3-D printer based on the Fused Deposition Method

Group Members

MUHAMMAD MOHSIN 190346
USMAN AHMAD 190368
HASSAN RIAZ KHAN 190382

BACHELORS OF ELECTRICAL ENGINEERING

(2019-2023)

Project Supervisor

DR. MUMAJJED UL MUDASSIR

Assistant Professor

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

FACULTY OF ENGINEERING

AIR UNIVERSITY, ISLAMABAD

A Final Year Project Report

presented to

AIR University

in partial fulfillment of the requirements

for the Degree of

Bachelor of Electrical Engineering

3-D printer based on the Fused Deposition Method

Submitted By

MUHAMMAD MOHSIN	**190346**
USMAN AHMAD	**190368**
HASSAN RIAZ KHAN	**190382**

(2019-2023)

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

June, 2023

Approval for Submission

It is to certify that the project report titled
3-D printer based on the Fused Deposition Method
has met the required standard for submission
in partial fulfillment of the requirements
for the award of degree of
Bachelor of Electrical Engineering
at Air University, Islamabad

Project Supervisor

Dr. Mumajjed UL Mudassir
Assistant Professor

Head of Department

Dr . Ashiq Hussain

Dedication

We dedicate this project to our professors, teachers for their unwavering support and guidance throughout our studies in engineering – as well as our supervisor Dr Mumajjed UI Mudassir his passion for the field and expertise in engineering have inspired us to pursue our own careers in this field and motivated to pursue with the project. Without the support and guidance of our faculty and supervisor, this project would not have been possible. We are forever grateful for their invaluable contribution to our success.

Acknowledgement

We, the undersigned, hereby confirm that the work presented in this final year project is the original work of our group and has not been submitted or published elsewhere. We also confirm that all sources and materials used in the project have been properly cited and referenced. We would like to express our sincere gratitude to our professor, teachers, and Supervisor Dr. Mumajjed UL Mudassir for their guidance and support throughout the completion of our final year project. Their expertise and valuable insights were instrumental in helping us navigate the complexities of our research and arrive at a successful conclusion. We would like to thank Air University for providing the necessary resources and support for the successful completion of this project.

Abstract

This engineering project aims to pioneer the development of a cutting-edge 3D printer based on Fused Deposition Modeling (FDM) technology, with the primary goal of revolutionizing additive manufacturing processes. Distinct from conventional printers limited to two-dimensional printing, our 3D printer showcases remarkable capabilities in fabricating intricate objects across three dimensions (X, Y, Z). By harnessing the power of additive manufacturing, which involves the meticulous deposition of materials layer by layer based on a digital model, our technology offers a faster, more cost-effective, and highly customizable approach to manufacturing.

In this project, we delve into the fundamental principles that govern the operation of 3D printers, shedding light on the indispensable role of specialized Computer-Aided Design (CAD) software. This software enables the generation of precise digital blueprints, serving as the guiding force behind the printer's systematic construction of objects layer by layer. Leveraging this digital blueprint, the printer expertly deposits materials, such as plastics, metals, or ceramics, with unparalleled precision onto a designated build platform. The material extrusion process entails the controlled melting of materials within a heated nozzle, ensuring the meticulous deposition required for complex object fabrication. Throughout our exploration, we investigate various 3D printing technologies, including photo polymerization, selective laser splintering (SLS), and binder jetting, each with its own unique characteristics and capabilities.

The immense potential of 3D printing transcends numerous industries, encompassing prototyping, product development, manufacturing, and the medical field. This transformative technology empowers users to rapidly and cost-effectively produce intricate, customized objects that were once unattainable using traditional manufacturing techniques. Furthermore, 3D printing has the capacity to significantly reduce waste and production costs, while empowering designers to create innovative objects that surpass the limitations of conventional methods. With its versatility and innovation, 3D printing redefines the entire paradigm of object conceptualization, design, and manufacturing.

By embarking on this ambitious engineering project, our aim is to push the boundaries

of additive manufacturing and highlight the exceptional capabilities of FDM-based 3D printers. Through the creation of a functional prototype, we seek to demonstrate the unparalleled versatility, efficiency, and ingenuity that this technology offers to the manufacturing industry and beyond.

Table of Contents

List of Figures

List of Tables

Chapter 1
Background

1.1 Introduction

The conventional strategies of manufacturing consumer goods are very time-consuming and less efficient with high assembling costs. The project aims not only to reduce the expense but also to work upon its precision, lessening time and errors. It likewise saves money on energy by 40 to 60 percent as it dispenses with transportation and other coordination exercises. Moreover, it enables clients to create objects with lesser material.

1.2 History

The modern-day 3D printer was first designed and invented by American Engineer Chuck Hull in 1986 – the technology was based upon the principle of stereo-lithography. 3D printers were made with the initial motive of mass-producing products in mega factories and industries such that the product doesn't account for any human error. The term "Additive Manufacturing" was coined for 3D printing due to its nature of procedurally producing products in an additive manner one layer at a time. It provides us with an alternative to the other ways of producing mass-produced products in industries that otherwise involve sculpting molding and folding (subtractive manufacturing); additive manufacturing simplifies this approach and employs a layer-by-layer approach for production. With innovation and enhancement in technology now an average consumer can easily get access to a 3D printer on a reasonable budget and possesses enough computing power to run it. The technology which was initially intended for industries that require high-end computers and an industrial-level budget is now accessible to an average user with an average device within a reasonable budget thanks to the technological revolution. [1]

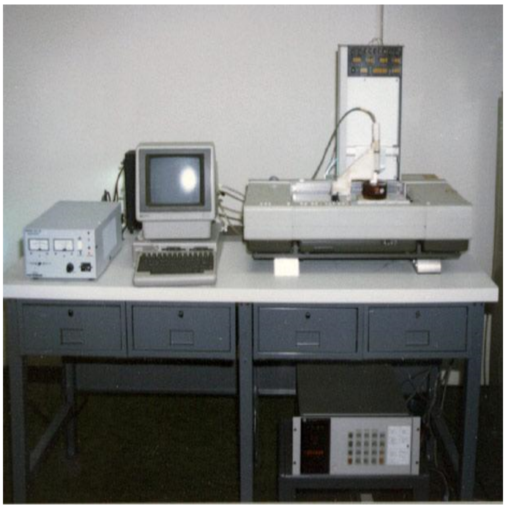


Figure 1.1: The first 3D printer designed by Chuck Hill

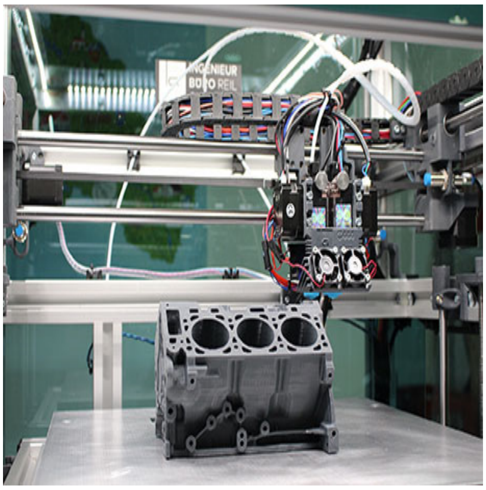


Figure 1.2: Industrial 3D printer

1.3 What is 2D and 3D printer?

A conventional printer (or a 2D printer) is fairly common in a professional and even home environment – it is a device that produces a hard-copy of digital data present on the computer – these can be PDF s, photographs, etc. We can say that a 2D printer can print digital data in the 2D plane (x and y plane) line on a flat paper – a 3D printer can be easily understood taking this analogy into context, a 3D printer can print digital data in the 3D plane (X Y Z plane). A 3D printer produces a 3D model of any digital model present on the computer using specialized polymer ink. We can print digital models of any complexity with any number of polygons, the printer would print the model and allow us to interact and use it in the real world. [2]

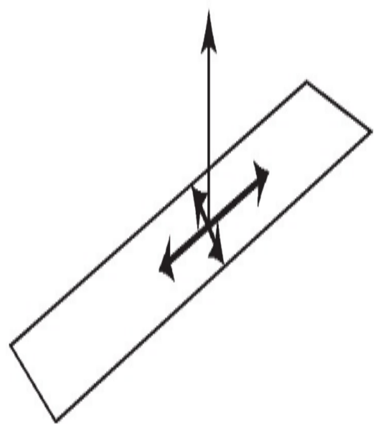


Figure 1.3: 3D

A conventional Printer (2D printer) can print in the 2D plane like flat page. It can print digital data in X and Y axis only.

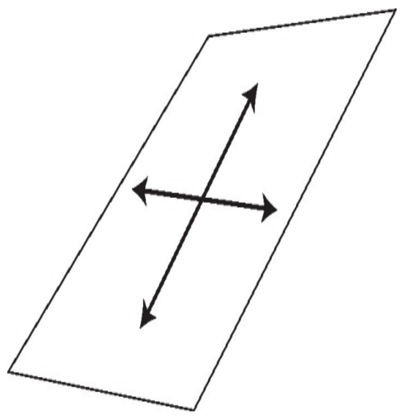


Figure 1.4: 2D

A 3D Printer can print digital data in the 3D plane. It can print produce any model in the X Y Z plane.

1.4 Computer-Aided Design (CAD)

The interface which allows us to create models and print them can be coined "CAD" which stands for "Computer-Aided Design". CAD is a software suite that allows us to model, design, and simulate 3D models and allow us to interface the computer with the printer – one can think of it as using a word processor such as Microsoft Word to print a Document using a conventional 2D printer, in easier to so we can say that CAD is to 3D printer what is word is to a 2D printer. CAD used to be extremely specialized packages of software that were not meant to be run on the ordinary home computers due to the processing power and the proprietary hardware required to run them, however, due to the technological revolution and the rise of computers in the reach of ordinary people CAD software soon found its place in home computers as well. Today we have much more powerful hardware – a typical consumer has more computing power than industrial-grade computers two decades ago – the rapid enhancement in technology has made it possible for users to run much more demanding programs on their computers. There exists a lot of CAD software now ranging from open-source free software like Blender to industry-leading paid software like 3D MAYA, AUTO-DESK 3D, and Z-brush. Due to the popularity of such 3D software, it was only natural for the consumers to demand an alternate solution to get their custom 3D models printed in their homes rather than commissioning them to some 3D printing services.

