CERTIFICATE

It is certified that the contents and form of thesis entitled **"3D Printer-Ecraft"** submitted by *Saleha Asad* **(***458***),** *Madiha Zahir Bokhari (449) and Umair Ali Zafar (416)* have been found satisfactory for the requirement of the degree.

Advisor: Mr. Kamran Zaidi

( )

Co-Advisor: Mr. Ansar Moughis

( )

DEDICATION

In the name of Allah, the Most Gracious,

the Most Merciful

To

Our Parents

&

Our Teachers

ACKNOWLEDGEMENTS

We would like to take this opportunity to pay our humble gratitude to Almighty Allah who blessed us with his kindness to complete this project.

We are highly thankful to our Advisor Mr. Kamran Zaidi and our Co-advisor Mr. Ansar Moughis for guiding us in the project and for encouraging us to achieve our goal. Their support has helped us complete many ambitious tasks and has shown us the right way whenever we were in lost direction.

Last but not the least we would like to offer appreciation to our parents. Their supplications, criticism and encouragements were the means by which we have completed our project in time.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

3D Printing is a new and creative step in the field of rapid manufacturing where people look for easy and quick solutions to their problems. The world has grown so fast that nobody has the time to go to the market to replace a faulty part, so a 3D printer provides a solution to it. A 3D printer is capable of printing objects in 3D using layer by layer deposition of a specific plastic material. The printer consists of a three axis assembly with an extruder attached in a 3D plane. The axes are controlled with the help of stepper motors and the extruder is moved along the three dimensions. The purging of the material from the extruder gives the required shape in the form of a 3D object. The control commands are sent through the computer software interfaced serially with Arduino platform. The GUI allows you to convert any 3D image into controller understandable commands which in turn moves the axes. It is a complete solution to rapid prototyping with hardware and software package. The major advantages of 3D prints include fast development of parts, flexible customization and ease of modification.

Chapter: *1*

1.0 INTRODUCTION

In this era of technological advancement, when everything is contained in your desktop PC and is easily available to you e.g. you want a document and you get it easily printed with your printer how nice it would be if you can also print objects.

3D printers are designed with the purpose of rapid prototyping. The main principle behind a 3D printer is the rapid deposition of layers of a particular material successively to form a specific structure. As with a 2D plotter the 3D printer also requires a similar working mechanical structure with the addition of third axis (for movement in the third dimension) as now instead of forming just an image we are forming a complete object. 3D printer is an affordable and a fast device as compared to other technologies for rapid prototyping. The user can print out different machine parts and also replicate objects of daily use.

1.1 PROBLEM STATEMENT:

In the current market, plastic things are made by melting plastic and then pressing it into a required shape. For this purpose, an outer shell of the material has to be provided which encapsulates the melted plastic. The problem in this process is that every shape requires a new shell to give the melted plastic a shape. Avoiding this process in industrial manufacturing can be carried out using a 3D printer.

The main aim of this project is to provide easy access of simple structures such as screws, hangers, boxes, screw driver etc to the user at his home. For this purpose a printer has to be designed that can structurally replicate 3D image and provide us a solid 3D model. The printer takes input from the computer and lays down a material in successive layers to form a solid object.

1.2 MOTIVATION:

In the world of modern technology where one needs to cope up with its pace, rapid prototyping is becoming one of the major areas of interest for most of the people and so it did for us as well. Nowadays, just as everyone has a laptop in his lap, soon everyone will be having his own 3D printer used for structural replication of objects of daily use. The world has grown so fast that nobody has the time to go to the market to replace a faulty part. This generates the need for rapid prototyping and manufacturing of different parts to save time and money spent in finding and buying a replacement. The major advantages of 3D printed products are that they are fast to develop, can be customized easily and can be shared with large designs in an open source.

1.3 DOMAIN:

The domain in which 3D printer falls is the control domain as the major part in this project is of controlling the motion of the print head in three dimensions depending upon the object to be printed. Also the material has to melt to a specific temperature before depositing it, this specific temperature has to be controlled and maintained by taking feedback. Electronics is also a big part in this project as micro controllers will be used and their circuit boards and different driver boards have to be designed.

1.4 GOALS:

The goals of this project are to construct an efficient machine that performs rapid prototyping to facilitate the user. Instead of going to the market again and again to buy items of daily use the user can easily produce them using the 3D printer. The goals of our project are summarized below:

1. To achieve rapid prototyping

2. To reduce the time to market
3. Reduce development cost
4. An easy and effective method of structural replication.
5. To achieve a simpler design of the machine.

1.5 SCOPE:

The scope of this project is to be constructing a machine that can replicate structures of simple shapes such as square, triangle. Initially we do not aim for complex structures but once this printer is made it can be modified and advanced to print complicated structures. There are some structures e.g. a sphere that also need a support when they are being formed using a printer otherwise the material would collapse. This support is also created by the printer and once the object is made this support is removed. In this project we do not aim for such type of objects, this machine will just produce the objects that do not require such support.

1.6 MODULES:

The project involves a complete hardware assembly to be made that can support motors on it for 3D motion. Also the programming and interfacing of the controller is an important part of the project. So the project is a blend of hardware and software. We have divided the project in three major modules that are as follows.

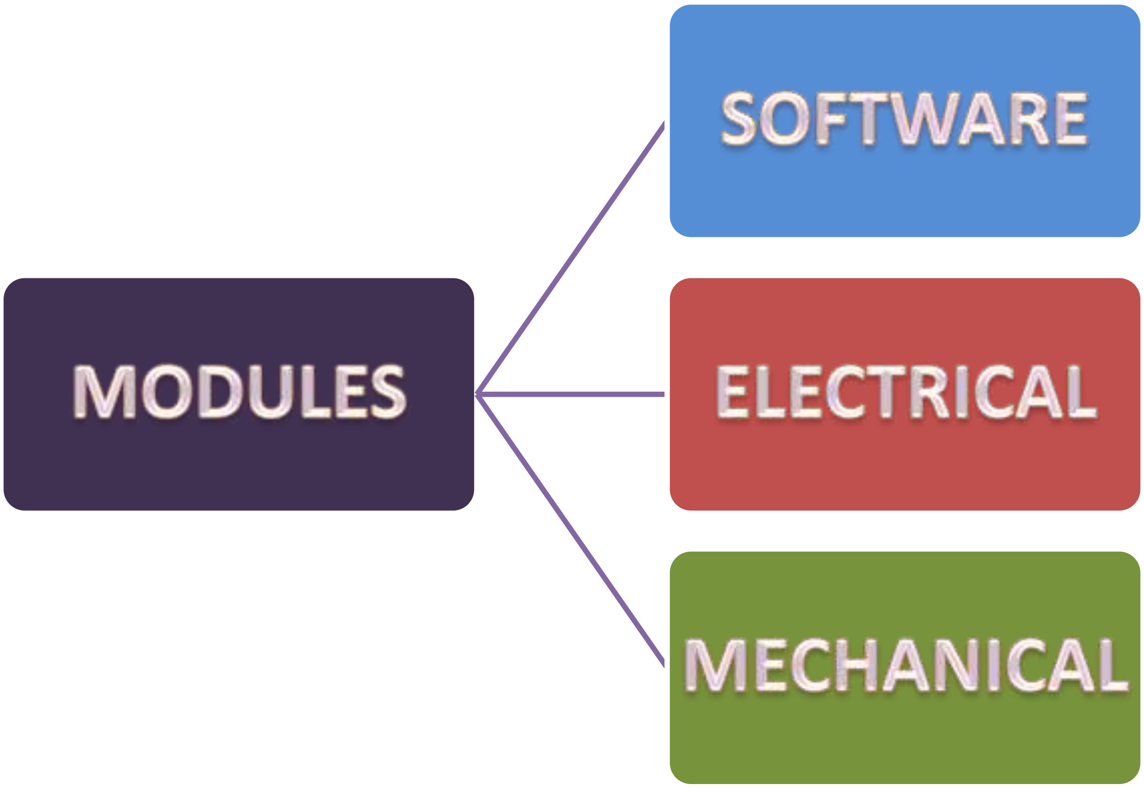


Figure 1: Project Modules

1.7 FLOW GRAM:

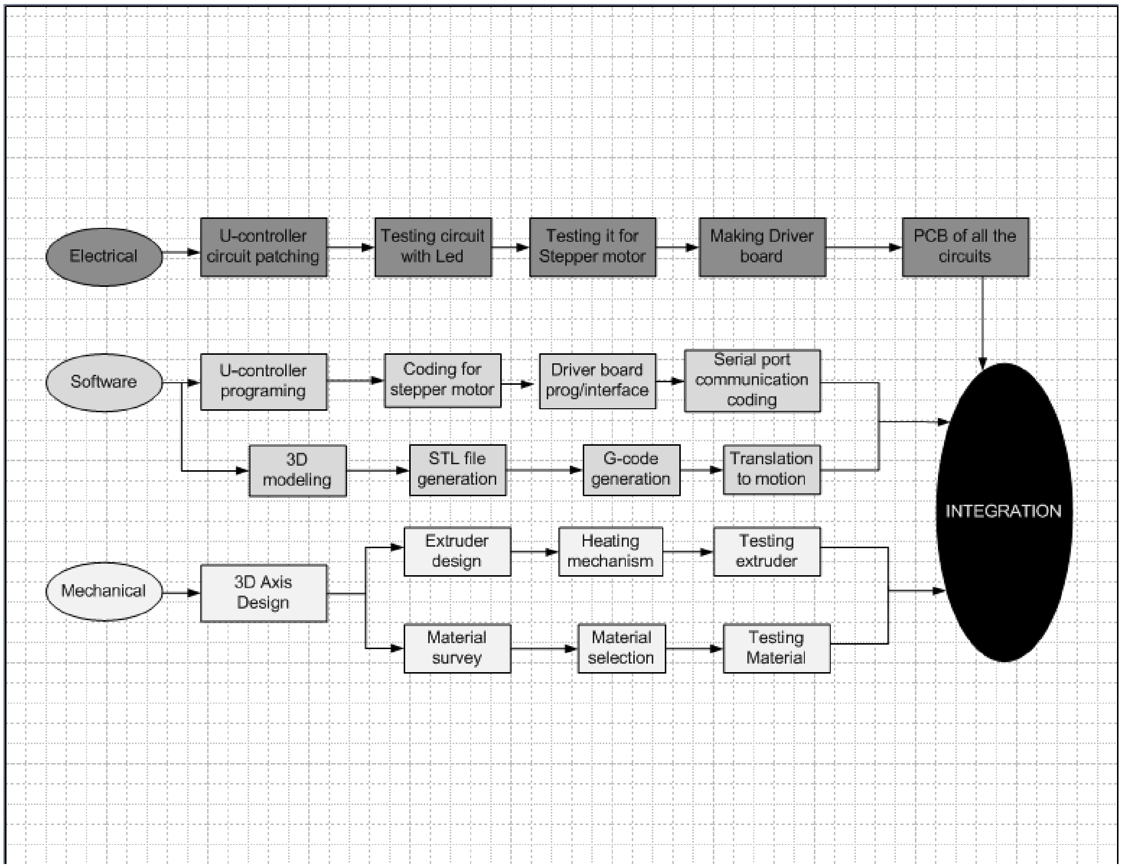


Figure 2: Flowgram

Chapter: *2*

2.0 LITERATURE REVIEW

2.1 HISTORY OF 3D PRINTERS:

In 1984 Charles Hull was the first person who gave the idea of producing 3D objects from a model and he gave this technique the name of Stereo lithography [2]. In 1986 he got the idea patent and after that he made his first 3D printing machine. The machine was delivered to a few customers and then with improvements another machine was created that was known as SLA-250. By the end of 1980s this technique had become popular.

In 1993 came another technique that was patented by Massachusetts Institute of Technology (MIT) and was known as the "3 Dimensional Printing techniques". Z-corporation got the license in 1995 to use this technique, which used a concept similar to that used a 2d printer. Z corporations then starting making 3D printers and in 2005 they launched one of their grand products which was a high definition color 3D printer. It was known as Spectrum Z. In 2006 a major contribution was done in this field by Reprap which was a project aimed at making desktop 3D printers. With this project the development of 3D printers increased and people started using these printers for their daily use. Up till now the trend of 3D printing is increasing day by day.

2.2 TECHNIQUES FOR 3D PRINTING:

Different technologies emerged to make 3D objects using printers. They are briefly discussed below:

- **Selective Laser Sintering**

In this technology laser is used to combine small particles of metal, glass or plastic. These particles fuse together and the fusion is done selectively according to the 3D model scanned. All this fusion happens on a powder bed and after the completion of one layer the powder bed is lowered.

- **Fused Deposition Modeling**

In this technique plastic material is melted and is extruded through a nozzle to form layers of the material according to the path specified by the 3d model and layer by layer forms the complete object. The nozzle can be moved horizontally and vertically and is controlled by Computer aided Manufacturing software (CAM).

- **Laminated Object manufacturing**

In this technique layers of material such as plastic or metal are coated with adhesive and are then glued together. To give the shape of the object it is then cut with a knife.

2.2.1 Reprap:

Reprap is an open source project which aims at constructing desktop 3D printers that are low in cost and can be replicated easily [1]. Plastic objects can be easily produced from this printer, even the complete 3D printer can also be printed using the Reprap machine.

2.2.1.1 Versions of Reprap:

The first version of Reprap is called the Darwin version. It is not used much and is just kept for legacy purposes. The second version of Reprap is Mendel which is of a small size so can be used as a desktop printer. Mendel has a number of improvements as compared to Darwin so it is a improved version. The improvements of Mendel are enlisted below:

- Simplified assembly
- Larger print area
- Portable and light in weight
- Higher axis efficiency

The third version of Reprap is Huxley which can also be called as Mini-Mendel as it is basically a Mendel with parts redesigned and options available to choose the firmware and host software.

The volume of Huxley parts is only 30% of the volume of the parts used in Mendel so this printer can be produced much faster. Prusa Mendel is also one of the versions of Reprap and is called the Ford Model T of 3D printers. It is one of the versions that are very easy to modify and repair.

2.2.1.2 Technique used by Reprap:

Reprap uses a technique that is a variant of fused deposition modeling. There is a thermoplastic extruder that is moved in a 3D plane made with steel rods. There are 3 axes defined and the motion of the extruder along the three axes is controlled using stepper motors. Along the x and y axis belts are used and along the z axis a lead screw is used. Stepper motors have been used instead of dc motors due their high precision due to the fact that they move in steps. Also stepper motors have the ability to start and stop quickly.

2.2.2 Makerbot Industries:

The products of MakerBot Industries are designed to be built by anyone equipped with basic technical skills. That is why they are known as ‘Do it yourself’ kits. Their products are not focused on the goal of self-replicating as in the RepRap. MakerBot printers currently print with acrylonitrile butadiene styrene (ABS) , high-density polyethylene (HDPE), polylactic acid (PLA), and polyvinyl alcohol (PVA).

2.3 SOFTWARE PART:

Software part includes:

- 3D modeling and slicing
- Gcode-translation to motion
- Coding of Micro controller (Arduino UNO)
- GRBL-Translation of G-code files to Arduino
- Serial Port Interfacing/communication

2.3.1 3D-Modelling & slicing:

3D modeling includes the specifications of the object that is to be replicated. It is the initial stage and includes specifying the dimensions of the object. A 3D image is generated in any 3D modeling software. The final object formed will be in accordance to the specification set here in the initial stage. Any changes made here will be reflected in the final product. So for a perfect product formation it is necessary to give accurate dimensions here and form a perfect design. Thickness and Elevation are the two keys factors explained here which will also decide for that how much material is to be deposited in later stages.

Thickness- It is the property that gives 3rd Dimension to the object. All the shapes get a height after providing them thickness.

Elevation- It is the distance from XY plane in the direction of Z axis. This feature actually provides the object with third dimension and makes it 3D.

There are number of softwares that are used for 3D modeling. The softwares generate different file formats depending upon the software being used.

2.3.1.1 File type extension for 3D modeling:

The following file types are used for 3D modeling generally in 3D printers.

- ***VRML or WRL*** – Similar to STL, but includes color. Different CAD systems and 3D rapid prototyping machines use these file types. In addition to it VRML models on the web use it as well.
- ***STL*** – Stereo Lithographic data format used by various CAD systems and stereo lithographic printing machines.
- ***AMF*** – Additive Manufacturing File Format

STL File Types:

STL is a file format native to the stereo lithography CAD software created by 3D Systems. Many softwares support this file format; it is widely used for rapid prototyping and computer-aided manufacturing. Only 3 dimension geometry is described by the STL files. Without any common CAD model attributes like color texture etc. ASCII and binary representations are both in STL file format. Binary files are more common, since they are more compact. These file formats are quick and easy in implementation and the binary format is very compact. For the basis of calculation many CAD need triangular models. From a Cad system the STL file format is always there and for importing triangular geometry it is a most quick.

2.3.1.2 Autocad:

Auto cad is very famous software used for 3D modeling. **Auto CAD** is a CAD software application for 2D and 3D design and drafting. It has the following features:

- 3D Free-Form Design Tools
- Auto desk Inventor Fusion
- Parametric Drawing
- Point Cloud Support
- Model Documentation Tools
- Associative Array Functionality
- PDF Support

As Autocad has more support and features for 3D modeling hence we have opted for Autocad as our design tool. The file generated by Autocad will be of STL format.

2.3.2 G-CODE:

G-code is the programming language used for Computer Numerical Control (CNC) machines to define the motion paths. In industry it is implemented in automation and 'computer-aided-engineering' [24]. G-code is a very limited language in which there are no proper structures or loops. Its format just contains alphabets and numbers which signify different types of motion for a CNC machine. It basically tells a CNC machine "what" to make and "how" to make by defining certain parameters such as:

- ❑ where to move to
- ❑ how fast to move
- ❑ through what path to move

2.3.2.1 Format:

In G-code programming there are certain words and these words signify different motion paths and different parameters. A word further has two parts:

- ➢ Letter
- ➢ Number

Letters is Gcode define different types of function

2.3.2.2 M-CODES:

Mcodes are the special type of function that are used for the movement of the 4th axis of the extruder. The Mcodes let the extruder turn on and off as required by the design of the object. While the g-code file is generated there are special functions M-codes that are also generated to control the turning on and off of the extruder.

