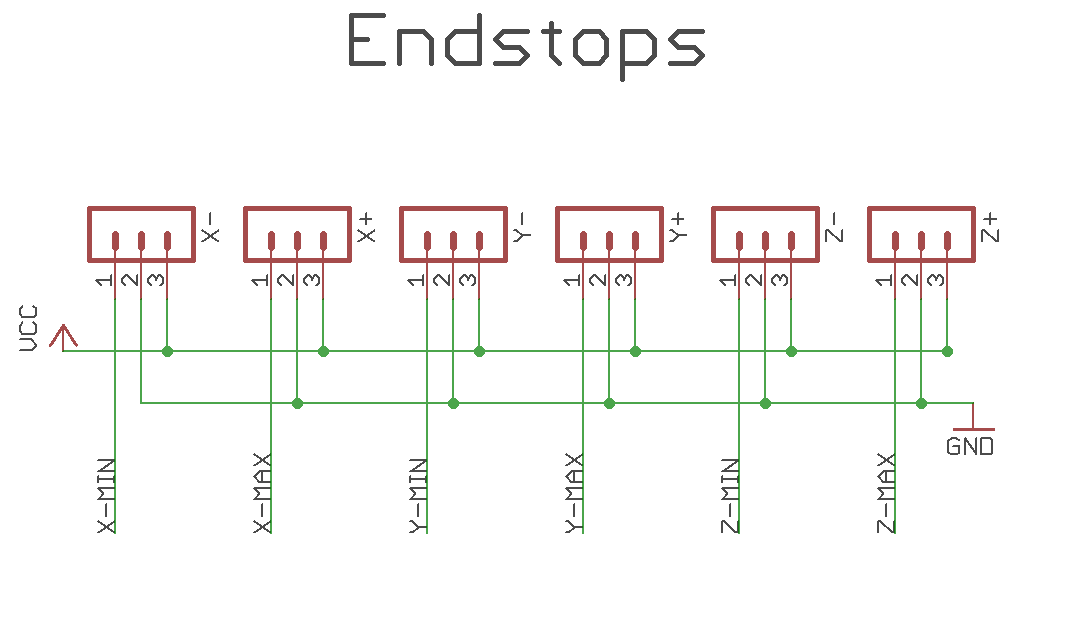
The A4988 carrier mentioned in the schematic is the actual stepper driver which is not solder onto the control board. The idea is for the stepper motor drivers to be connected to the control board via a series of driver sockets which will be input pins above all the components so that the small A4988 boards can hover over the control board allowing for easy wiring to the stepper motors and provide adequate passive cooling to the stepper drivers which are prone to overheating. The schematic is not very complicated to understand, power is delivered from the power section of the control board onto the stepper motor drivers. Each stepper motor driver received three commands from the microcontroller, for example for the X axis, the X-EN pin enables the motor, the Z-Step pin manages the amount of step the motor should do, and the X-DIR pin simply states the direction of the motor. Only two pull down resistor are required per stepper motor driver.



**Figure 4.1.2.3-1:** *Stepper Motor Driver Schematic*

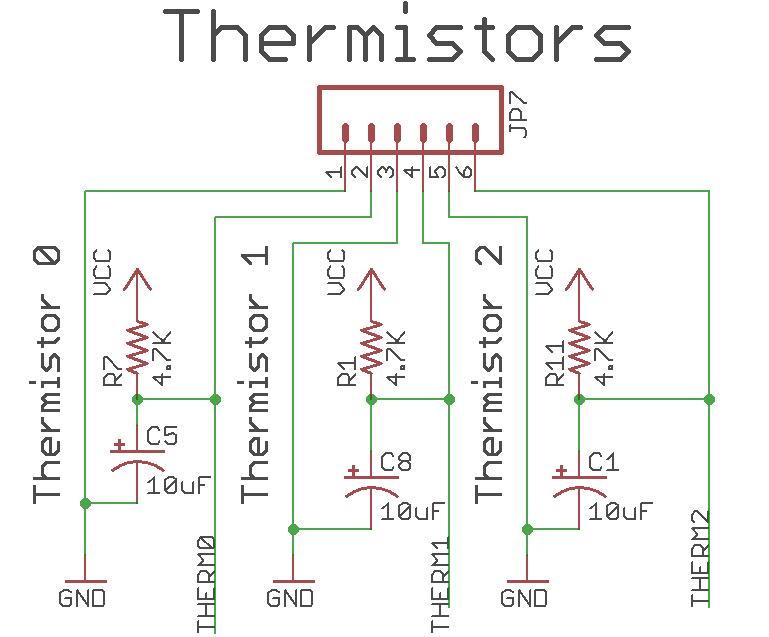
**4.1.2.4 End Stops and Thermistors**

The more analog portion of a 3D printer comes in the form of end stops and thermistors. An end stop goes at end of an axis and send a signal to the microcontroller that a motor has moved too far in one direction and has reached its bounds. Two end stops are used per axis and their use are highly recommended for calibration and unaltered performance of a 3D printer. An end stop only requires three wires, signal, VCC, and ground. The way it works is if VCC is shorted then a signal is send to the microcontroller to note that a motor is at the end of the axis. The schematic for end stops is very simple, the 2 connectors that go out to the physical end stop mechanism are soldered right onto the control board as pictured in Figure 4.2.1.4-1.



**Figure 4.1.2.4-1:** *Schematic for Mechanical End Stop*

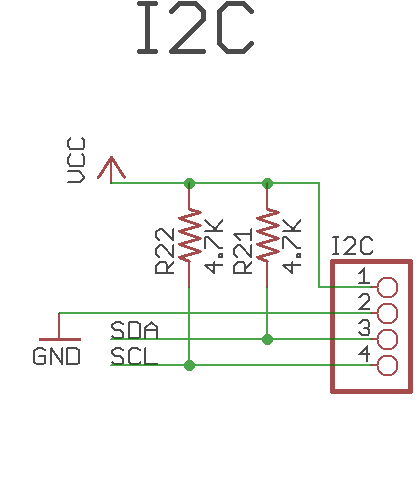
Thermistors, schematic seen in figure 4.2.1.4-2, have the important task of relaying temperature information back to the microcontroller, depending on the thermistor the microcontroller will know if the appropriate temperature has been reached to begin extruding the melted plastic. A faulty thermistor will lead to many jams and will probably destroy the extruder in the process. Three thermistors are present, two for each extruder and one for the heat bed. Power comes from the power section of the control board and is shared with the mosfets in charge of heating. Only a single signal goes to and from the microcontroller and is labeled THERMO for each respective thermostat.



**Figure 4.2.1.4-2:** *Thermistor Input*

**4.1.2.5 Control Board – I2C**

For the purpose of expansion the Pegasus Producer control board will include an I2C port. I2C is a popular protocol to attached low speed peripherals to embedded processors such as the ATmega644p, it uses a standard master and salve design using the SCL and SDA lines for 7-bit addressing and can be seen in figure 4.2.1.3-1. With an I2C connector the Pegasus Producer has room for expansion for LCD or simple button controllers, the ideas are limitless. Along with the support for SPI and Serial with the ATmega644p UART the Pegasus Producer has ample support for future additions such as Bluetooth, SD card slots, 802.11 wireless. The addition for I2C was straightforward and did not take too much board space so the design is relatively simple. Two 4.7k resistors were used as pull down resistor for the SDA and SCL lines, on the PCB the I2C connector will be bare until an expansion is introduced.



**Figure 4.1.2.5-1:** *I2C wiring*

**4.1.3 Motor Selection**

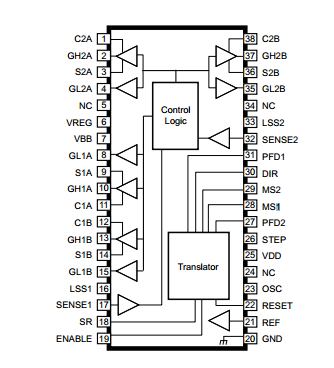
After carefully analyzing the options with some of the motors that the Pegasus Producer is able to utilized, stepper motors came on top with its ability to offer precise precision by having small increments in its operation. The stepper motors are used widely in the robotic industry, stepper motors can operate with each pulse of power that is to be provided from the RAMP 1.4 driver. The ability for the stepper motor to move in such a small increment makes this the ideal candidate for the Pegasus Producer since it is going to require a lot of repeatable positioning.

**4.1.4 Motor Drivers**

Stepper drivers keep the power that drives the motors separate from the power that runs the ATmega644p. Due to the need for high amperage the microcontroller is unable to run stepper motors directly. Without the use of motor drivers fractional steps on stepper motors would be impossible which is required for precision. As mentioned in the control board section, the motor drivers hover on top of the control board and connected via pins, this requires for the use of a specific PCB with the entire motor driver circuit and chip on it. The reason for purchasing the entire assembly rather than designing and constructing one is due to the difficulty in sourcing a chip capable for acting as a dual full-bridge gate driver with an integrated micro stepping translator. As stated in the research for motor drivers, directly chips that meets the requirements for the Pegasus Producer was not an option unless bought in large quantities overseas. Online retailers offered entire motor driver assemblies that can be used so the design of the control board was adjusted to accommodate them. In total, five stepper motor drivers are used.