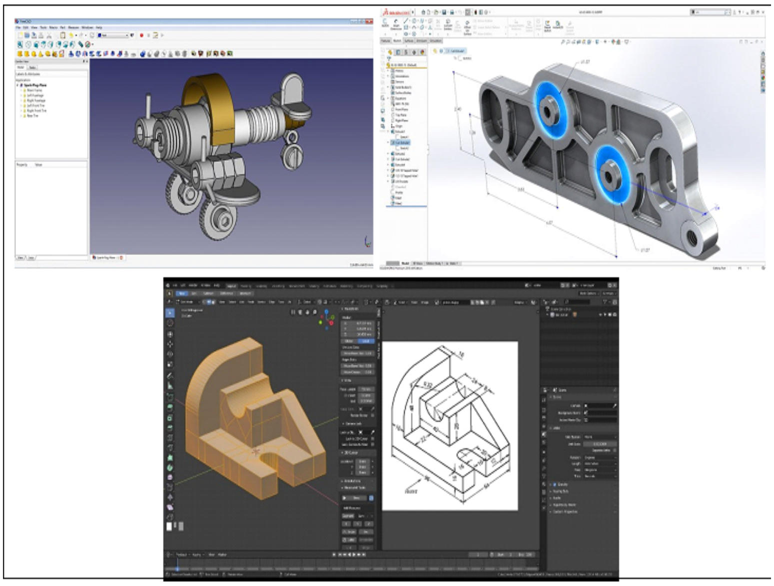


Figure 1.5: Computer Aided Drawing (CAD)

1.5 3D Printer Types

A 2D printer has various types like dot-matrix, ink-jet, etc. A 3D printer also has several types. We must develop an understanding of the nomenclature and types of 3D printers and their working before continuing. The three most common Additive Manufacturing types of 3D printing techniques are

- FDM
- DLP
- SLA

1.5.1 FDM

FDM stands for Fused Decomposition modeling – this method can be understood by imagining a simple glue gun; where a solid plastic filament is pushed to the heated nozzle which then extrudes a fine stream of molten, the molten plastic. An FDM-based 3D printer uses solid polymer filament – this filament is pushed by a stepper motor into the extruder which consists of a heated nozzle – the nozzle is usually heated to 200 degrees Celsius. The extruder is manipulated by 3 stepper-motor on the X Y and Z axis. The motors that manipulate the extruder in the X Y Z plane are given instructions by the computer to define a path for proper printing. The resolution of the printer depends upon the diameter of the nozzle and the precision of stepper motors. This is one of the most common and cost-effective methods to 3D print an object – and this is the method we shall be using.

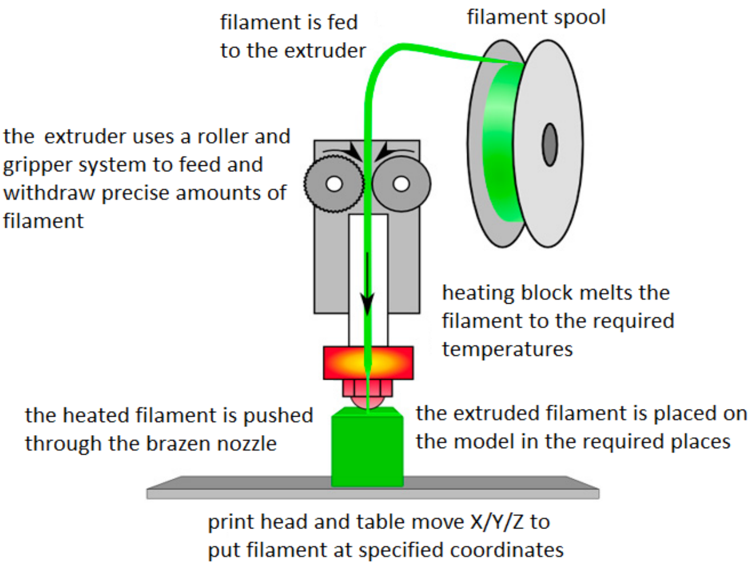


Figure 1.6: FDM

1.5.2 DLP

DLP stands for Digital Light processing. DLP unlike FDM makes use of liquid-based resin for its printing. The liquid resins solidify when exposed to Ultra Violet light. Because such a printer uses liquid resin modelling, the result is extremely detailed. The working behind the DLP printer is based upon the curing phenomenon where the exposure to resin to UV light hardens it and the resin turns into a solid model – such models have finer details than FDM-based models. Such printers print the model upside down – by this we mean the model is PUSHED upwards using the stepper motors as the model is created the suspended in the tank of resin. As the resin is solidified in the tank using UV light, the bed is raised, creating a new layer. A projector underneath projects the image for the entire layer for the curing process. This process usually takes 10 seconds before the next layer starts. DLP printers can simultaneously print multiple models which are exposed to UV lamps – this is not possible in SLA.

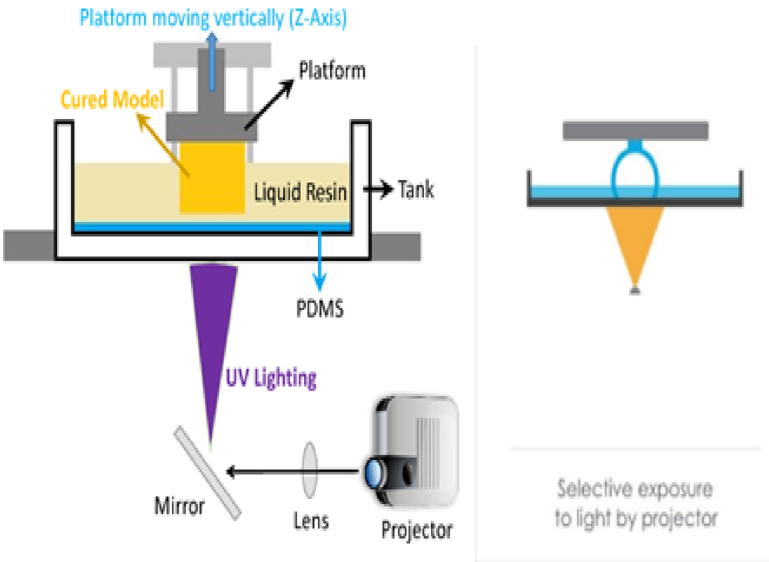


Figure 1.7: DLP

1.5.3 SLA

SLA stands for stereo-lithography apparatus. Such printers like DLP use liquid resins for their work. Like DLP printers they also print the model upside down – by this we mean the model is made starting from the bottom. The working is similar to DLP – where the model is made in incremental layers in a suspended resin bath which is then pushed up for the next layer – however, the SLA printer uses an Ultra Violet LASER rather than light which allows for even higher precision at the expense of time. The curing process in SLA takes longer than in DLP however it makes up for the result where the result from SLA has even finer details than DLP. SLA design does not allow multiple models to be made using a single firing LASER.

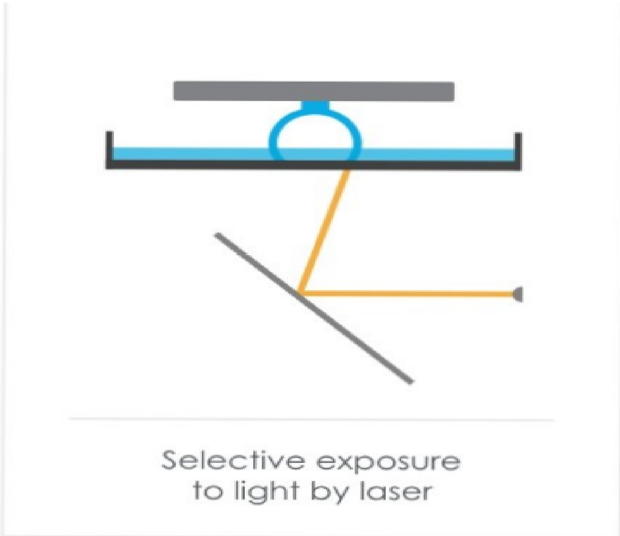


Figure 1.8: SLA

1.6 Comparison table

The following table gives a brief comparison of the technologies.

Technology	Cost	Resolution	Speed	Maintenance
FDM	Low	High	Medium	Low
SLA	High	High	Slow	High
DLP	Medium	Low	Fast	High

Table 1.1: Comparison of technologies

Chapter 2
Literature Review

2.1 Review of article

- This book consists of seven different parts, the most important among these are 1st, 2nd, and 4th parts that deal with hardware, software, and materials respectively. The hardware part is about choosing the printer. The software part deals with the three types of software that are used in 3D printing as mentioned below

  - CAD (Computer-Aided Design)
  - CAM (Computer-Aided Manufacturing)
  - PCS (Printer Control Software)

  CAD software like Auto CAD and Solid works is used to design a 3D model. CAM and printer control software like Slicer and Slic3r is used to generate G-code and instructions to control the printing assembly. The materials part explains the materials (poly-lactic acid, Poly-carbonate, Nylon, etc.) that are used in 3D modelling. This part also talks about the required bed temperatures and nozzle temperatures for each of the materials.
  Publisher: Maker Media, Inc, 1005 Gravenstein Highway North, Sebastopol, CA 95472.
  Compiler: Anna Kaziunas France
  ISBN: 978-1-457-18293-8 [4]

- This paper is by Ottnad, Irlinger, and Lueth, titled "Design, construction, and verification of a printhead - tolerant towards bubbles - dosing liquid wax using rapid prototyping techniques," presented at the 2011 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), describes the development of a printhead for dosing liquid wax that is tolerant towards bubbles.To precisely distribute the liquid wax, the authors carefully studied the printhead's design, including the material choice, printhead size and shape, and nozzle design. The printhead could be produced quickly and affordably because to the utilisation of rapid prototyping techniques.The experimental outcomes show that the printhead can effectively dispense the liquid wax even when there are bubbles, which makes it a promising technology for application in printing procedures that need for accurate dosing of liquid materials. This study has significant ramifications for a variety of applications, including 3D printing, microfluidics, and printing electrical circuits, and it offers insightful information regarding the behaviour of liquid wax droplets during the printing process.

