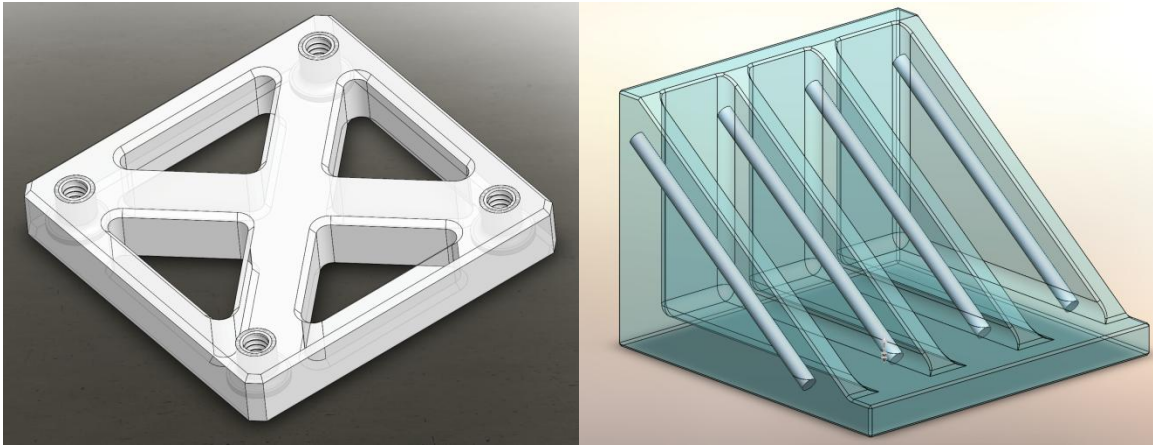


10/2/2015

ADD_IN

Conceptual Design Review



TEAM F

- Nikhil Baheti
- Daniel Berman
- Ihsane Debbache
- Astha Prasad

Contents

1. Project Description

2. Use Case

3. System Level Requirements

3.1 Software

3.2 Hardware

4. Functional Architecture

4.1 Slicing Software

4.2 Printer Interface Software

4.3 Printer (Electromechanical System)

5. System Level Trade Study

6. Cyberphysical Architecture

6.1 Makergear M2

6.2 Slicer Software

7. Subsystem Description

7.1 Slicer Subsystem

7.2 Microcontroller Subsystem

7.3 Joint Subsystem

7.4 Rotary Stage Subsystem

7.5 Nozzle Subsystem

8. Project Management

8.1 Work Plan and Tasks

8.2 Schedule and Milestones

8.3 System Validation Experiments

8.4 Team Member Responsibilities

8.5 Part List and Budget

8.6 Risk Management

9. References

Appendix

1. Project Description

The ADD_IN team aims to develop a 3D printer technology (named 4DOF) capable of producing printed parts which incorporate commercial off the shelf (COTS) components. These COTS items can include components such as threaded inserts, structural reinforcement (stiffeners), and electronics. Incorporating these components enables inclusion of geometrically precise features (such as threads), production of stronger parts (via the use of stiffeners), and reduced overall production time (since post-printing operations such as tapping and gluing can be eliminated).

The parts to be incorporated will be generally cylindrical or rectangular prism in shape, can optionally extend up to 1" above the top layer of the part, and must have sides which are orthogonal to the print surface. The 3D print material will be precisely deposited along the perimeter of the COTS item, thus encapsulating it within the 3D printed part.

Installation of COTS parts during the printing process will be performed by a human operator. When the 3D printer reaches the print layer on which a part is to be installed, it will 'pause', move to a safe and accessible configuration for COTS item insertion, and await a command from the user that the part has been successfully installed before resuming printing. The system will be able to produce printed parts incorporating multiple COTS items at varying locations and orientations, provided that the orientations and clearances between the parts do not exceed certain geometry restrictions.

2. Use Case

An RC plane engineering company is in a race to develop a new fixed-wing airplane. The general configuration of the airplane has been determined but continued iteration on the structure and aerodynamic surfaces is needed. Up until this point the airplane has been prototyped using individually cut pieces of foam, balsa wood, and aluminum. Even with the aid of a laser cutter and CNC hot-wire foam cutter this has been a time consuming process since many of the parts are made from vastly different materials and are difficult to join together. Because of this the engineering team is often using glues and composites to hold the parts together, which makes it near impossible to replace or modify parts thus making design iteration difficult.

As with their other fixed wing airplanes, and as is an industry norm, the company eventually aims to mass produce the design from aluminum and polystyrene. Aluminum provides the structural rigidity necessary to form a solid airframe, especially for enduring the high stresses present at the motor mounts. Polystyrene is a light-weight plastic which can be easily molded into the complex shapes necessary to produce the aerodynamic surfaces of the aircraft. Furthermore, polystyrene can be injection molded to contain thread inserts to provide



Figure 2.1: Injection molded part with threaded insert.
(<http://precision.bc.ca/photos/>)

attachment points to the airframe and other internal hardware. This however, is only cost effective for large production runs.

The design team has previously tried using 3D printers to more rapidly produce and test structural components. While helpful in some instances, the team has generally found that 3D printed parts are not sufficiently strong for constructing load bearing airframe components, and prone to mechanical failure at the interfaces between components. Efforts to work around this problem have been made, but usually involve designing a special ‘thickened’ version of the part for 3D printing, which must then be redesigned when transitioning to large volume manufacturing using polystyrene. This extra design step uses up valuable team member time and upsets the weight, balance, and aeroelastic properties of the aircraft. The team however has remarked that 3D printing’s ability to easily incorporate more complex features, such as wire runs, electronics mounts, and access ports into their project is a great benefit.

Recently, as part of an experiment to improve their design process, the aircraft design team purchased an ADD_IN 4DOF 3D printer. This printer contains an extra mechanical degree of freedom and a special slicing software to enable 3D printing of complex parts that include embedded COTS items. The cost of the printer was comparable to other FDM 3D printers, and thus was not a significant burden to the small company. Since the printer looked and operated in nearly the same manner as their other 3D printers, the employees were immediately able to start using the printer without any additional training.



Figure 2.2 Example of failed 3D printed motor mount and a less desirable ‘thickened’ replacement necessary to support the loads.
(<http://www.wattflyer.com>)

The first project the 4DOF printer was applied to was development of the aircraft’s motor mount. The motor mount is a particularly challenging component since it must be strong enough to withstand large forces and moments imparted by the motor, propeller, and occasional crash landings, but also be light weight since it is the most forward part of the aircraft where weight is a premium. In these regards 3D printed parts had previously failed due to their poor material properties, especially with the motor mounting bolts loosening, stripping and pulling out from the plastic part. Using the ADD_IN printer the team was able to produce a 3D printed part directly from the airplane’s CAD design, and include threaded inserts which could securely attach the motor and distribute the forces within the motor mount thus preventing material failure. The new motor mount was produced in half the time and with fewer resources than the previous aluminum-balsa wood-glue version, and thus enabled the team to conduct more frequent tests and design iterations.

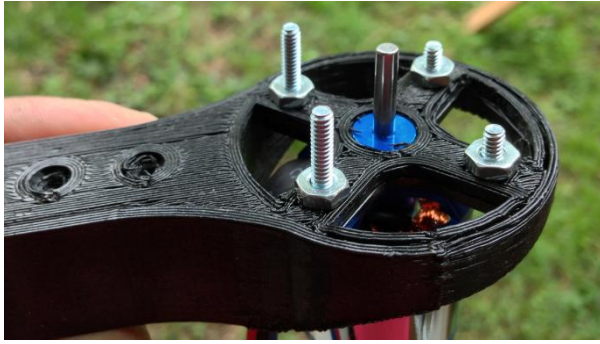


Figure 2.3: Example 3D printed motor mount with stripped threads (<https://backyardrc.wordpress.com/category/quadcopter/>)



Figure 2.4: Time intensive aluminum and balsa motor mount. (<http://www.rcgroups.com>)



Figure 2.5: A 3D printed motor mount with internal stiffeners and screw inserts printed by 4DOF

In another instance, the engineering team decided to experiment with using the 4DOF printer to produce a wing rib. This particular wing rib had been causing difficulty since it established both the aerodynamic profile of the wing, but also was the structural mount for the mechanism that actuated the ailerons. Because of this it needed to be particularly rigid, and provide mounting for numerous bearings, sensors, and the aileron motor, and have a precise complex outer geometry to define the airfoil shape of the wing. All of these requirements had made prototyping in balsa-wood particularly difficult, and often the team had to pay large sums to have the part CNC machined.

Immediately, the 4DOF printer was able to overcome these issues. Incorporation of screw inserts, aluminum stiffeners, and a special aluminum socket for connection with the wing spar provided the structural rigidity needed. Because 4DOF printers can produce complex 3-dimensional parts, the bearing and motor mounts needed were not only incorporated into the wing rib, but actually replaced with integrated features in the rib which directly held the bearings

and motor and reduced the overall part count. Finally, since the 4DOF printer can print around complex shapes, even the sensors used in the aileron actuation mechanism could be incorporated directly into the wing rib, thus greatly simplifying the whole assembly. Again, producing the wing rib required only a fraction of the time as needed for former methods, and the result achieved was much higher than previously achievable. Rapid design iteration perfected the part, and helped to significantly reduce the time-to-market of the RC airplane engineering company's newest airplane.