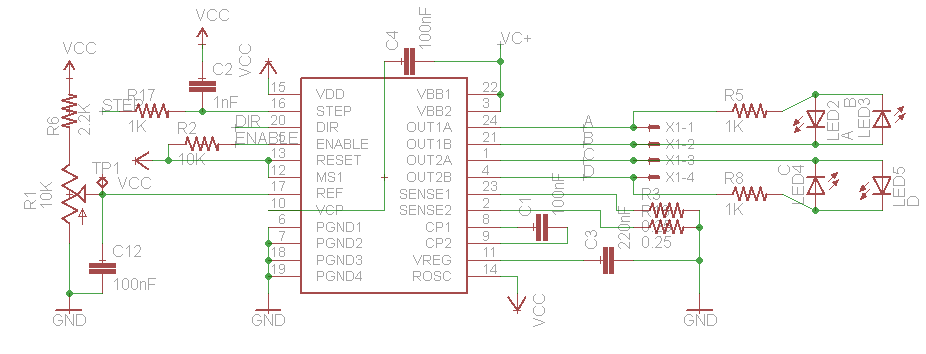
The stepper motor driver chip chosen for the Pegasus Producer is the Allegro A4989. The Allegro A4989 is dual full-bridge mosfet driver with a micro stepping translator that features 2-wire step and direction interface, synchronous rectification, cross-conduction decay and operates with 12 volt to 50 volt power supplies. For this chip, micro stepping is controlled by a two-wire and direction interface which provides micro stepping at full, quarter, and sixteenth step resolutions. The most valuable aspect of the A4989 is the inclusion of the microstepping translator which allows for signals straight from the microcontroller to control the stepper motor. The chosen motor drivers is powered off N-Channel mosfets such as the one used in the control board design (STP55NF06L) allowing for a cost effective solution for high precision motor driving required for 3D printing. The A4989’s main disadvantage is it heat output, it can quickly reach its thermal shutdown temperature of 165C without adequate cooling.

Figure 4.1.4-1 shows the pin out of the A4989 that is placed on a pre-bought PCB, the PCB includes the necessary inputs and outputs to interface the control board with the stepper motor. Not using this chip would require extensive design to create a microstepping translator which would be out of the scope of the Pegasus Producer in terms of engineering and cost.



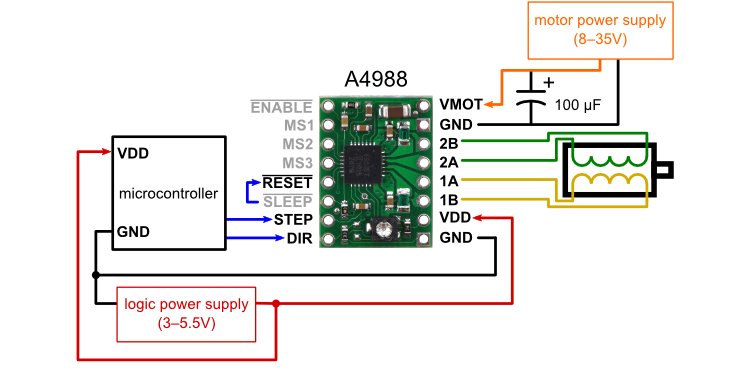
**Figure 4.1.4-1:** *Allegro A4989*

In general, the pre-assembled A4989 PCB consists of an interface for the stepper motor, a power circuit for power regulation, minimum and maximum sensors, and a driver circuit which can be seen in figure 4.1.4-2. The PCB includes holes for soldering of wire or pins that then can be placed on top of the control board. The idea of implementing the entire driver circuit on the main control board was discussed but due to the nature of the chips overheating and becoming inoperable, the idea was discarded, the ability to be able to change the degree of microstepping by utilizing another chip also played a factor in the decision.



**Figure 4.1.4-2:** *Driver Circuit for A4989*

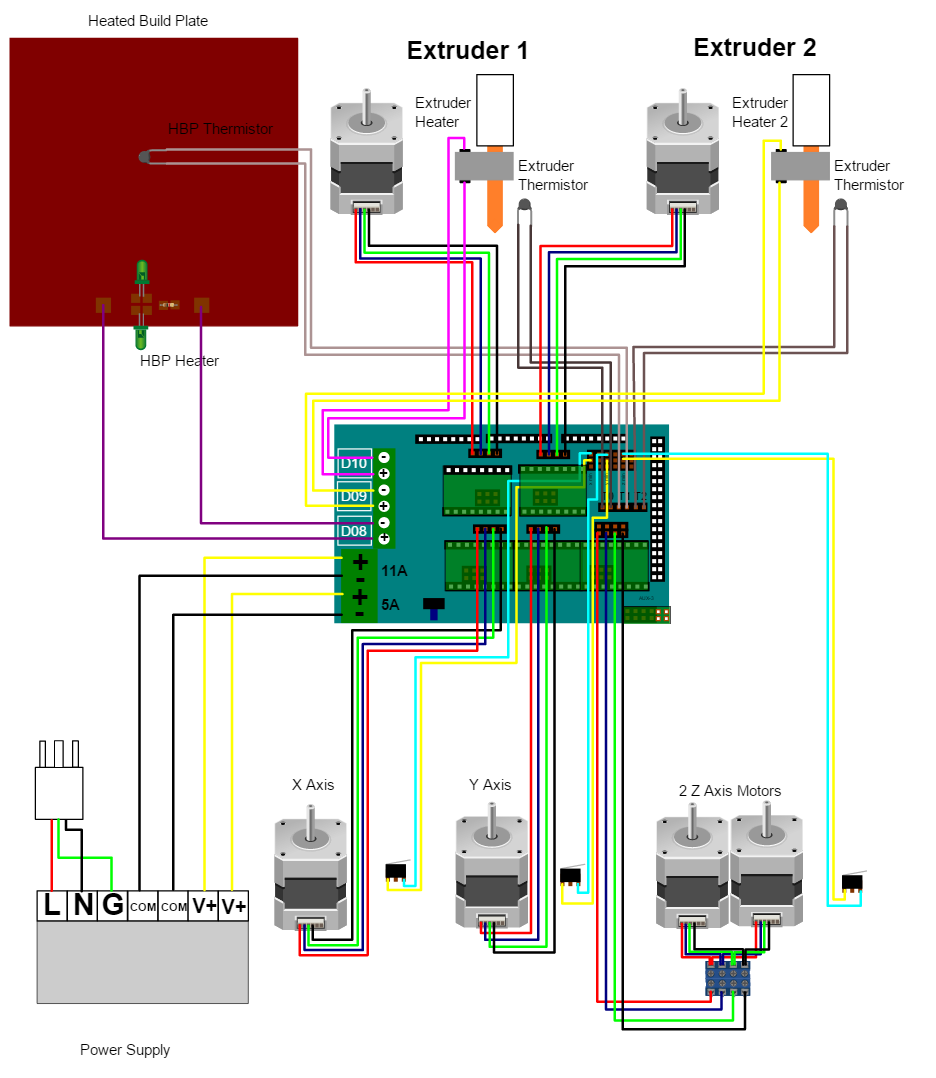
Since no A4989 were available for purchase in small amount a stepper driver chip that had a similar design to Figure 14 was sourced online. The Pegasus producer control board was also designed to fit the pre-bought PCB. The entire assembly for the motor driver is only 0.6” wide and 0.8” in length and typically cost under $10, since 5 drivers are used, a small PCB was a requirement. Building and construction a similar PCB would require a much larger control board to accommodate the 5 driver assemblies. The entire motor driver assembly that has been purchased is shown in figure 4.1.4-3.



**Figure 4.1.4-3 -** *Motor driver containing the A4988*

**4.1.5 Wiring**

The “all-in-one” design of the control board allowed for some very simple external wiring. The main wires that will be present are the power wires going from the power supply to the control board and microcontroller, the control wires from the stepper motor to the motor drivers, end stop wires, and the thermostat wires. For aesthetic purposes the wires are wrapped and sealed in heat shrink adhesive for the final design. All wires will consist of 24 gauge copper wire, every wire will be color coded for easier implementation and testing. No wire will exceed the length of 24inches for the entire 3D printer. The entire wiring assembly is detailed below in Figure 4.1.5-1.



**Figure 4.1.5-1:** *Electrical wiring for the Pegasus Producer 3D*

**4.2 Power Supply**

**4.2.1 AC circuit**

The Power supply circuit is to be divided into different parts. The Transformer, Rectifiers, and the DC components. All these components would work together to deliver a DC steady output voltage.

**4.2.1.1 Multiple winding Transformers**

The transformer is the most important piece of circuitry in the Pegasus producer because it’s the piece of hardware that does the most amount of work in the circuit. Also without the proper conversion form AC to DC the electronics that operate for a smooth DC input wouldn’t be able to operate properly. The multiple winding transformer can properly connects the primary winding to a higher voltage and cant output differently at its secondary. In order to be able to utilize this transformer the voltage and current rating must be identical in order to be properly adjusted to the design. Better observed in Figure 4.2.1.1-1.

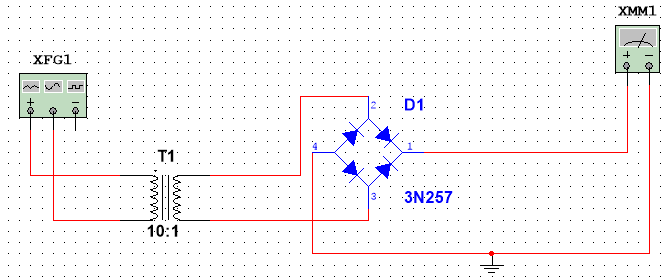


**Figure 4.2.1.1-1:** Multiple Transformer Winding

The downside of utilizing at multiple winding transformers is that it normally increases cost because of the fact that it increases in size. When looking at production cost the multiple winding would be more expensive. The solution to this circuit would be to use a full wave bridge rectifier.

**4.2.1.2 Full-Wave Bridge Rectifier Circuit Design**

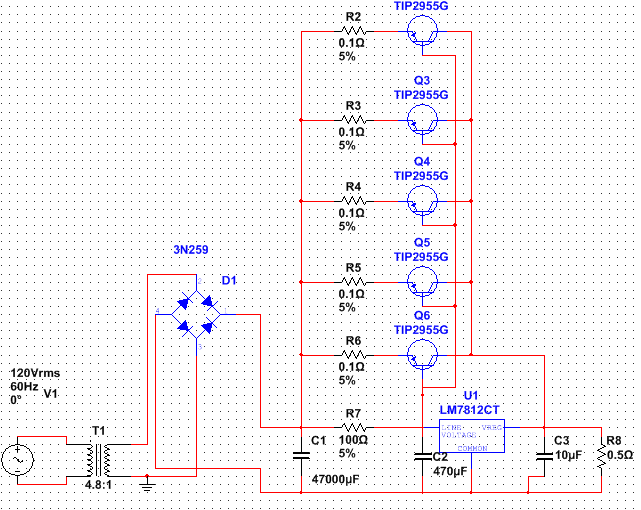
The Full wave rectifier would be the power circuit utilizes four diodes connected in a closed loop. The secondary winding is then connected to the diode and the load is then follow by the bridge connection. At any time, there are two of the diodes are on for each of the half cycle.