- The paper by Soriano Heras, Blaya Haro, de Agustín del Burgo, and Islán Marcos, titled "Plate auto-level system for fused deposition modelling (FDM) 3D printers," published in the Rapid Prototyping Journal in 2017, describes the development of an automatic leveling system for the build plate in FDM 3D printers.The authors emphasise the need of a level build plate in producing prints of a high calibre and minimising the need for post-processing. They give a thorough explanation of the development and design of the auto-leveling system, which modifies the height of the build plate using a capacitive sensor and a microcontroller. The trial results show how well the system works to level the build plate, producing high-quality prints with a better surface finish. This article offers insightful information on the creation of automatic levelling systems for FDM 3D printers, which can enhance the speed and calibre of the printing procedure. [3]

- This paper is by Jahandardoost and Milani, titled "Multiphysics Modeling and Experimental Investigation of the Deposition Process in Fused Filament Fabrication Method, under High-Viscosity and Non-Newtonian Material Flow," published in the Journal of Materials Engineering and Performance in 2021, presents a comprehensive study of the deposition process in Fused Filament Fabrication (FFF) 3D printing. The authors concentrate on non-Newtonian and high-viscosity materials, which are crucial for 3D printing applications. To better understand the deposition process, the work combines computational modelling and experimental research. The findings offer perceptions into how process variables affect deposition quality, which can help in the creation of the best printing settings for highly viscous and non-Newtonian materials. This study makes a significant contribution to the field of 3D printing since it solves a critical problem with the printing procedure and offers insightful information about how non-Newtonian and high-viscosity materials behave during the deposition process.

Chapter 3

Methodology

3.1 Implementation

A literature view has been done in Chapter 2. After Literature Survey, a 3D printer is selected having a key feature that is cost-effective easily constructible, and portable. Our objective is to make a high-precision 3D printer. There are many types of making a 3-D printer, we selected the additive manufacturing type fused deposition modelling-based printer because it meets our requirement of cost-effective and high precision. The concept behind the making of an FDM 3D printer is that it uses thermoplastic material that melts and is forced out to make the desired shape. The process starts with the CAD modelling of any object, once we get a hands-on 3D structure, we will convert that into instructions and feed it to the printer by coding using Arduino at mega which is the controller we will use. RAMPS 1.4 will control the movement of stepper motors. RAMPS is also used for the interference between the computer and the controller. When the 3D printer gets the instruction from the controller it starts making the work-piece. The 3D object is made layer by layer such that when the instructions are fed to the printer, the extruder pushes the plastic filament the and hot end melts it, producing the layer-by-layer 3D object on the printer bed. [5]

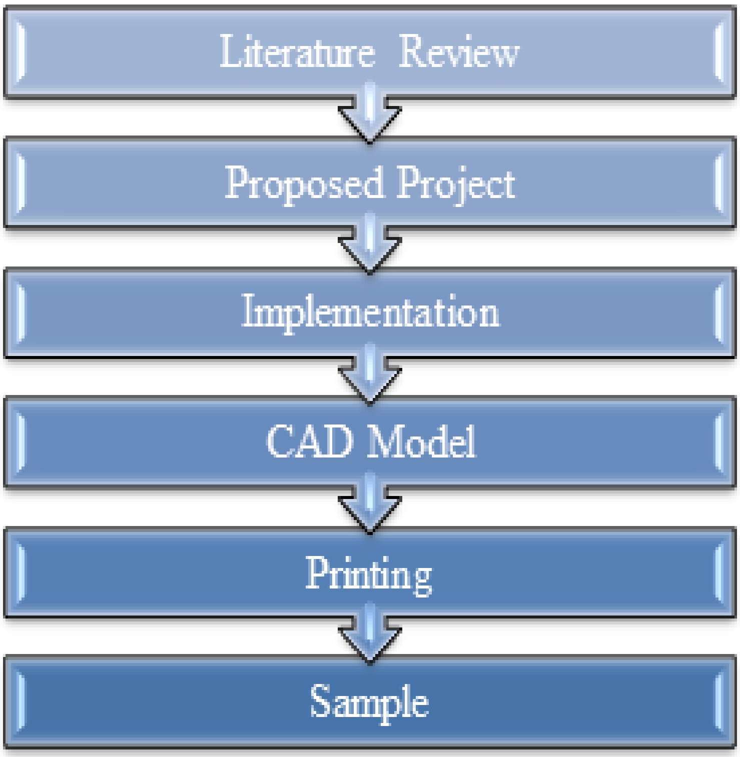


Figure 3.1: Methodology

3.2 Working

The 3D printer works on the basic principle i.e.; a digital model is converted into an actual three-dimensional object by layering material on top of it. An extrusion nozzle slides over a build platform in FDM 3D printing, which operates both horizontally and vertically. The procedure includes layering a 3D object made of a thermoplastic material that reaches a melting point and is then forced out. FDM can be understood by imagining a simple glue gun; where a solid plastic filament is pushed to the heated nozzle which then extrudes a fine stream of molten, the molten plastic. An FDM-based 3D printer uses solid polymer filament – this filament is pushed by a stepper motor into the extruder which consists of a heated nozzle – the nozzle is usually heated to 200 degrees Celsius. The extruder is manipulated by 3 stepper-motor on the X Y and Z axis. The motors that manipulate the extruder in the X Y Z plane are given instructions by the computer to define a path for proper printing. The resolution of the printer depends upon the diameter of the nozzle and the precision of stepper motors. This is one of the most common and cost-effective methods to 3D print an object.