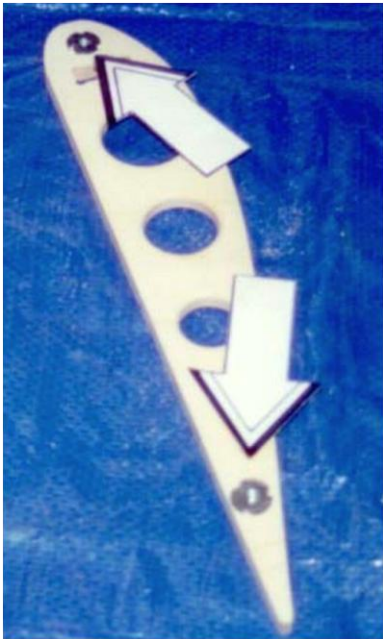


Figure 2.6: Plastic wing rib with added screw inserts. 4DOF printed ribs can have both more complex geometries and integrated screw inserts (<https://sidewalkfliersag.wordpress.com/>)

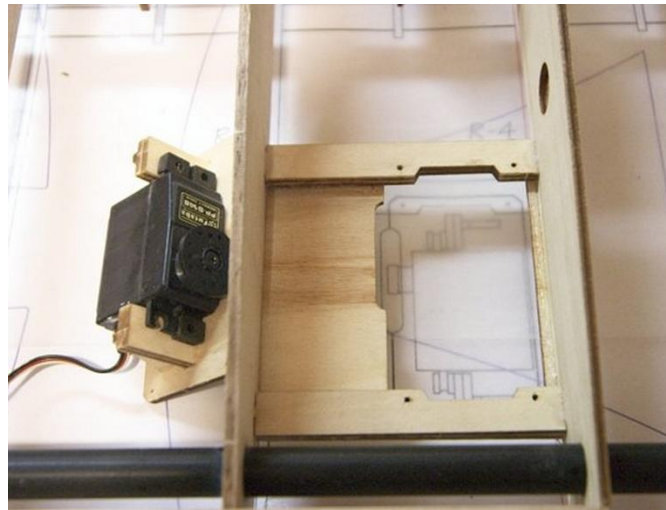


Figure 2.7: Balsa wood wing rib with complex aileron-servo mount

The case of the fixed-wing drone manufacturer is not unique. Thousands of companies today extensively rely on 3D printing to rapidly prototype parts and iterate designs. Improving the efficiency of this process even a small amount can have multiplicative effects on the quality of a resulting product. Strong, lightweight 3D printed parts and parts with integrated sensors and electronics represent a significant new advance in 3D printing technology that will help to provide rapid prototyping, mass customization, and lower cost parts to a variety of both businesses and consumers.

Graphical Use Case

After power up and initialization, the user will find the 4DOF printer in its 'idle' state. The heaters will be off and the motor axes will be disabled making it possible to manually reposition the build platform by hand.

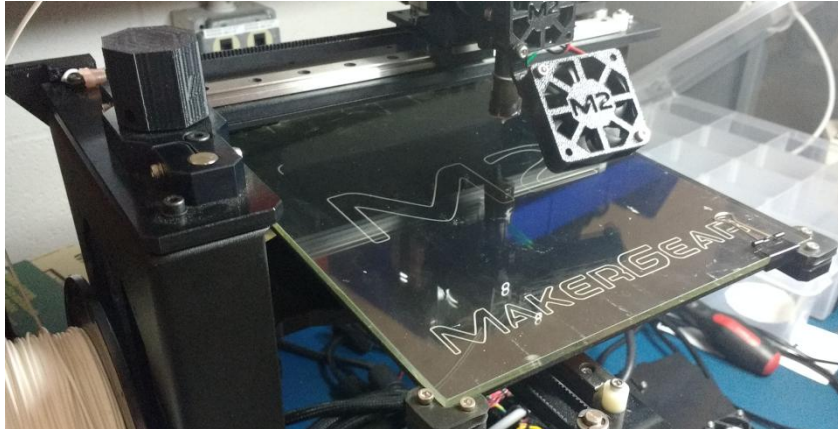


Figure 2.8: Printer in idle configuration

As with traditional 3D printers, the user will use a slicer application [1] to decompose a 3D files into layers and generate a G-Code file which can be sent to the printer. The only difference is that the user will also select insertion layers for the print to pause at and COTS items to be added.

After using the slicer program to generate the G-Code file, the file will be opened in a printer interface software from which the printer can be controlled and instructed to print the file.

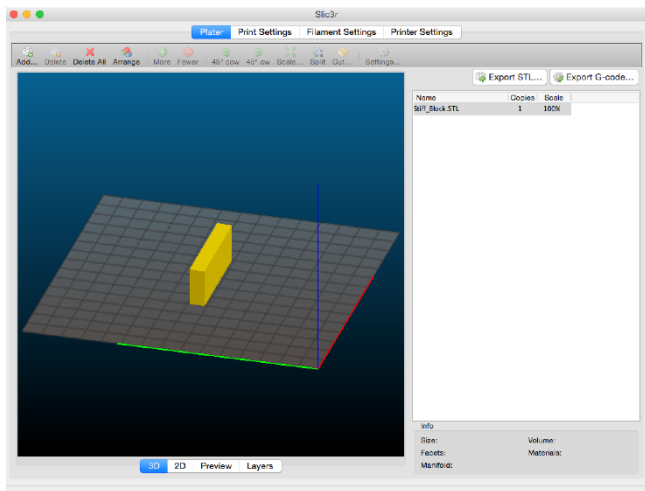


Figure 2.9: Screenshot of slicer application. The part to be printed is displayed and a G-Code file to print it can be exported. [1]

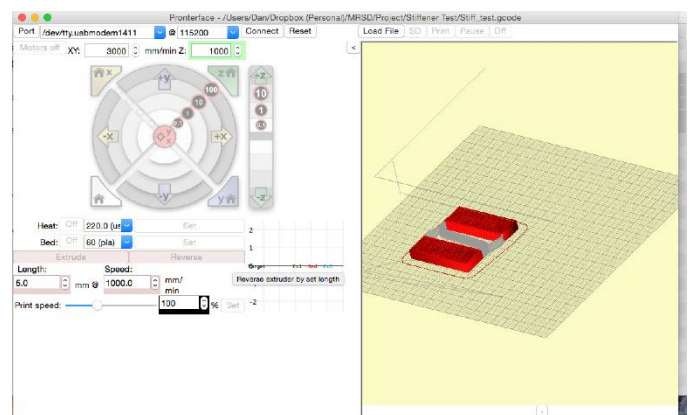


Figure 2.10: Printer interface application. The part to be printed is displayed and controls for the printer's movement and heater settings are available

After homing the printer and verifying it is correctly operating, the user instructs the printer to begin printing. The printer prints the part up until the selected print layer.

Upon reaching the selected insertion layer, the printer pauses and moves the part to the ‘insertion configuration’ to allow a user easily access the print bed and to install the COTS items.

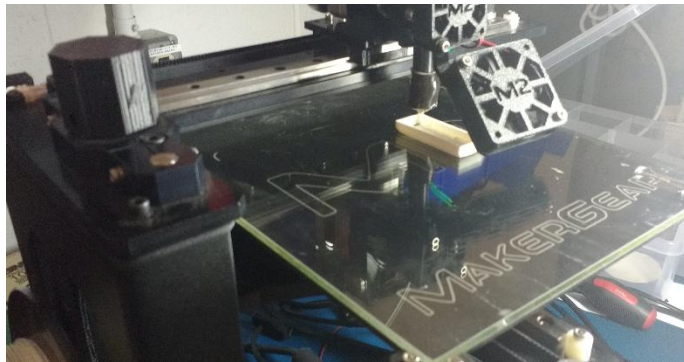


Figure 2.11: 4DOF printing the 3D part



Figure 2.12: Printer in insertion configuration awaiting COTS item installation

At this point the user would then manually install the COTS items.