**Figure 4.2.1.2-2:** Schematics for the AC to DC full wave rectifier

Unlike the a single diode this setup allows for a larger mean DC output value that are smooth by the capacitor that’s normally placed in parallel. The determined capacitor is determined by the working voltage needed to be less than the load output of the rectifier and the capacitance value needs to be larger to eliminate the fluctuation on the voltage. The schematic can be seen in figure 4.2.1.2-2.

**4.2.2 DC circuit**

****

**Figure 4.2.2:** Project Schematic

After the transformer the rectifier worked together to output a smooth DC voltage we have arrived at the section that would require regulations and DC circuitry in order to provide 12V that’ fully rectified.

Having an understanding on the functionality of a power supply is very important implementing the design of a power source that’s adequate for the Pegasus Producer 3D printer. There are many options that are great considering for the 3D printer and a proper understanding on the functionality and process of the different implementations is very important. As the name implies the primary function of a power supply is to supply power that is in a certain form, and convert it to a specified form, that best fits the design specification, hence why they are called “Electric Power Converters”. In the conversion process every power supply consumes the energy that’s demanded in addition to the power dissipated by the internal components. There are some general classifications of the power supplies;

Function

These are the types that carry functional features integrated such are regulating and maintaining a steady voltage/current even when there’s a change on its input whereas the unregulated power supply functions differently and it fluctuates with the change of conditions.

Mechanical; A bench source of power supply that tends to be used for purposes such as testing a circuit, this circuitry can be found on mounting bases or even sometimes built in machinery

Power conversion; These supplies can be divided into linear and switching types. The more common one linear uses the input power directly along with the transformer and other components such as full wave rectifiers, differently from the switching power supplies; it convers AC to DC in form of pulses before actually processing and these tend to be components that operate predominantly in non-linear modes. Switching coverts are usually more efficiently since components spend less time operating.

When building the power supply there were certain things that had to consider such as the filtering capacitor to make sure that the ripple current was within safe range going into the input to the microprocessor.

**4.2.2.1 Capacitor Selection**

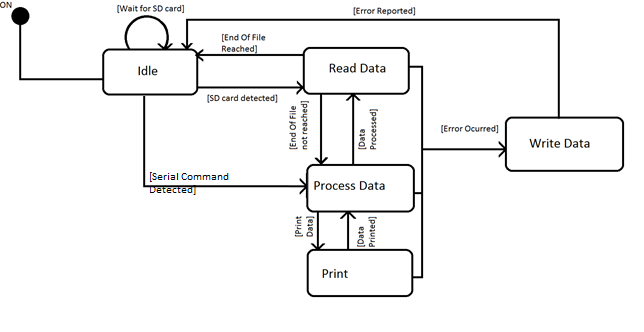
In the Pegasus Producer it was imperative to keep the rectifier ripple to 0.5V or less to avoid the alternating lost in the sinusoidal wave from dipping into the microcontroller. For the sake of planning for full load the team assumed a constant load current the voltage on the cap will decrease linearly (ΔV = I\*Δt/C). For a full wave rectifier the time between the peaks is 8.3ms (1/120Hz). So, to keep the rectifier ripple voltage to less than 0.5V at a 15A load there must be roughly 47,000uF or larger cap (15A\*8.33ms/0.5V)= 47,000uF which case the capacitors were put in parallel in order to reach this desired high capacitance.

The input transformer is likely to be the most expensive part of the entire project. The input voltage to the regulator after it has been fully rectified must be at least several volts higher than the output voltage (12V) so that the regulator can maintain its output, in the process all the components have to be accounted for along with their drops at full load. The rectifier diodes for the Pegasus Producer has to be rate enough so that it’s capable of passing a very high peak forward current, typically 100amps or more because of the high capacitance in the circuit It has an Initial spike. The 7812 IC will only pass 1 amp or less of the output current, the remainder being fed by the outboard pass transistors. The circuit is designed to handle loads of up to 20 amps, and so five TIP2955 are wired in parallel to meet this demand. The dissipation in each power transistor is one sixth of the total load, just to minimize having to run components on their max load. Maximum load current will generate maximum dissipation, so a very decent size heat sink was placed in order to account for that required. Because of the design of the circuit in the event that the power transistors should fail, then the regulator would have to supply full load.

**4.3 Software**

The design of the software depended heavily on what features were decided that the Pegasus Producer would have. It was necessary first to make a variety of decisions in the features, specifications, dimensions, and microcontroller selection before start developing the software.

However, the design of the software can be, and was, performed since some functions will be needed in all, or some, of the cases. Taking into account that the software will deal with multiple elements, it is a good practice to divide the code into multiple functions that perform different duties.



**Figure 4.3-1:** *State diagram of the printer*

Before the software functions are analyzed it is necessary to understand what the functional requirements of the printer are. The best way to do this is to design a state diagram as the one presented on Figure 4.3-1. By doing so, it was possible to analyze what functions are required to reach to printing actions, and it gives a better perception of what is going on during the process. Often, some states are not included or some unnecessary ones are added. For this reason, the state diagram must be tracked through all the possible routes. Specifically for the project, some transitions to the diagram were added, such as the one from the idle state to the process data state.

When the printer is first turned on, it is in idle mode. In this mode the printer only waits until an SD card has been introduced, or a command is received through serial communication. If a card has been introduced, it will go to the process of reading data. As it will be discussed later on, not all the data can be read right ahead. Therefore, there will be multiple change of states between reading the data and processing the data. Once the end of the file has been reached, all the data has been read and the state of the printer goes to idle.

Once it reaches the process data state, the printer needs to translate the data into instructions to give to the axis’ motors, extruders’ motors, and temperature control mechanisms in order to print the object.

If there is any error in any of this states, or in the transition of this states, the printer will write detailed data about the error and will return to the idle state. However, it will not recognize the SD card, as it has not been introduced after the idle state has been reached. This means that in order to print multiple elements, the user has to remove and reintroduce the SD card between printing jobs.

**4.3.1 Reading from SD Card**

The G-Code that is derived from the CAD object is usually thousands of lines long. To provide some perspective, the example shown in the Figure 4.11.4.2.2-1. The G-Code for the robot CAD file is exactly one hundred fifteen thousand seventy two lines long.

Consequently, it is unlikely that all this code could be stored into the random access memory (RAM) that is provided for our processor. Thus, the data could be divided into smaller pieces called packets that will be read from the SD card as the packets have been processed and printed.

Because of the microprocessor selected, it was important to format the SD card that will be used into FAT16 format. So that it converts data into two bytes segments.

**4.3.1.1 Communication**

To retrieve data from the SD card, a library that communicates with the SD card is used. The library extracts one character at a time from the card. To communicate directly with the computer through the USB interface, serial communication is implemented. To achieve this it was necessary to establish a baudrate and use the Arduino Serial class to read and write commands. The communication could be divided into, at least, two different functions, write and read.

The difference be

**4.3.1.1.1 Writing Data**

To write data to the SD card the function will receive an array of characters as a parameter, and will send one by one to its destination. This function can be useful to write information back to the SD card for error reports and troubleshooting purposes.

On the other hand, to send data back to the computer, it is only necessary to use the print and print line functions from the Serial class. This was done accordingly, so that the desired output was achieved.

**4.3.1.1.2 Reading Data**

There were two different options of how to handle the incoming data. The first one reads the characters and stores them into an array of characters until it encounters an end of line. Then it returns the array, so that the G-Code interpreter handle the string as a whole. This might be inefficient because this can lead to read the comment lines that Cura automatically produces, and store them into the RAM.

The other option is to have two different functions. The first one will specifically read a character, while the other one will read a number. Having each one of them two different return values: a character and a float number.

The second method provides efficiency in terms of memory handling not only because it will not store the comments in memory, but because the numbers will not need to be stored twice, once as a string and once as a float. It will be responsibility of the G-Code interpreter function to use these two reading functions correctly for this method to work without problems.

What was done, was a combination of both of them. The printer reads and stores the commands character by character and line by line except when the comment signed is received –then it only reads until the end of the line. It does this process whenever there is space in the buffer for the commands. Once the information is stored, it reads each command stored character by character and transforms the necessary character sequences into floats.

**4.3.2 Controlling the Motors**

There were multiple ways to control the motors. It could be done as single function, requiring parameters that specify not only the length of the movement, in that specific axis, but in which axis the movement will be done.

An easier to read version would be to divide this into different functions. Thus, allowing each function to control the corresponding motors for the axis in which the movement must be performed. The speed of movement and direction were stablished in the reception, and preparation of the command, and are not needed as parameters.

**4.3.2.1 The Z-axis’ Motor**

This function controls the motors that are in charge of the z-axis movement. It does not need any parameters since everything needed is already saved in the command buffer. It calculates how long the motors must be moved in order to achieve the distance. This is a void function, since no feedback from this is needed.

**4.3.2.2 The X-axis’ motor**

Like the previous function this receives no parameters. It will calculate for how long the motor that is dedicated to this axis must be on to achieve this, and then it will actually turn it on. The difference with the previous function resides not only in the fact that it is controlling a different motor, but that the delay time actually changes because of the motor specification, lack of gravity and lack of threaded rod to improve precision.

**4.3.2.3 The Y-axis’ Motor**

This function calculates how many steps the motor must perform in order to achieve the distance, and, finally, it will turn the motor on for the calculated number of times.

It is important to point out that the number of steps is not significantly different for the same distance in the three functions previously mentioned.