M3 = Spindle on clock wise

M4 = Spindle on counter clock wise

```
G1 X-5.76 Y-4.392 Z0.72 F60.0
M101
G1 X-5.76 Y1.512 Z0.72 F45.6999
G1 X-2.88 Y2.952 Z0.72 F45.6999
G1 X-2.88 Y-4.392 Z0.72 F45.6999
G1 X0.0 Y-4.392 Z0.72 F45.6999
G1 X0.0 Y2.952 Z0.72 F45.6999
G1 X2.88 Y2.952 Z0.72 F45.6999
G1 X2.88 Y-4.392 Z0.72 F45.6999
G1 X5.76 Y-4.392 Z0.72 F45.6999
G1 X5.76 Y1.512 Z0.72 F45.6999
M103

G1 X-5.796 Y0.0 Z1.18 F960.0
M101
G1 X5.796 Y0.0 Z1.18 F69.6088
G1 X5.796 Y-1.44 Z1.18 F69.6088
G1 X-5.796 Y-1.44 Z1.18 F69.6088
G1 X-5.796 Y-2.88 Z1.18 F69.6088
G1 X5.796 Y-2.88 Z1.18 F69.6088
G1 X5.796 Y-4.32 Z1.18 F69.6088
G1 X-5.796 Y-4.32 Z1.18 F69.6088
M103
G1 X-5.796 Y1.44 Z1.18 F960.0
M101
G1 X5.796 Y1.44 Z1.18 F69.6088
M103
G1 X2.916 Y2.88 Z1.18 F960.0
M101
G1 X-2.916 Y2.88 Z1.18 F69.6088
M103

G1 X-5.796 Y0.0 Z1.58 F613.8795
M101
G1 X5.796 Y0.0 Z1.58 F69.0523
G1 X5.796 Y-1.44 Z1.58 F69.0523
```

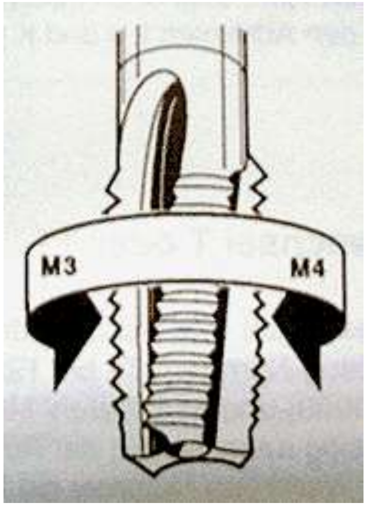


Figure 3 Spindle movement [33]

G1=Straight line

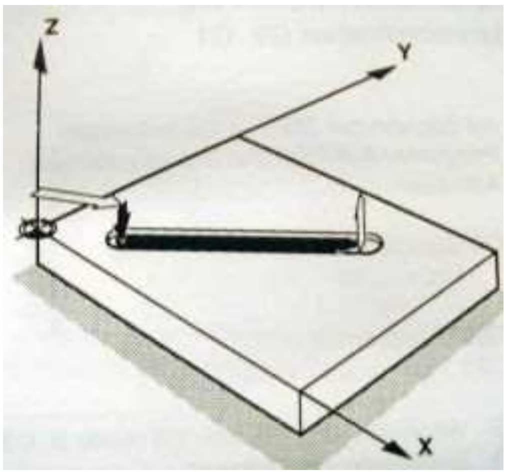


Figure 4 Linear motion in Gcode [33]

The following table contains the Letters and their intended functions.

Letter	Meaning
A	A axis of machine
B	B axis of machine
C	C axis of machine
D	Tool radius compensation number
F	Feed rate
G	General function
H	Tool length offset index
I	X offset for arcs and G87 canned cycles
J	Y offset for arcs and G87 canned cycles
K	Z offset for arcs and G87 canned cycles. Spindle-Motion Ratio for G33 synchronized movements.
M	Miscellaneous function
N	Line number
P	Dwell time in canned cycles and with G4. Key used with G10.
Q	Feed increment in G73, G83 canned cycles
R	Arc radius or canned cycle plane
S	Spindle speed
T	Tool selection
U	U axis of machine
V	V axis of machine
W	W axis of machine
X	X axis of machine
Y	Y axis of machine
Z	Z axis of machine

Table 1 Functions of letters in G-code

The numbers preceding the letters specify further different sub-functions. For example [25]:

- Coordinate system selection: G54, G55, G56, G57, G58, G59, G59.1, G59.2, G59.3
- Spindle on or off : M3, M4, M5
- Coolant on or off: M7, M8, M9

- Set length units: G20, G21
- Perform motion: G0 to G3, G33, G73, G76, G80 to G89
- Stop: M0, M1, M2, M30, M60

2.3.2.3 Skeinforge-Gcode Generator:

Skeinforge is software that contains a tool chain composed of ***Python scripts*** that converts 3D model into G-Code instructions for a 3D Printer. It takes a 3D image file as an input and then performs certain functions on it to convert it into a G-code file.

2.3.2.3.1. File formats:

Skeinforge takes the following files as input [26]:

- .gts Gnu Triangulated Surfaces
- .obj Wavefron 3D OBJ
- .stl Stereolithography
- .svg Scalable Vector Graphics
- .xml Extensible Markup Language

2.3.2.3.2. Installation:

For running Skeinforge there are some softwares that need to installed:

1. Python needs to be installed in the first place as Skeinforge runs the scripts that are written in Python. We are using the Python version 2.7.2.
2. After installing python there is optional software known as Psyco that is used to speed up Python execution which normally takes a lot of time to execute.
3. After installing the preliminary softwares Skeinforge can be installed.

The interface of this software contains two windows, one is the preferences window and the other one is the black shell window.

Figure 5 Black shell window

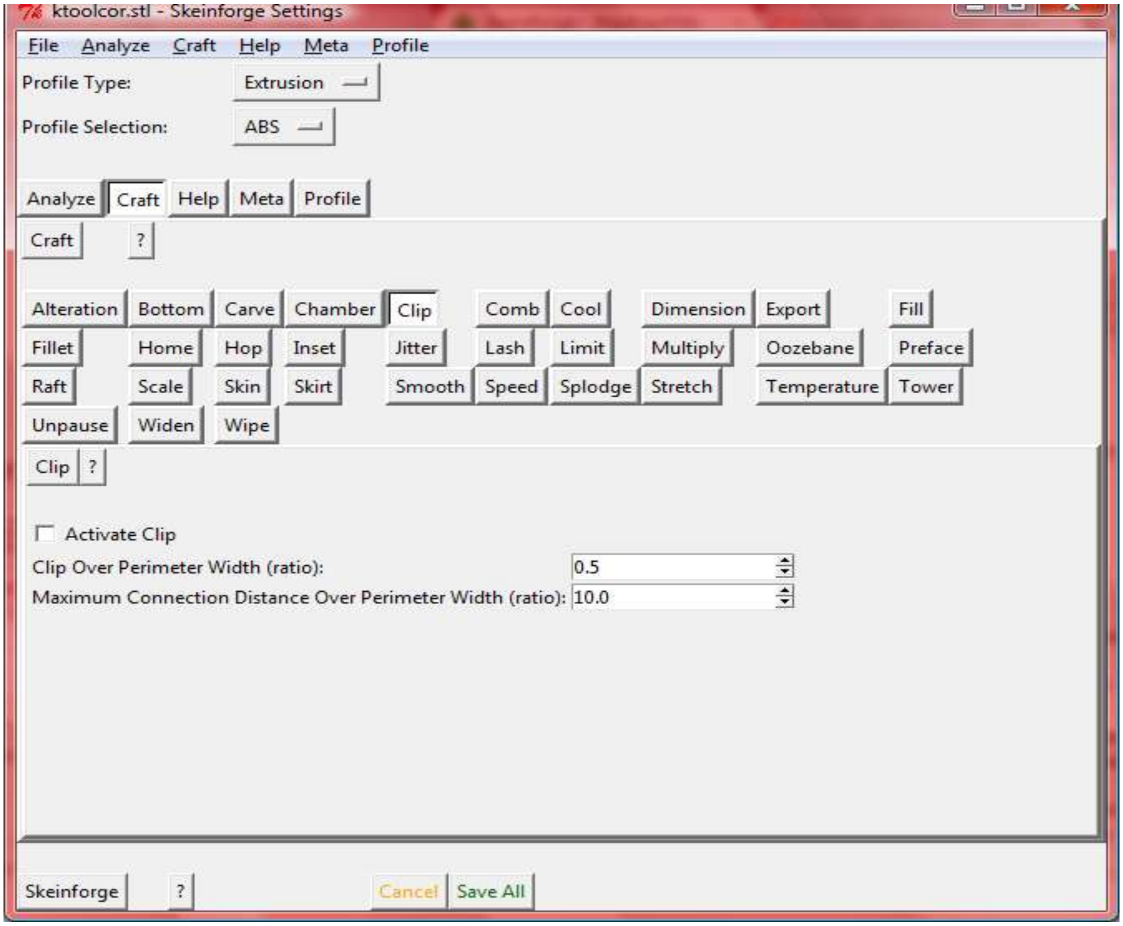


Figure 6 Preferences window

In the preferences window there are a number of settings which we can configure according to our need. Few of the settings are listed below:

- Carve
- Chamber
- Clip
- Comb
- Cool
- Dimension
- Export
- Fill
- Fillet

In the black shell each of the command selected in the preferences window is visible being executed.

There are four types of profiles in this software:

- Extrusion
- Winding
- Milling
- Winding

In our project we will select the Extrusion option as we have to extrude plastic material out of the nozzle. Also there are options to select the plastic material. The available options for material are:

- ABS
- PCL
- PLA
- HDPE

We are using the ABS plastic. With the help of Skeinforge button in the preferences window .stl file is selected and on the black shell window the proceeding are seen of the conversion of the .stl to Gcode. Once the Gcode is formed it is stored at the same location where the .stl file was located.

2.3.3. MICRO CONTROLLER PROGRAMMING

The **Arduino programming environment** is written in Java. It provides with a syntax which is very much similar to C++. Its environment is compatible with Windows, Mac OS, and Linux. The code can be compiled and uploaded to the Arduino board easily using this environment. For programming the Arduino two functions have to be defined.

setup()

loop()

2.3.3.1. PROGRAMMING STRUCTURE:

The programming structure has two parts:

```
void setup()
{
statements;
}
void loop()
{
statements;
}
```

In the setup() function all declarations are made . The pins are declared as inputs and outputs in this function. In the loop() function all the major work is done and the code is executed continuously. Let us see an example:

```
#define LED_PIN 13

void setup () {
 pinMode (LED_PIN, OUTPUT);    // enable pin 13 for digital output
}

void loop () {
 digital Write (LED_PIN, HIGH);  // turn on the LED
        delay (1000);                 // wait one second (1000 milliseconds)
```

```
digital Write (LED_PIN, LOW);   // turn off the LED
        delay (1000);                // wait one second}
```

This is a simple program to turn the LED connected to the pin13 of the controller on and off with a delay of 1s. In the setup() function the pin13 is declared as output by using the pin mode command. Using the same command the LED is given a zero signal after a delay of 1000ms. The loop() function then repeats again and again.

2.3.4. GRBL:

Grbl is free, open source, high performance **CNC** milling controller written in optimized C that will run on a straight Arduino. GRBL provides an alternate to the Arduino programming each time for controlling the 3 axes movement with the help of motors. Instead of writing a code in Arduino the GRBL is configured and then GRBL.HEX is uploaded to the controller after setting some specifications. The Gcode generated must now be sent to the Arduino micro controller for the movement of the axis. But the controller has its own language and does not understands the G-code itself. Hence GRBL firmware is written. It is a C code written in AVR studio which explains the meaning of the G code commands. Each time a Gcode and command is sent the GRBl interprets it and sends specific signals. The firmware is a collection of C and header files.

CONFIG.h, GCODE.h, EEPROM.h, MAIN.c, GCODE.c, CONFIG.c. All the setting of the commands that must be set and read by the controller are specified in Gcode.c. The main.c file contains all the functions and their calls required to generate the signals for the movement of the motors of all the axes. Config.h contains the pin configs of all the connection between the motors and Arduino.

The complete process is described shown below:

i. Setting up Arduino
ii. Configuring GRBL
iii. Extract the G-code generated for the sketches
iv. Uploading code to Board

2.3.4.1. Configuring GRBL

2.3.4.1.1. G-code sender:

GRBl is configured in the G-code sender. It is a GUI that connects to the micro controller Arduino board, and then sends a G-code file to it. It has a visual representation that shows the progress. Moreover it also shows what lines it has already sent and the replies back from the Arduino.

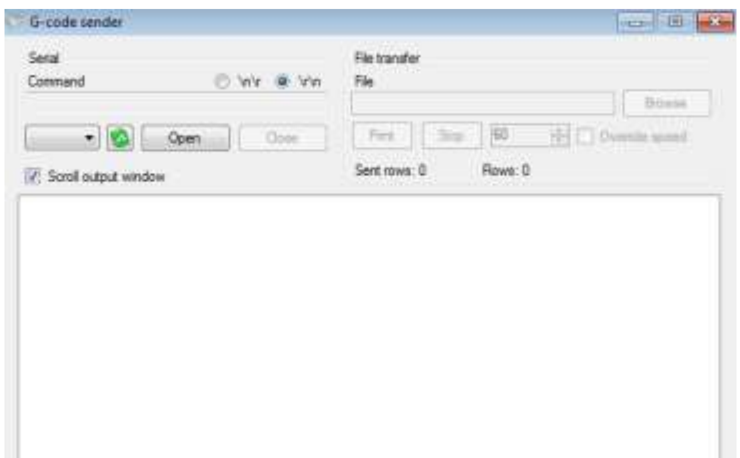


Figure 7 Gcode sender

i. Select COM port and press Open.
ii. Enter a $ sign in the command box.
iii. You will get all the settings for GRBL.
iv. You can test out your stepper motors by entering Gcode.

By Configuring GRBL it is meant that the GRBl needs to know how far each step will take the tool in reality. The settings are stored in EEPROM and will be retained until you change them.

2.3.4.1.2. Parameters:

$0, $1 and $2	**– Steps/mm;**
$3	Microseconds/step pulse
$4	Default feed rate
$5	Default seek rate
$6	Mm/arc segment
$7	Invert mask
$8	Acceleration
$9	Max instant cornering speed change

Table 2 GRBL Configuration commands

To calculate steps/mm for EACH axis of your machine, you need to know:

- turns per mm of lead screw
- full steps per revolution of steppers (typically 200

- Micro steps per step of controller -GRBL assumes it is 1/8 micro step driver.

And then the steps/mm can then be calculated as follows:

steps_per_mm = (steps_per_revolution*microsteps)/turns_per_mm

2.3.4.1.3. Extract the G-code generated for the sketches:

G-codes that are generated using skeinforge as mentioned above are added to the G-code sender environment and then the process is completed. After the process has been completed a GRBl. Hex file is generated which is used by the microcontroller.

2.3.5 Serial Port Communication Coding:

For communication, serial port communication is a physical interface through which information can be transferred in form of bit from and through computer. Data transfer through serial ports connects the computer to devices such as terminals and various peripherals and this method of communication has been used through ages. Nowadays, computers usually do not have a serial port for communication and is replaced by usb port, still a serial-to-USB converters allow compatibility with RS 232 serial devices. Serial ports are used in applications such as industrial automation systems, scientific instruments, shop till systems and some industrial and consumer products. Serial ports are still used in these areas as they are simple, cheap and console functions are standardized. A serial ports simpler and requires very little supporting software from the host system which reduces its complexity.

It is also used for communication between the Arduino board and a computer or other devices. All Arduino boards have at least one serial port also known as a UART or USART. It communicates on digital pins 0 (RX) and 1 (TX) as well as with the computer via USB. So if these pins are used for communication then we cannot also use pins 0 and 1 for digital input or output. Arduino environment has a built-in serial monitor to communicate with an Arduino board.

A sample program for serial communication for data in and out transfer

```
////////////////////////////////////////////////////////////////////////////////
Programing structure:

void setup()
{
Serial.begin(9600); // open the arduino serial port
}


void loop()
{
if(Serial.available()) // check to see if there's serial data in the buffer
{
serialvalue = Serial.read(); // read a byte of serial data
started = 1; // set the started flag to on
}
if(started) { // loop once serial data has been received
randomvalue = random(1000); // pick a new random number
Serial.print(countervalue); // print the counter
Serial.print(" "); // print a space
Serial.print(randomvalue); // print the random value
```

```
Serial.print(" "); // print a space
Serial.print(serialvalue); // echo the received serial value
Serial.println(); // print a line-feed
countervalue = (countervalue+1)%1000; // increment the counter
delay(100); // pause
}
}
////////////////////////////////////////////////////////////
```

2.4 ELECTRICAL PART:

2.4.1 Microcontroller:

Microcontroller is an integrated chip containing a processor, memory and peripherals. Most of the microcontrollers are programmable thus increasing the number of applications in which they can be used. The main function to use a microcontroller is to have a low cost device that can be rapidly programmed and is used to control different events. “In short the microcontrollers are the heart and soul of many everyday appliances.” [8]

There are a number of microcontrollers available in market e.g. ARM processors, Atmel AVR, Power PC, PIC and many others. All of these

microcontrollers perform the function of taking input from the sensors and controlling the desired output. In the following diagram the interfacing of a microcontroller is shown

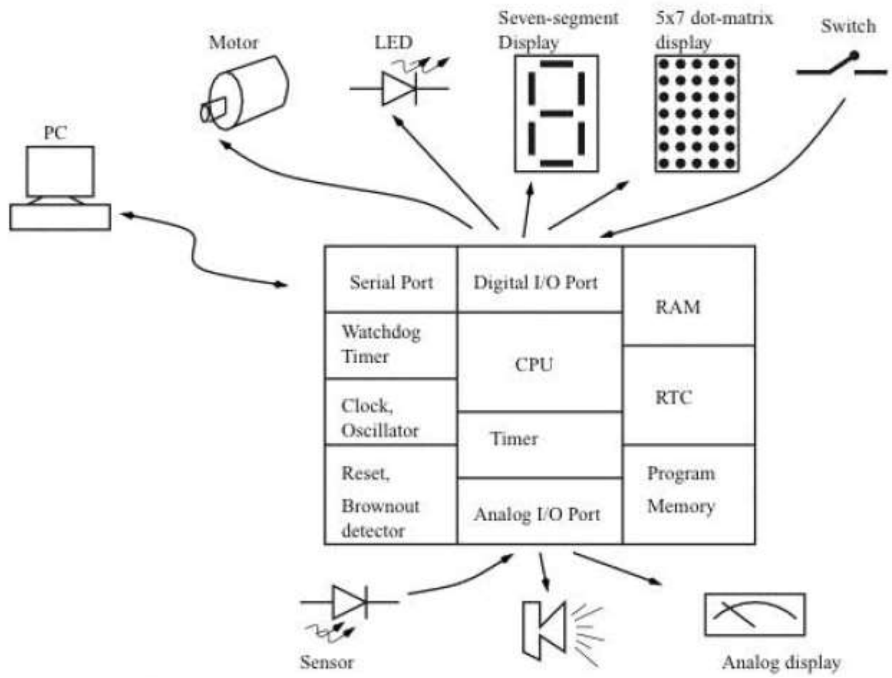


Figure 8 Microcontroller applications [9]

2.4.2 Arduino:

Arduino is an open-source platform for controlling different parameters by receiving inputs from different sensors. The hardware for this microcontroller consists of a board with AVR Atmel processor and peripherals for I/O. The Arduino project is a descendant of Wiring platform project which was a project done as a master thesis at *Interaction Design Institute Ivrea* and it involves processing using electronics[13]. It can be used to make a lot of interactive projects. The board is can be made or can also bought.

2.4.2.1 Advantages:

The reasons for selecting this microcontroller are the numerous advantages of Arduino over other controllers such as 89C51, PIC etc.