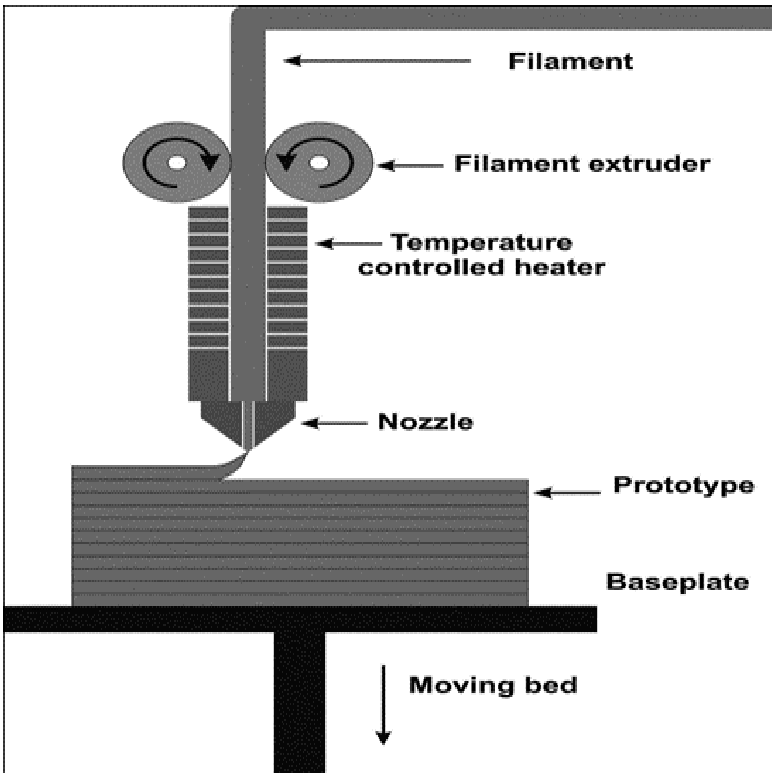


Figure 3.2: Working

3.3 Block diagram

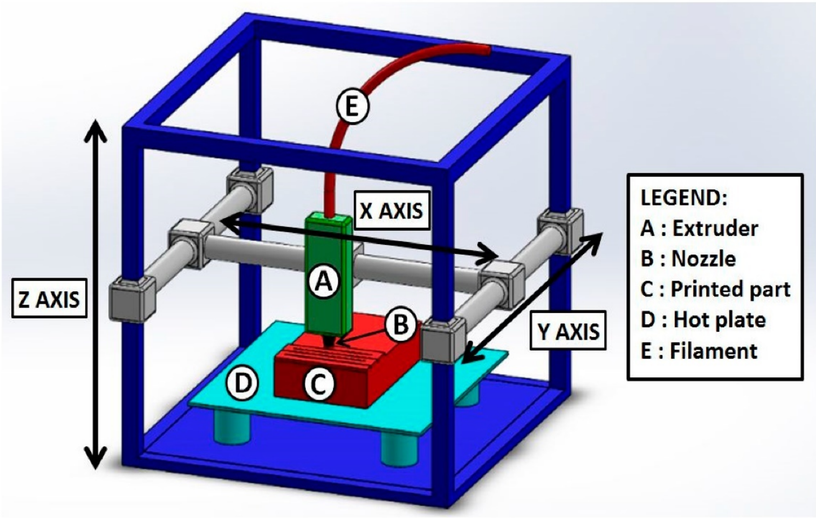


Figure 3.3: Working Prototype

3.3.1 Functional Block diagram

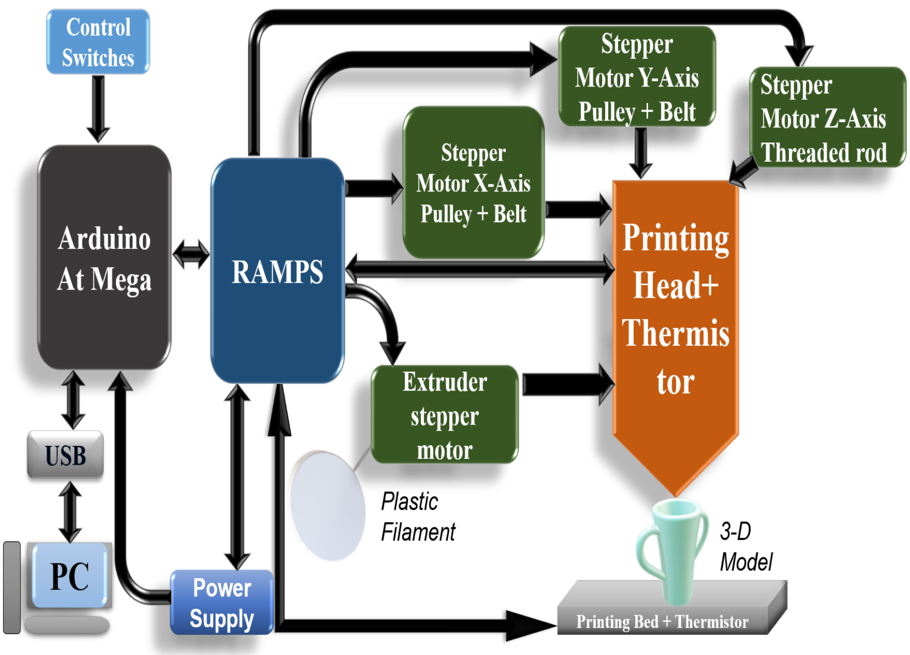


Figure 3.4: Functional Block diagram

3.3.2 Process Flow Chart

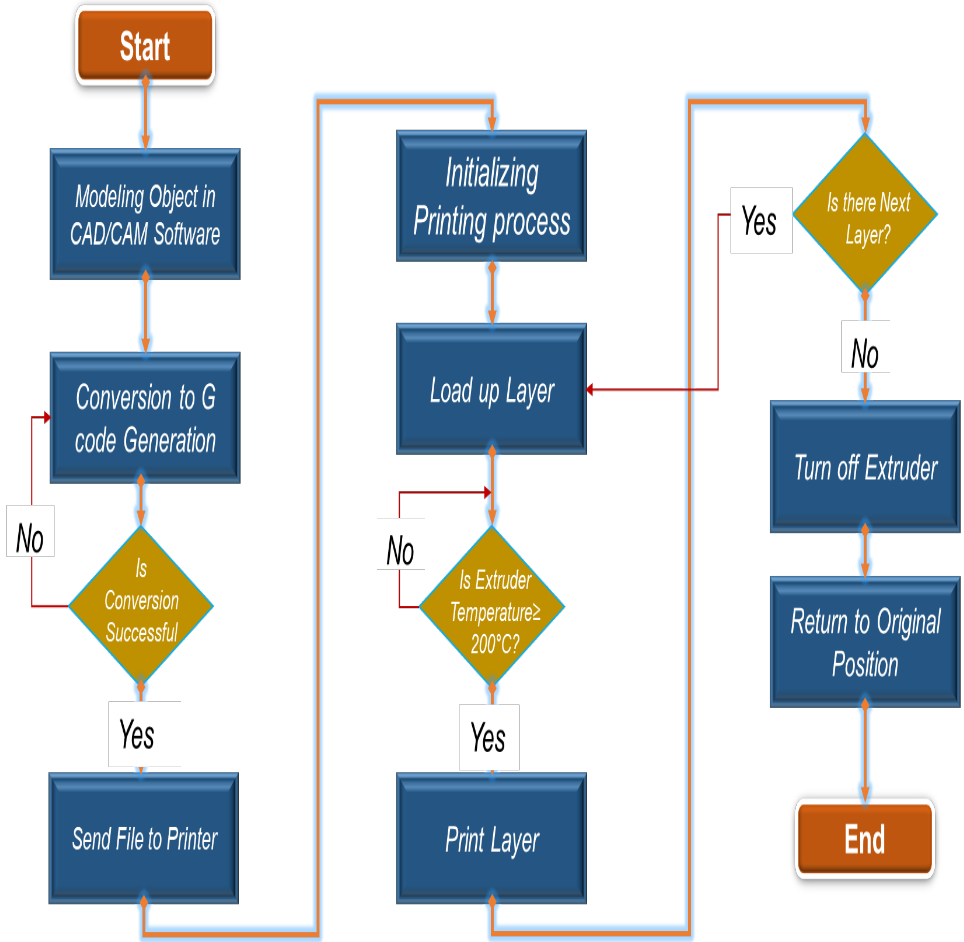


Figure 3.5: Flow Chart Diagram

Chapter 4
Components

4.1 Electrical Components

4.1.1 Print Bed

This is the surface that the printed part lies on while being produced. Ambient or heated bed temperatures are both options. A heated bed will maintain a temperature between 40°C and 110°C throughout the print, depending on the material.

Figure 4.1: Print Bed

4.1.2 Extruder

The component that squirts the plastic out is not the extruder. The component that feeds plastic filament into the hot end is the extruder. Extruders can be integrated into the hot end or remote, and they commonly push the filament into the hot end using a stiff PTFE

(Teflon) tube (also known as the Bowden cable). We can print two different materials or colours at once with a dual extruder.

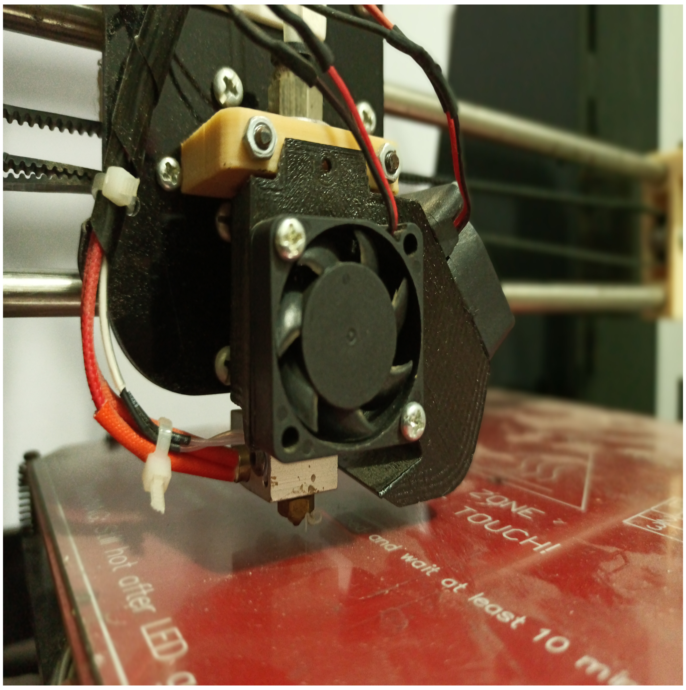


Figure 4.2: Extruder

4.1.3 Hot end

E3D v6 is a specific kind of hot end, a part of a 3D printer that is in charge of melting and extruding the printing substance. The United Kingdom-based company E3D is the one who makes it. High-performance hot ends, like the E3D v6 hot end, are made to work with a variety of printing materials, like PLA, ABS, PETG, and nylon. Experienced 3D printer users and qualified 3D printing service providers frequently employ it because of its reputation for dependability, durability, and consistency.A stainless-steel core, a high-temperature PTFE liner, and a thermistor for precise temperature detection are some of the features that make the E3D v6 hot end ideal for high-quality 3D printing. Usually, a controller board like an Arduino Mega 2560 and a firmware like Marlin or Repetier are used with it.

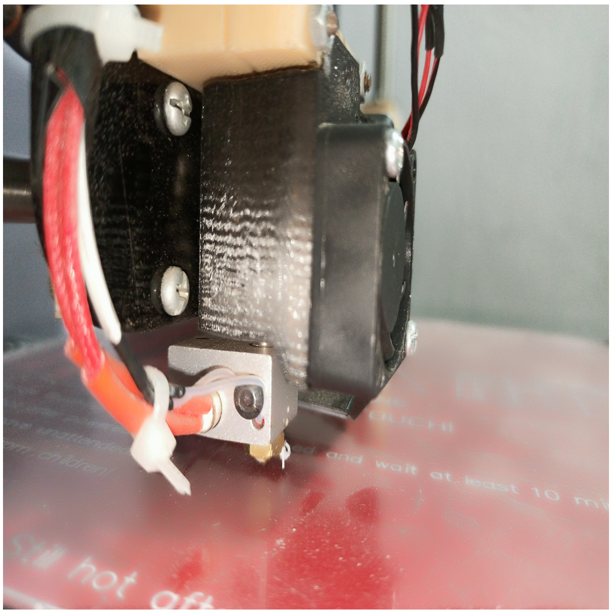


Figure 4.3: Hot end

4.1.4 *Stepper motor*

A brushless DC electric motor that divides a whole rotation into a number of equal steps is called a stepper motor. Association of National Electrical Manufacturers. Electronics technical standards are created by NEMA.Stepping motor NEMA 17 has a 1.8° step angle (200 steps per revolution). There are four phases and a 12 V DC rated voltage. [6]

Figure 4.4: Stepper motor

4.1.5 Arduino Atmega 2560

Arduino is an embedded device that is used for controlling different types of electronic functions. A micro controller board called the Arduino Mega 2560 is based on the Atmega2560 micro controller. It has a 16 MHz crystal oscillator, 54 digital input/output pins, 16 analogue inputs, 4 hardware serial ports (UARTs), a USB connector, a power jack, an ICSP header, and a reset button. Most Arduino software, including the Integrated Development Environment (IDE) for writing and uploading code to the board, is compatible with it. For tasks like large-scale projects, robotics, or 3D printing that call for more memory or digital I/O ports than the Arduino Uno can provide, the Arduino Mega 2560 is frequently employed. The Arduino Mega, which the Mega 2560 succeeds, has been updated.

Figure 4.5: Arduino Atmega 2560

4.1.6 RAMPS

Ramps 1.4 (RepRap Arduino Mega Polulu Shield) is a controller board for 3D printers, which is based on the Arduino Mega 2560 microcontroller. It can support a variety of 3D printer setups and is made to control up to 5 stepper motors (for the X, Y, Z, E0, and E1 axes). A high-current output for driving stepper motors, several end stop connectors for determining the location of the print head, and a heat bed connector for heating the print bed are just a few of the features of the Ramps 1.4 board. For creating and uploading code to the board, it is generally used in conjunction with the Arduino Mega

2560 and the Arduino Integrated Development Environment (IDE). The abbreviation RepRap stands for ”replicating rapid prototype.” The interface between the electronic components of a RepRap 3D printer and the controller computer, the Arduino Mega, is a board called RAMPS. The computer takes data from files containing information about the thing you want to print and converts it into digital events, such applying voltage to a particular pin. 1. Supply Voltage: 12 V 2. Fused at 5A for additional safety and component protection 3. 3. Heated bed control with additional 11A fuse 4. 4. Pololu boards are on pin header sockets so they can be replaced easily or removed for use in future designs. 5. 5. I2C and SPI pins left available for future expansion. 6. 2 stepper motor for Z axis in parallel.