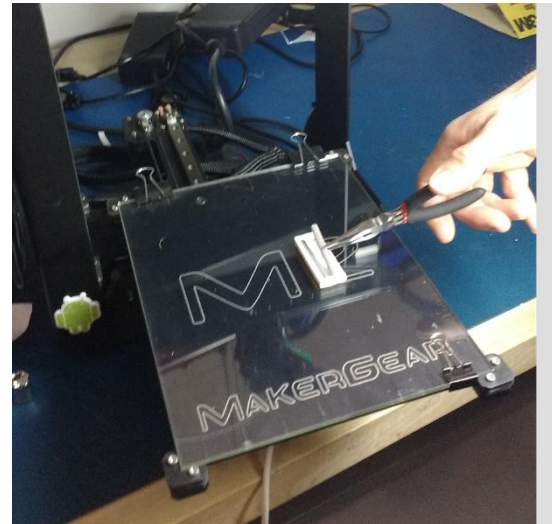
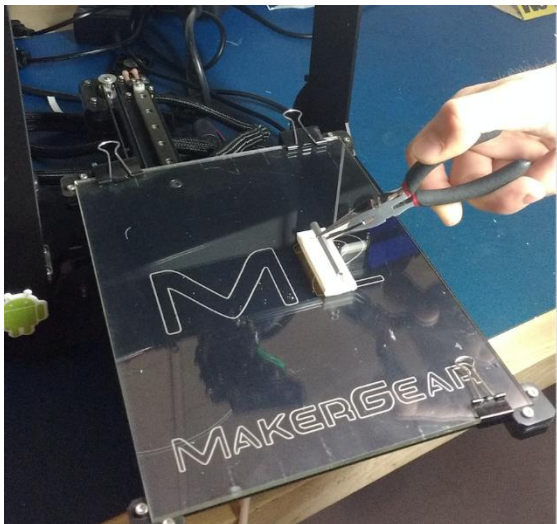


Figure 2.13 User inserting COTS items into 3D printed part

After installing the COTS items, the user instructs the printer to continue printing via the LCD interface.



Figure 2.14: 4DOF printer interface

The printer resumes the print and begins to enclose the COTS items

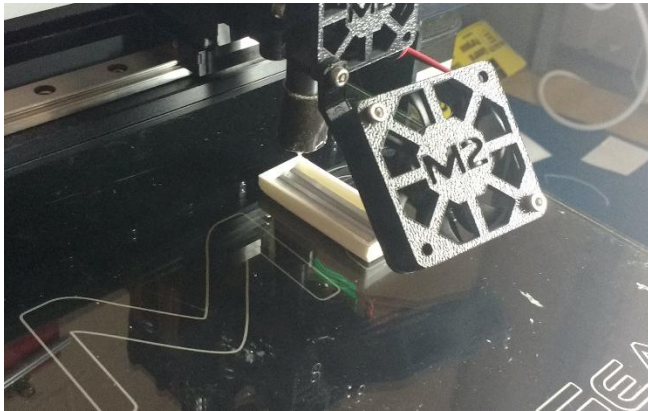


Figure 2.15: 4DOF printer enclosing COTS items. Path planning algorithms executed in the slicer ensure that the extruder nozzle doesn't collide with the COTS items.

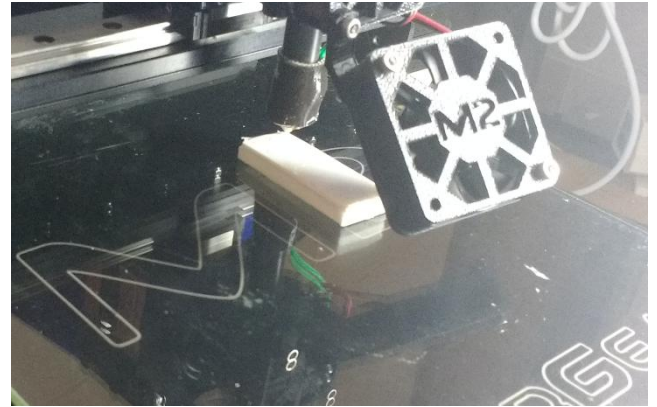


Figure 2.16: 4DOF printer fully encapsulating the added COTS stiffeners.

The printer continues printing until the part is fully enclosed. It then cools down and returns to its idle position so the user can extract the finished part.

3. System Level Requirements

The system is divided into two primary focuses, namely the ‘Software’ and ‘Hardware’. System requirements are mapped to either of these sections. Within these sections, they are grouped under the categories listed in Table 4.1.

Table 3.1 Legend

ID	Category
M	Mandatory (Threshold requirement)
D	Desirable (Stretch Goal)
F	Functional
N	Non Functional

3.1 Software

The Software System shall –

Table 3.2 Software Requirements

ID	Requirement	Deadline
Software Mandatory Functional Requirements		
M.F.1	- Receive 3D files The execution shall begin once the 3D file of the assembly (part to be printed + COTS item(s)) has been fed to the Slicer program.	Fall
	- Prompt user for insertion layer Slicer shall request the user to input the decided insertion layer. Upon reaching this layer, the printer will pause and move to a safe position and allow the COTS item to be inserted.	
M.F.3	- Create collision free path With knowledge of the dimensions and position of the COTS item, the Slicer program shall create a collision free path around the same.	Fall - Spring
	- Generate 4DOF G-Code The G-Code is comprised of movement commands for each of the 4 DOFs The printer interface software will load this G-Code before printing.	
M.F.4	- Generate 4DOF G-Code The G-Code is comprised of movement commands for each of the 4 DOFs The printer interface software will load this G-Code before printing.	Spring
Software Mandatory Nonfunctional Requirements		
M.N.1	- Work with standard 3D print files The slicing program shall be compatible with various standard 3D files Examples include .stl , .obj and .amf	Fall
Software Desirable Functional Requirement		
D.F.1	- Assign insertion layer A desirable feature of the slicing program would be the capability to assign insertion layers autonomously based on the dimensions and orientation of the COTS item. This would greatly reduce the program’s dependence on humans.	Spring

Software Desirable Non Functional		
D.N.1	- Easy-to-use interface	Spring
	Since the system requires user interaction, it is necessary to have an intuitive graphical user interface.	

3.2 Hardware

The Hardware System shall –

Table 3.3 Hardware Requirements

Hardware Mandatory Functional Requirements		
M.F.1	- Print layers of material	Fall
	The primary function of the 3D printer, upon which our system is developed, must be upheld post alterations to the manipulator and other system components.	
M.F.2	- Print locating features for COTS items	Fall
	Accurate positioning of an item on a partial print is achieved by printing locating features tailored to the base dimensions of the said item.	
M.F.3	- Go to safe configuration during insertion	Fall
	Since the addition of the COTS part involves a human placing the part on a partial print, measures shall be taken to ensure that the hot nozzle is out of the way and that the partially printed part is easily accessible. The above shall include: <ul style="list-style-type: none"> i. Moving the nozzle to safe location (away from printed part) ii. Fully lowering the bed to the insertion position 	
M.F.4	- Enclose COTS item with print material	Spring
	The printer shall deposit print material close to the added part such that there is no wiggle room around the piece for it to move, allowing for a snug fit.	
M.F.5	- Avoid Collisions	Spring
	Since the added COTS part will be protruding above the top surface of the part, the movement of the printer nozzle will be controlled to avoid collisions with it.	

Hardware Mandatory Non Functional Requirements		
M.N.1	- Provide user feedback during printing	Fall
	The user will be provided with a real time display of the progression of the print as the material gets deposited along the coordinates provided by the G-Code.	

Hardware Desirable Functional Requirements		
D.F.1	- Print between parts	Spring
	The nozzle should be capable of maneuvering between two added COTS items. The items shall not be more than 2 inches apart .	
D.F.2	- Print close to the COTS item	Spring
	The printer should print within 0.1 mm of the COTS part	

Hardware Desirable Non Functional Requirements		
D.N.1	- Maintain print speed	Spring
	The 4DOF printer should not have a print speed lower than that of the unmodified printer.	