1. It provides easy programming and we do not have to deal with bits masks which are a little complicated to use.

2. I/Os can be easily read and controlled.

3. Power and reset circuitry is already available on the Arduino board.

4. It allows serial communication.

5. Arduino has a number of libraries which makes the programming very easy.

2.4.2.2 Hardware:

The hardware contains a 8-bit AVR Atmel processor such as Atmega8, Atmega168, Atmega328 or Atmega1280. On the board other components are present for programming and I/O. A 16MHz crystal is used by this controller and the board also contains a linear voltage regulator. The board is programmed using a RS-232 connection. There are thirteen versions of Arduino available in the market few of which are enlisted below:

Ø Arduino Uno

Ø Arduino Mini

Ø Arduino Mega

Ø Arduino Serial

2.4.2.3 Arduino UNO:

The Arduino Uno is a microcontroller board based on the ATmega328. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

- Current limiter resistor in pin13
- TX and RX onboard LEDs

It has the following schematic.

Figure 9 Microcontroller circuit

ATEMGA328 has been used in this board.

2.4.2.4 ATMEGA328:

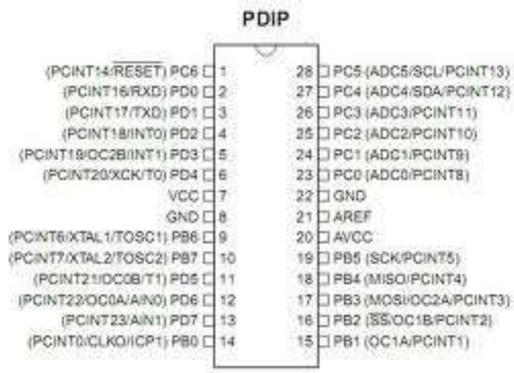


Figure 10 Pin outs of Atmega328 [11]

Atmega328 belongs to the AVR series of low power processors and has a RISC based architecture [12]. This processor gives a throughput of 1MIPS for 1MHz thus optimizing the power consumptions. Some features of this processor are enlisted below:

➢ Microcontroller	ATmega328
➢ Operating Voltage	5V
➢ Input Voltage (recommended)	7-12V
➢ Input Voltage (limits)	6-20V
➢ Digital I/O Pins	14 (of which 6 provide PWM output)
➢ Analog Input Pins	6
➢ DC Current per I/O Pin	40 mA
➢ DC Current for 3.3V Pin	50 mA
➢ Flash Memory	32 KB (ATmega328) of which 0.5 KB used by bootloader
➢ SRAM	2 KB (ATmega328)

- EEPROM 1 KB (ATmega328)
- Clock Speed 16 MHz

2.4.3 UART:

UART stands for Universal Asynchronous Receiver/ Transmitter and is used for transmitting data in serial form. The data is transmitted bitwise serially. The fundamental method that is used to convert data to serial form is by using a shift register. This mode of transferring data is much more cost effective than using parallel transmission. The main characteristics of UART that it is asynchronous i.e. no clock signal is transmitted and that it has a relatively simple hardware.

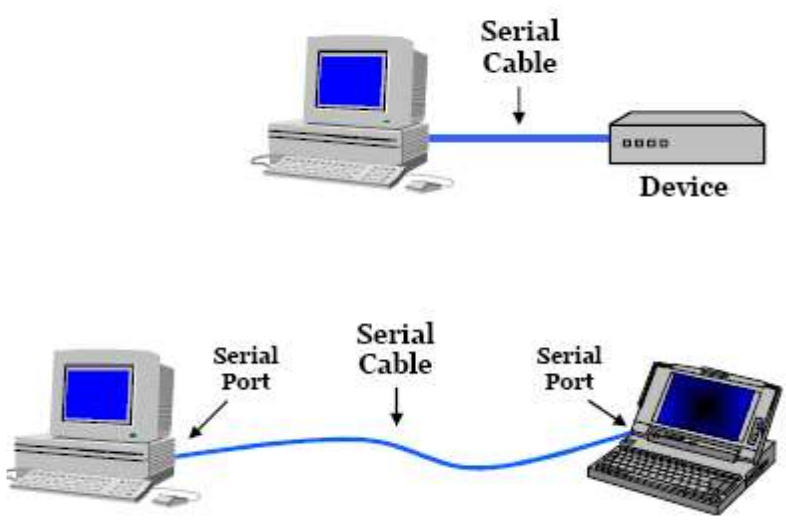


Figure 11 Serial port communication [14]

2.4.3.1 RS 232:

The following diagram shows the pinout configuration of an RS-232 connector used for serial communication.

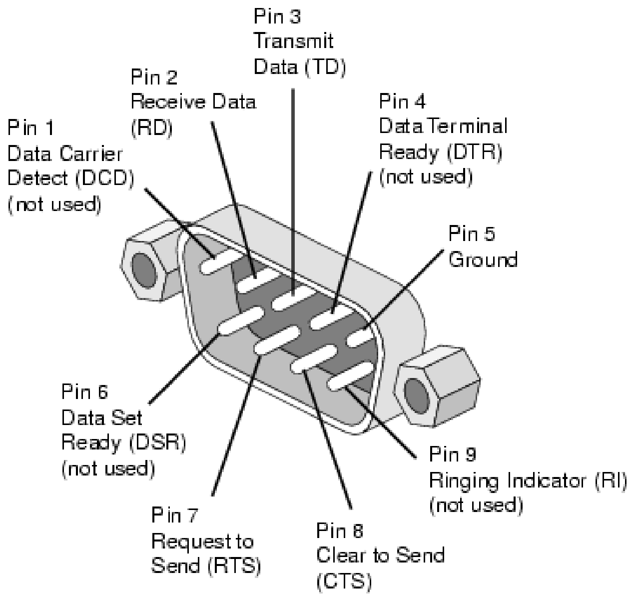


Figure 12 RS32 [15]

The pin out description of RS-232 is given below:

• **TD:**

Serial data output line

• **RD:**

Serial data input line

• **RTS:**

It is a handshake line. Indicates that transmitting device is **ready to send**

• **CTS:**

It is a handshake line. Indicates that receiving line is **ready to receive**

• **DTR:**

It is output from computer. Indicates that data terminal is **ready/ ON**

• **DCD:**

It is input to computer. Indicates that the other device is **ready / ON**

2.4.4 Motors:

Motors have to be used for controlling the motion of the extruder along the three axes. There are two types of motors from among which we had to select one:

- DC Motors
- Stepper Motors

2.4.5 Dc Motor:

It is a motor that runs on dc electricity. .This is accomplished by forcing current through a coil and producing a magnetic field that spins the motor. The three main parts of this motor are the internal commutation, rotating electrical magnets and stationary magnets. In this type of motor brushes are connected with the ends of a coil and make a temporary connection with a voltage source. After every 180 degrees the contacts of the brushes are exchanged. . This interaction results in a force to be exerted on each segment, resulting in a torque to be generated, causing the coil to rotate.

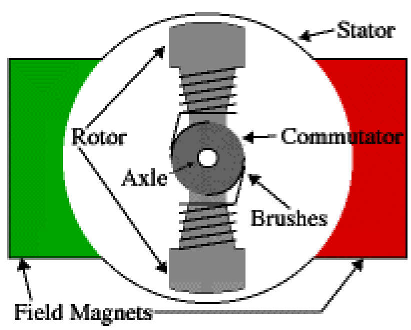


Figure 13 DC Motor

2.4.6 Stepper Motor:

A stepper motor has a rotor which moves in steps. Resolution is the steps per revolution or degrees travelled in one step for a stepper motor. Generally the most common stepper motors have 200 steps per revolution which gives us 1.8 degree per step resolution. There are two types of stepper motors:

- ***Unipolar Stepper motor:***—A unipolar stepper motor has generally six wires. Internally there are two coils in the unipolar stepper motors with the center tap.
- ***Bipolar Stepper motor:*** — A bipolar stepper motor has generally four wires coming out of it. There are two coils inside the bipolar stepper motor without a center tap.

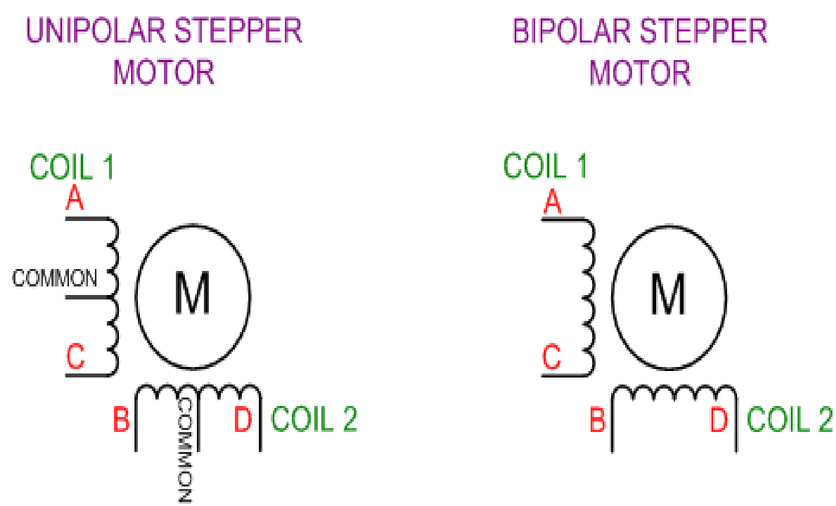


Figure 14 Stepper motor types

2.4.6.1 Driving the motor:

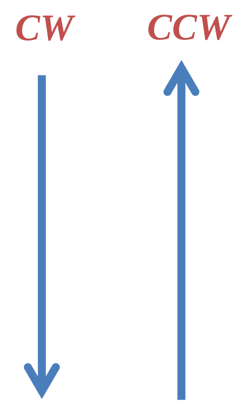
In order to drive the unipolar stepper motor, we need to control the sequence of the four wires coming out of the motor. We apply the power to the common terminals.

The control of bipolar stepper motors is a little difficult than the unipolar motors. We need to change the direction of the current in the windings in order to drive the bipolar stepper motor. Since there are two windings, we need to use two H-bridge circuits in order to control one bipolar stepper motor. IC L298 contains dual H-bridge circuits and can be used to drive a bipolar stepper motor or two DC motors alternatively.

There is a certain sequence of control signals that should be given to the motor by the microcontroller for motion in the clockwise or anticlockwise direction.

2.4.6.2 Driving sequence for bipolar motor:

COIL 1 Current Polarity	COIL 2 Current Polarity
+	+
-	+
-	-
+	-
+	+



2.4.6.3 Driving sequence for unipolar motor:

A	B	C	D
1	1	0	0
0	1	1	0
0	0	1	1
1	0	0	1
1	1	0	0
0	1	1	0
0	0	1	1

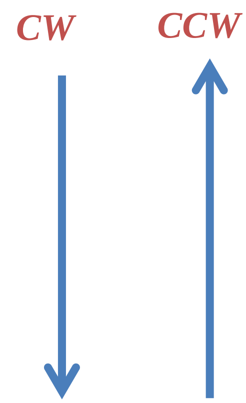


Table 3 Motor control sequence

2.4.6.4 Advantages of a Stepper Motor:

- Each step has a rotation angle that is proportional to the input.
- Stops and starts quickly.
- Full torque when the motor is at standstill.
- This motor has the speed proportional to the frequency of input so a wide range of speeds can be realized.
- High precision and repeatability of motion.
- Reliable motor as the lifetime of the motor just depends upon the life of bearing.

2.4.7 Motor Selection:

In our project we are using the *Stepper motor* as we want the quick response of the report by quick starting and stopping. Also the precision required is very high as the 3D object created should be of a defined and regular structure.

We will use a *Bipolar stepper motor* as the windings in a bipolar motor are better utilized and they are more powerful than a unipolar motor.

2.4.7.1 Comparison of Dc and Stepper Motor:

No.	DC MOTOR	STEPPER MOTOR
1	Relatively cheap and easily available in the market in various specifications, different power ratings and torque	These are relatively expensive and are not available in the required specifications and high torque easily
2	Comparatively easier control circuitry	Difficult control circuitry
3	To fill the requirement of knowing the location of the extruder at all times, a Feedback system would be required	The number of steps, and hence the fractional rotations the motors revolves when translated into linear motion can determine the location of the extruder.

4	Due to the need of the feedback system, it would require a high frequency of operation.	Due to no compulsion of feedback, the system requires a relatively low frequency to operate.

Table 4 Motor comparison

2.5 MECHANICAL PART:

2.5.1 Sliding Contact:

The mechanical structure needs to have minimum lag and friction. This requires a suitable sliding mechanism for the sliding contacts. The mechanism must provide for maximum efficiency, precision and accuracy as the motion needs to be precisely streamlined. The following options were considered for this purpose.

2.5.2 Sliding Windows:

Sliding aluminum windows provide a simple daily life solution for this problem. The contacts consist of small balls in order to reduce the friction in the contacts. They also help to keep the motion streamlined and balanced.

Visiting a dealer and a brief study showed that this mechanism, however cheap, comes with some issues. The reason is that it offers much more friction than we can afford. Moreover, the requirement for the motion to be straight and streamlined is dependent on the length of the contacts and the weight it supports. In our case, the weight was much lesser than used usually and the length of the contacts was also small. Thus, it was predicted to lose stability while moving. Thus, it's a tradeoff between stability and cost.

2.5.3 Ball Bearing:

Figure 15 Ball Bearings [17]

A rolling element using a ball to maintain separation between moving parts of a bearing is called a ball bearing. It is an engineering term and is used to support axial and radial loads, and reduce rotational friction. At least two races are used for transmitting the loads through the balls and to contain the balls where one of them is held fixed. The coefficient of friction would be higher if two flat surfaces were rotating, but, the rolling of the ball bearings reduces it.

In the ball bearing, the outer race transmits the load to the ball and the ball transmits the load to the inner race. The spherical shape of the ball bearing reduces the area of contact to very small points. This also helps the ball bearing to spin smoothly. However, if the weight of the load is large, it can result in deforming or squishing of the ball and thus, ruining the bearing. The reason for the deforming is that the contact area is very less and thus, the weight is distributed on very small areas. Ball bearings are the least expensive types compared to other bearings, because of the low cost of production of the balls being used. **[18]**

2.5.4 Translating Rotational Motion to Linear Motion:

As the motors produce rotational motion, it needs to be translated into linear motion that is required to cover the slide on each axis. This can be done in several ways, discussed in brief detail below:

2.5.4.1 Lead Screw:

A lead screw is often called a power screw or a translation screw and it is used to translate rotational motion into linear motion.

A lead screw has a relatively higher friction compared to parts that mate with rolling surfaces such as bearings, due to the reason that it mates with rubbing surfaces. The efficiency of a lead screw is typically somewhere between 25 to 70%. The screws with higher pitch are more efficient than those with lower pitch. The ball screw is a higher performing alternative but it is also more expensive.

Figure 16 Lead Screw [19]

The lead screw systems are not capable of working at high speeds for long periods of time because of the high internal friction, which causes them to overheat. They are often used in applications where back driving is unacceptable, like holding vertical loads or in hand cranked machines. This is because of the inherent high friction which causes the typical screw to self-lock.

Lead screws need to be used well-greased. However, an appropriate nut can also be run on the screw dry, even with higher friction. Manufacturers often specify the choice of screws and nuts as a combination.

Backlash in a lead screw can be reduced by cutting the nut along a radius and clamping that cut back together to create a static loading force, called preload. Or, a second nut can be used for this purpose.

The advantages of a lead screw are:

- Large load carrying capability
- Compact
- Simple to design
- Precise and accurate linear motion
- Smooth, quiet, and low maintenance
- Minimal number of parts
- Most are self-locking

The disadvantages are:

- Not very efficient.
- High degree for friction on the threads, which can wear the threads out quickly.
- Need maintenance and oiling. **[20]**

2.5.4.2 Ball Screw:

Another mechanical device for translation rotational motion into linear motion is a ball screw. A spiral raceway is provided for ball bearings by a threaded shaft and it acts as a precision screw. Ball screws are able to apply and withstand high thrust loads with minimum internal friction. They are suitable for use in situations requiring high precision because they are manufactured to close tolerances. The threaded shaft acts as a screw whereas the ball assembly acts as the nut. There is a need for a mechanism that re circulates the balls, and it makes the ball screws bulky as compared to conventional lead screws.

Figure 17 Ball Screw [21]

Contamination with dirt and abrasive particles need to be avoided at all costs because they pose a hindrance in maintaining the inherent accuracy and in long life of ball screws. Ball screws effectively eliminate backlash between input and output due to their ability to operate with some preload.

A ball screw provides higher mechanical efficiency compared to alternatives because of low friction. Generally, ball screws can be efficient up to 90 percent, which is much higher as compared to the 50 percent accuracy of lead screws of equal size. The cost of ball screws is higher but it can be compensated by lower power requirements for equal performance. **[22]**

2.5.4.3 Belts:

A simpler and less complex way of achieving linear motion using motors is through the use of timing belts. A free pulley is attached to one end of the motor and the rotor of the motor is attached to the other end. In order to move it linearly, the belt can be attached to the sliding mount at any point.

Different sizes and strengths of belts are available in the market. The material used in the manufacturing of the belt decides the strength of the belt. Little slits are made on the surface of the belts, which are called teeth of the belt. The teeth help in ensuring grip of the belt on the pulley and the motor. Exactly matched teeth sizes for both the motor and the belt are used for maximum grip. The belt should be selected keeping in view the load it will be used to handle.

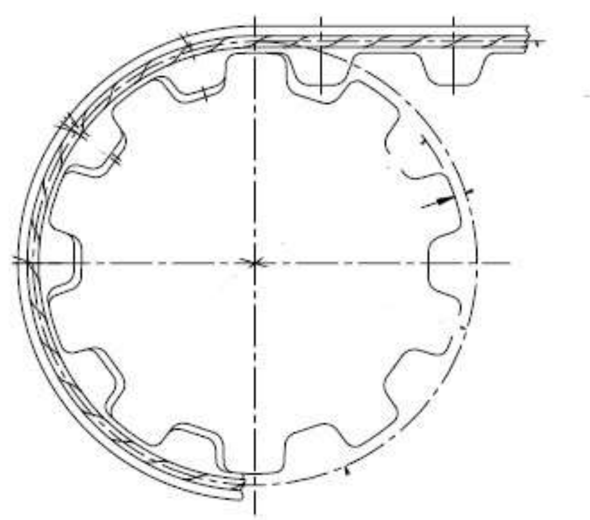


Figure 18 Gear and Belt [23]

For not so heavy loads, belts are a relatively cheap alternative and provide very accurate performance.

2.5.5 Summary and Comparison:

The following table summarizes the advantages and disadvantages:

	Lead screw	**Ball screw**	**Belts**
Cost	High	High	Low
Weight	Heavy	Heavy	Light
Efficiency	Low	High	Reasonable

Table 5 Summary and Comparison

2.5.6 Dimensions:

The dimensions of the printer will have a high impact on the design considerations. The width and length of the structure govern changes in the weight the sidebars will have to handle. This will require in a change of material of the side bars, requiring more strength from the side bars. Furthermore, this change will increase the weight of the bars too, and it will require a larger torque from the motors.