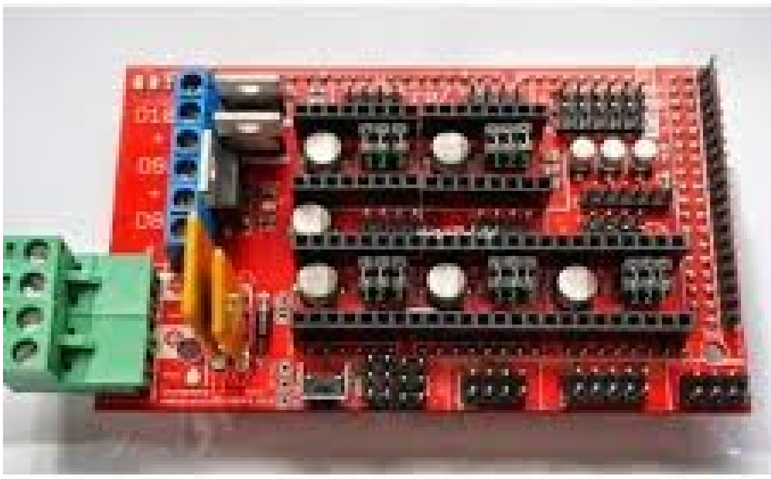


Figure 4.6: RAMPS 1.4

4.1.7 Optical End Stop Switch

An optical end stop switch is a type of switch that determines the location of an object using both a light source and a light sensor. The print head, the component of the printer that goes back and forth to deposit the printing material, is often utilised with an optical end stop switch to detect its position.The optical end stop switch activates when the print head approaches the limit of its range of motion, telling the printer to cease going in that direction. Because they are precise, dependable, and simple to install, optical end stop switches are frequently utilised in 3D printers. They are frequently chosen over mechanical end stop switches since they don't degrade over time and don't need to be in direct contact with the printer's moving parts. [7]

Figure 4.7: Optical End Stop Switch

4.1.8 PCB Heat Bed

The heated works are based on the idea that the amount of heat produced by an electric current running through a conductor is inversely proportional to the amperes that conductor is carrying. A heated build plate that is frequently used in 3D printers is a PCB heat bed. The construction platform is heated to the proper temperature for printing using a printed circuit board (PCB) with inbuilt heating components. A substance like

BuildTak or PEI is generally coated on the heat bed, which is normally made of a sturdy, heat-resistant material like aluminum or borosilicate glass, to aid in the printed objects' adhesion to the build platform. The controller board of the 3D printer, which controls the heat bed's temperature, is connected to the heat bed. A PCB heat bed can help to lessen warping and enhance the quality of the printed products in addition to giving a stable, heated surface for printing.

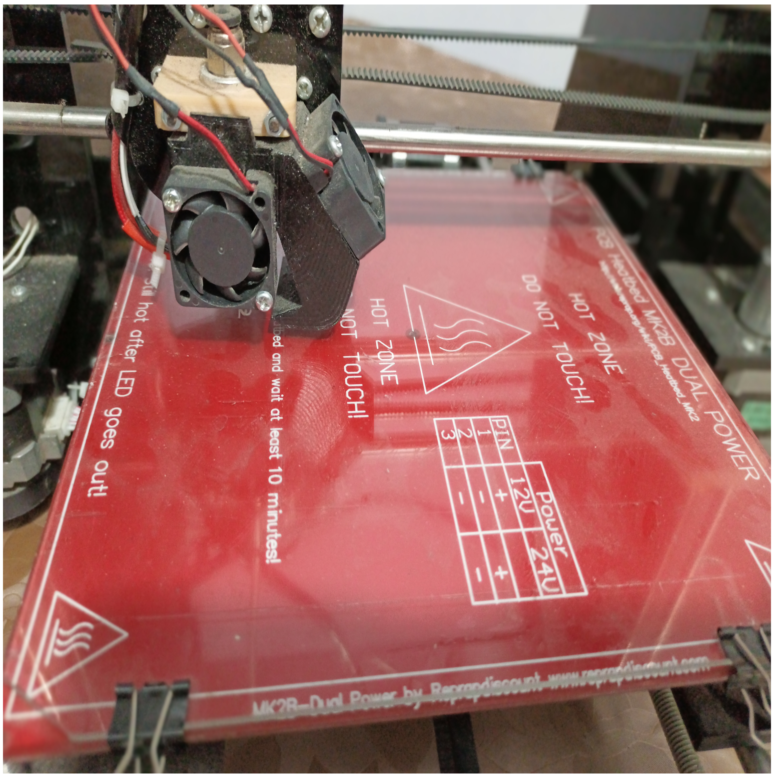


Figure 4.8: PCB Heat Bed

4.1.9 A4988 Stepper motor driver

The A4988 is a stepper motor driver that is frequently used in 3D printers and other applications that need exact control over the motor's position and speed. The manufacturer is Allegro Micro-systems. The A4988 can translate a high-level step and direction signal into the proper low-level signals to drive the stepper motor because it is a comprehensive micro stepping motor driver with built-in translator. Additionally, it has on-board voltage regulators and several safety features to guarantee the motor driver's dependability. The Arduino Mega 2560 is compatible with the A4988, which is simple to use and works with a variety of microcontrollers.

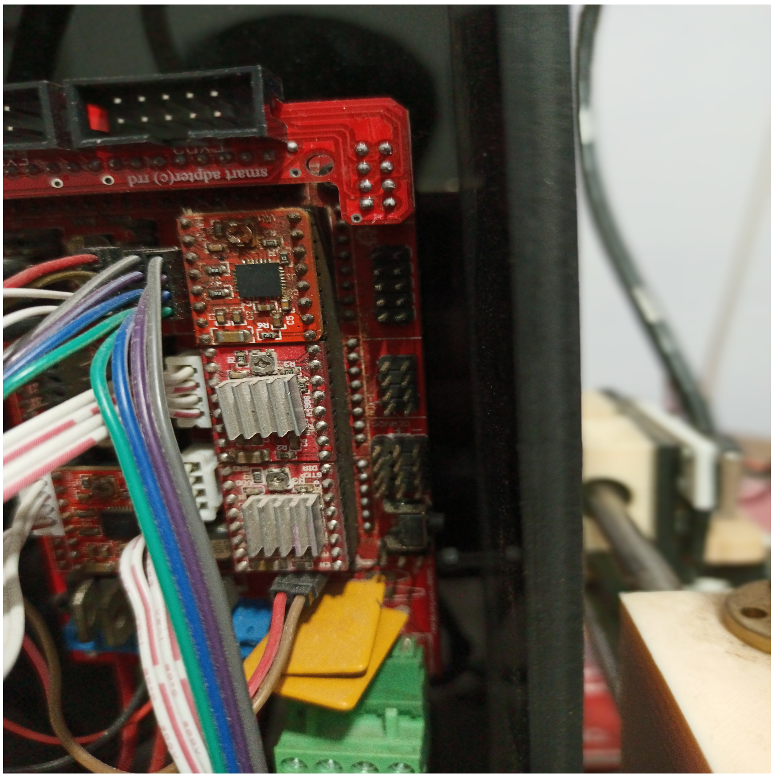


Figure 4.9: Stepper motor driver

4.1.10 Power Supply 12V/15A

A power supply that is rated at 12V/20A can deliver a maximum current of 20A while maintaining a consistent output voltage of 12V. Applications that call for a moderate quantity of power, such 3D printers, small appliances, and LED lighting systems, frequently use this kind of power source.

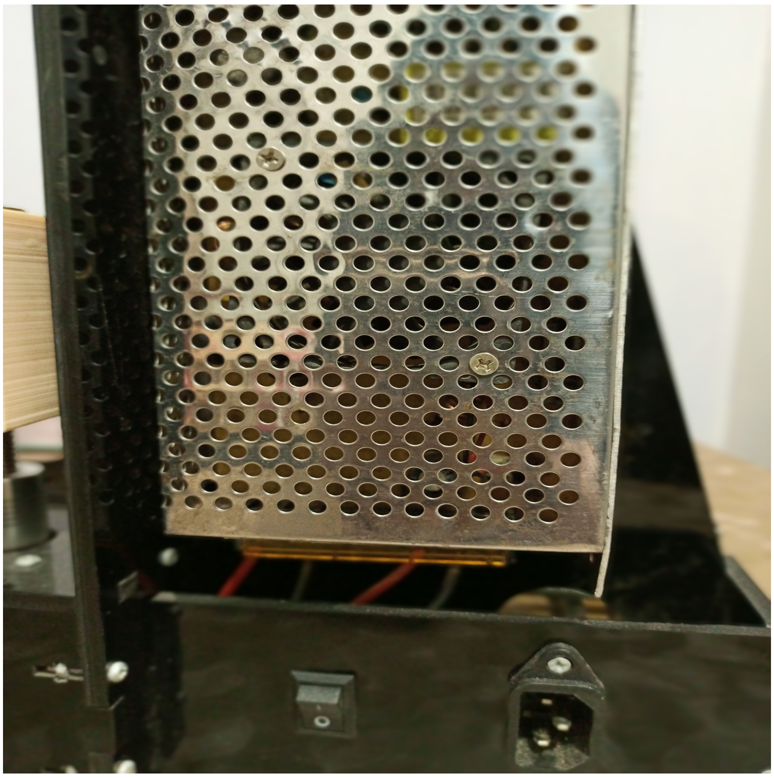


Figure 4.10: Power Supply

4.2 Mechanical Components

4.2.1 Gantry

The bed only travels along the z-axis whereas the extruder moves along the x- and y-axes. It is a component of the printer's frame that allows for horizontal motion. In other words, a gantry is a 3D printer structure that holds a sliding part.