4. Functional Architecture

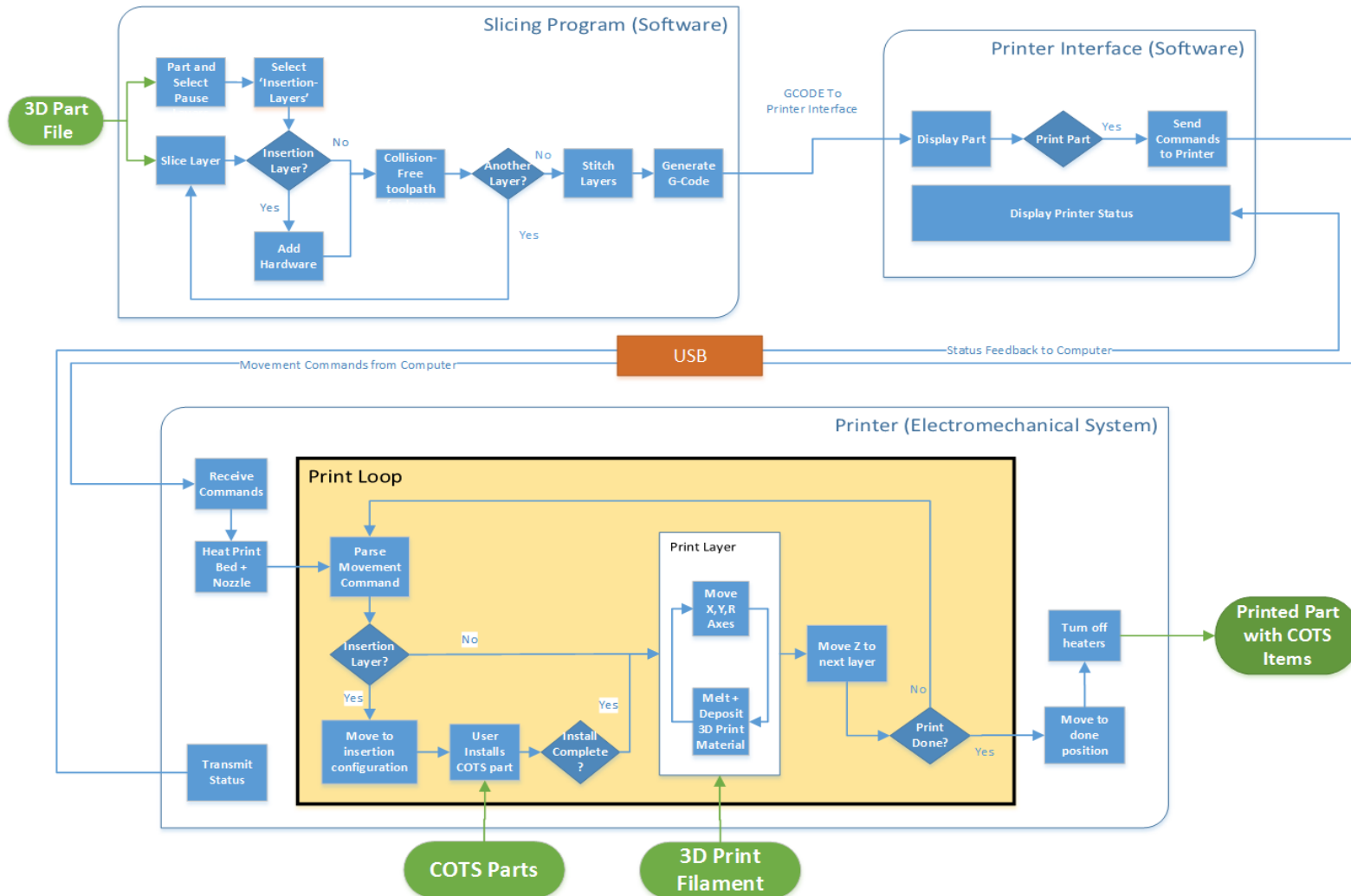


Figure 4.1: Functional Architecture

4.1 Slicing Software

This subsystem is responsible for generating the G-Code that is fed to the printer. G-Code consists of a list of coordinates that constitute a path the printer nozzle must navigate while depositing the print material. A collision avoidance algorithm will be put in place so as to generate a path around the obstructing COTS items.

The following is an example of G-Code command. It includes a positioning command ‘G00’ followed by coordinate locations for the XYZ axes.

```
G00 X0. Y0. Z0.25
```

This subsystem receives a 3D model file of the part to be printed as an input. The slicing software displays this 3D model on screen and prompts the user to select insertion layers where the print must stop to enable insertion of the COTS item. The program then proceeds to slice the model layer by layer iteratively from bottom up, generating a layer wise collision free path for the manipulator of the 4DOF printer to traverse. With each iteration, the program checks if the current layer has been designated to be an insertion layer by the user. If it has, the program acknowledges that a COTS item has been added and continues to generate a collision free path.

Once the entire model has been sliced, the layers are stitched together and the program converts the information to G-Code which is in turn fed to the Printer Interface Software.

4.2 Printer Interface Software

The Printer Interface loads the G-Code generated by the slicing software and displays the 3D sliced model on the computer screen. It communicates with the microcontroller mounted on the printer over USB. The microcontroller receives instructions to initiate the print when the user hits ‘Print’. The interface relays the coordinate and pose information of the extruder to the microcontroller. As the printing ensues, the status is transmitted back from the printer to the software. This is displayed on the screen, giving the user real time feedback on the print progress.

4.3 Printer (Electromechanical System)

The printer is an electromechanical system that is responsible for physically producing the required 3D assemble. As soon as the microcontroller on board receives commands to initiate the print, it first begins to heat the bed and nozzle head to predefined temperatures. The printer then begins the print and the base structure layer by layer. If and when it encounters an insertion layer, the hot nozzle moves to the safe configuration and alerts the user to place the COTS item in its feature. The user then inserts the item and instructs the printer to continue printing. Layer by layer, the printer continues to deposit material on the collision free path, steadily encasing the COTS item within the material.

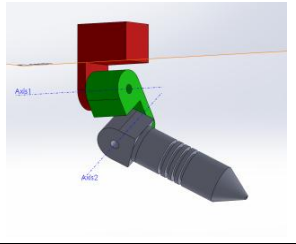
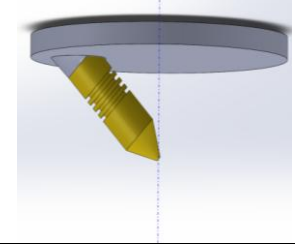
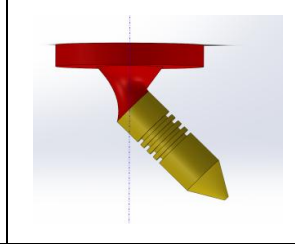
Once the printing is complete, the extruder moves to the home position and the heaters are turned off. This concludes the printing process, leaving the user with a completed 3D part.

5. Systems-Level Trade Study

To build the proposed system we have performed a systems level trade study on the construction of the extruder nozzle and on the 3D printer platform we should use.

The *extruder nozzle* is the crux of the project and it must be carefully designed. The three designs used for the trade study are shown in Table 5.1. Design D1 uses two series connected orthogonal rotational joints. Design D2 is a single axis rotational joint. Here the nozzle is inclined at an angle and the axis of rotation coincides with the nozzle tip. Design D3 is also a single axis rotational joint. However, the axis of rotation is along the nozzle mount end such that the nozzle tip rotates in a circle.

Table 5.1 Trade Study of the Extruder Nozzle

Parameter	Weight	Two Axis of rotation (D1)	Axis of Rotation is along nozzle print axis (D2)	Axis of Rotation is along nozzle head (D3)
				
Accuracy	12	6	8	7
Speed	8	6	9	8
Ease of Control	18	7	9	8
Mechanical Simplicity	18	5	7	9
Small Size	12	5	8	8
Avoids Filament Damage	20	7	6	9
Light Weight	12	5	9	9
Weighted Sum	10	5.96	7.8	8.38

The designs have been tagged by their design names in the parenthesis. The selection criteria of the system trade study are the accuracy, avoiding filament damage, ease of control and mechanical simplicity. Accuracy of the design is measured as how precisely can the extruder nozzle be positioned. When design D1 is used, the accuracy is low because the mechanical slop in each joint will add to create a cumulative error larger than that in other designs. Also multiple degrees of freedom may not be needed since the team has specified that added hardware must be orthogonal to the print surface plane. Thus, placing the nozzle at a fixed angle in the z-x plane and rotating it about the z axis should be suitable for the design.

Another, important parameter to be considered is that the rotation along the joint does not twist, bend, kink or tangle the filament that is being extruded into the nozzle. This was the major drawback of design D2 since it would likely require a ‘bowden tube’ configuration to feed the filament into end of the print nozzle. Design D3 avoids this problem by placing the path of the

extrusion filament concentric with the rotation axis, and thus preventing the possibility of filament wrapping around the rotation axis. However, this design does still twist the filament.

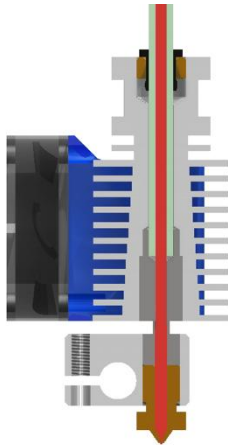


Figure 5.1 Example of a Bowden tube nozzle. Filament (red) is fed to the nozzle through a flexible PTFE tube (green), thus enabling the nozzle to move while the extruder remains stationary. (community.robo3d.com)

In design D3 the mechanical complexity is minimized, although the control complexity is slightly higher because of the offset between the tip and the axis of rotation. However, this challenge was determined to be less significant than the mechanical complexity incurred in D2.

The 3D printer is the base upon which we will build our system, so we conducted a trade study to assess the most appropriate 3D printer for our project as shown in table 5.2. We considered 3 of the most popular desktop 3D printers on the market. A Makergear M2 [2] printer was made available to us by our sponsor, but we directed the trade study to make sure it was indeed the best choice for our project, and that availability was represented in the cost parameter.

Other important parameters for our project are open-source software and hardware, the community support, the stability of the bed (This parameter is important for our project since we will add COTS items, having a stable bed is more reliable), mechanical simplicity, Nozzle design and filament damage all refer to the adaptability of the printer's design to our 4DOF printer.

Table 5.2 Trade Study of the 3D printer

Parameter	Weight	Ultimaker 2	Makerbot Replicator 2	Makergear M2
Open source (Firmware/ hardware)	20	8	6	9
Community support	12	9	9	8
Bed stability	8	8	8	7
Mechanical Simplicity	18	5	7	9
Nozzle design	12	5	6	8
Avoids Filament Damage	18	7	6	9
Cost	12	5	5	9
Weighted Sum	10	6.68	6.58	8.60

After conducting the trade study we concluded that the Makergear M2 was indeed the most appropriate printer platform for our project.