2.5.7 Heaters/Heating Mechanism:

The heating mechanism is dependent upon the type of material used. Heater can be of any kind but its size and temperature material are of prime importance. It is very important that the heaters are in accordance with the motor movement and control and they have a quick change of temperature so as to settle the material deposited and to get switch off so that no longer the material purges out.

2.5.7.1 Block heater:

This device uses a resistor embedded in a metal block to heat the extruder nozzle. The resistor is at the left of the picture. The block has a thermistor embedded in it to measure its temperature - the thermistor is the small device with the fine wires extending from it. Both of the resistors listed above have a power rating of about 3 watts. You are going to run them at about 20 watts, but don't worry - the power rating given is for when they run in air and you are going to run them in a metal block which will take the heat away. They also have a top temperature of 200ºC and you are going to run them a few tens of degrees hotter.

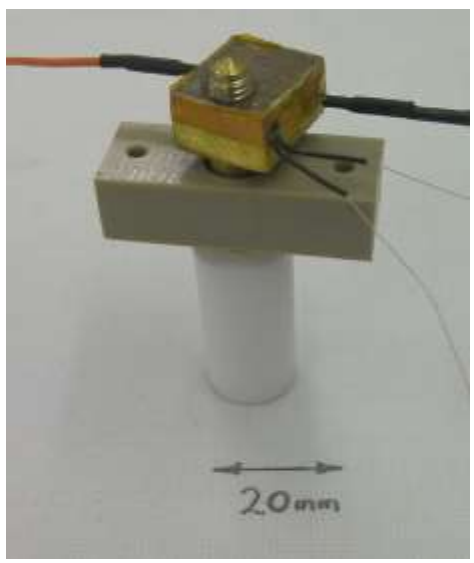


Figure 19 Extruder heating mechanism [16]

2.5.7.2 Domestic Heating Mechanism:

There is a special type of heating mechanism used in small cities in cottage industry specially Gujranwala. It is an extruder type assembly used to melt materials and deposit it. We carried out its survey and studied its mechanism if similar one could be used in our extruder assembly. This mechanism works with on gas or electricity for heating up and setting the required temperature.

2.5.7.2.1 Principle:

Small beads of the plastic material are poured in from the top. The pressure of the Gas or the electric power is adjusted so as to set the required temperature. After melting, the material is purged out of the nozzle downwards and it sets down immediately as well.

The size of the nozzle and the temperature variations is subjective in nature and depends on the material to be used.

Below are shown some of the shots of the ***extruder assembly.***

Figure 20 Domestic Extruder

Figure 21 Gas connection Domestic Extruder

Figure 22 Nozzle-Domestic Extruder

Chapter: *3*

3.0 FUNCTIONALITY AND DESIGN

In this section the complete architecture and the design of the 3D printer will be discussed.

3.1. SYSTEM LEVEL ARCHITECTURE:

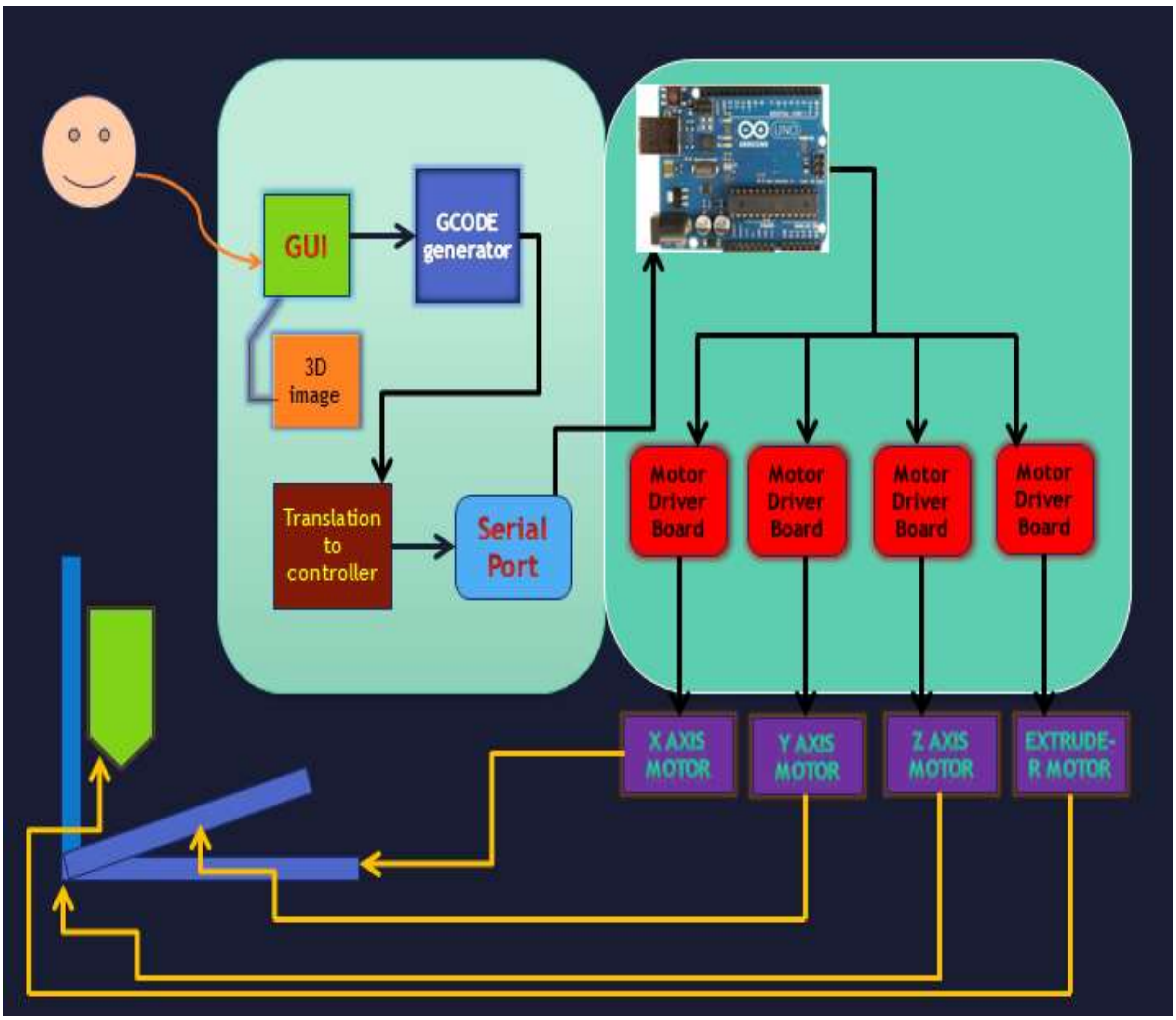


Figure 23 Architectural Level Diagram

3.2. ELECTRICAL:

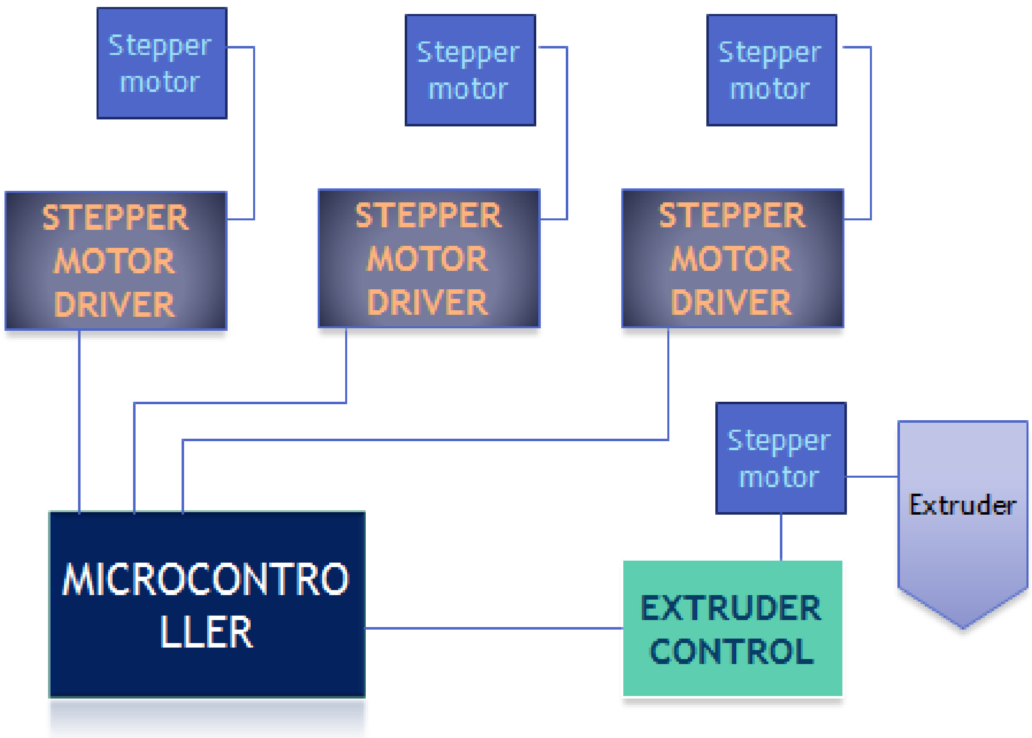


Figure 24 Electrical part

3.2.1. DRIVER BOARD FOR STEPPER MOTOR:

The stepper motor requires a proper sequence to run and also a regulator to regulate the current flowing in the motor. For this purpose the driver circuit of the motor contains two ICs: L297 and L298. L297 is the sequence generator IC and L298 is the current regulator.

3.2.1.1. L297:

It generates the sequence to run the Bipolar stepper motor. The main advantage of using the L297 IC is that it contains two chopper circuits which help in attaining high speed of motor without a lot of heat dissipation in the current regulator. The diagram contains the pin configuration of the IC:

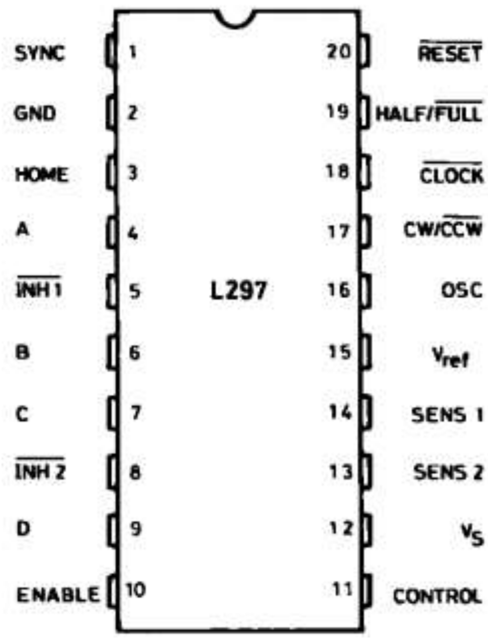


Figure 25 Pinouts of L297

On the pin 18 clock signal is given and the frequency of the clock decides the speed of the motor. For a stepper motor there are two types of stepping, Half step and Full step. In half step stepper motor moves 0.9 degrees in a single step and in full stepping the motor moves 1.8 degrees.

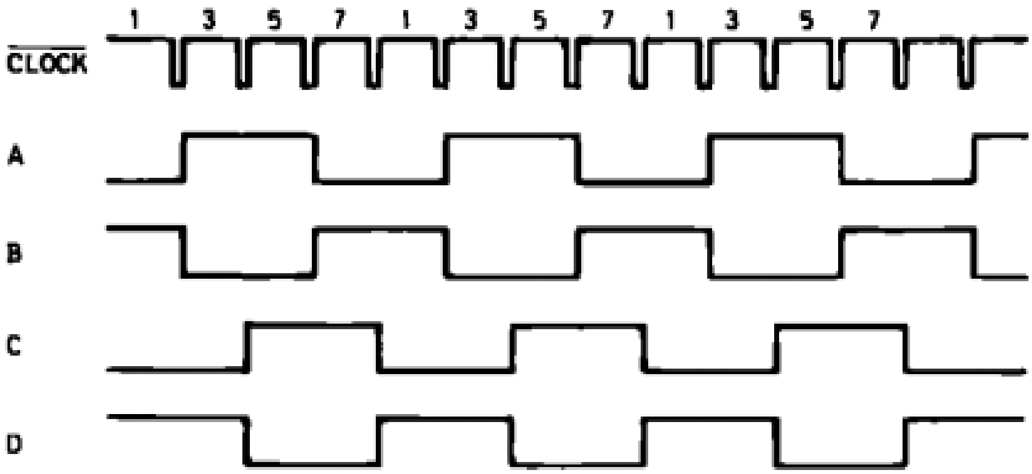


Figure 26 Output waveforms of L297

Sense1 and Sense2 inputs provide a protection mechanism when the current exceeds a certain limit. The output is switched off when the limit is exceeded and then after certain delays determined by the chopper rate the output is switched on again. With this switching the motor operates on a constant current.

3.2.1.2. L298:

It is a high voltage and high current IC that contains two H-bridges to regulate the current flowing through the motor. The sequence generated by the L297 IC is fed L298 which feeds the sequence to the motor after current regulation. It gives a total output current of 4A.

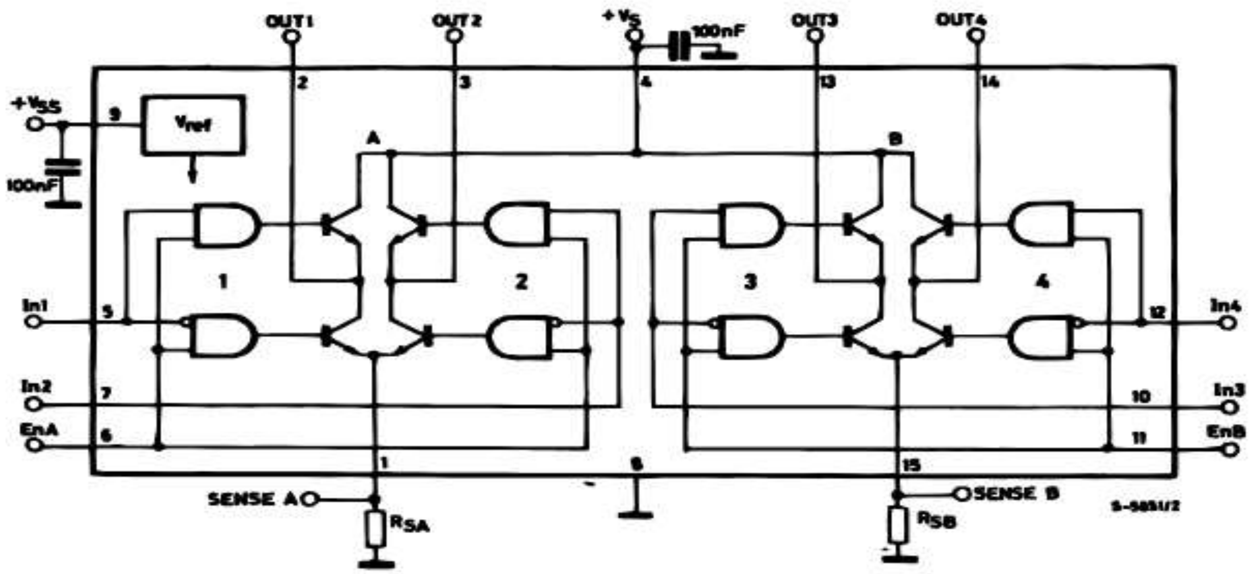


Figure 27 Internal structure of L298 [31]

The following circuit is the complete driver circuit for the Bipolar Stepper motor. In this circuit a heat sink is also attached on the L298 IC as this IC gets heated up while driving the motor and there is a possibility that the IC could burn for this purpose heat sink is used.

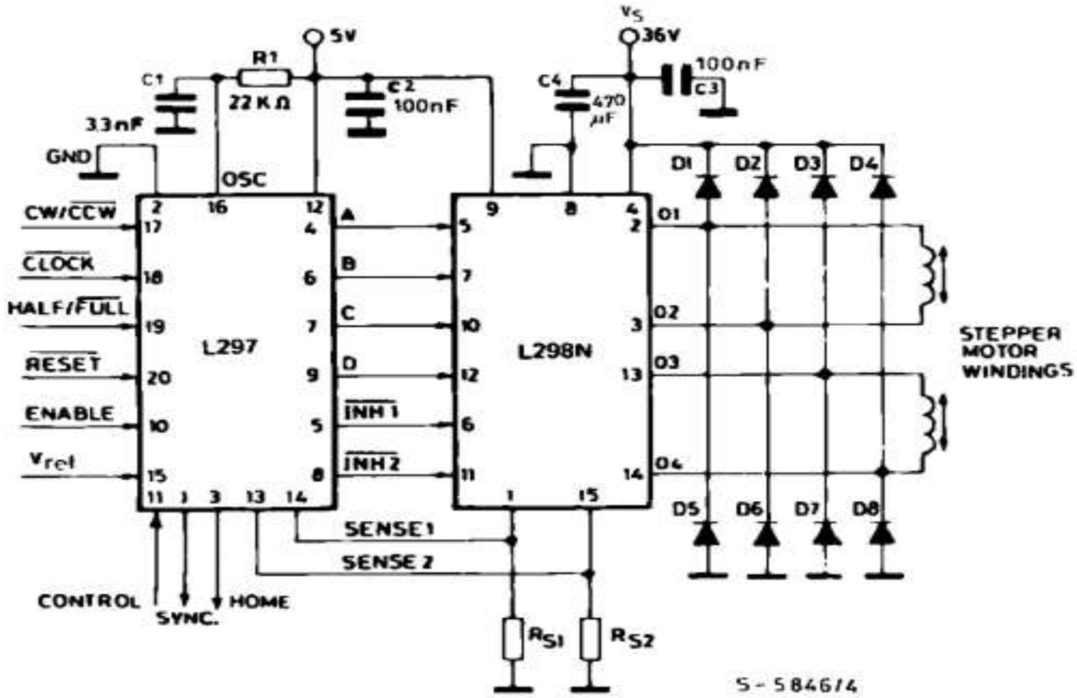


Figure 28 Schematic of Circuit Board

Free wheeling diodes of very high current ratings are connected at the output to finish the spikes in current due to the inductive effect of motor.

3.2.2. Sequences for Bipolar Stepper Motor:

A bipolar motor has two coils. The controller of a bipolar motor should be able to reverse the direction of current through the coils by reversing the polarity. For this purpose an H-bridge circuit is required. The following figure depicts an H-bridge.

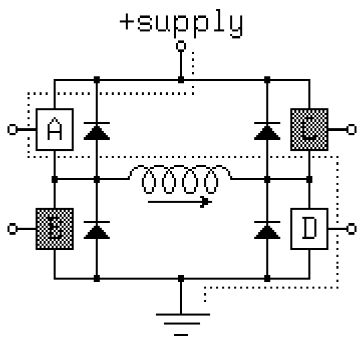


Figure 29 H-bridge

The current can flow in one direction by connecting A and D in the figure and can flow in the other direction by connecting B and C. It is called an H-bridge because its shape is in the form of a bridge. As each half of the bridge can either act as a source or a sink of current so it is also known as the push-pull configuration. To drive the bipolar motor we require two such H-bridges as there are two coils in the Bipolar motor.