Figure 4.11: Gantry

4.2.2 Plastic Filament PLA

Usually manufactured from corn or potatoes, PLA (poly-lactic acid or polylactide) is a biodegradable material. Without a heated bed, PLA filament can be extruded at a lower temperature of 160 to 220°C.

Figure 4.12: Plastic Filament

4.2.3 SC8UU Linear Motion Ball Bearing Slide Bushing

A typical linear motion ball-bearing slide bushing for 3D printers and other precise motion control applications is the SC8UU. It is produced in China by a business called SBR. The SC8UU sliding bushing has an 8mm diameter and can withstand stresses of up to 2000N. It comprises of a precision-machined aluminum housing that houses a linear ball bearing and a low-friction sliding bushing. A retainer separates the slide bushing from the ball bearing, keeping the ball bearing in place and lowering the possibility that it may come loose. It is simple to install and compatible with a variety of linear motion systems, including those used in 3D printers, is the SC8UU slide bushing.

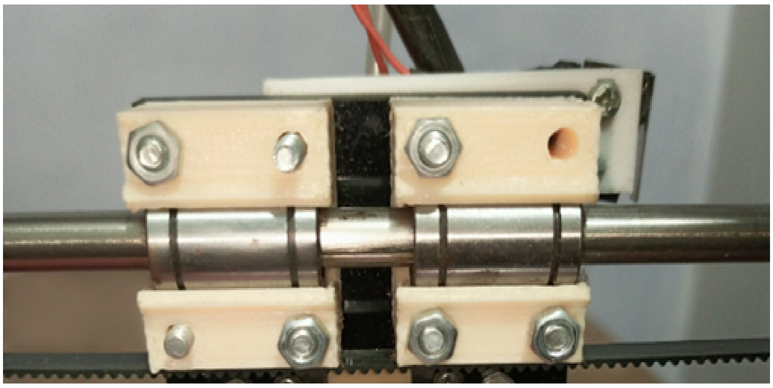


Figure 4.13: Linear Motion Ball Bearing Slide Bushing

4.2.4 SC12UU Linear Motion Ball Bearing Slide Bushing

A common type of linear rail shaft rod used in 3D printers and other precise motion control applications is the SK12UU. It is produced in China by a business called SBR. The 12mm-diameter SK12UU rail shaft rod can withstand loads of up to 5000N in weight. It is constructed from premium hardened steel, and its smooth, polished surface lessens wear. The 3D printer's moving parts are supported and guided by the rail shaft rod, which is frequently used in conjunction with linear bearings or sliding bushings. The SK12UU rail shaft rod is easy to install and is compatible with a wide range of linear motion systems, including the ones used in 3D printers.Easy to install and suitable for a variety of linear motion systems, including those found in 3D printers, the SK12UU rail shaft rod.

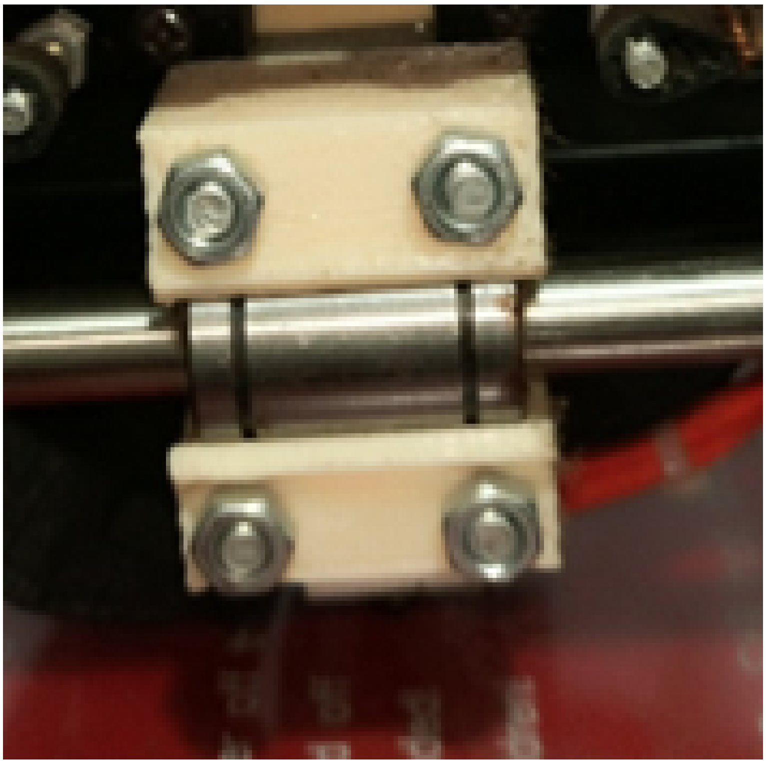


Figure 4.14: Linear Motion Ball Bearing Slide Bushing

4.2.5 Pedestal bearing

A revolving or oscillating shaft is supported by a particular type of bearing called a pedestal bearing. A pedestal or other type of support structure is often used to mount it, giving the bearing a secure and sturdy base. In situations where the shaft is subjected to heavy loads, rapid speeds, or misalignment, pedestal bearings are frequently used. They are frequently utilised in situations where the shaft is not exactly straight or where the thermal expansion coefficients of the bearing housing and the shaft differ. To accommodate various uses, pedestal bearings come in a variety of sizes, designs, and materials. Ball bearings, roller bearings, and hydrodynamic bearings are a few typical types of pedestal bearings.

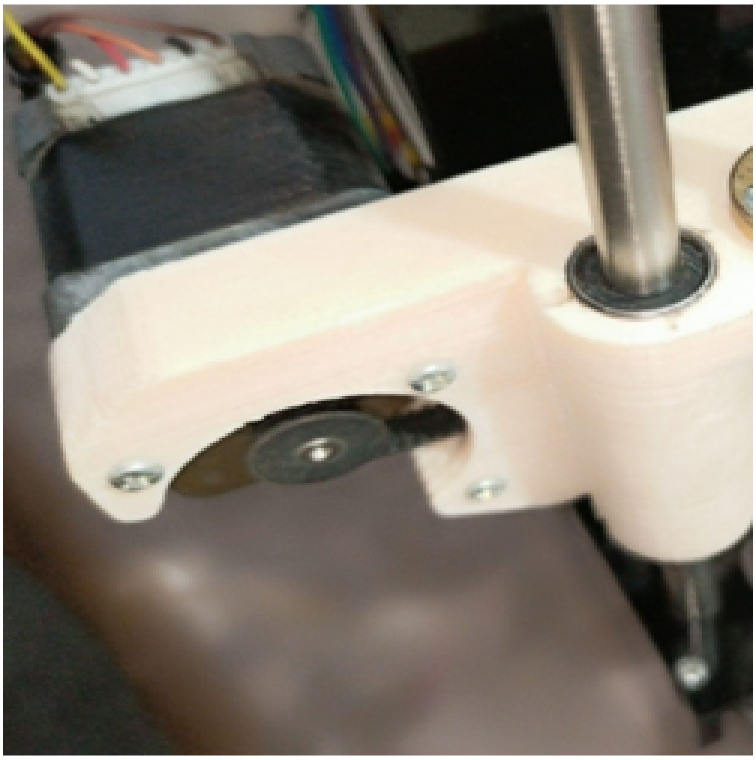


Figure 4.15: Pedestal bearing

4.2.6 T8 8mm Lead Screw with Nut

In 3D printers and other precision motion control applications, a type of screw and nut assembly known as a T8 8mm lead screw with nut is frequently utilised. The T8 screw moves by 2mm for each complete spin thanks to its 8mm diameter and 2mm pitch. Usually constructed of high-grade, hardened steel, the screw has a trapezoidal thread shape that offers a large carrying capacity and less friction. The nut is often made of brass or another low-friction material and is engineered to precisely and smoothly fit the T8 screw. The T8 8mm lead screw with nut, which is frequently used to drive the moving components of a 3D printer, such as the print head and the print bed, transforms rotational action into linear motion. The rotational force needed to move the screw is often supplied by a stepper motor or other sort of actuator, which is attached to it.

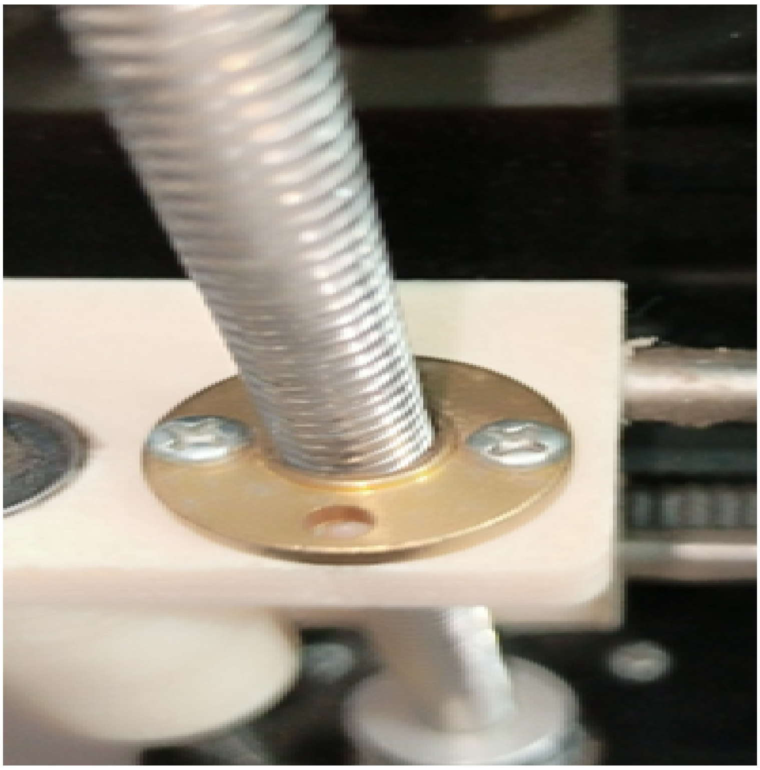


Figure 4.16: Lead Screw with Nut

4.2.7 Stepper motor coupler

A stepper motor coupler is used in 3D printers to join the extruder-driven stepper motor to the filament drive gear. The filament drive gear is the device that forces the filament into the extruder, which is the component of the 3D printer that melts the filament and puts it onto the print bed. To enable the stepper motor to control the movement of the filament, a stepper motor coupler is utilised to transmit torque from the motor to the filament driving gear. A 3D printer can employ a variety of stepper motor coupler types, and the choice of coupler will rely on the particulars of the printer's design and the needs of the application. [8]

Figure 4.17: Stepper motor coupler

4.2.8 GT2 20-tooth flanged pulley

A common kind of pulley used in 3D printing and other processes that demand exact positioning and control is the GT2 20 tooth flanged pulley. The ”20 tooth” component of the name relates to the number of teeth on the pulley, and GT2 is the name of a standard for pulleys and belts used in motion control systems. A flanged pulley is a particular kind of pulley that has a projecting rim or flange on one side. The belt is supported and guided by this flange, helping to maintain the belt’s perfect alignment with the pulley. The flange also aids in preventing the belt from coming loose from the pulley, which is necessary to maintain precise and reliable motion. In a 3D printer, a GT2 20 tooth flanged pulley is often used on the extruder stepper motor, the bed carriage, and other parts of the printer that require precise movement.The extruder stepper motor, the bed carriage, and other components of a 3D printer that need precise movement frequently employ a GT2 20 teeth flanged pulley.