6. Cyberphysical Architecture

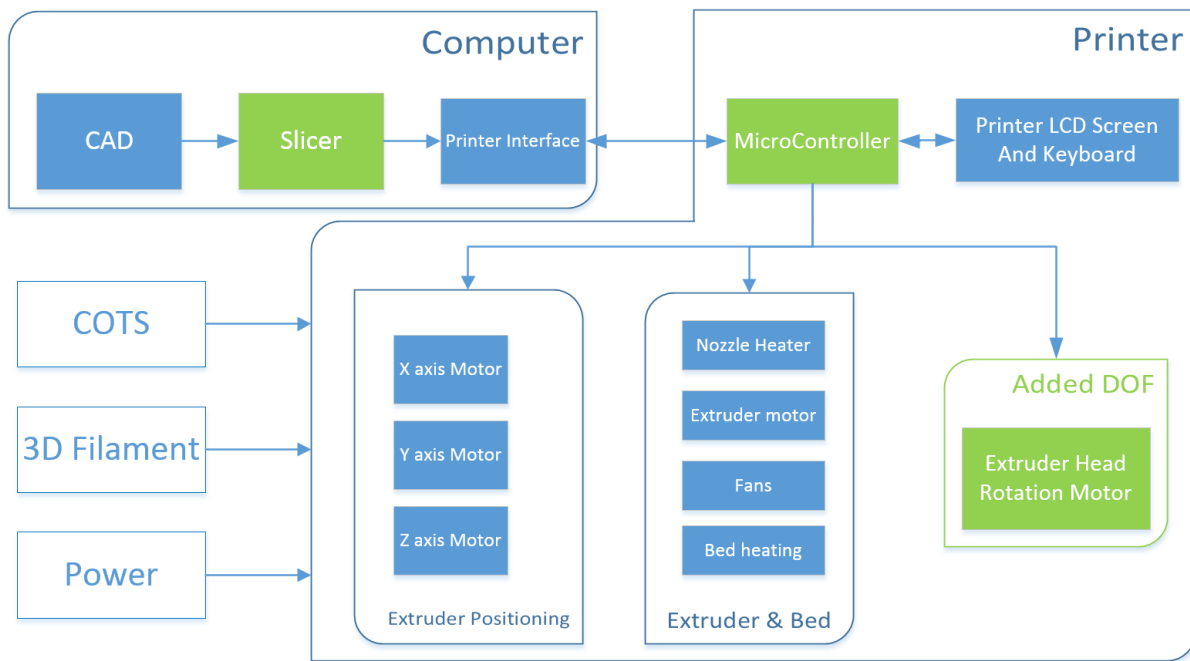


Figure 6.1: ADD_IN Cyberphysical architecture

6.1 Overall Architecture

The cyberphysical architecture of the 4DOF system, shown in Figure 6-1 below, is inspired by the functional architecture and standard 3D printers. It comprises a software component, namely the slicer and printer interface, and hardware component which is the 4DOF printer.

Our system is based on the **Makergear M2** [2] commercial 3D printer that was provided by our sponsor and already contains many of the necessary system components (shown in blue in Figure 6.1) Our project will focus on adding and redesigning specific aspects (shown in Figure 6.1 as green blocks) and integrating them with the rest of the system. Thus the portions of the cyberphysical architecture which will be developed by the team are:

The slicer software: Must to be extended to generate a 4 DOF path

The printer controller: Embedded system that must be modified to control all four DOF.

The R axis joint: Must to be implemented to enable collision avoidance with COTS items.

The Nozzle head: Must to be redesigned to enable printing at an angle and must be integrated with the R axis joint.

6.1. The Makergear M2

- Open Source Software and Electronics
- 8" x 10" x 8" Build Area
- Weight: ~28 lbs (12.7kg)
- Min. Resolution: X/Y 0.01mm. Z: 0.0025mm
- Nozzle orifice diameter: 0.35mm
- Heated bed
- Stepper motor actuated

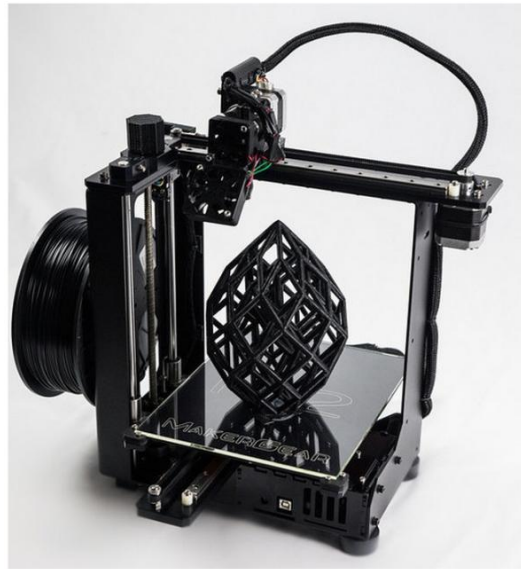


Figure 6.2: Makergear M2 (<http://www.makergear.com/>)

6.2 The Slicer Software

The slicer software is responsible for translating the 3D cad model to G-code interpretable by the printer. For our system we will work with one of the open-source slicer softwares available. The two industry standards are Slic3r and Cura [3], and an inspection of the code and extensive trade study will have to be conducted before we select the one most compatible with our project.

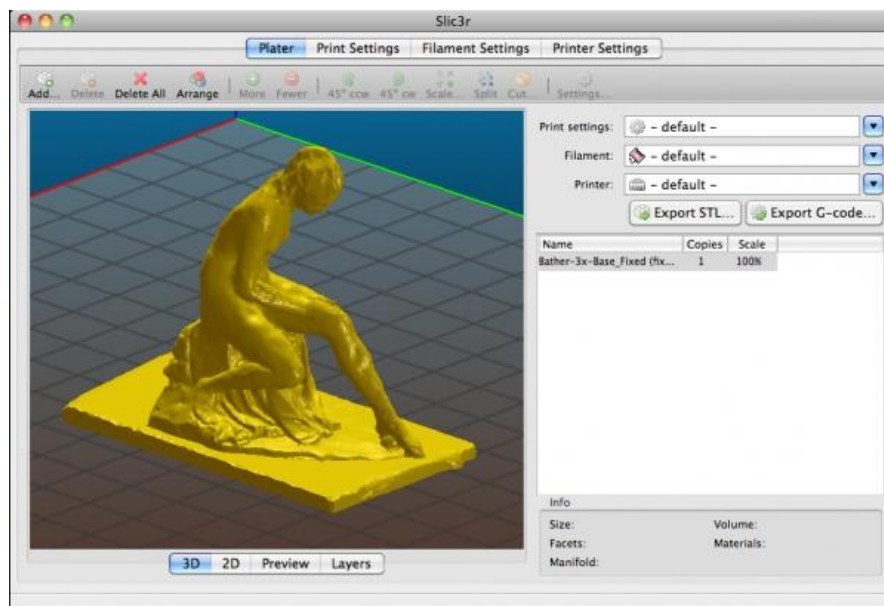


Figure 6.3: Slic3r user interface (<http://slic3r.org/>)

7. Subsystem Descriptions

7.1 Slicer Subsystem

A slicer program [1] is the software application that converts a model of a 3D part (most commonly a stl, .obj, or .amf file) into machine instructions (usually G-Code) that can be executed by the physical printer. To do this the slicer will decompose the part into horizontal layers, invoke a path planning algorithm to determine the nozzle movements and filament feed rates to produce each layer, and then generate a list of G-Code commands which can be sent directly to the printer.

In addition, Slicer programs also typically implement a myriad of other features such as part visualization, infill control, and platter layout:

Part visualization enables viewing of the original and sliced part – an important verification step so that the user can ensure the geometry of the sliced part is consistent with their original design.

Infill control, enables the interior of 3D printed parts to be partially hollow. This saves print material, reduces print time, and also increases the geometric precision of the part. Typical slicer programs enable control of infill density (usually as a percentage) and infill geometry pattern (common patterns: honeycomb, rectilinear, concentric rectangles, lines, and Hilbert curves).

Platter layout is the process of locating the modeled 3D part on the physical print area of the printer. Often the orientation of the part on the print surface will affect the print time and accuracy of certain features within the part.

Since the ADD_IN printer uses a 4th degree of freedom to enable printing around COTS items, the path planning portion of traditional slicer programs must be extended to determine both a translational path for the extruder but also an extruder pose (rotation angle) for each point along the path. The added COTS items can be considered obstacles that the extruder must traverse around, thus producing a more complicated optimization problem. The algorithm for converting the planned path to G-Code must also be modified to conform to G-Code standards for four axis machines.

Furthermore, the ADD_IN printer requires the periodic ‘pausing’ of the print and movement of the printer axes to a configuration that allows an operator to install COTS items. Determination of the correct step in the print sequence to pause requires information about the location of the COTS items being added. This information will be provided by the user through an ‘insertion layer selection’ interface that enables the user to specify the layers for the printer to ‘pause’ at. For each of these insertion layers, the slicer program will then generate a special set of G-Code commands to stop the extruder, move the printer into its part insertion configuration, and prompt the user to install the COTS item. The G-Code language already contains the necessary flow-control operations to allow the user to resume the print once the COTS item has been installed.