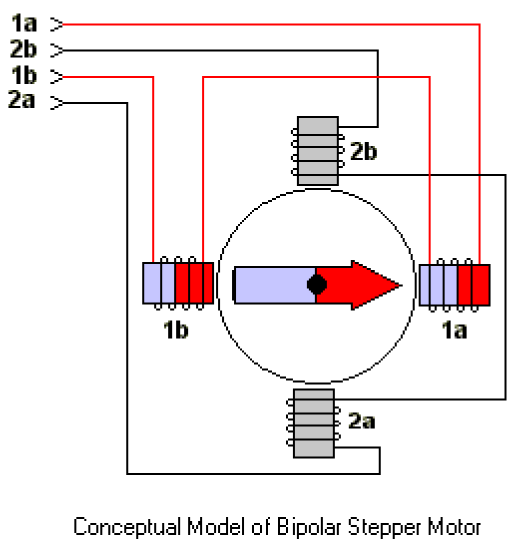


Figure 30 Stepper motor

In this figure when the line 1a is positive than 1b the rotor will point in the East direction and when 1b is positive than 1a the rotor will point in the West direction. This conceptual model of the motor has a 90 degree step size

The three useful sequences of driving the stepper motors are given below:

Sequence	Name	Description
0001 0010 0100 1000	Wave Drive, One-Phase	This stepping ensures the accuracy of position. At one time only one phase is energized.
0011 0110 1100 1001	Hi-Torque, Two-Phase	Two phases are energized at one time to give high torque.
0001 0011 0010 0110 0100 1100 1000 1001	Half-Step	In this stepping the resolution of the motor is doubled.

Table 6 Sequences to drive Bipolar Stepper Motors

3.3. SOFTWARE:

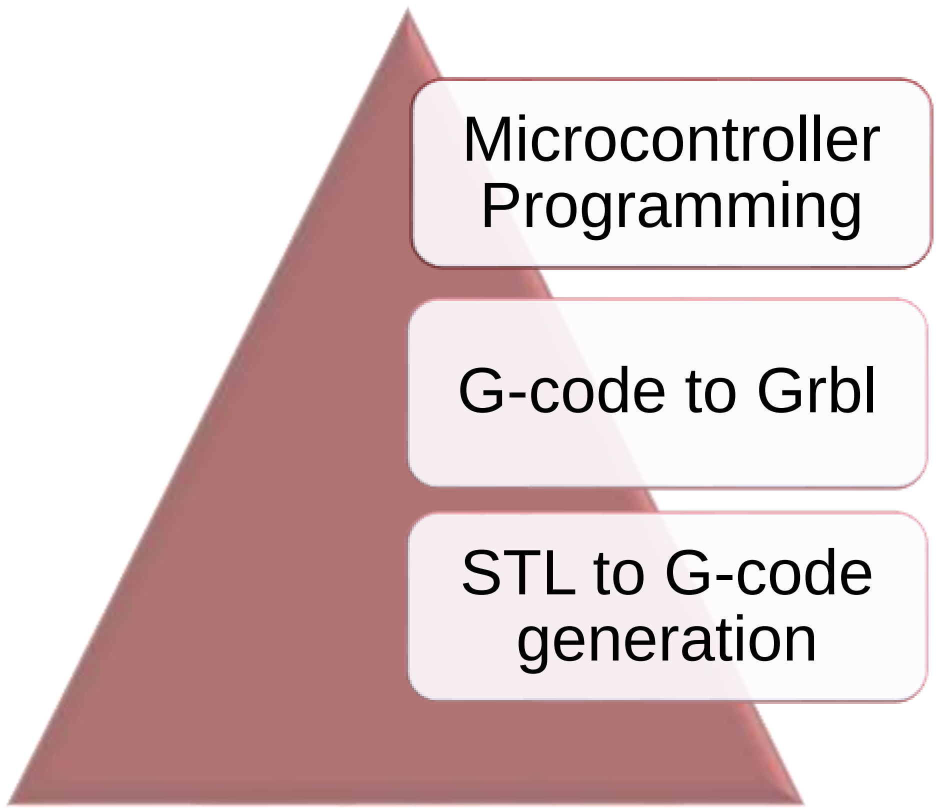


Figure 31 Software Flowchart

The 3D Printer's entire assembly will be controlled using the software. It includes various stages staring from the initial design of the 3-D object needs to be replicated. The following figure shows how the different software stages are achieved and how they work and interface with eachother.

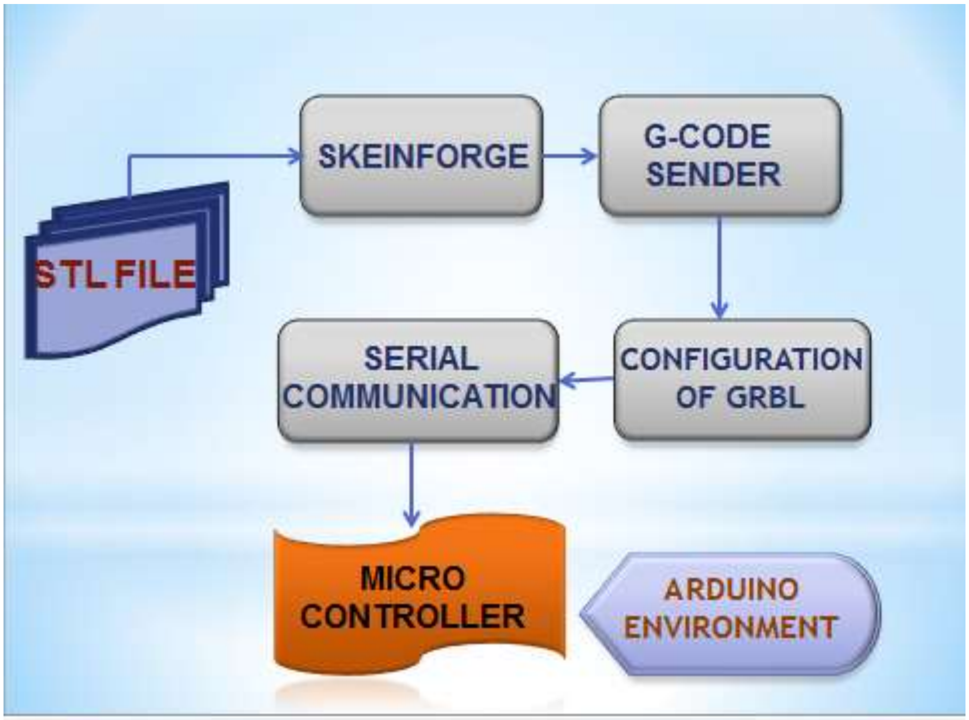


Figure 32 Software Flowchart

The printer starts from the input of the STL file which contains all the dimensions of the object to be replicated. This file is used by the skein forge with its path to generate the G-code. After the g-code s generated it needs to be converted into a form that is understandable by the micro controller. Hence GRBl is needed to do so. GRBL is an open source firmware that will convert the G-code into a hex file using the source programs in it about the movement of the stepper motor and its Control. Using the G-code sender the .hex file is sent to the microcontroller and hence the 3 axes motor starts to move.

3.3.1 Stl to G-Code (Skeinforge):

G-code generator software that converts the STL file into G-code.3D printer needs to be given a machine understandable language which is not the STL file itself; hence it has been converted to G-Code.

3.3.2 G-Code to Grbl (Microcontroller):

G-code is converted to a language that is understandable by the micro controller. Arduino has a special programming environment that accepts code through serial communication or USB connection. GRBl firmware is used to do so.it helps in converting the G-code to .hex file that is uploaded to the board through serial communication

3.4. MECHANICAL:

The mechanical part comprises of details of the 3D structure of the printer, the heating mechanism, extruder and the material used to make models.

3.4.1. 3d Structure:

The 3D structure contains 3 stepper motors. Each of the motors controls one axis of the structure. The rotation of the motors moves the extruder on the required axis.

3.4.2. Slides:

A slider bearing is used for reducing friction in the sliding motion of the rods. The slider bearing provides smooth movement of the axes rods and reduces frictions. This makes it easier for the motors to move the load on the axes. Two linear bearings are being used in series which removes any chance of angular disorientation in the direction of motion. Moreover, it increases the precision by removing any angular movement and making it sure that the movement is as required.

3.4.3. Translation of Rotation into Linear Motion:

The rotational motion of motors needs to be translated into linear motion to move the axes. In order to accomplish this task, belts are used on two axes and lead screw is used in the third axis. The z-axis goes up and down using a lead screw. The linear movement of the extruder on the rods is carried out using belts. The motors are connected with gears having teeth. The belts also contain teeth, which lock with the teeth of the gears connected with the motors, in order to avoid slipping of the belts under heavy load.

3.4.4. Heating Mechanism:

The heater uses the element of a soldering iron to convert electrical energy to heat energy. A copper pipe is inserted inside the element to make a nozzle. The plastic is inserted in the form of a wire from the top of the pipe. The heater is plugged into 220V AC power and it starts transferring heat to the pipe. The pipe elevates the temperature of the plastic to its glass transition temperature and it moves down through the nozzle. The element we are using is using 45W of power and can achieve temperatures up to 500 C. The glass transition temperature of the ABS plastic (the material we are using) is around 105 C. Thus the heater can provide sufficient heat to melt the plastic.

Figure 33 Element of soldering iron [27]

3.4.5. Extruder:

The extruder design contains a stepper motor and gears connected to a bolt. There is a groove cutout which leads to the copper nozzle that is heated. The motor movement makes the gears move which in turn rotate the bolt. This motion makes the plastic to be pushed forward and thus it is inserted into the copper nozzle. The heated nozzle melts the plastic which flows down to the base of the printer. When the plastic is not to be sent down, the motor of the extruder is stopped. This makes all the plastic inside the nozzle to go down and plastic extrusion stops. When the plastic is to be extruded again, the motor is started again, which pushes the plastic forward into the nozzle and extrusion starts again.

Figure 34 Reprap Extruder [30]

3.4.6. Material:

The material we are using for printing is Acrylonitrile butadiene styrene (ABS) Plastic. ABS is a "polymerized alloy" of the tree materials acrylonitrile, butadiene and styrene. The material is located under the group styrene plastic. Styrene plastics are in volume one of the most used plastics. Acrylonitrile butadiene styrene (ABS) (chemical formula $(C_8H_8)_x\cdot(C_4H_6)_y\cdot(C_3H_3N)_z$) is a common thermoplastic. Its glass transition temperature (ABS is amorphous and therefore has no true melting point) is approximately 105 °C (221 °F).[27]

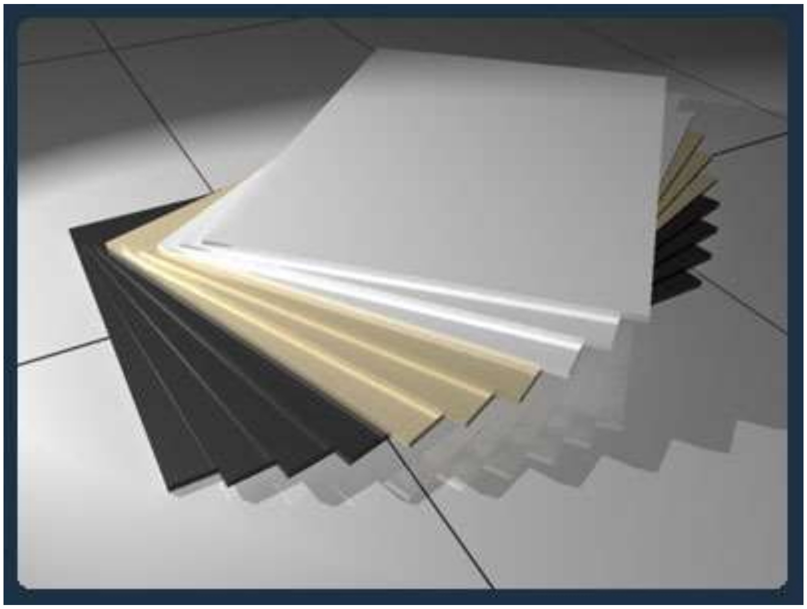


Figure 35 ABS Plastic Sheets [28]

The most important mechanical properties of ABS are impact resistance and toughness. A variety of modifications can be made to improve impact resistance, toughness, and heat resistance. The impact resistance can be amplified by increasing the proportions of polybutadiene in relation to styrene and also acrylonitrile, although this causes changes in other properties. Impact resistance does not fall off rapidly at lower temperatures. Stability under load is excellent with limited loads. Thus, changing the proportions of its components ABS can be prepared in different grades. Two major categories could be ABS for extrusion and ABS for injection moulding, then high and medium impact resistance. Generally ABS would have useful characteristics within a temperature range from -20 to 80 °C (-4 to 176 °F). [28]

ABS's light weight and ability to be injection molded and extruded make it useful in manufacturing products such as drain-waste-vent (DWV) pipe systems, musical instruments (recorders, plastic clarinets, and piano movements), golf club heads (due to its good shock absorbance), automotive trim components, automotive bumper bars, enclosures for electrical and electronic assemblies, protective headgear, whitewater canoes, buffer edging for furniture and joinery panels, luggage and protective carrying cases, small kitchen appliances, and toys, including Lego bricks.[29]

ABS is also commonly used in rapid prototyping extrusion-based 3D printers. Its glass transition temperature makes it a material of choice for rapid prototyping — relatively high as to reduce unwanted deformation at slightly elevated temperatures but low enough to be safely attainable with standard extrusion setups.

ABS plastic is available in the market in the form of sheets, granules or wire. In Pakistan, however, ABS is not available in the form of wire in the local market. If it is required in wire form, it has to be ordered specifically to recyclers.

The material that we used has been taken from a Plastic factory in Lahore where strands of plastic wire were available. These strands of wire were the scrape material so we took it from that factory. A flaw in this material was that it was non-uniform, i.e. there was no specific width of the material. This material has been shown in the figure below

Figure 36 Plastic material

Chapter: *4*

4.0 IMPLEMENTATION & RESULT DISCUSSION

In this chapter the integration of all modules and final implementation of the project will be discussed which results into final product.

4.1 MODULE INTEGRATION:

The individual modules include software, electrical and the mechanical parts. These parts when assembled together give a 3D printer. The combination of all these modules provides a complete solution. The integration of the modules includes interfaces between the hardware and the software. The hardware also includes the electrical and the mechanical part so both have different interfaces through which they communicate.

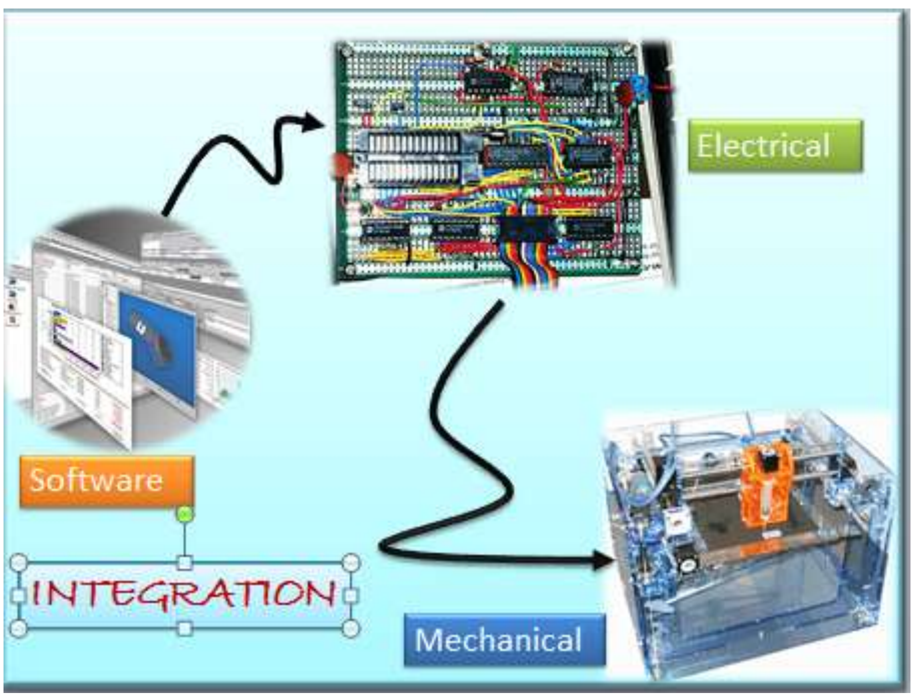


Figure 37 Module integration

4.2 ASSEMBLING THE PRINTER:

Once the modules are tested thoroughly, the entire assembly must be connected together. Assembling the product is an important part. It involves joining different parts together and making necessary mechanical and electrical connections. The GUI is at display that operated the software part. Microcontroller is connected with the PC. The micro controller is connected with a USB to serial interface cable. This serial communication helps in sending commands to from Pc to controller. This communication is important because the system works on the basis of the signals that are sent.

4.3 SERIAL INTERFACING AND MICROCONTROLLER:

Arduino uses a USB to serial cable attached with it for serial communication with the PC.The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus. For SPI communication, SPI library is used.

Figure 38 Serial Cable

Through this cable the software is connected to the hardware i.e. the software package with the motors through the micro controller.

The motors which are responsible for driving the mechanical structure are driven through the signals generated by the micro controller. Digital pins of Arduino board generate signals of clock and direction for the motors. These signals are responsible for moving the motors in forward and reverse directions. The X, Y and Z axes motors clock is connected to digital pin 2, 3, 4 and their direction is connected to digital pin 5,6 and 7 respectively. When the G-code is sent form the G-code sender GRBL interprets it and generates the required signals on these pins for motor movement.

4.4 EXTRUDER CIRCUITRY:

The extruder runs with the aid of the motor attached to it. This motor needs a continuous clock in one direction such that the wheels move inwards pushing the material downwards to the heater.

A pressure rubber is attached with the wheels of the extruder in order to apply pressure to the material fed in.