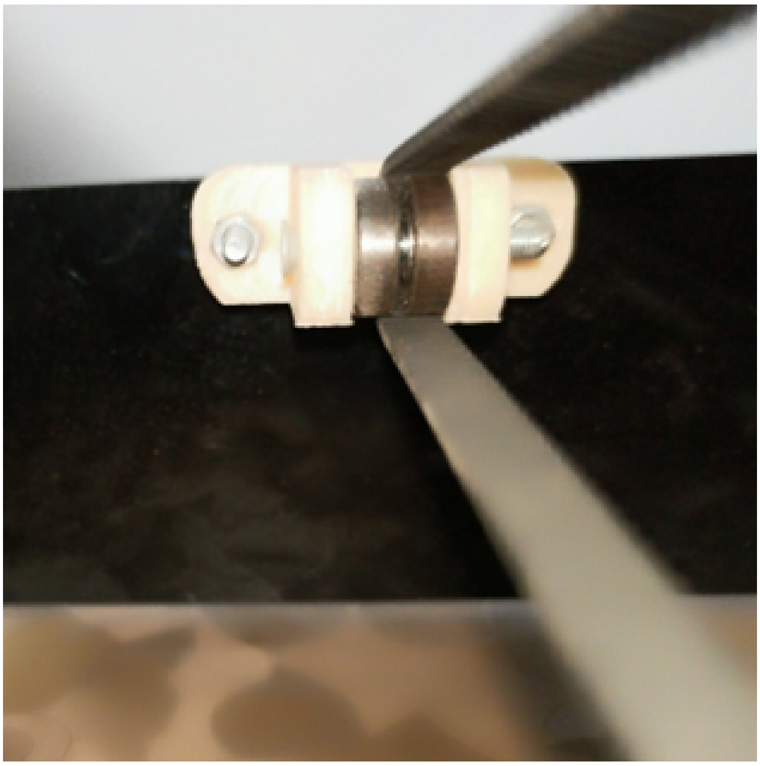


Figure 4.18: GT2 20-tooth flanged pulley

4.2.9 Timing Belt for GT2 pulley

A form of toothed belt called a timing belt for a GT2 pulley is intended for use with GT2 pulleys in motion control systems. In 3D printing and other applications that call for exact placement and control, pulleys and belts that adhere to the GT2 standard are frequently utilised. The belt's toothed shape helps it to grab the pulley teeth, keeping it securely in place and preventing slipping. For the system to continue to move accurately and consistently, this is crucial. Neoprene or polyurethane are common materials used to make timing belts for GT2 pulleys. These belts come in a variety of lengths and widths to suit a variety of applications.

Figure 4.19: Timing Belt for GT2 pulley

Chapter 5

Effect on Environment and Sustainability

3-D printers are widely used nowadays. The most common type of 3D printer used is the Additive type manufactured (AM) by additive manufacturing we mean that the 3D models are made layer by layer. They have a variety of benefits like they are used to design 3D objects and, their prototype could be helpful for industrial projects. Industrial projects on large scale are first prototype into 3D models for the demonstration which helps in better understanding. We can see that any project which is made is modelled in 3-D so that viewers can have a visual understanding of how their project will look once it is made public. Our project is a fused deposition modelling-based (FDM) 3D printer that has a great impact on society, it is an environmentally friendlier printer. First of all, it's a portable 3D printer by portable we mean that one carries it anywhere in their homes, and the education sector and can use it for office work too. This printer is the low cost that's what our one of the objectives, its price is low as compared to other 3 D printers so that a common citizen could afford this. Also, FDM 3D printers are environment friendlier in a way that for the production of 3D models they use plastic polymers. This plastic material is biodegradable i.e., it can be reusable and they have low melting point than other fossils-based plastics. PLA a plastic polymer is one of the two most used plastics for 3D printing i.e., it has a 45 percent market share because of its low cost and low melting point. [9]

- **Reduce waste and overproduction:**The 3D printer enables customized and on-demand production, reducing the amount of waste and resources used in traditional manufacturing processes.
- **Promote sustainable manufacturing:** The use of open source components and software promotes collaboration and innovation in the field of 3D printing, enabling the development of more sustainable manufacturing practices.

- **Encourage economic growth:** The affordability and accessibility of this 3D printer technology can promote economic growth by providing access to manufacturing capabilities and job opportunities.
- **Support education and research:** The lightweight and portable design of the printer makes it ideal for use in educational institutions, promoting STEM education and research.
- **Enable innovation:** The customizable design and dual extruder capabilities of this 3D printer enable the development of new applications and industries, driving innovation and creating new job opportunities.Reduce carbon footprint: By enabling localized production, this 3D printer reduces the need for long-distance shipping and transportation of goods, thereby reducing the associated carbon footprint.
- **Promote circular economy:** The ability to use recycled materials in 3D printing can promote a circular economy, where waste is minimized and resources are reused.
- **Increase accessibility:** The affordability and accessibility of this 3D printer technology can promote inclusivity and reduce the digital divide, providing more people with access to manufacturing capabilities and job opportunities.
- **Address healthcare needs:** 3D printing technology can be used to produce medical devices and prosthetics, improving healthcare access and affordability in underprivileged areas.
- **Empower local communities:** The ability to customize and produce products locally can empower local communities, enabling them to become more self-sufficient and reducing their dependence on external resources.

Chapter 6

Schematics

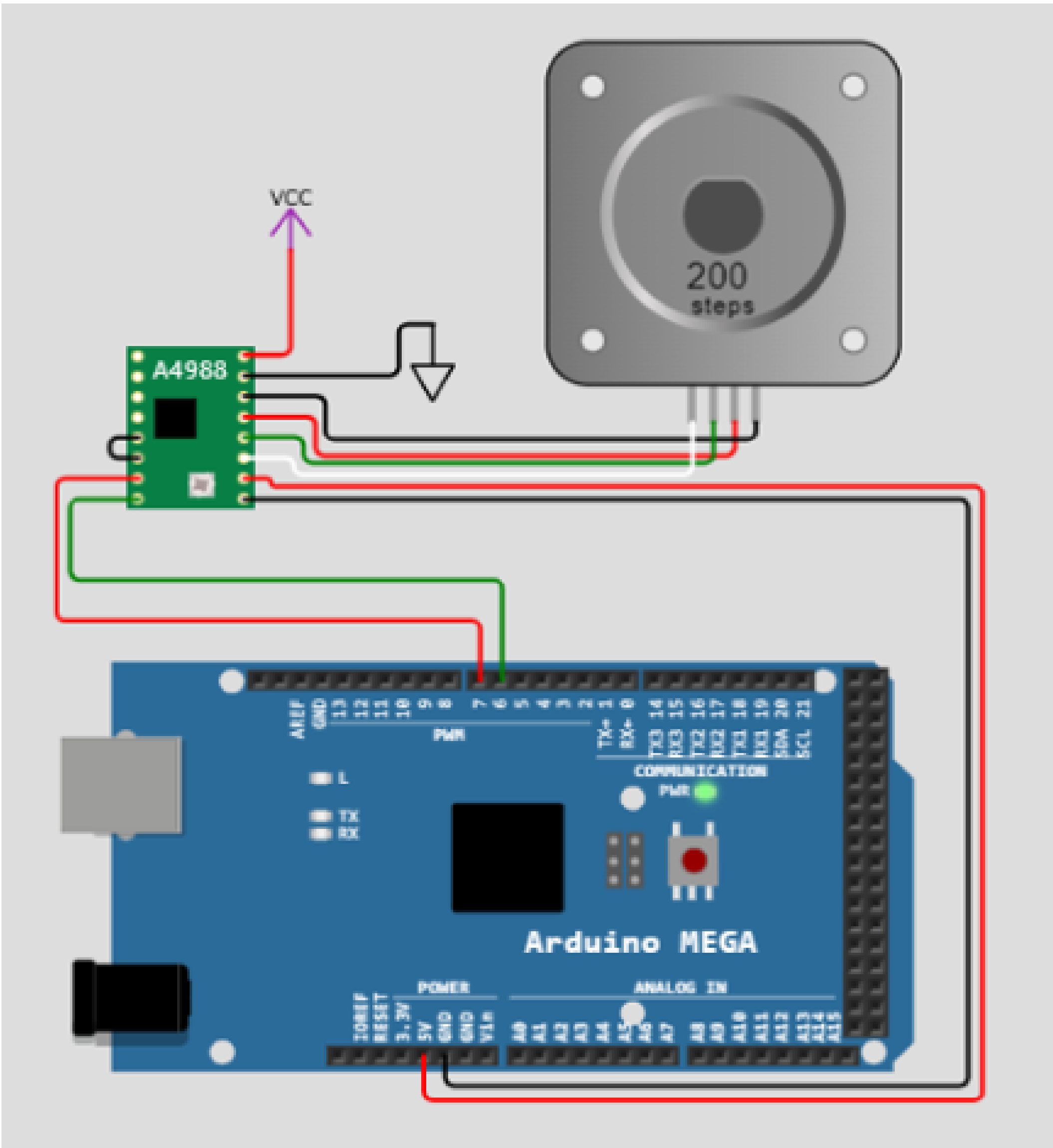


Figure 6.1: calibration of driver

The A4988 stepper motor driver can be calibrated to ensure that the stepper motor moves accurately and smoothly.