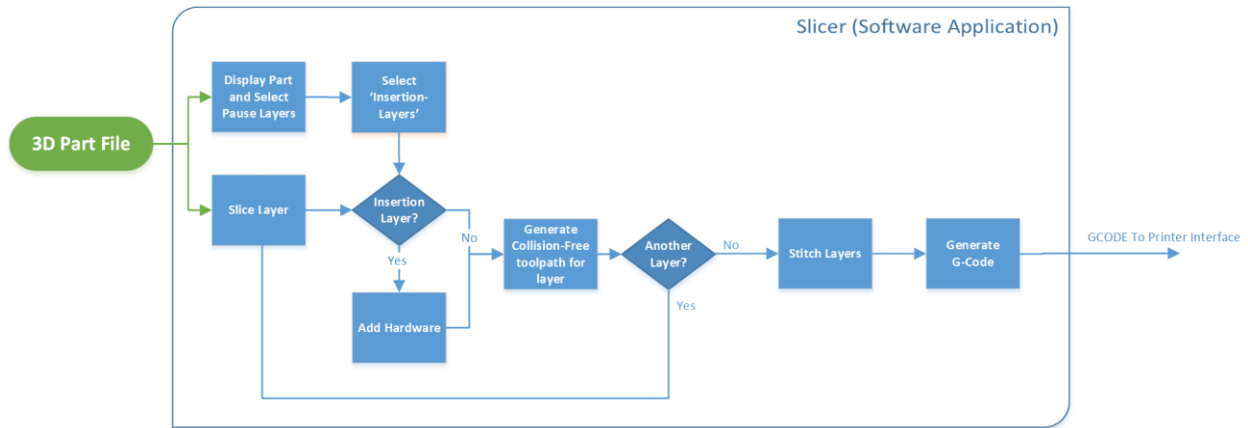


Figure 7.1: Functional Overview of Slicer Program

Currently, there are a variety of open source slicing software for low cost¹ FDM 3D printers that provide the features described above. The most notable examples are Slic3r and Cura, both of which are well documented and have a large technical support network. ADD_IN aims to modify one of these existing slicing programs to incorporate the more complex path planning algorithm and insertion layer selection tool necessary for the printing of parts that include COTS items.

The 4-DOF path planning portion of the software is predicted to be the most technically challenging aspect of the slicer subsystem. Fortunately, the problem is very similar to the path planning and collision avoidance algorithms that are utilized in CAM software for controlling four and five axis CNC machines which has been extensively researched and numerous solutions have been published. [4](Tang, 2014) provides an overview of the 11 most commonly used algorithms, their implementations, and the pros and cons of each.

7.2 Printer Controller subsystem:

The Microcontroller board that powers the MakerGear M2 we are using is a Rambo 1.1B Controller Board [5]. Rambo stands for RepRap Arduino-Mega BOard. As shown in Figure 7.2 the Rambo board is an all-in-one electronics board that communicates with the computer through the printer interface and is used to control all components of the 3D printer.

Electrical connections: Conveniently, this board can control five stepper motors (two ports are connected together for the Z axis) but only four steppers are currently used on the MakerGear M2. The unused control axis (rightmost port in Figure 7.) is typically used only in printers with dual extruders for control the second extruder stepper motor. ADD_IN will use this port in our project to control the R joint necessary to rotate the nozzle. To implement the endstops for that axis we will use currently unassigned pins from the expansion port.

¹ 'Low-Cost' approximately refers to the \$500-\$3000 range of 3D printers most commonly used by hobbyists, product designers, and small businesses. Common examples include MakerGear, Makerbot, Ultimaker, and RepRap machines.

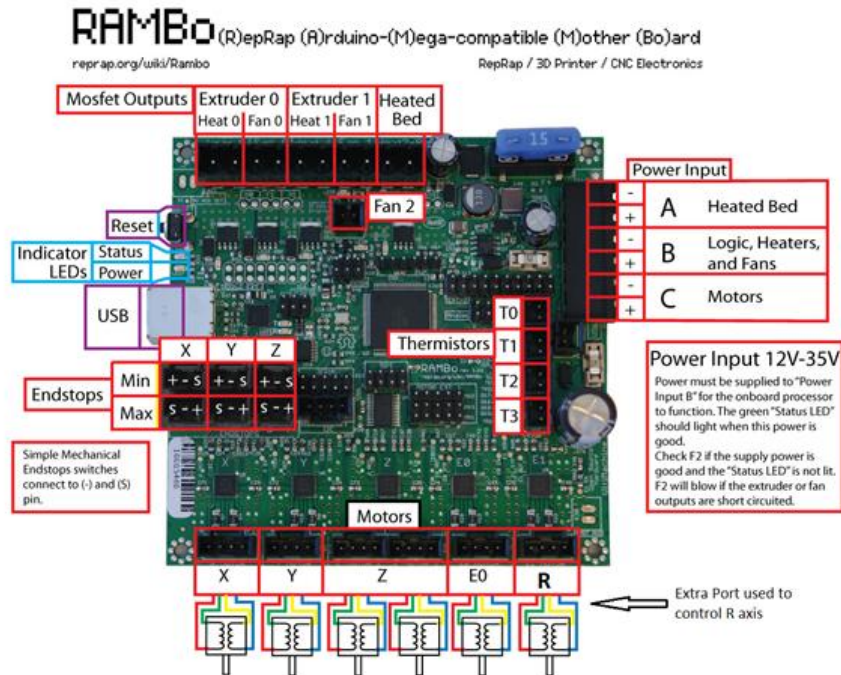


Figure 7.2: Rambo Control Board
 (http://reprap.org/wiki/Rambo_v1.1)

Firmware: The board on our MakerGear 2 is loaded with the Merlin Firmware, which is Arduino based and entirely open-source. The role of the Firmware is to interpret the G-code commands and control the printer's heaters and motions accordingly. Each firmware version is specific to the printer, so it contains the parameters such as the dimensions, ports and heating settings.

We have obtained the source code for the version running on our printer, and ran tests by uploading it to the board and testing the print result. We also successfully modified the code to send control signals from the fifth motor axis thereby confirming the board's ability to control an additional stepper motor. The objectives of this subsystem are to extend the code to control the R axis and enable parsing of G-Code commands for all four axes. Since the G-code commands received by the microcontroller only describe discrete point, line, and circular arc movements, the firmware must use motion planning algorithms to synchronize the movement of all axes.

7.3 R Joint subsystem:

The R joint connects the extruder with the nozzle and enables the latter to rotate during the print process, necessary for collision avoidance. Designing it represents one of the most challenging aspects of the project since it must avoid filament kinks and tangles caused by rotation. To alleviate this, the filament will be fed through the joint's rotation axis and melted to produce a sort of 'filament slip ring'. The team's research has not revealed an instance in which this has been done before, thus the work schedule has been designed to allow multiple iterations of the joint design.

7.4 Rotary stage subsystem:

Purpose: The rotary stage is the drive of our fourth degree of freedom. It will be used to rotate the R joint precisely so that it is able to print from various orientation at each point in the XYZ space. The R joint will be inside the whole of the stage with its top part fixed and lower part connected with the rotary stage

Implementation: The joint will be implemented using a high-end precision rotary stage with a clear aperture, as illustrated by Figure 8.3. We will buy the stage off-the-shelf and make custom mounts and connections.

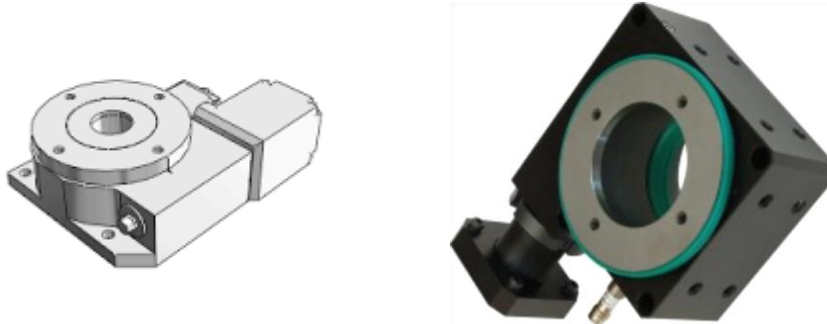


Figure 7.3 Rotary stage example design- (Illustration only, final design might differ)

7.5 Nozzle head subsystem:

Overview:

The nozzle is the heated orifice through which molten plastic flows out. In the ADD_IN system a custom nozzle is needed that is inclined relative to the print surface to be able to print material around COTS items.



Figure 7.4: View of standard 3D print nozzle (left) and rendering of inclined nozzle concept for 4DOF

The Figure 7.4 shows the typical components of a standard nozzle (left), and illustrates the design of the 4DOF nozzle concept decided after conducting a trade study(right). The inclination angle of the nozzle will be fixed so research will be conducted to determine the optimal angle for printing and avoiding direct contact with any COTS part.

8. Project Management

8.1 Work Plan and Tasks

Figure 8.1 represents the work breakdown structure of the project in a tree format. A bottom up approach is used here, as the sub-systems described in section 8 will be developed and then integrated to form the entire system. Each box represents the task that an individual/group will perform. The responsibility of each task is assigned to the team member(s) whose first letter is represented in the parenthesis.