**4.3.2.4 The Extruders’ Motors**

Since, unlike it was originally planned, the Pegasus Producer uses a single extruder head, the function takes into account the length of filament to be extruded by the hot end. Just as it was done previously, the information is taken from the buffer eliminating the need for parameters.

**4.3.3 Controlling the Temperature**

When printing, there must be a control of the temperature of the extruder. When the extruder is not printing, the temperature will drop to a specific temperature known as idle conversely, if temperature is not regulated for a long length of time it will over-melt the plastic within the extruder.

**4.3.3.1 Extruder’s temperature**

This function takes into account to what target temperature the extruder must be set on. Similarly to the movement functions, the information is available in a global variable, and it is obtained when the command is being interpreted. When the extruder is not being used, it goes to a stand by temperature, unless the print job has been finished in which case the extruder’s hot end will be completely turned off. When it is first turned on, it goes to a target temperature, by entering while loop, and staying there until it has reached a value greater or equal to the minimum value. In order to maintain the extruder within the desired temperature range, a controlling function is called periodically. It turns on, or off the extruder depending on where it is in relation to the range.

There is a need for a helper function here, which will determine the current temperature of the extruder by translating the number received from the thermistor with the use of thermistor tables.

**4.3.4 Configuration Information**

There are multiple values that are be constant, and they depended on the parts selection and chassis design of the Pegasus Producer. They are defined using C macro *#define*, so that the can be analyzed by the preprocessor. This information is used by most of the discussed functions. These are very important for the printer, and were only changed while performing the calibration process, which is when most of this values will be figured out. As a rough estimate, at the beginning the Pegasus Producer was using similar information that can be found online for the chassis that was selected. However, this drastically changde due to the difference in motors, and other aspects.

**4.3.4.1 Dimensions**

The dimensions are three float values that define the dimensions of the printer in terms of x, y, and z. The functions that control the motors for the different axis will use the corresponding value to make sure that the desired movement does not exceed the corresponding maximum value specified here, preventing possible damage to the motors and belts.

**4.3.4.2 Temperature**

There are three values needed for the extruder. They are the idle temperature, the temperature of operation, and the maximum temperature. All three of these are important to make sure that the printer acts the way it is supposed to.

**4.3.4.3 Origin**

This represents where in the coordinates the origin can be found, or where the extruders start and have to move to when the printing is finished. It is important to hold three variables that will be modified with each movement. They will represent the absolute positioning for the nozzle heads. Mainly, for the finalizer function. However, this became obsolete after the minstops where implemented.

**4.3.4.4 Minimum Step Length for Motors**

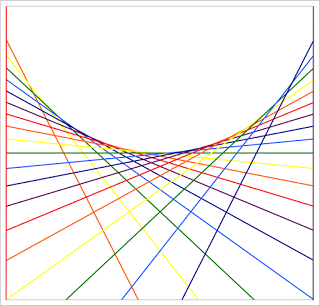
One value for each motor will be present. It represents the minimum step size that a motor can perform. The values stored will be an integer that represents the minimum length in millimeters that the nozzle heads will move in the corresponding direction, or the minimum length in millimeters of material that can be extruded.

**4.3.5 Arc Function**

Since there is a G-Code instruction that indicates an arc movement with constant radius, there needs to be a function that can interpret itand perform it.

The specific instruction is *G2 X90.6 Y13.8 I5 J10 E22.4*. The movement that describes it is a clockwise function from the current (xc, yc) point to the point (90.6, 13.8) while maintaining a constant distance from the point (5, 10). Extruding 22.4 mm of material in the process.

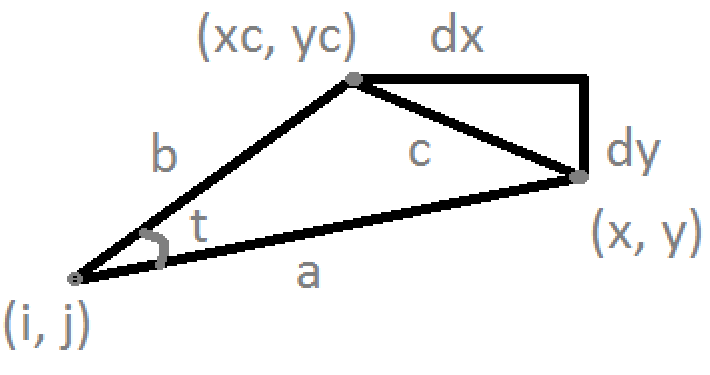
Considering the concept of tangents, which state that there is a single straight line that touches a circle in one point, and one point only. Then, it can be said that the circle can be drawn by an infinite number of tangent lines that will all touch different points of the circle, as can be appreciated in the Figure 4.3.5-1. From this image it can be concluded that the greater the number of lines, the sharper the circle is going to be.



**Figure 4.3.5-1:** *Tangents of a circle17. Permission granted by* *Professor Hubbard*

This concept will be used to interpret the arc instruction from G-Code. First of all, the length of the arc. In order to do this the distance between the current point (xc, yc) and the final point (x, y) has to be calculated through the Pythagorean Theorem. This distance (c) will then be. Once the distance is obtained, it is possible calculate the angle t using the law of cosines:

Finally, to calculate the arc length (l) the formula will be used. Where. Since both points share the same distance to the point (i, j) as stated in the documentation of the G-Code instruction.



**Figure 4.3.5-2:** *Values needed to calculate the arc length between the points (xc, yc) and (x, y).*

Once the arc length is calculated, the maximum number of straight lines that can be drawn will be calculated. This heavily depends on the minimum step in both the x and y axis. In order to calculate this, the formulas for the conversion between Cartesian coordinates, and polar coordinates must be taken into consideration.

Finally, the functions that control the movement both in x and y axis must be called to draw each of the straight lines that are printed.

**4.3.6 G-Code Interpreter**

What the G-Code interpreter will do is receive the first character of the G-Code instruction, this will tell the function what type of instruction it is so it can prepare to read an expected value using the read data function.

Because of the format of G-Code, the next value will always be an integer. This will tell the function what specific instruction is going to be received. Therefore, it can prepare itself to receive the remaining parameters by using the serial communication read data function.

The microcontroller will then check the temperature of the extruder used with the previously mentioned function before returning to the read from the SD function. Which will determine if the end of file was reached, and call the terminating function if it is the case.

**4.3.7 Initializer**

What the initializer function does is set up the position of the nozzle heads, and the temperature of the extruders and heatbed to the designated temperature. The extruders should be set to their respective idle temperature while the heatbed is set to its operating temperature. All this data is available in the configuration information. It should not change unless the materials used or the parts of the printer change. In which case, the code should be uploaded again to the microcontroller.

This function should be called as soon as the printer detects an object to be printed, so that all the elements are ready to start the process before being called. Thus, there must be a timeout at the end of this function to let the elements arrive to their expected values.

**4.3.8 Terminating Function**

This function, also referred to as the finilizer function, should be printed at the end of every printing process. What it should do is turn off the heating bed and the heater at the extruders, and return the nozzle heads to the origin position specified in the configuration information.

It will be called at the end of the function that reads from the SD card, and will return ultimately to the main function or the main loop.

**4.3.9 Main**

The main function is what will run first in any C program. What this specific function will do is an infinite loop, which will wait until it detects an SD card. It will then proceed to check that it is correctly formatted. If it is, it will call the initializer function to prepare the printer. This will start the printing process. Once it is finished, the control will come back to the main function which will wait one minute until it starts the loop again.

The timeout is of one minute to give the user enough time to remove the SD card, once the printing job is finished. If it is not removed, the printer will detect the SD card and print the object again.

For simplicity purposes. This function will look for a file called PEGASUS.txt. This way, it is not necessary to send parameters in a text file.

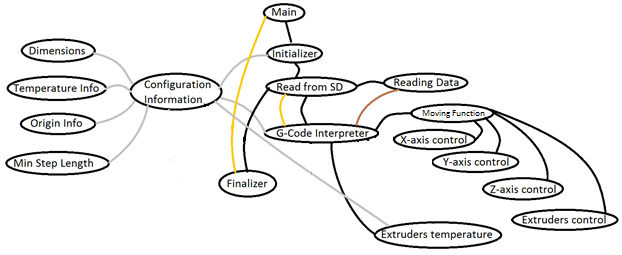
**4.3.10 Software flow diagram**

The in Figure 4.3.10-1 describes the flow path that the code follows in order to print a CAD object. The gray lines represent the configuration information, and what functions use them. The black lines represent a flow from the top element to the bottom one. The brown line represents the G-Code Interpreter calling the Reading Data function. Finally, the yellow lines represent the returning of the method the calling function.

What stands out the most is that the finalizer (terminating) returns directly to main while the G-Code Interpreter return to Read from SD, which was the function that actually called it. This is because Read from SD may call G-Code Interpreter again, while Finalizer is only called at the end of Read from SD, which in turn is only called at the end of Initializer.

It is important to point out that if there occurs an error in any of the functions, they will use the write data function to write an error report to the file “errors.txt” on the SD card. Allowing the user to identify what the error was when printing, and perform the necessary adjustments (if possible).

The fact that the G-Code interpreter calls the moving function stands out. This is not really happening since the moving function is activated through a timer, it checks if there are any commands available and processed in the buffer, and then it proceeds to call the corresponding control function for each axis. The arc function is already called if necessary through the G-Code interpreter if necessary.