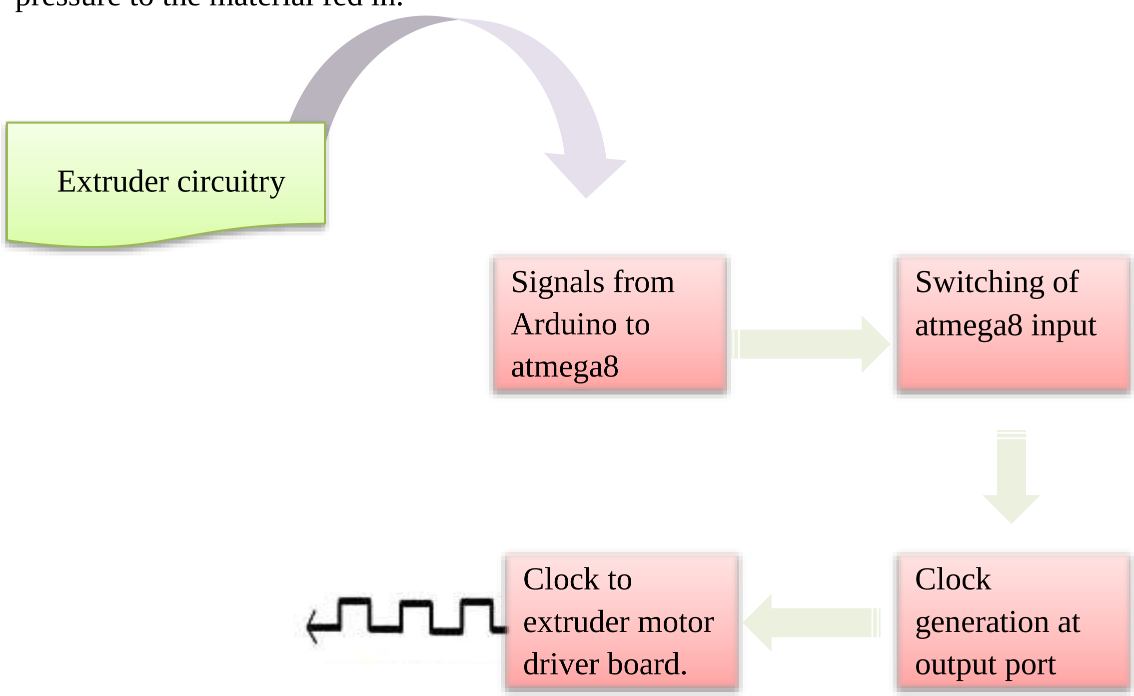


Figure 39 Extruder control

4.4.1 Connections of Extruder circuitry:

Signal generated from GRBL is 0 for M103 and 1 for M101 at digital pin12.The connections are made as follows in the circuit.

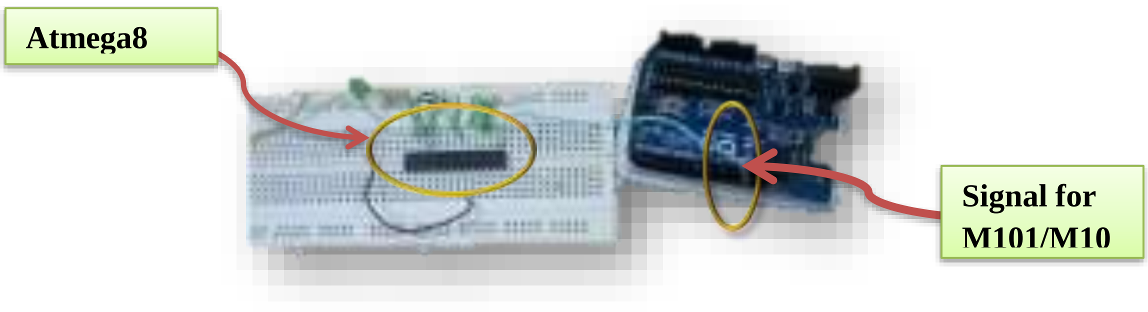


Figure 40 Extruder Circuitry

Atmega8 is connected with port C as an output port and port D as the input port.PD0 is kept as the switching pin and rest of port D is grounded.as soon as PD0 is switched on clock of frequency100Hz is generated at the output port C. Simulation of the circuit is done on Proteus for testing purpose.

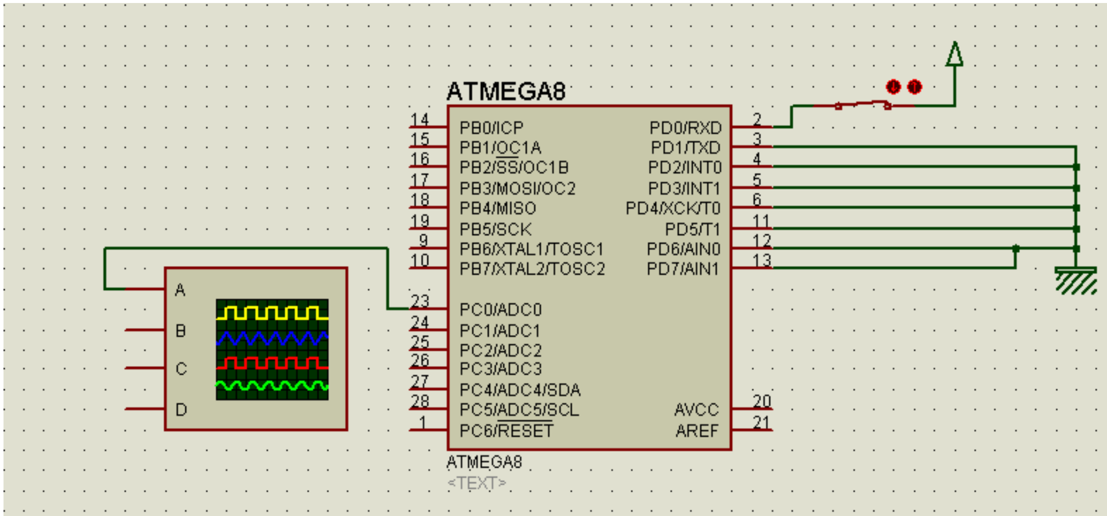


Figure 41 Extruder Circuitry simulation

4.4.2 Microcontroller Atmega8 Programming:

100Hz clock is generated to run the extruder motor. The motor is tested with the trainer board initially in order to determine the clock speed. A C++ code is written to program atmeg8 and clock signal is generated on triggering of the input signal.

```
/*
 #define F_CPU 1000000UL /* 1 MHz Internal Oscillator */

#include <avr/io.h>
#include <util/delay.h>

int main(void)
{ DDRC=255;
       PORTC=0;
       DDRD=0;
       PORTD=0;
       int i;
    while(1)
```

```
    {
              //if (PIND==00000001)
              //{
              _delay_ms(5);
              PORTC=0;
              _delay_ms(5);
              PORTC=255;
              //}
              //else
              //PORTC=0;


              }
}
```

4.5 POWER REQUIREMENTS:

Each motor is attached with a power supply with an input current of approximately **2.5-3A.** For 3 Axes motors and extruder we need 4 power supplies to run the system. Each driver board is connected to the power supplies for motor and also for energizing L297 for generating input sequence required by the motor.

4.6 MECHANICAL STRUCTURE:

The mechanical structure of the printer is shown in the figure below. Each part of the structure is explained in detail.

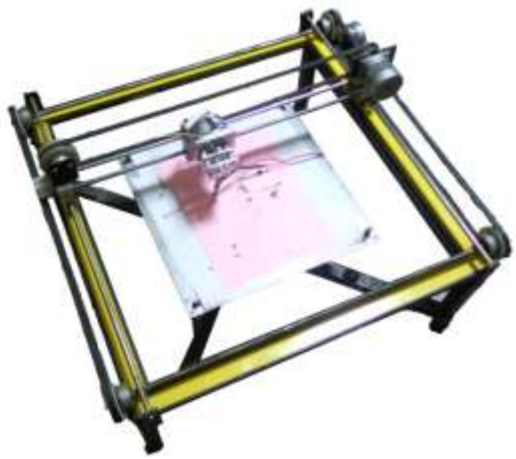


Figure 42 Mechanical structure

4.6.1 The Print Bed:

The print bed is the plane where the material falls down after melting from the extruder. The print bed is supported by four pillars along which it moves up and down. It has an area of 14x14 sq. inches. The effective printing area is 12x12 sq. inches calculated by ignoring the area consumed by the support pillars. It has a supporting structure below that is attached to the z-axis motor. The material falls on the print bed and it then successively moves down with every layer. This gives the object the third dimension of height.

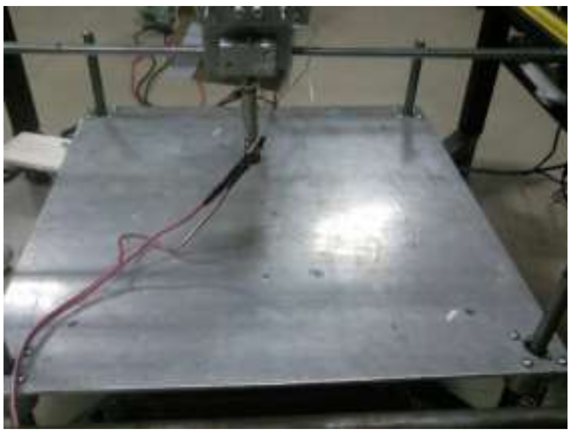


Figure 43 Print Bed

4.6.2. The Extruder:

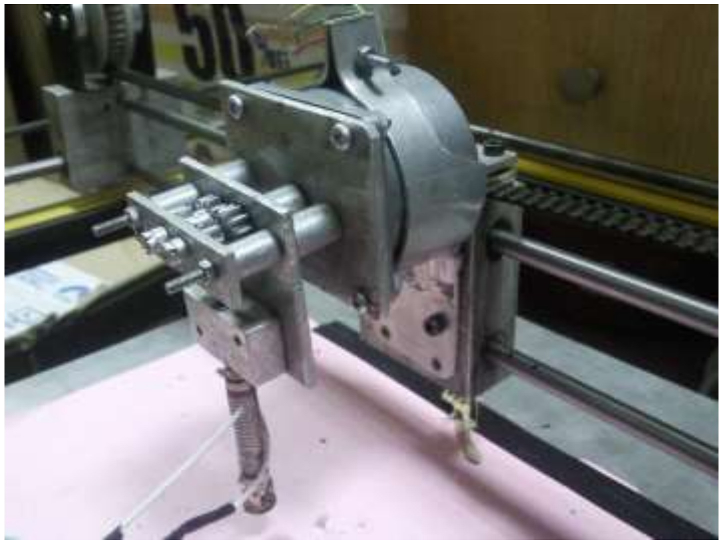


Figure 44 Extruder

The extruder can be seen in the figure mounted on the x-axis. It consists of a motor attached with gears, a heater and a nozzle. The motor attached to the extruder has small gears attached to it. These gears push the material in with the help of a rubber ring. The ring takes into account minor changes in the width of the material by expanding or shrinking. It also provides the force required to push the material inside.

Figure 45 Gears and Rubber ring

Beneath the gears is the nozzle. The nozzle is wider (5mm) from the upper side and has a welded connection with its lower part. The lower part of the nozzle is narrower (3.5 mm) than the upper side. At the end, it is made even narrower for the material extruded to be fine. The changes in nozzle width help in avoiding heat transfer between the heater and motor assembly. If that happens, it will make the material stick inside the nozzle due to insufficient heat for melting.

Figure 46 Nozzle and Heater

The nozzle is surrounded by the heating element. It is an element of the soldering iron. It works on 60W and 220V AC power supply. The heater is wide enough to touch the nozzle and transfer heat to it. The nozzle, in turn, heats the material and melts it. There are two screws on the heater that help it strengthen its grasp on the nozzle. They also make sure that the nozzle is heated appropriately.

4.6.3. The X-axis:

The x-axis moves along the length of the object. It has a motor mounted on it that is connected to a gear. A belt runs over the gear and translates the rotatory movement of the motor into linear motion. Two rods are connected to the base of the axis to bear any load. On those two rods, the extruder is mounted. The rods are connected to the belt. There are slider bearings inside the mount which runs over the rod. These slider bearings reduce the friction of the load moving on the rods. The motor has a current rating of 1.4 A/ Phase and a voltage rating of 2.8V/Phase. It is a stepper motor with each step of 1.8 degrees. The calibration of the motor yielded that 1.15 steps of the motor translate into 1mm of linear movement.

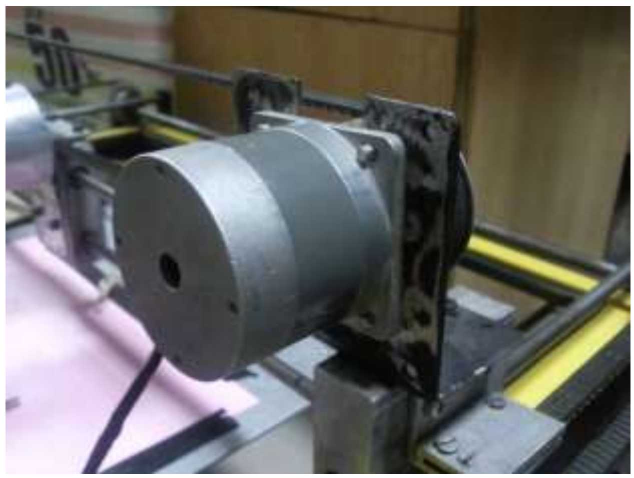


Figure 47 X axis stepper motor

4.6.4. The Y-Axis:

The y-axis moves along the width of the object. It has a motor attached to it along with two gears. Two belts are connected, one on each corner of the x-axis, to the gears and translate the rotation of the motor into linear motion. The belt also contains the base of the x-axis. The weight of the extruder and the x-axis, thus, falls on the belt of the y-axis. One rod running parallel to the belts and one running perpendicular to them supports the motion of the y-axis. The rod running perpendicular connects two gears each supporting a belt and making the translation of motion possible. Slider bearings are used to reduce friction between the base of x-axis and the rod on which it is mounted. The motor has a current rating of 1.1A/Phase and a voltage rating of 3.5V/Phase. It is a stepper motor with 1.8 degrees of rotation per step of movement.

The extruder moves 1mm in the y direction for every 1.7 steps movement of the motor.

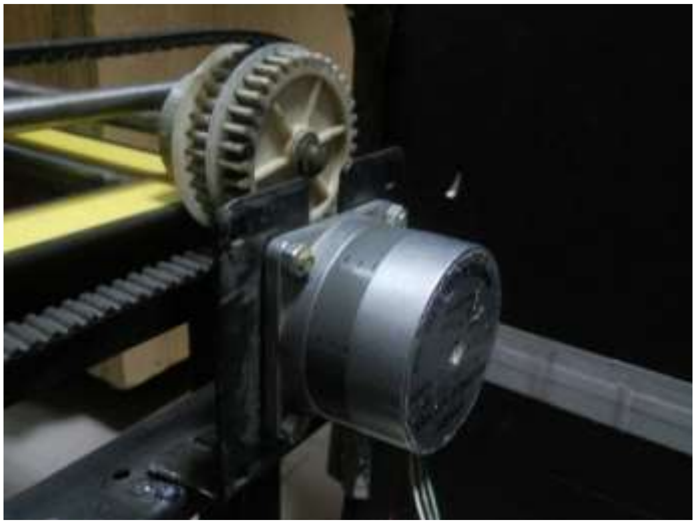


Figure 48 Y axis stepper motor

4.6.5. The Z-Axis:

The z-axis moves the print bed down, when one layer of the object is completed. It has a motor with a lead screw. The lead screw moves with the rotation of the motor.

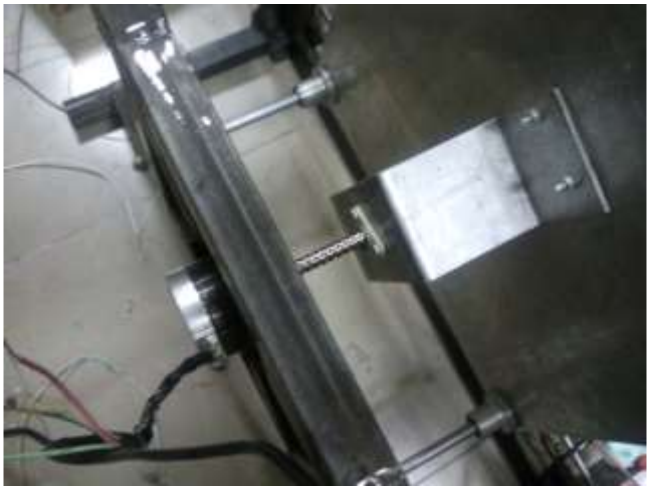


Figure 49 Z axis stepper motor

The lead screw, in turn, moves the base of the print bed. The print bed thus moves down with every layer. The print bed can move a maximum distance of 3 inches up linearly. The effective movement for printing is 2.5 inches. The motor is responsible for the movement of the print bed and its support.

Figure 50 Lead Screw

The motor is rated at a current of 1.5A/Phase and a voltage of 2.8V/Phase. It is a stepper motor with a 1.8 degrees rotation per step of movement. The calibration yielded a result of 7.5 steps per mm of linear motion of the z-axis.

4.7 CALIBRATION:

The axes need to be calibrated in accordance with the printing area and the maximum size of the object to be printed. The calibration settings are made using the

Gcode sender and then these settings are stored in the GRBL firmware installed in the Arduino board. The following settings were made using the Gcode sender:

No of steps/mm on x-axis:1.15

No of steps/mm on y-axis:1.7

No of steps/mm on z-axis:7.5

The commands used to do these settings on the x, y and z axes are $0, $1, $2 respectively.

The method we used to calibrate the axis was by determining that how much the axis moved when we kept the no of steps per mm equal to 1. The distance covered is measured. Along with the distance we asked the motor to move, we can find out the steps per mm of the motor using ratios.

Suppose we commanded the motor to move 5mm while steps per mm is set to 1. This means that the motor moves 5 steps. Now, we measure the distance covered by the motor. Suppose the motor covers 2mm of distance. This means it took the motor 5 steps to move 2mm. hence, it can be calculated that the motor should be calibrated at 5/2=2.5 steps per mm.

4.8 WORKING OF THE PRINTER:

The complete system works with the integration of all the 3 modules. We start up with the GUI .The GUI takes in .STL file as the input. Once the design file has been given its G-code is generated using the Skeinforge command. The G-code contains the G codes as well as the M-codes. The M-codes that are used in our case are M101 and M103.G-code is sent to the microcontroller using the G-code sender. The microcontroller then generates the signals accordingly after the GRBL burnt in it interprets the G-code for the controller. Each axis picks its commands from Arduino and starts its movement according to the imaginary x-y-z Grid. The axes movement is totally dependent upon the G-code generated. The axes are capable of moving one at a time and simultaneously as well. The signals for the extruder connected to the 4^{th} motor are sent through form the Arduino board according to the M codes. When M101 is sent 1 signal is sent to the AVR microcontroller. After receiving 1 as the input it generates a clock of 100 Hz at its output port. This clock is connected to the driver board which drives the extruder motor. As soon as the G-code sender sends M103 commands this clock signal is stopped and the extruder no longer purges the material down. This is done several times according to the design requirements. The heater attached with the extruder nozzle provides suitable temperature for the material to melt the wire material being fed into the extruder. The material is fed manually in to the extruder wheels but the extruder wheels help in applying a specific pressure to make its way inside the heater. The extruder purges the material until all the movements have been done according to the G-code file hence making a 3D object.

4.9 PROBLEMS IN INTEGRATION:

4.9.1 Axes Vibrations:

While testing the mechanical structure we noticed that there were certain vibrations in the system. These vibrations were generated when the axes moved and as a result the accuracy of the object being printed was disturbed. The material coming out of the extruder at the time of vibration was a little shaky which altered the shape of the object. To end this problem we looked for the sources of vibration and tightened all the screws which may be leading to it. Doing so improved the results and lessened the instability of the system to a great extent.

Figure 51 Axes Vibration problem

4.9.2 Non-uniform material:

The width and the shape of the material was the greatest problem that we encountered while printing the material. The material that we are using has been taken from the scrape of a plastic factor so it had no uniform thickness whereas our extruder was designed for a material of a given thickness. So our extruder let only the material of the thickness it was designed for it to pass through it and all other material got stuck in it and the gears of the extruder stopped working. To end this problem we needed a material of very uniform thickness which could be easily passed through the extruder.