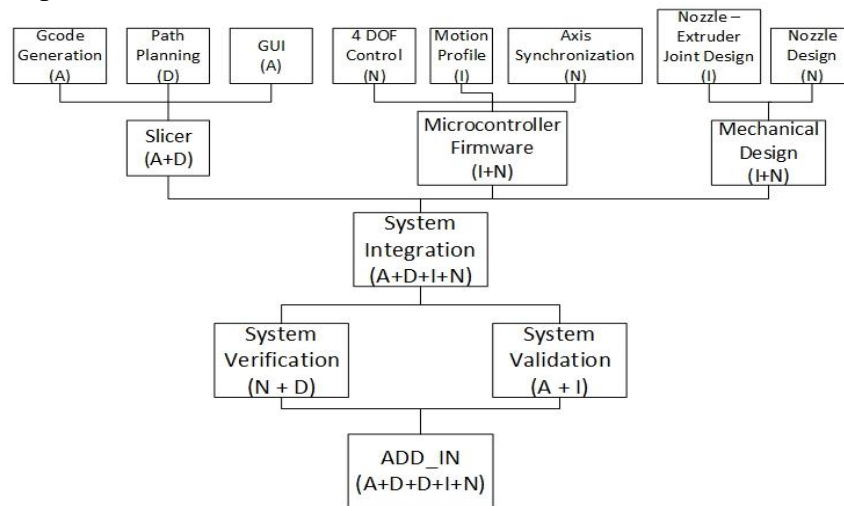


Figure 8.1: Work Breakdown Tree

8.2 Schedule and Milestones

Figure 8.2 shows the Gantt chart which represents the project schedule. The milestones are highlighted by black diamond shaped objects in the Gantt chart.

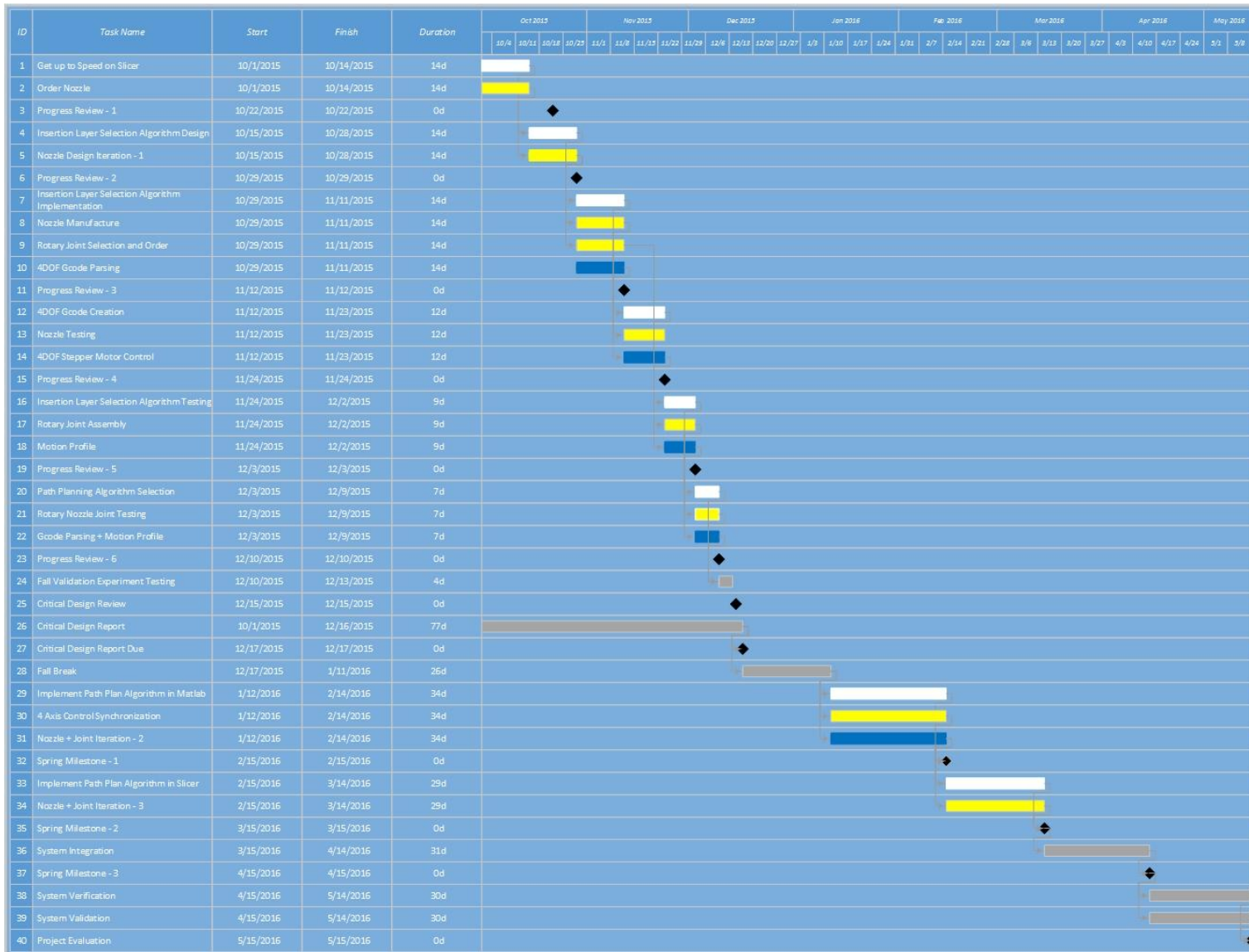


Figure 8.2: Work Schedule represented by a Gantt chart (White Bars - Software Section; Yellow Bars - Mechanical Section; Blue Bars - Controls Section; Brown Bars-Systems Integration; Diamonds-Milestones)

8.3 System Validation Experiments

The ADD_IN team's fall validation experiment is aimed at testing the function of the two 'highest risk' subsystems. These are the ability to extrude 3D print filament from an inclined rotating nozzle and the ability to select insertion layers within the slicer software.

Test 1: Verification of nozzle print capability

Test Objective: To evaluate the ability of the custom print nozzle to extrude and deposit 3D print filament while rotating.

Test Summary: The print nozzle assembly (print nozzle, nozzle joint, and rotary stage) will extrude plastic onto a build platform while rotating in a circular path

Required Materials:

1. Laptop computer with necessary control software
2. 1.75mm 3D print filament (material type to be chosen by team prior to test)
3. Printer electronics for controlling extruder, rotary stage motors, and nozzle heater
4. Print nozzle assembly mounted on printer or test fixture
5. Control file defining extruder motor and rotation motor movement commands

Test Sequence:

1. The electronics will be powered on and the nozzle allowed time to warm up
2. The computer will command the extrusion motor to start extruding filament. Simultaneously the rotation motor will be commanded to begin rotating the nozzle
3. After approximately 1 rotation the motors will be commanded to stop.
4. The extruded filament will be inspected to ensure it is continuous and flowing smoothly through the nozzle joint and from the nozzle
5. The test will be repeated at least 3 times, with the nozzle rotating in the same direction to demonstrate that the nozzle joint is allowing infinite continuous rotation without disrupting the filament flow.

Test 2: Insertion Layer Selection

Test Objective: Demonstrate the ability of a slicer program to allow selection of insertion layers and for the 4DOF printer to 'pause' at the selected layers.

Test Summary: An example part will be loaded into the slicer application and the user will be prompted to select insertion layers. The slicer will then be invoked and the G-Code file sent to the printer. The printer will then control the X, Y and Z axes, simulating the printing of a part, and pausing to enter a 'safe configuration' when reaching the insertion layer. After receiving a 'continue' command from the user, the printer will return to its previous location and continue the print simulation.

Required Materials:

1. Laptop computer with printer interface and slicer software
2. 4DOF 3D printer and associated power supplies
3. .stl, .obj, or .amf file for simulated part to be printed

Test Sequence:

1. The part file will be loaded into the slicer program

2. Pause layers will be selected. This may be arbitrary.
3. The slicer will be invoked to generate G-Code for the part. It will include instructions to pause the print at the selected layers.
4. Depending on the progress of the nozzle development, the team may optionally choose to physically print the part or just simulate printing by appropriately moving the X, Y and Z axes.
5. Using either a printer interface program of the team's choosing, the G-code file will be sent to the printer.
6. The printer will simulate printing of the part, and must enter a 'part insertion configuration' at the beginning of the correct layers.
7. The printer must resume printing after the operator has 'Confirmed COTS installation' via the printer's user interface

8.4 Team Member Responsibilities

Table 8.1: Team Member Responsibilities for Technical Tasks

Roles	Slicer	Controls	Extruder-Nozzle Construction
Primary	Astha	Nikhil	Ihsane
Secondary	Dan	Ihsane	Dan

The non-technical tasks have been assigned as follows:

- Website Management: Astha Prasad
- Sponsor Coordination and Budget Management: Ihsane Debbache
- Documentation and Presentation Review: Dan Berman
- Project Management: Nikhil Baheti

8.5 Part List and Budget

Table 8.2: Part List and Provisional Budget

Part	Quantity	Expense	Total Price	Details
Printer	1	\$0	\$0	Makergear M2
Spare Printer Parts	2	\$110	\$220	3D printer parts such as the print nozzle will be modified and may be damaged so to avoid delays in project the team will have spares
Custom print nozzle and extra DOF	1	\$2000	\$2000	Design and construction of the rotation stage and nozzle assembly will be an iterative process. Manufacturing and machining costs for the rotation joint and nozzle construction are included.
Print Materials	4	\$100	\$400	Various print materials such as PLA, ABS, and polycarbonate will be tested during the implementation
Rambo Board	2	\$300	\$600	Rambo interface board may get damaged during the testing phase and thus spare boards must be available to avoid delay in development time.