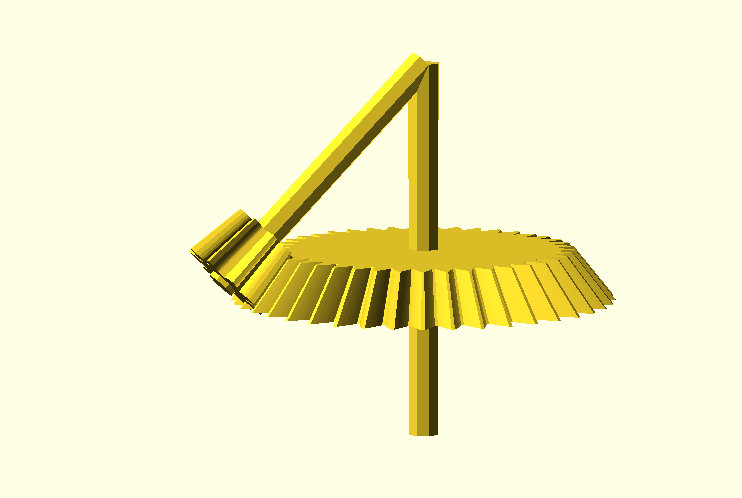
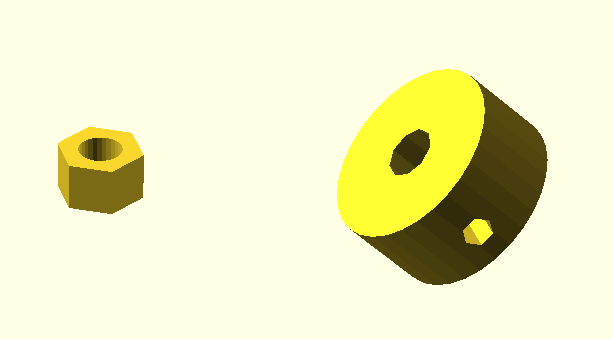


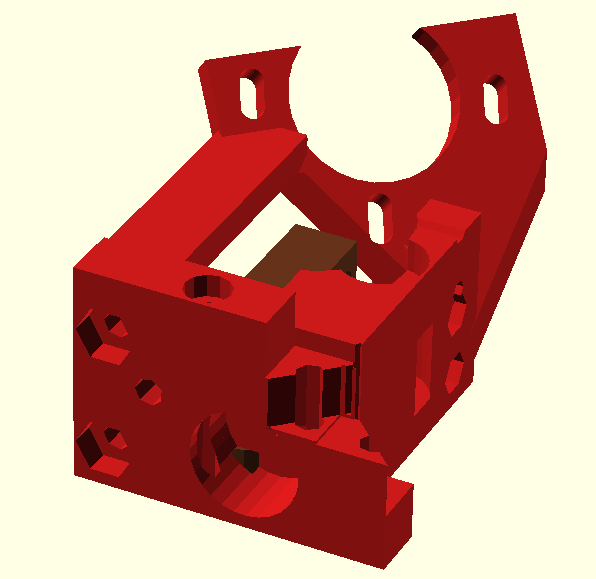
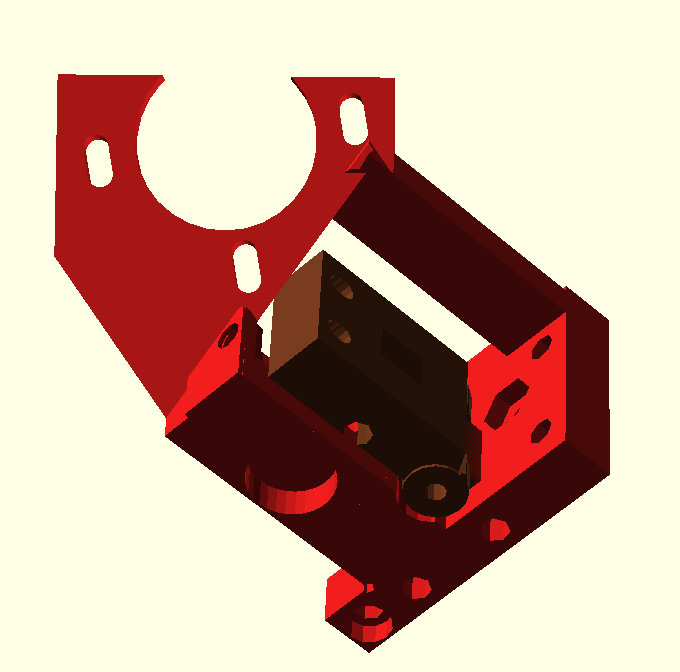
**Figure 4.3.10-1:** *Flow diagram of the software*

**4.4 Extruder**

What really differentiates a 3D printer from a CNC machine is the extruder, the extruder is what determines each 3D printers’ speed, printing material, and personality. An extruder consists of two main parts, the cold end and hot end. The cold end is in charge of holding the motors, connecting to the axis, and pushing the plastic filament into the hot end. The hot end contains the nozzle and contains the heating element required to melt the plastic for extrusion. Due to the availability of 3D printers many reference design are available on the internet and many of them can be printed from another 3D printer.

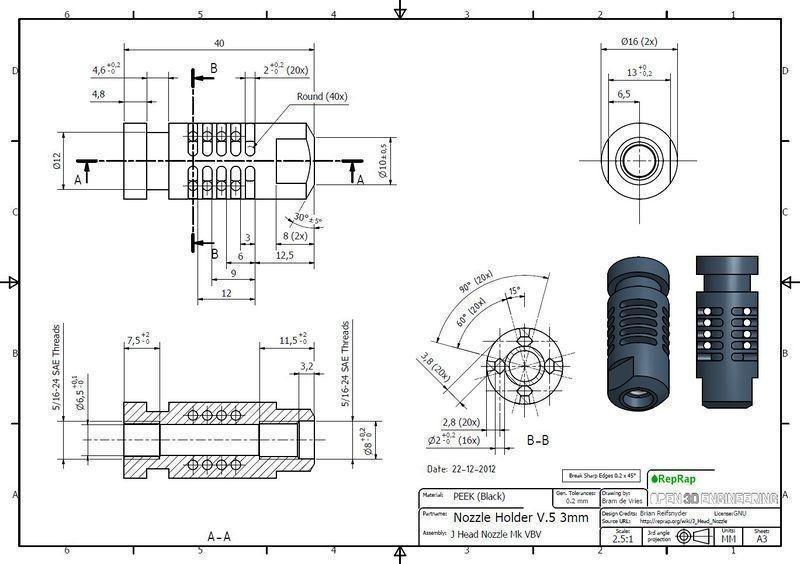
The single extruder for the Pegasus Producer will be mostly 3D printed from the UCF 3D printer available in the innovation lab. The cold end, cogs, and wheels can entirely be 3D printed. The socket head bolts, nozzle, and heating element must be purchased. The hot end will consist of a 1.75mm J-Head MK-V Hot End 0.4mm Nozzle. Everything will be put together using various M4 and M3 size socket head and screws. Most extruders follow the same basic steps for construction so design discussion will be limited to the designed 3D parts which will be printed and specifications on the chosen nozzle hot end. The cold end will be 3D printed from the following designs in figure 4.4-1.

**Figure 5.1-1:** *Entire Cold End Assembly*

The Hot end will be purchased and will be a J-Head MK-V Hot End 0.4mm Nozzle, it is based off an open sourced rep rap design. A hot end is a precision instruments that cannot be manufactured with household tools. The hot end that will be used is pictured below in figure 5.1-2.

**Figure 5.1-2:** *J-Head Nozzle*

**5 Project Testing and Prototyping**

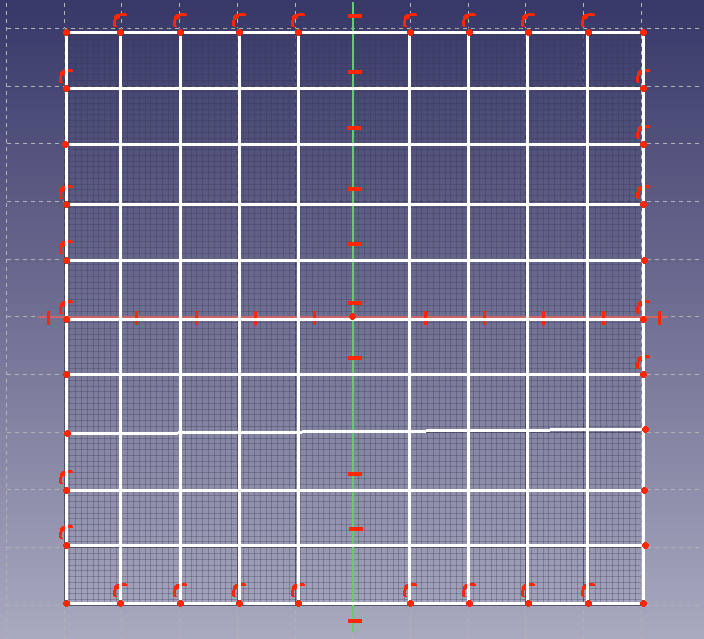
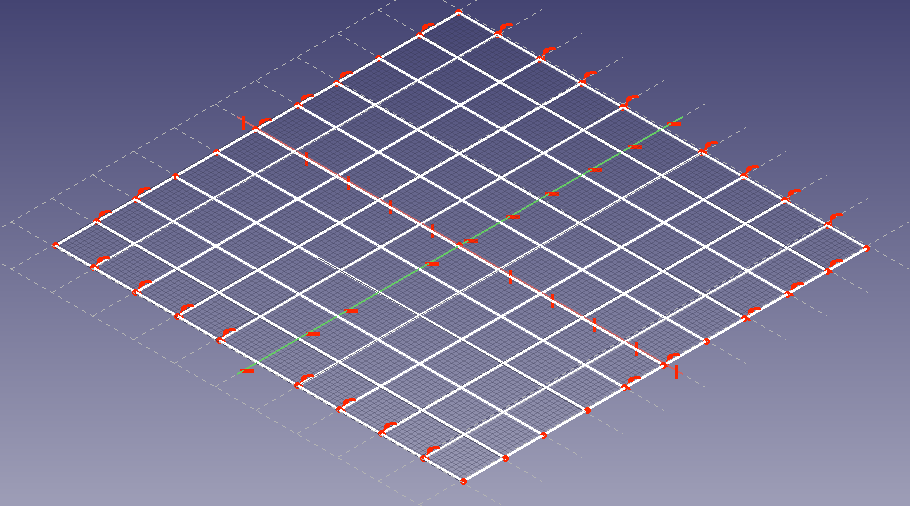
The testing will be divided into two different categories. The first one will perform the same procedures on each of the axis on the x-y plane; while the second one will perform procedures specific for the z axis.