4.9.3 Unable to Connect to Single Power Source:

In the mechanical assembly for the printer we had used four stepper motors i.e. three for the axes and one for the extruder. Each motor had a current requirement of almost 2A and other then that each motor driver board required a 0.09A current supply as well for the L297 IC. To address this issue we used the power supply of the PC and connected all the four motors with it. With doing so the motors did not draw a sufficient amount of current and did not function. The current was sinking somewhere in the circuitry and was not being supplied to the motors appropriately. We tried a lot to debug the issue but it led to no solution so we used an individual power supply for each motor.

4.9.4 Breakage of Pressure Rubber:

A rubber is installed between the gears of the extruder which pulled the material inside the pipe of the heater. Sometimes due to the pressure of the entering material the rubber broke up. Without this rubber there was not the enough thrust for material to enter the pipe. When the rubber broke it had to be replaced with a new one to continue with the printing.

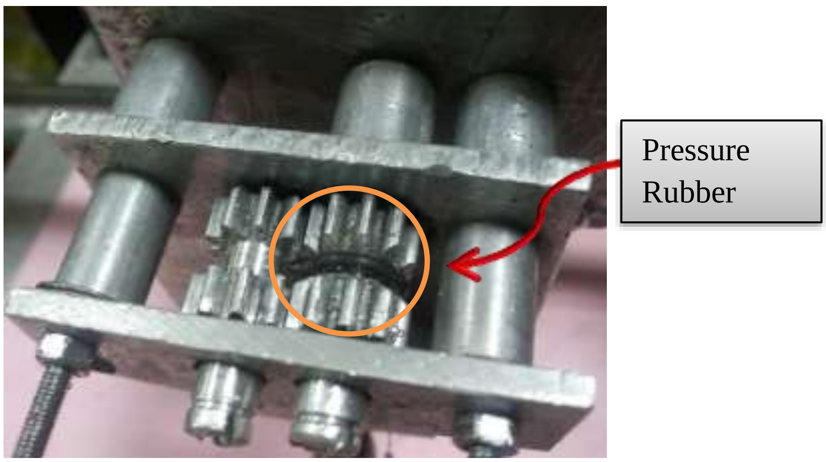


Figure 52 Rubber Breakage problem

4.10 GRAPHICAL USER INETRFACE (GUI):

Graphical User Interface (GUI) is the interface which the user uses to give commands to the system to perform some certain tasks. As the user is the one who interacts with the GUI so the GUI should be user-friendly and easy to understand as the user does not gave low-level information about the system and he just requires a proper output.

For our printer we made a GUI which allows the printing of any selected object. The input to the GUI is a 3D file and as the output the GUI runs the printer to print the object. While designing the GUI the ease for the user has been kept in mind.

4.10.1 Layout of GUI:

The basic layout of the GUI included a window with two frames and four buttons. These four buttons provide all the functionality required to run and the printer and print 3D objects.

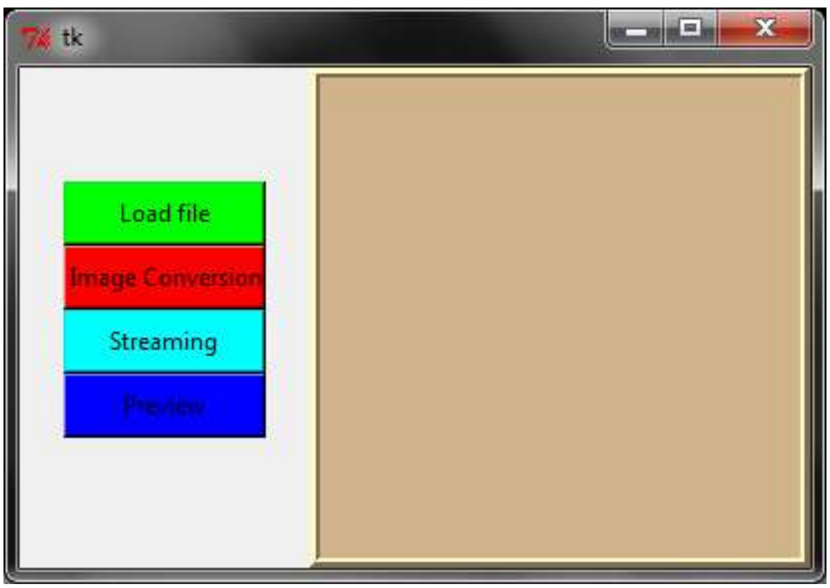


Figure 53 GUI Layout

The first button of ***'Load image'*** when pressed opens up a window through which you can browse to a 3D image .stl file. Select the file which you want to be printed. Press 'Open' in the below window and your selected file will be loaded

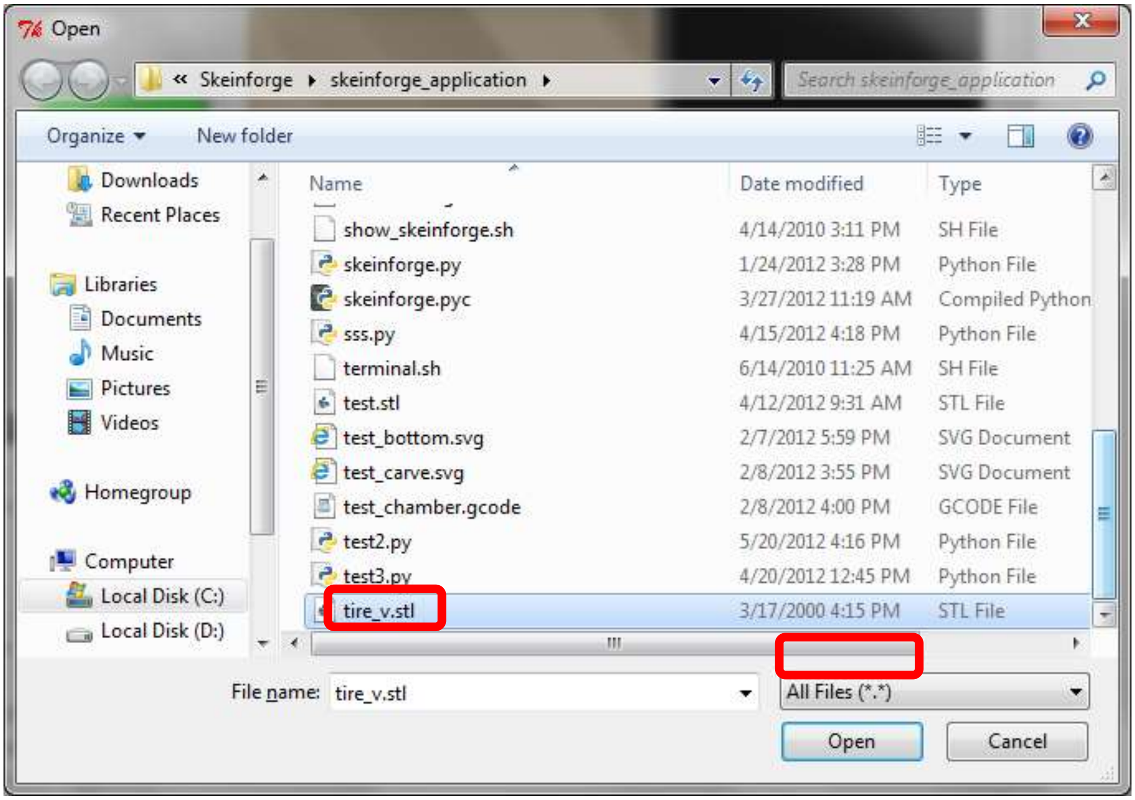


Figure 54 Stl File selection

The second button of *’Image Conversion’* when pressed converts the 3D image to a form that can be sent to the printer. This button converts the image to the Gcode file.

The third button is of *‘Streaming’* which sends the converted image to the printer through a serial port. Pressing this button will open the Gcode sender program. When the Gcode sender is opened the first step is to select the COM port. Once the COM port is selected you press the **‘Open’** button as shown in the below window. Doing so will open the serial port and commands can be sent to the printer over a serial interface.

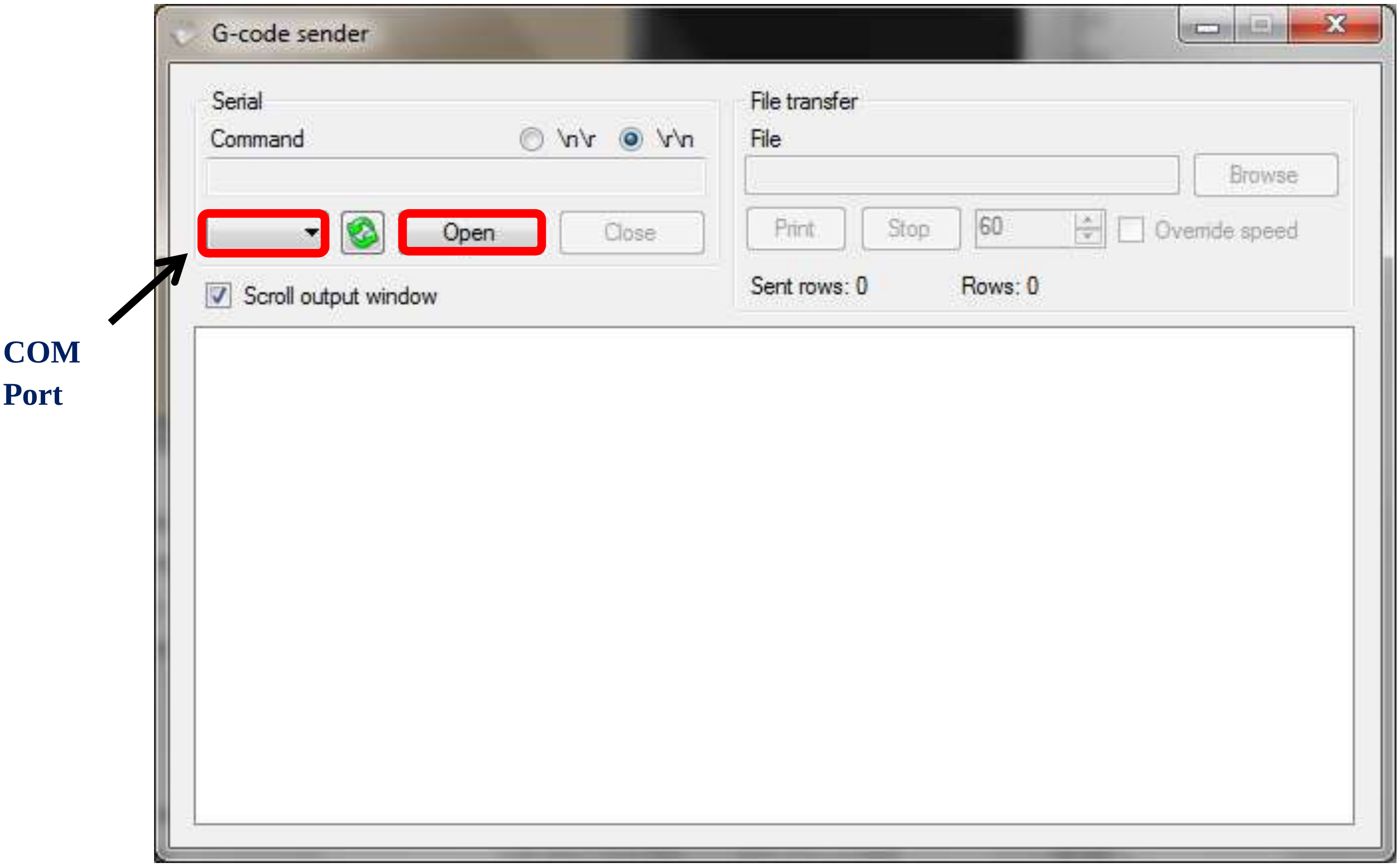


Figure 55 Gcode sender

Now the next step is to select the gcode file created using the second button and send it to the controller. Use the Browse button to browse to the gcode file. This gcode file will be placed at the same location where the .stl file located which you loaded into the GUI. After browsing the file press the **'Print'** button. The printer will be activated and will start printing the object. If you want to Abort the program press the 'Stop' button, the printer will be deactivated.

The ***'Preview'*** button is used to view the object that you are printing in 3D. Pressing this button will open the 3D view of the object. This is shown in the screen shot below.

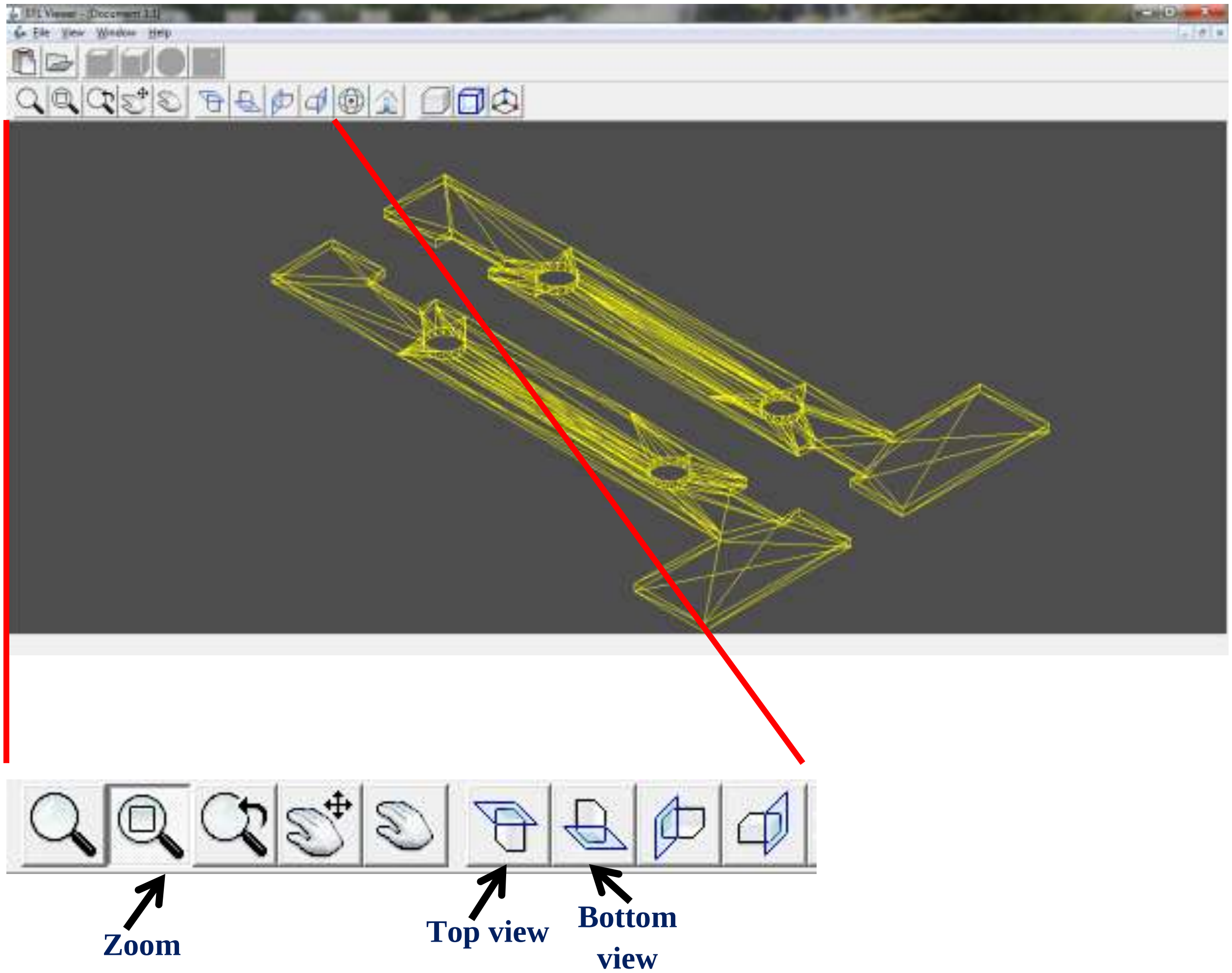


Figure 56 STL file Viewer

While viewing the object in 3D there are a number of options available to view the object from different angles and also to zoom the object.

4.10.2 Implementation of GUI:

The GUI made for the printer integrates a number of softwares to perform all the steps from conversion of the 3D image to the final step of printing. So we needed a platform which could easily integrate all the softwares and provide us with a GUI that is easy to use. After a thorough research we decided to use ***Python*** as a platform for GUI development due to the following reasons:

- Provides a library known as 'Tkinter' for GUI development
- Provides an easy interface as Skeinforge is also written in Python so it could be easily integrated to form a GUI
- Availability of a number of built in functions and libraries which simplify a lot of complex programming
- It is a portable language and has the same interface on multiple platforms. The same code can be run in Mac, Linux or Windows

4.10.2.1 Tkinter library:

Tkinter is the standard GUI library in Python and is also known as the ***'Tk interface'***. It provides with a number of functions which help in making a good GUI. The input to the GUI may be mouse clicks or keyboard press. The GUI making in this library involves the following steps:

- Import the *Tkinter* module.
- Create the GUI window
- Add widgets and containers to the GUI. These widgets include buttons and the containers are frames that contain these buttons. Frame is the canvas for your GUI.
- Bind the buttons with the main event that you want to perform

4.10.2.1.1 Programming in Tkinter:

The major concepts in Tkinter programming are:

- Create a GUI object and then associate it with a parent

- Making the widgets and containers, and then packing them. Packing is the process to set up the relationship between the GUI component and parent.

First of all the library has to be imported in the code using the following command.

from Tkinter import *

The second task is to create an instance of the class Tkinter.Tk. This instance is the Top level window and by convention this top level window is named as 'root'. A mainloop has to be added for the root. The function of the mainloop is to wait for the event to occur, when the event occurs it is handled and then the main loop waits for the second event to occur.

```
from Tkinter import *

root = Tk()
root.mainloop()
```

We make a class which defines the GUI and the code for the complete GUI is written in the constructor (__init__) method of the class.

```
class MyApp:

def __init__(self, parent):

 self.myParent = parent
```

Now we create frames as an object to the parent. Frames are provided by Tkinter in a class called *"Frame"*. The command we use is:

```
self.myContainer1 = Frame(parent)
```

Then we pack this frame with the following command:

self.myContainer1.pack()

To add buttons we use the *"Button"* class of Tkinter:

self.button1 = Button(self.buttons_frame, command=self.button1Click)

'command=self.button1Click' specifies that the button will run the event associated with it with the left click of a mouse. The colour, borders, shape of the button can be modified using the configure command *'self.button1.configure(text="Load file", background= "green")'.*

Then packing the button is also important. In this command the location of the button in the frame can also be specified to be in the top, bottom, right or left.

self.button1.pack(side=TOP)

There are number of other libraries that we have used in the code as well such as

from tkFileDialog import askopenfilename

import skeinforge

import os

import shutil

'askopenfilename' has been used for in the 'Load image' button. This is the command that allows for the file to be browsed. *'os'* allows the system commands to run and *'shutil'* is for copying a file into some other file. All these commands have been used in our code.