8.6 Risk Management

The following are the risks that are expected to occur in the course of the project:

1. **Nozzle Design Complexity:** The nozzle is a highly engineered component of the 3D printer requiring mechanical design, material science, and thermal analysis. Thus, we expect needing multiple design iterations before achieving a fully functional version
2. **Kinks and jams in the extruder filament:** If the nozzle-extruder joint is not designed carefully it will lead to kinks and jams in the filament affecting the print quality. To simplify the nozzle design and therefore minimize risk the filament is path positioned along the axis of rotation and thus only twisting forces will need to be mitigated.
3. **Critical Component Damage:** Since there will be rigorous testing and modification of the 3D printer, there will be a high likelihood that components may get damaged. This could significantly impede team progress until replacement parts arrive. To avoid such risks, two sets of spares parts for the most commonly damaged printer components will be ordered and kept for immediate repairs.
4. **Shipping Delays:** To avoid shipping delays every week the team will discuss the components that might be required in the following two weeks for the system that they are working on. 15 minutes will be allocated in one of their twice weekly team meetings.

Since certain aspects of the 4DOF printer have not been previously implemented or researched, preliminary tests have been performed to provide proof of concept validation and mitigate the risk of key project aspects being unachievable.

1. **Printing at an Incline:** It is important that the nozzle can print at the same quality when at an inclination. To test this, a component was printed with the printer at both normal and inclined configurations. Figure 8.3 shows the object being printed at an approximately 45° incline. The quality of the print remains the same if the entire object is printed at the same inclination. Minor textural changes are noticeable if the printer inclination is moved during printing.

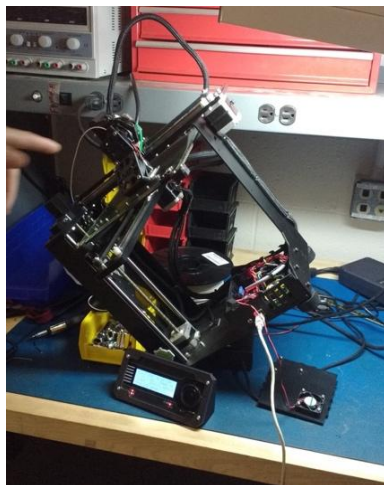


Figure 8.3: Printing at an Incline of 45 degrees

- 2. Rotating an extra Stepper Motor:** To save time of implementation on a new board and make maximum use of the controller board provided by the 3D printer, a test was conducted to check if the Rambo Board can control another stepper motor. This stepper motor will be responsible for controlling the axis of rotation of the extra DOF. The Rambo Board can drive five stepper motors, only four of which are used in the original printer configuration. To validate the usability of the extra stepper control axes, the y-axis stepper motor was rewired to one of the unused stepper drivers and the firmware updated. In this configuration a part was printed and did not have any detectable change in the quality of print. Therefore, the team is confident that the current printer board can control the extra stepper needed for the rotational axis. Figure 8.4 shows the printer controlling the y-axis using the extruder1 port.

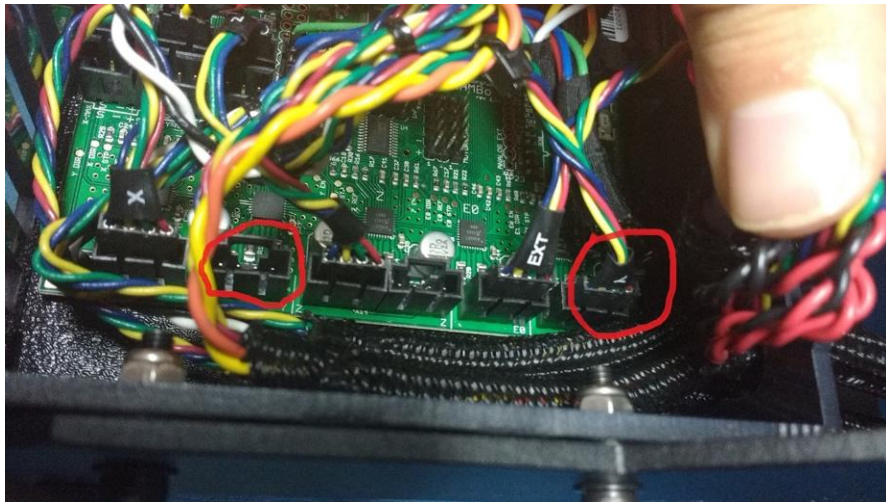


Figure 8.4: Y-axis motor controlled to E1 port

9. References

- [1] Slic3r software: <http://slic3r.org/>
- [2] Makergear M2 3D printer: <http://www.makergear.com/products/m-series-3d-printers>
- [3] Cura software: <https://ultimaker.com/en/products/cura-software>
- [4] Tang, Tran Duc, "Algorithms for collision detection and avoidance for five-axis NC machining: A state of the art review" J. Computer Aided-Design pp.1-16, 2014.
- [5] Rambo Development Board: http://reprap.org/wiki/Rambo_v1.1

Appendix

A1: Glossary

COTS	Commercial Off The Shelf (referring to the type of components 4DOF will include in prints)
4DOF	The name of ADD_IN's printer
Slicer	The software program that converts a 3D part file into machine control commands that can be executed by the printer
G-Code	The functional language describing the control commands interpreted by the 3D printer
Extruder	The electromechanical component that feeds 3D print filament into the nozzle.
Nozzle Hot End	The nozzle through which 3D print material is extruded to produce the part
3D Print filament	3D print material, as it is purchased, as a 1.75mm filament which can be fed through the extruder
ABS, PLA	Two of the most common types of plastic used for 3D print filament
R Axis Rotation Axis	The additional rotational degree of freedom that separates 4DOF printers from traditional FDM printers
FDM	Fused Deposition Modeling. The 3D print process used by the 4DOF printer whereby a heated nozzle extrudes a 'thread' of filament which is successively laid down to produce print layers
Print Layer	One layer of a 3D printed part, produced by manipulating the X and Y axes while the Z axis remains fixed
Print Bed Print Surface	The glass surface upon which a 3D part is printed. It is heated to prevent warpage of the part during print.
Safe Position Safe Configuration Insertion configuration	The X,Y,Z and R axis positions that the 4DOF printer is moved to, for the installation of COTS items into the partially printed part
Insertion Layer Pause layer	The print layer at which 4DOF will pause printing and move to a safe configuration to enable insertion of COTS items
Rambo	The name of the open source electronics board used for controlling the 3D printer

A2: Work Breakdown Structure

Task 1.x relates to software task; Task 2.x relates to Hardware; Task 3.x relates to control; Task 4.x relates to system

	Task	Days	Notes
1.1	Get up to speed on slicer	14	Become familiar with open source Slic3r code and program operation
1.2	Path Planning	60	Develop path planning algorithm to control 4DOF and avoid COTS items
1.3	Insertion Layer Selection	10	System for assigning insertion layer
1.4	G-Code Generation	7	System for generating G-Code to control 4 axes
1.5	Stitch Layers	7	Integration of planned path for each sliced layer
1.6	Part Display	7	System to display part and insertion layers to user
1.7	GUI	15	GUI modifications necessary to accommodate 4th DOF
1.8	Simulation	20	Simulator to display part and extruder path
1.9	Slicer Integration	15	Integrate all Slicer Parts
2.1	G-Code Parsing	7	Make sure the controller reads and interprets the 4DOF control code
2.2	4 DOF Control	14	Hardware implementation of 4th axis motor controller
2.3	Motion Profile	21	Position/Speed/Acceleration control of fourth axis
2.4	Axis Synchronization	28	Synchronized motion of all axes
2.5	Firmware Integration	15	Integrate all control parts
3.1	Nozzle Research/Design	60	Mechanical design of nozzle to achieve proper filament flow properties
3.2	Nozzle Joint Research/Design	60	Mechanical design of joint between extruder and nozzle through which print filament must flow
3.3	Rotation Stage Design/Integration	30	Design and selection of rotation DOF components
3.4	Nozzle/Printer Integration	12	Integration of nozzle including 4th DOF into printer
4.1	Subsystem Integration	30	Integration of all systems into working printer (.stl -> physical part)
4.2	Verification + Validation	30	Verifying that the design works as the system and validating that all user requirements have been met.