**5.1 X and Y Axis Calibration**

For the XY plane, the first procedure will be to print a straight line on each of the limits of the axis. Therefore, two straight line will be print at x=0, and x=X\_MAX\_VAL. Printing two straight lines along the y-axis. The same will be done to test the remaining axis.

This procedure evaluates the chassis limitations. It checks if there is any problems when reaching any of the extreme spots in the plane. If so, the software should be modified to allow a smaller printing space, which was the case.

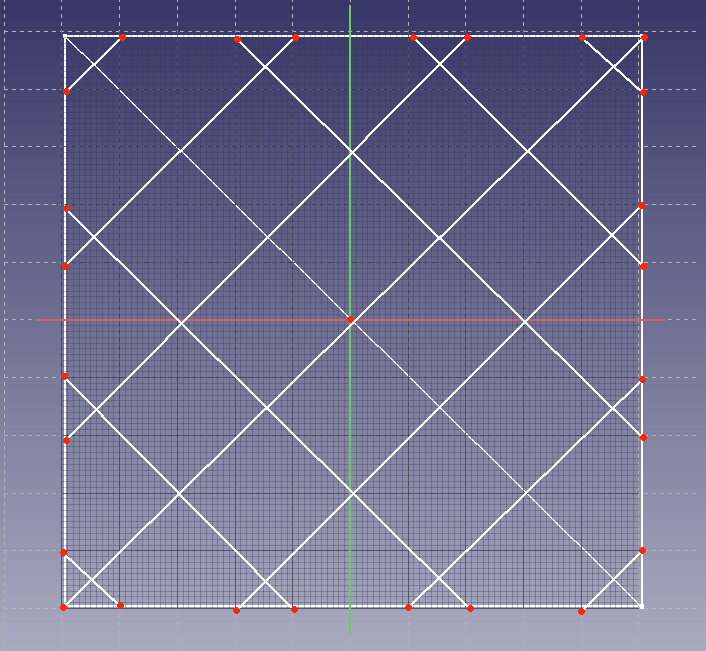
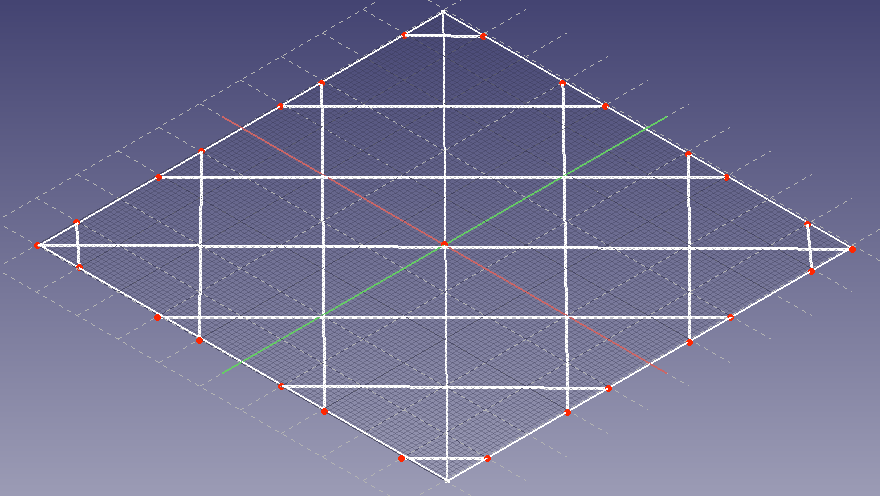
The second procedure consists on printing parallel straight lines along each of the axis with 3 cm of separation between one and other. This will test that no there no errors in printing straight lines.

**Figure 5.1-1:** *Top view of the* **Figure 5.1-2:** *3D view of the* *combination of the first two test* *of the first two test procedures*

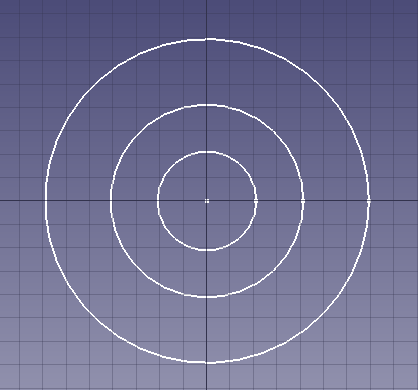
*procedures*

In the third procedure, parallel diagonal lines will be printed. First in one direction, and then in the other. This will ensure that the method that is used to print lines that are not along the axis are done so correctly by the software, and to check the number of steps for each motor is measured correctly.

**Figure 5.1-3:** *Top view of the* **Figure 5.1-4:** *3D view of the diagonals diagonals combined*

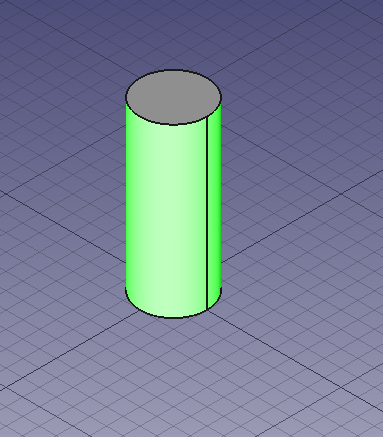
Finally, the last procedure consisted of printing circles of different diameter on the XY plane. This will ensure that the arc function is working properly, and providing the best detail possible. It is necessary to keep write down the radius of the circles, so that the can later be measured. Note that by printing a circle with a small radius we can test the limitations of this function.



**Figure 5.1-5:** *Top view of the printed circles procedure*.

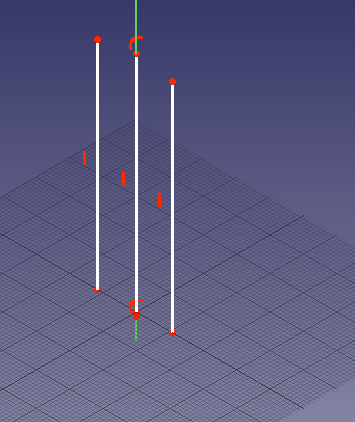
**5.2 Z Axis Calibration**

There were three tests for the Z axis. The first one printed a cylinder with the maximum height permitted to confirm that the Pegasus Producer is able to reach the height. It was necessary to make the diameter of the circle long enough so that the cylinder does not lose stability and fall down before reaching the height. Even though that the fact that the printer tape hold the object in place



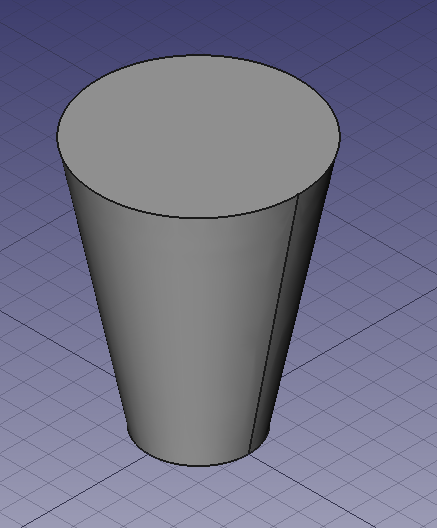
**Figure 5.2-1:** The cylinder that will be printed in order to test the height capabilities of the printer

The second one will consist of printing a line in the z axis. The g-code will tell the printer to keep going until the maximum height, but for obvious reasons the structure will not be stable and will probably fall down before reaching it. The purpose of this specific test is to know at what specific height the line will fall. This will be a factor to take into account when dealing with small details like eyelashes.



**Figure 5.2-2:** *Lines on the z-axis*

The final test consisted in an inverted cone. However, the base is be a circle. This structure will test if there is any problem in terms of stability derived from the printer itself. In other words, it will test if the Pegasus Producer is able to print a relatively unstable structure. Moreover, we wanted to test how much the movement affected the printing precision and stability. The painters tape helped the printer to pass this test successfully.



**Figure 5.2-3:** *Inverted cone*

**5.3 Extruder Calibration**

The extruder is where the mechanical meets the software, when both are not in sync the 3D printer will not function correctly. Without calibration the flow of plastic out of the extruder will be disrupted and prints will come out incorrectly. Firstly, it is important to understand the following three concepts, thread width depends on three things, the distance from the nozzle to the heated bed, the speed of travel, and the amount of plastic being extruded. Less distance from the nozzle to the bed creates a wider thread due to the plastic being squished. A faster moving extruder will cause less plastic to be dispensed at a single point making a thinner thread and conversely, high plastic flow rate will provide a wider thread. Balancing the right amount layer height, feed rate, and flow rate is how an extruder is calibrated, this requires the use of software and external tools. Tools and knowledge required for extruder calibration include a measuring device. Such as a Vernier caliper, able to precisely measure 100mm and .5mm lengths and widths, knowledge of stepper motors’ full steps per revolution value, knowledge of the stepper driver micro-steps setting, the number of teeth on a pulley, teeth on extruder gear, and belt pitch. Software such as Slic3r can determine flow rate and adjust it accordingly by giving it appropriate values such as filament diameter, nozzle size, and speed the filament is being extruded. Taking everything mentioned above into account one can begin proper testing and calibrating of the extruder.

The process for testing whether extruder needs calibration is quite simple and should take no longer than 30 minutes. Prior to starting one would need the tools mentioned above and access to the extruder. Using software such as Repetier Host, software for manual control of the extruder, set the hot end to begin heating up to melting temperature. The second step is to locate the filament just above the extruder and mark a portion of 100mm with a marker. Using Repetier Host manually extrude 1mm of filament until the bottom mark on the filament is level with the extruder. The next step is to set extrude length to 100mm on Repetier Host and click to extrude, a calibrated extruder would extrude until the second mark on the filament is level with the extruder, if this fails the extruder needs calibration.

The process of calibration is continued if the marked part of the filament is not level with the top of the extruder, at this point a second mark must be put where it actually landed and the difference in length must be measured. Going back to software, on Repetier Host one must access the EEPROM settings and alter the steps per mm. Creating a mathematical ratio from the expected extrusion length of 100mm to the result received one can calculate the new steps requires.