4.11 RESULTS:

After integration of all the parts of the printer, the testing phase started in which all the parts of the printer were tested individually to see if they are giving the desired results. After testing all the parts individually we started printing 3D objects. The results of testing and printing are summarized in this chapter.

4.11.1 Testing of Individual Parts:

4.11.1.1 Testing the motor driver boards:

Driver boards are the main circuitry of the printer so there correct functioning is very important. To test the driver boards they are connected to the power supply and the motor is attached. The motor should draw an appropriate amount of current if all the components are functioning correctly. The clock can be given to the board from a Trainer board or a microcontroller and the direction can be controlled with the help of a switch. L298 IC gets overheated if the current exceeds the rated value, for preventing these heating heat sinks are used on this IC.

TEST	DESIRED RESULT	ACTUAL RESULT
Testing of Stepper Motor Driver boards to see that motors run properly and L298 does not get overheated.	Motor works and draws an appropriate amount of current	The motor is working and is taking correct amount of current. L298 is not getting overheated as the excess heat is being absorbed by the heat sinks

4.11.1.2 Testing the serial port:

The serial port is tested in order to check if the data being sent to the Arduino board is received properly. For this purpose open the Serial monitor in the Arduino IDE and run the code for serial transmission. The two LEDs on the Arduino board for transmission and reception will blink whenever the data is transmitted or received respectively.

TEST	DESIRED RESULT	ACTUAL RESULT
Testing reception and transmission of data over the serial port	Correct transfer of data	The data was transmitted correctly. By connecting the receive pin with the transmit pin, the hyper terminal is receiving what is being typed through keyboard

4.11.1.3 Testing the axis movement:

The axes of the assembly need to be tested to see if they are moving properly and with correct dimensions. Each of the three axes was tested individually. Driver boards were connected to the axes and the signals of clock and direction were given by Arduino which was controlled using the Gcode Sender. Simple commands of absolute motion were sent such as X 10, Y -5, Z 2 to see the movement of the axes.

TEST	DESIRED RESULT	ACTUAL RESULT
Testing the movement of axes. **It is highly imperative that the motors are**	Axes should move with exact calibration	Axes are moving perfectly and are following the exact dimensions being sent over the serial port

working properly because the major functionality of this project is dependent on the smooth working of the motors.	

4.11.1.4 Testing the Mcodes:

The Mcodes are controlling the turning on and off of the extruder which in turn controls the flow of the material. To test that the Arduino board is sending the correct signals when it receives the Mcode the pin 13 and pin 12 of the Arduino board are shorted. When M101 is sent the LED on the board is turned on and when M103 is sent the LED turns off.

TEST	DESIRED RESULT	ACTUAL RESULT
Testing the Mcodes which are controlling the motion of the extruder.	Mcodes are working fine on Arduino board	Mcodes are working perfect. M101 turns on the extruder and M103 turns it off.

4.11.1.5 Testing the extruder:

The extruder is tested to see if the material is flowing correctly through the opening of the heater without getting stuck. The smooth and non-stop flow of the material is very necessary for maintaining the correct shape of the object. The extruder is tested by giving the motor connected to the extruder an appropriate clock and setting the direction. The gears of the extruder start moving in the right direction and pull the material inside the heater with a thrust. The material starts extruding from the nozzle.

TEST	DESIRED RESULT	ACTUAL RESULT
Testing the extruder to check the flow of material. The flow of material is important for the correct shape of the object	Material should come out in a continuous manner.	The material extruded properly and is non-stop so the object can be formed correctly

4.11.1.6 Testing the GUI:

The GUI needs to be tested to see that all the softwares are working correctly in integration with each other. The Arduino is connected with the laptop and the GUI is activated by opening the .py file of GUI. Once the GUI is started the ‘Load’ button is used to load the .stl file of a 3D image. The second button coverts the 3d file to gcode and then the third button opens the Gcode sender. In the Gcode sender the gcode file made in the second step is loaded and then is streamed to Arduino.

TEST	DESIRED RESULT	ACTUAL RESULT
The GUI is tested for its correct functioning as it is the integration of all softwares and should be easy to use and understandable	GUI is user-friendly and working properly	GUI is performing all the functions correctly

4.11.2 Printing:

After testing all the parts of the printer for their correct functioning we started with printing. First we started off with printing simple shapes and then moved towards a little complex objects.

4.11.2.1 Test-1:

The first shape we printed was of a Square. We just made the boundaries to test whether the extruder is moving along the correct dimensions. The boundary had two layers of material deposited.

TEST	DESIRED RESULT	ACTUAL RESULT
A square is formed to test that the extruder is moving along the correct dimensions	A square is formed	Square shape was achieved proving that the extruder was moving in accordance with the commands sent from the controller

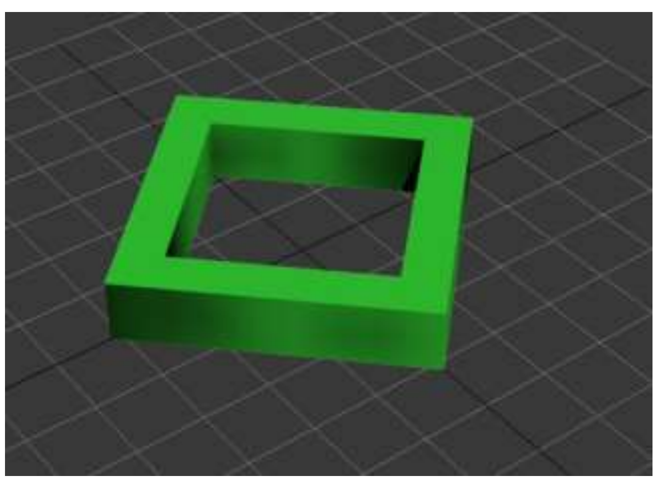
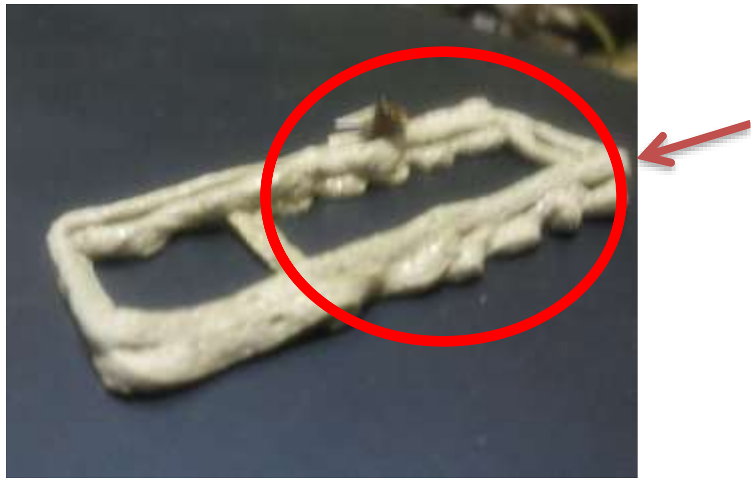


Figure 57 .stl file of square



The shape was a little distorted due to axes vibrations

Figure 58 Printed square

4.11.2.2 Test-2:

The second test performed was to form a triangular shape. This was test was important because to form a triangle the x and y axes had to move together rather than moving one by one as in the case of the square. The triangle was formed with few layers of the material deposited. We painted the printed triangle to give it a good look.

TEST	DESIRED RESULT	ACTUAL RESULT
To make a triangular shape to test the mutual movement of x and y axes	A triangle is obtained	Triangle was formed with few layers of the material deposited confirming that x and y axes can move together to form diagonals and produce good results

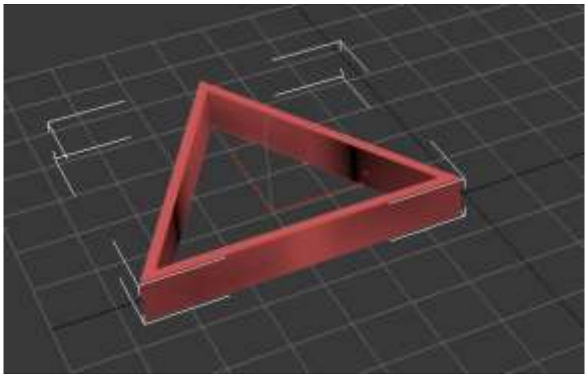


Figure 59 .stl file of triangle

Figure 60 Printed Triangle

4.11.2.3 Test-3:

After forming the basic shapes we moved towards some complicated shapes. A star was formed. . stl file of a star was given and the result was obtained.

TEST	DESIRED RESULT	ACTUAL RESULT
Forming a star to test the ability of the printer	A star is obtained	A star of a certain height is formed.

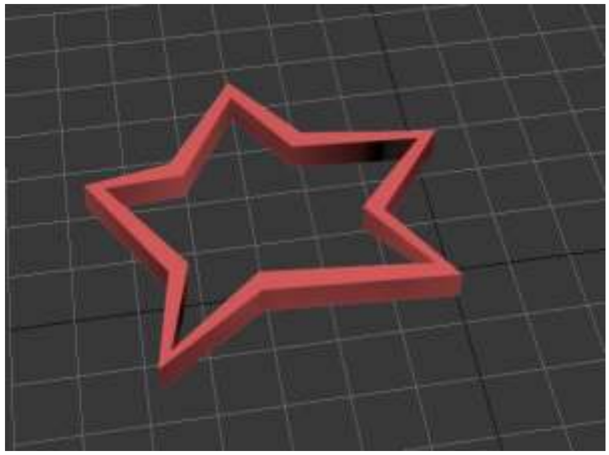


Figure 61 .stl file of star

Figure 62 Printed star

4.11.2.4 Test-4:

To further test the functionality of the printer we printed a hexagon by giving a 3D model of the hexagon to the printer.

TEST	DESIRED RESULT	ACTUAL RESULT
We tried printing a	A hexagon is obtained	A hexagon of a certain

hexagon to see if the printer is capable of printing different shapes	height is formed.

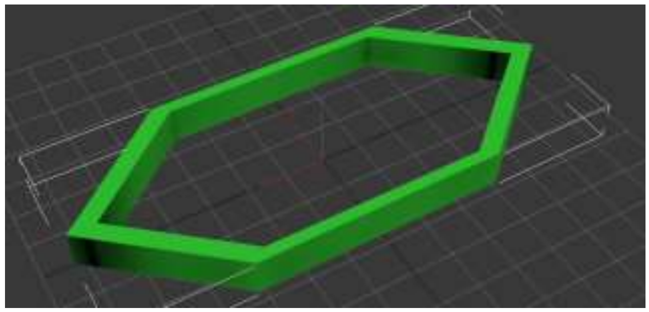


Figure 63 .stl file of hexagon

Shape is distorted due to non-uniform material

Figure 64 Printed hexagon

Chapter: *5*

5.0 CONCLUSION & RECOMMENDATIONS

3D printer is fully capable of making 3d prints of a desired object. The printer is a cost-effective solution for rapid prototyping. The main aim of this project was to achieve the capability of replicating simple objects of daily use. The printer we made successfully achieved the desired aim. The three axes motion and the extruder movement are calibrated to get the desired object formed with a certain amount of precision.

The GUI is user friendly and converts any .stl file to gcode that is them transmitted to the controller. The printer is a controlled hardware with a software package to control and use it. The objects printed are robust and durable. The printer can find a lot of usage in industries where plastic parts of machines are required; also the printer replaces molds to form different shapes.

The 3D printer can be modified in some ways to achieve advanced functionalities as well. The printer is designed to produce objects that have a flat bottom so that they do not require any supporting material while being formed, such as a sphere would need a base as a support over which the complete sphere will be printed. By introducing a second extruder in the 3D printer sphere and other support requiring objects can also be formed. The second extruder will be used to deposit the support material and the other extruder could perform its regular function of forming the object.

Other amendments can also be made to this machine. The printer can also be used as CNC machine if we use a cutter instead of an extruder which would cut in the desired shape and also drilling can be performed by replacing the extruder with a drill.

To improve the appearance of the objects being formed by the printer colored material can be used which would give a good look to the prints. Furthermore the precision of the objects being formed can be improved by controlling the feed rate of the material; this could improve the shape of the object.

REFRENCES:

self-replicating machines[1]http://reprap.org/wiki/Main_Page

3Dprinter-history[2]http://www.mahalo.com/3d-printers/

Serialport-communication[3]http://en.wikipedia.org/wiki/Serial_port

Arduino;G-code-interpreter[4]http://reprap.org/wiki/Arduino_GCode_Interpreter

GRBl[5]http://dank.bengler.no/-/page/show/5470_grbl?ref=mst

Thing-o-matic[6]http://www.prestonlee.com/2011/01/18/makerbot-thing-o-matic-3d-printer-print-pictures-product-review/

Makerbot[7]http://en.wikipedia.org/wiki/Makerbot

[8] Dhananjay V. Gadre, *Programming and Customizing The AVR Microcontroller* [e-book]. India: McGraw Hill pp. 11.

[9] A Microcontroller Interfaced with other Components, Dhananjay V. Gadre, *Programming and Customizing The AVR Microcontroller* [e-book]. India: McGraw Hill pp. 12.

CAD [10]http://en.wikipedia.org/wiki/List_of_file_formats#Computer-aided_design_.28CAD.29

Atmega8-pinouts[11] http://arduino.cc/en/uploads/Hacking/Arduino-To-Atmega8-Pins.png

Atmel[12] http://www.atmel.com/dyn/products/product_card.asp?part_id=

Arduino[13] http://en.wikipedia.org/wiki/Arduino

[14] UART Using Serial cable And Serial Port, Embedded Systems, Lecture 7. Available: http://sites.google.com/site/esdfall2008/.

RS-232pinouts[15] http://www.noktaendustriyel.com/eng/pratik/0x1/rs232_pinout.gif

Extruder nozzle[16] http://reprap.org/wiki/Geared_extruder_nozzle

Ball Bearings [17]http://www.designworldonline.com/uploads/ImageGallery/igus-bearing.jpg.

Ball Bearings [18] www.wikipedia.com.

Lead screw[19]http://en.wikipedia.org/wiki/Lead_screw.

Lead Screw [20]http://en.wikipedia.org/wiki/Lead_screw.

Ball screw [21]http://eolsurplus.com/images/BeaverLeadscrew.jpg.

Ball Screw[22]http://en.wikipedia.org/wiki/Ball_screw.

Timing Belt [23]http://www.roymech.co.uk/images11/timing_belt_2.gif.

G-code [24] http://reprap.org/wiki/G-code

G-code [25] http://linuxcnc.org/docs/html/gcode_overview.html

Skeinforge [26] http://wiki.bitsfrombytes.com/index.php/Skeinforge

Heating element [27] http://www.sourcingmap.com/replacing-part-soldering-iron-30w-heating-element-core-solder-p-168631.html

ABS sheet [28]
http://www.archplastics.com/us_california_plastic_distributor/abs_sheet_plastic.html

Reprap extruder [29] http://www.make-something.com/index.php/3d-printing-home/building-a-reprap-printer/reprap-assembly-part-8-extruder-body/

Bipolar Stepper motor [30] http://www.make-something.com/files/cache/2a4b7e9dd63132512b36f548e377e1ad.jpg

L298 dataheet [31]
http://www.sparkfun.com/datasheets/Robotics/L298_H_Bridge.pdf

Stepper motor tutorial[32] http://www.stepperworld.com/Tutorials/bridge.gif

Gcode commads [33] http://engraving.majosoft.com/assets/images/DSCF0092.gif

APPENDIX :

GUI CODE :

```
from Tkinter import *
from tkFileDialog import askopenfilename
import skeinforge
import os
import shutil


class MyApp:
  def __init__(self, parent):

    self.myParent = parent

    ### Our topmost frame is called myContainer1
    self.myContainer1 = Frame(parent) ###
    self.myContainer1.pack()

    #------ constants for controlling layout ------
    button_width = 11     ### (1)

    button_padx = "2m"    ### (2)
    button_pady = "1m"    ### (2)

    buttons_frame_padx =  "3m"   ### (3)
    buttons_frame_pady =  "2m"   ### (3)
    buttons_frame_ipadx = "3m"   ### (3)
    buttons_frame_ipady = "1m"   ### (3)
    # -------------- end constants ----------------

    ### We will use VERTICAL (top/bottom) orientation inside myContainer1.
    ### Inside myContainer1, first we create buttons_frame.
    ### Then we create top_frame and bottom_frame.
    ### These will be our demonstration frames.

    # buttons frame
    self.buttons_frame = Frame(self.myContainer1) ###
    self.buttons_frame.pack(
      side=LEFT,   ###
      ipadx=buttons_frame_ipadx,
      ipady=buttons_frame_ipady,
      padx=buttons_frame_padx,
      pady=buttons_frame_pady,
      )

    # top frame
    self.top_frame = Frame(self.myContainer1)
    self.top_frame.pack(side=TOP,
      fill=BOTH,
      expand=YES,
      )  ###
```

```
### Now we will put two more frames, left_frame and right_frame,
### inside top_frame.  We will use HORIZONTAL (left/right)
### orientation within top_frame.


### right_frame
self.right_frame = Frame(self.top_frame, background="tan",
   borderwidth=5,  relief=RIDGE,
   height=250,
   width=250,
   )
self.right_frame.pack(side=RIGHT,
   fill=BOTH,
   expand=YES,
   )  ###


# now we add the buttons to the buttons_frame
self.button1 = Button(self.buttons_frame, command=self.button1Click)
self.button1.configure(text="Load file", background= "green")

self.button1.configure(
   width=button_width,  ### (1)
   padx=button_padx,    ### (2)
   pady=button_pady     ### (2)
   )

self.button1.pack(side=TOP)


self.button2 = Button(self.buttons_frame, command=self.button2Click)
self.button2.configure(text="Image Conversion", background="red")
self.button2.configure(
   width=button_width,  ### (1)
   padx=button_padx,    ### (2)
   pady=button_pady     ### (2)
   )

self.button2.pack(side=TOP)
self.button2.bind("<Return>", self.button2Click_a)



self.button3 = Button(self.buttons_frame, command=self.button3Click)
self.button3.configure(text="Streaming", background="cyan")
self.button3.configure(
   width=button_width,  ### (1)
   padx=button_padx,    ### (2)
   pady=button_pady     ### (2)
   )
```

```
        self.button3.pack(side=TOP)

        self.button4 = Button(self.buttons_frame, command=self.button4Click)
        self.button4.configure(text="Preview", background="blue")
        self.button4.configure(
            width=button_width,  ### (1)
            padx=button_padx,    ### (2)
            pady=button_pady     ### (2)
            )

        self.button4.pack(side=TOP)
#self.button4.bind("<Return>", self.button4Click_a)

    def button1Click(self):
      a = askopenfilename()

      shutil.copyfile( a , remote )


    def button2Click(self):
       os.system('python skeinforge.py remote.stl')

    def button3Click(self):
       os.system('gcodesender.exe')

    def button4Click(self):
       os.system('remote.stl')


    def button2Click_a(self, event):
        self.button2Click()

    # def button4Click_a(self, event):
        # self.button4Click()
root = Tk()

myapp = MyApp(root)

remote = "remote.stl"

root.mainloop()
```