***(Length Expected)x(Steps)=(Length expected +/- Measured Distance)x(New Steps)***

Knowing the New Steps required one can input the number of steps into the software and save it onto the ATmega2560’s EEPROM memory. Other methods of testing can be used as well to test stability and to evaluate performance of the extruder. One method is simply printing a dot after dot on top of each other without any structural aid, a well calibrated extruder can achieve a respectable height.

**5.4 Project Operation**

This section is dedictaed to provide the user with a concise instruction manual on how to use the Pegasus Producer. The steps are outlined below:

* Choose one of the following two options to generate a stl file to be printed
  + Design the object using FreeCAD (open source) or a similar software that allows the user to design 3D models, and export to stl format
  + Download 3D models designed by third parties. They are available in websites such as thingiverse
* Choose Cura or Slic3r to import the object(s), place them where the user want to print the object(s) in the printer, scale, and/or rotate them. Then proceed to generate the G-Code by following the next steps
  + If the chosen software was Cura, click on ‘File’ and then proceed to click on ‘Save G-Code’. This will generate a file in the selected location with the name assigned by the user.
  + If the chosen software was Slic3r, click on the ‘Slice’ button. This will generate the G-Code in the text area in the software
* Create a text file called ‘PEGASUS.txt’
* Copy the generated G-Code file and paste it into the newly created text file.
* Insert the SD card into the computer.
* Copy the ‘PEGASUS.txt’ file into the SD card. If another file exists, replace it.
* Eject the SD card from the computer
* Insert the card into the SD slot of the printer
* Turn on the printer
* Wait between twenty seconds and a minute for the printer to move to the origin. If this does not happen, either the file was not found or there was an error. In which case the ‘error.txt’ file has to be checked for a report
* Wait until the printer finishes the job. It will move all the axis to the origin when this has been done. Notice that the hot end of the nozzle will cooldown to at least 100 °C before the axis can be moved for safety reasons.
* Remember to turn off the computer.

**6 Administrative Content**

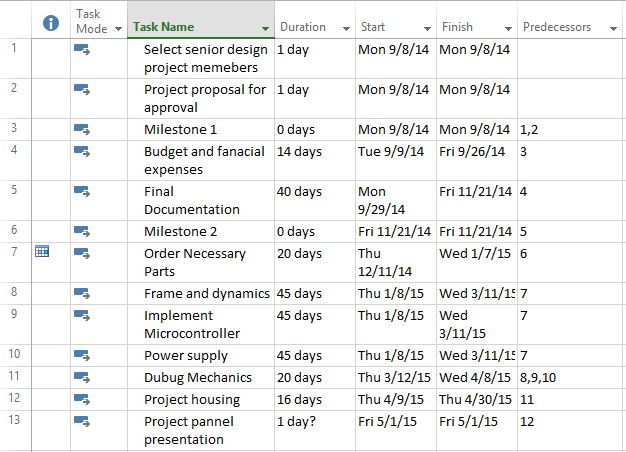
**6.1 Milestones**

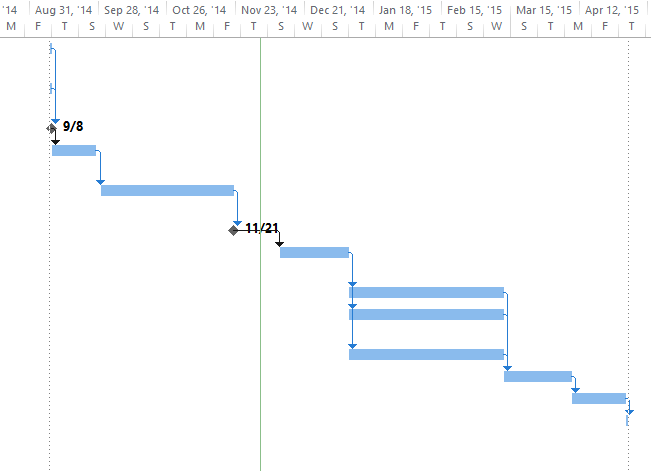
To guarantee that the project is finalized and completed in time, the milestones was created as to how each member of the team though would be realistic dates to complete each of the prototyping phases that were established. In here, not only the tasks identified and listed, but the tentative dates were also assigned to each task. In each of the phases, there will be a great amount of research, design, construction and testing. In this section of the milestones the projected completion goals an schedules are defined in detail for both Senior Design 1 and Senior Design 2.

The initial stage of research was the longest and most time consuming, as it ran the entire length of the semester. In this particular initial phase, a good suggestion was to look into different designs and implementations with the purpose of sparking innovations and grasping a better understanding of the final results.

**6.1.1 Senior Design 1**

Below is a list of milestones and dates that we came up with at the beginning of Senior Design 1





**6.1.2 Senior Design 2**

In order to design and build the Pegasus Producer the tasks were grouped for different areas of expertise. These being: power supply, microcontroller and control board, and software.

**6.2 Budget and Finance**

Two of the main goals for the Pegasus Producer is to build a 3D Printer that is low cost and efficient. All the components have been thoroughly researched and picked out based on their price, availability, and performance. Due to the extensive list all components have been split up between subsystems and type. Budget has remained within an acceptable amount of the initial cost analysis.

Tables 6.2-1, 6.2-2, 6.2-3 display all of the components for the power supply subsystem, table 6.2-4 displays the total cost. The PCB for the power supply will be chemically etched with spare copper boards and will therefore not be included.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Part** | **Manufacturers** | **Part Number** | | **Quantity** | **Unit Price** | **Total Price** |
| Cboot1 | TDK | C1005X5R1A104K | | 1 | $0.01 | $0.01 |
| Cboot2 | TDK | C1005X5R1A104K | | 1 | $0.01 | $0.01 |
| Ccomp1 | Yageo America | CC0805KRX7R9BB102 | | 2 | $0.01 | $0.02 |
| Ccomp2 | Yageo America | CC0805JRNP09BN101 | | 2 | $0.01 | $0.02 |
| Cin | Panasonic | 35SVPF82M | | 1 | $0.61 | $0.61 |
| Cinx | Kemet | C0805C104K5RACTU | | 2 | $0.01 | $0.02 |
| Cout1 | Panasonic | 16SVP180M | | 2 | $0.29 | $0.58 |
| Cramp1 | Yageo America | CC0805KRX7R9BB821 | | 1 | $0.01 | $0.01 |
| Cramp2 | Yageo America | CC0805KRX7R9BB821 | | 1 | $0.01 | $0.01 |
| Cres | MuRata | GRM155C80G474KE01D | | 2 | $0.01 | $0.02 |
| Css1 | Yageo America | CC0805KRX7R9BB153 | | 1 | $0.01 | $0.01 |
| Css2 | Yageo America | CC0805KRX7R9BB153 | | 1 | $0.01 | $0.01 |
| Cvcc1 | MuRata | GRM155R61A474KE15D | | 1 | $0.01 | $0.01 |
| Cvcc2 | MuRata | GRM155R61A474KE15D | | 1 | $0.01 | $0.01 |
| Cboot | AVX | 08053C104KAT2A |  | 1 | $0.01 | $0.01 |
| Cin | MuRata | GRM31CR71H475KA12L | | 3 | $0.10 | $0.30 |
|  |  |  | |  | **Total:** | $1.66 |

**Table** *6.2-1.* **Power Supply- Capacitors**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Part** | **Manufacturers** | **Part Number** | **Quantity** | **Unit Price** | **Total Price** |
| Rcomp1 | Vishay-Dale | CRCW040240K2FKED | 1 | $0.01 | $0.01 |
| Rfb1 | Vishay-Dale | CRCW04021K00FKED | 1 | $0.01 | $0.01 |
| Rfb2 | Vishay-Dale | CRCW040214K0FKED | 1 | $0.01 | $0.01 |
| Rramp1 | Vishay-Dale | CRCW040268K1FKED | 1 | $0.01 | $0.01 |
| Rramp2 | Vishay-Dale | CRCW040268K1FKED | 1 | $0.01 | $0.01 |
| Rsense1 | Susumu Co Ltd | PRL1632-R008-F-T1 | 1 | $0.19 | $0.19 |
| Rsense2 | Susumu Co Ltd | PRL1632-R008-F-T1 | 1 | $0.19 | $0.19 |
| Rt | Vishay-Dale | CRCW040215K4FKED | 1 | $0.01 | $0.01 |
| Ruv1 | Vishay-Dale | CRCW040254K9FKED | 1 | $0.01 | $0.01 |
| Ruv2 | Vishay-Dale | CRCW04026K34FKED | 1 | $0.01 | $0.01 |
|  |  |  |  | **Total** | $0.46 |

**Table** *6.2-2.* **Power Supply- Resistors**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Part** | **Manufacturers** | **Part Number** | **Quantity** | **Unit Price** | **Total Price** |
| D1 | Diodes Inc. | SDM10U45-7-F | 1 | $0.06 | $0.06 |
| D2 | Diodes Inc. | SDM10U45-7-F | 1 | $0.06 | $0.06 |
| L1 | Coilcraft | XAL8080-682MEB | 1 | $1.05 | $1.05 |
| L2 | Coilcraft | XAL8080-682MEB | 1 | $1.05 | $1.05 |
| M1 | Texas Instruments | CSD18504Q5A | 1 | $0.56 | $0.56 |
| M2 | Infineon Technologies | BSZ040N04LS G | 1 | $0.44 | $0.44 |
| M3 | Texas Instruments | CSD18504Q5A | 1 | $0.56 | $0.56 |
| M4 | Infineon Technologies | BSZ040N04LS G | 1 | $0.44 | $0.44 |
| U1 | Texas Instruments | LM25119PSQ/NOPB | 1 | $2.60 | $2.60 |
| Dvcc | ON Semiconductor | MBR0520LT1G | 1 | $0.06 | $0.06 |
| U1 | Texas Instruments | LMR14006YDDCR | 1 | $1.10 | $1.10 |
| N/A | NTE Electronics | NTE5322 | 1 | $5.49 | $5.49 |
| N/A | Fulham Pony | Electronic Transformer | 1 | $105.40 | $105.40 |
|  |  |  |  | **Total** | $118.87 |

**Table** *6.2-3.* **Power Supply- Miscellaneous Components**

|  |  |
| --- | --- |
| **Category** | **Total** |
| Capacitors | $1.66 |
| Resistors | $0.46 |
| Rest | $118.87 |
| **Total** | $120.99 |

**Table** *6.2-4.* **Power Supply- Total Cost**

The electronics subsystem of the Pegasus Producer is more expensive that thought, this portion includes both the Atmel2560 microcontroller and the entire Control Board assembly. Precise components had to be chosen for the electronics to work correctly and in many cases cheaper components could not be found. Pricing may change once official quotes from the third party PCB manufactured are attained, currently PCB pricing is based off estimates found on the manufacture’s website.

Most of the components will be purchased from Digi key with an exception of a few which will be bought from eBay. Wiring pricing is not included in this section due to salvageable wiring available from other electronics.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Part** | **Manufacturers** | **Part Number** | **Quantity** | **Unit Price** | **Total Price** |
| ATMEGA 2560 | ATMEL | ATMEGA2560-16AU | 1 | $16.72 | $16.72 |
| ARDUINO MEGA (Prototyping) | ARDUINO | ARDUINO | 1 | $16.00 | $16.00 |
| PCB MICROCONTROLLER | OSH PARK | N/A | 1 | $30.00 | $30.00 |
| PCB CONTROL BOARD | OSH PARK | N/A | 1 | $30.00 | $30.00 |
| ASSEMBLED J-HEAD HOT END NOZZLE | N/A | N/A | 1 | $24.48 | $24.48 |
| MK2B DUAL POWER PCB HEATBED | N/A | N/A | 1 | $12.53 | $12.53 |
| NEMA 17 STEPPER MOTORS\* | NATIONAL INSTRUMENTS | 780067-01 | 5 | $0.00 | $0.00 |
|  |  |  |  | **Total** | $129.73 |

**Table** *6.2-5.* **Microcontroller**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Part** | **Manufacturers** | **Part Number** | | **Quantity** | **Unit Price** | **Total Price** |
| CAP .1UF 100V 20% | KEMET | | 399-4265-ND | 1 | $0.31 | $0.31 |
| CAP ALUM 10UF 35V ELECT RAD | KEMET | | 338-1667-ND | 2 | $0.36 | $0.72 |
| CAP 100UF 10V ALUM LYTIC RADIAL | PANASONIC ELECTRIC | | P5123-ND | 1 | $0.20 | $0.20 |
| RES 4.7K OHM 1/8W 5% CF AXIAL | STACKPOLE ELECTRONICS | | CF18JT4K70CT-ND | 2 | $0.18 | $0.36 |
| RES 100K OHM 1/4W 5% CARBON FILM | STACKPOLE ELECTRONICS | | CF14JT100KCT-ND | 7 | $0.56 | $3.92 |
| MOSFET N-CH 60V 55A TO-220 | STMICROELECTRONICS | | STP55NF06L | 3 | $4.44 | $13.32 |
| DIODE GEN PURPOSE 400V 1A DO41 | FAIRCHILD SEMICONDUCTOR | | 1N4004FSCT-ND | 1 | $0.34 | $0.34 |
| FUSE PTC RESETTABLE 5A HOLD | BOURNS INC | | MF-R500-ND | 1 | $0.76 | $0.76 |
| TERM BLOCK 5.08 2POS VERT | MOLEX | | WM4393-ND | 1 | $0.85 | $0.85 |
| LED SS 3MM 568NM GRN DIFF | KINGBRIGHT | | 754-1244-ND | 1 | $0.10 | $0.10 |
| SWITCH TACT RA H=6.35MM | TE CONNECTIVITY | | 450-1177-ND | 1 | $0.21 | $0.21 |
| CONN HEADER RT ANG 2POS 5.08MM | PHOENIX CONTACT | | 277-1106-ND | 1 | $0.47 | $0.47 |
| CONN TERM BLOCK PLUG 2POS 5.08MM | PHOENIX CONTACT | | 277-1011-ND | 1 | $1.47 | $1.47 |
| CONN HEADER 50POS .100" SNGL TIN | SAMTEC | | SAM1035-50-ND | 3 | $6.12 | $18.36 |
| CONN HEADER FMAL 36PS.1" DL GOLD | SULLINS CONNECTOR | | S7121-ND | 1 | $2.87 | $2.87 |
| CONN HEADER FEM 8POS .1" SGL GLD | SULLINS CONNECTOR | | S4108-ND | 5 | $3.05 | $15.25 |
| CONN HEADER FEMALE 6POS .1" | SULLINS CONNECTOR | | S7039-ND | 1 | $0.61 | $0.61 |
| CONN HEADER FMALE 16POS .1" GOLD | SULLINS CONNECTOR | | S7049-ND | 4 | $5.12 | $20.48 |
| SHUNT JUMPER .1" BLACK GOLD | 3M | | 3M9580-ND | 12 | $0.61 | $7.32 |
|  |  | |  |  | **Total** | $87.92 |

**Table** 6.2-6. **Control Board**

|  |  |
| --- | --- |
| **Category** | **Total** |
| Control Board | $87.92 |
| Microcontroller | $129.73 |
| **Total** | $217.64 |

**Table** *6.2-7.* **Electronics-Total Cost**

The more mechanical subsystem for the Pegasus Producer comes from the chassis, it has been decided that the chassis will consist entirely of laser cut acrylic and metal bolts and bearings to hold it together, the majority will be 3D printer for free. If for any reason the chassis cannot be built with on hand parts, a complete chassis build has been source online. Figure 6.2-7 shows all the components for the chassis.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Part** | **Manufacturers** | **Part Number** | **Quantity** | **Unit Price** | **Total Price** |
| Arcrylic frome kit | Sintron | 16282 | 1 | $14.00 | $14.00 |
| Smooth and threaded kit | Sintron | 16282 | 1 | $26.00 | $26.00 |
| Printed plastic kit , made of PLA | Sintron | 16282 | 1 | $10.00 | $10.00 |
| Pcs of GT2 pulley + GT2 2m belt | Sintron | 16282 | 1 | $15.00 | $15.00 |
| Pcs LM8UU bearings | Sintron | 16282 | 1 | $25.00 | $25.00 |
| Pcs MF1045 bearings | Sintron | 16282 | 1 | $30.00 | $30.00 |
| Pcs coupling | Sintron | 16282 | 1 | $20.00 | $20.00 |
|  |  |  |  | **Total** | $140.00 |

**Table** *6.2-7.* **Chassis**

**7 SOURCES**

1. "Hot End Design on 3D Printer Extruder." Airtrippers 3D Printer and Arduino Blog. N.p., n.d. Web. 03 Dec. 2014.
2. "3D Printering: Electronics Boards." Hackaday. N.p., n.d. Web. 02 Dec. 2014.
3. "Materials - 3D Printing Industry." 3D Printing Industry Materials Category. N.p., n.d. Web. 21 Nov. 2014.
4. "Www.3ders.org." 3ders.org. N.p., n.d. Web. 14 Nov. 2014.
5. "Reprap Development and Further Adventures in DIY 3D Printing." : Universal Paste Extruder. N.p., n.d. Web. 17 Nov. 2014.
6. "Trials and Tribulations of Making a 3D Printer." Let's Make Robots! N.p., n.d. Web. 03 Nov. 2014.
7. "Start 3D Printing." Start 3D Printing. N.p., n.d. Web. 12 Nov. 2014.
8. *Print the Legend*. Dir. Luis Lopez and J. Clay Tweel. Netflix, 2014. Netflix.
9. Russon, Mary-Ann. "3D Printers Could Be Banned by 2016 for Bioprinting Human Organs." *International Business Times RSS*. N.p., 29 Jan. 2014. Web. 28 Nov. 2014.
10. Mack, Eric. "Here's the 3D-printing Institute in Obama's State of the Union."*CNET*. CNET, 12 Feb. 2013. Web. 26 Nov. 2014.
11. Presto, Suzanne. "Space, the Final Frontier for 3-D Printing." *CNN*. Cable News Network, 28 Nov. 2014. Web. 26 Nov. 2014.
12. Molitch, Michael. "11 Food 3D Printers from the Future - 3D Printing Industry." *3D Printing Industry*. 3D Printing Industry, 9 Nov. 2014. Web. 27 Nov. 2014.
13. Sher, Davide. "UNYQ New Camouflage for Troops." *3D Printing Industry*. 3D Printing Industry, 17 Nov. 2014. Web. 26 Nov. 2014.
14. Chavez, Evan. "Oreo Can Play Again with His 3D Printed Patella." *3D Printing Industry*. 3D Printing Industry, 15 Nov. 2014. Web. 27 Nov. 2014.
15. Molitch, Michael. "Man’s Livelihood Restored by EU’s Approval of 3D Printed Cranial Implant." *3D Printing Industry*. 3D Printing Industry, 12 Nov. 2014. Web. 25 Nov. 2014.
16. Kelsey. "Best 3D Printing Articles of 2013 | Solid Concepts Blog." *Solid Concepts Blog*. Stratasys Direct Manufacturing, 30 Dec. 2013. Web. 28 Nov. 2014.
17. Prof. Hubbard. "Tangents of the Lower Half of a Circle." *Math Year 2013*. Blogspot, 27 July 2013. Web. 26 Nov. 2014. <<http://mathyear2013.blogspot.com/2013/07/tangents-of-lower-half-of-circle.html>>.
18. "What Is Python? Executive Summary." *Python.org*. Python Software Foundation, n.d. Web. 1 Dec. 2014.
19. "Multiple Winding Transformer and Multicoil Transformers." *Basic Electronics Tutorials*. N.p., n.d. Web. 04 Dec. 2014.

**8 COPYRIGHT PERMISSIONS**