- **V**CC and GND to 5V and GND on the Arduino board respectively. MS1, MS2, and MS3 to digital pins on the Arduino board. These pins determine the microstepping resolution of the driver. STEP and DIR to digital pins on the Arduino board. These pins control the direction and speed of the motor.

- **S**et the microstepping resolution by setting the MS1, MS2, and MS3 pins to the appropriate digital values. The following table shows the microstepping resolution for different combinations of MS1, MS2, and MS3:

MS1	MS2	MS3	Microstepping Resolution
0	0	0	Full step
1	0	0	Half step
0	1	0	Quarter step
1	1	0	Eighth step
1	1	1	Sixteenth step

- **C**onnect a multimeter to the VREF pin of the A4988 driver. VREF is the reference voltage used to set the current limit for the motor. The current limit should be set to the rated current of the motor.

- **A**pply power to the Arduino board and the A4988 driver.

- **S**et the current limit by adjusting the VREF voltage using a potentiometer connected to the VREF pin.

- **T**est the motor by running a simple program that moves the motor back and forth. Adjust the STEP and DIR pins to control the direction and speed of the motor. If the motor is not moving smoothly, adjust the VREF voltage until the motor moves smoothly.

- **O**nce the motor is moving smoothly, you can fine-tune the VREF voltage to optimize the motor performance. The optimal VREF voltage is the lowest voltage that allows the motor to move smoothly without skipping steps or overheating the driver.

[10]

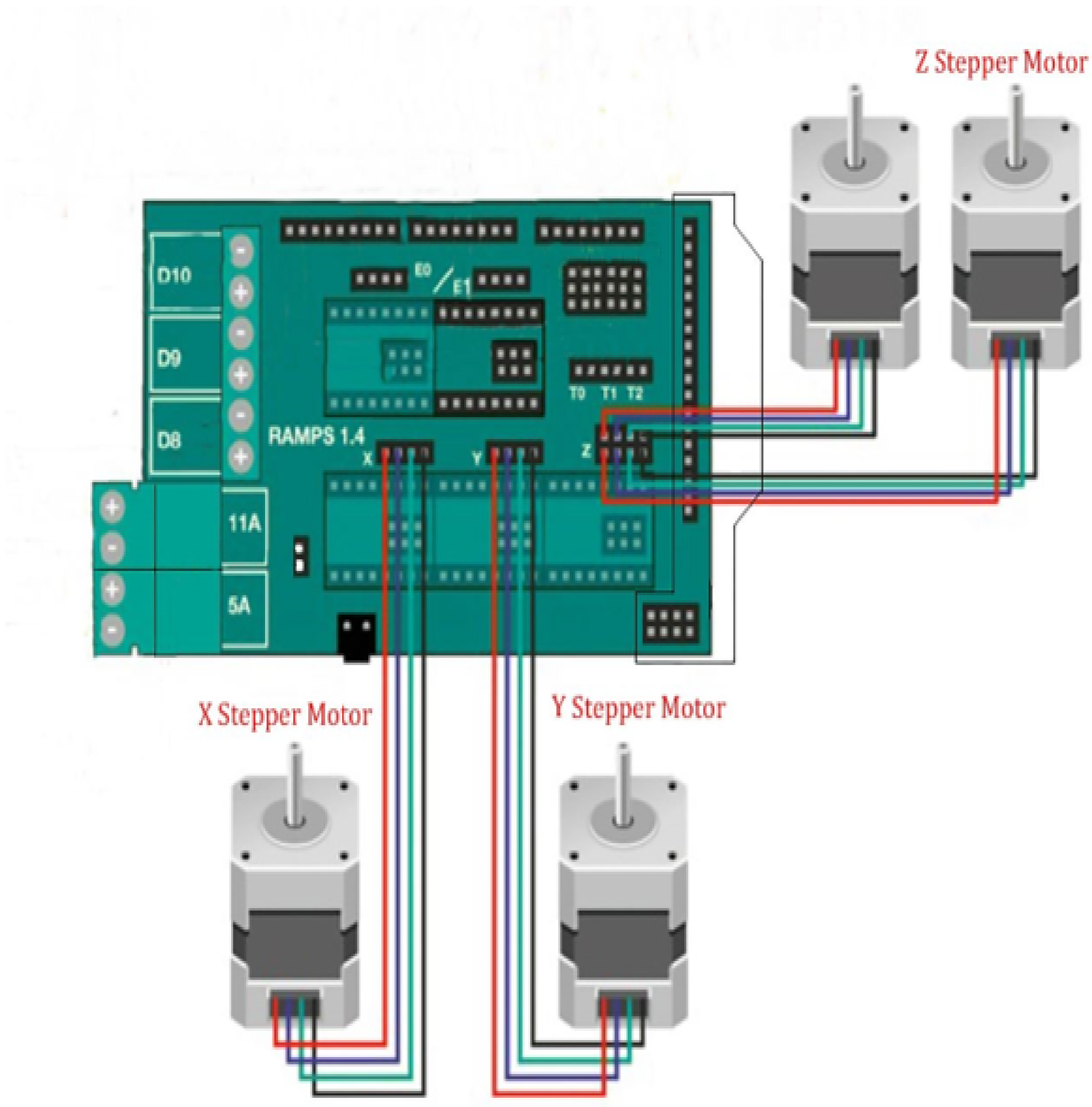


Figure 6.2: Stepper Motor for X, Y, Z Direction

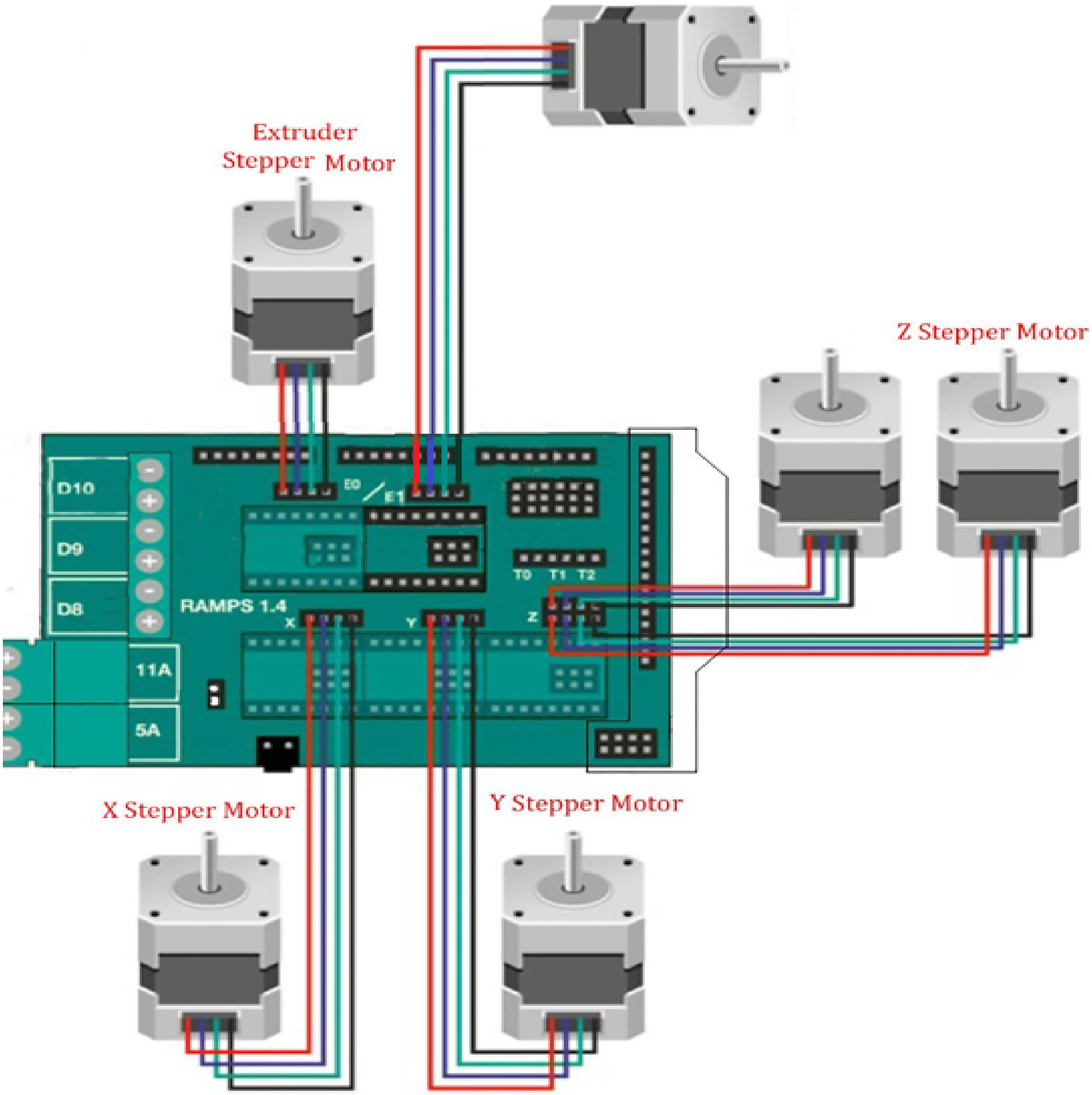


Figure 6.3: Extruder Motors

The RAMPS board is a popular open-source electronics platform used to control 3D printers.The RAMPS board, which is in the middle of this interface, is connected to a number of components, including stepper motor drivers, endstop switches, and a heated bed. The stepper motor drivers are in charge of regulating the movement of the printer's axes, and the endstop switches aid in precisely locating the printer's position. The build plate is maintained at a constant temperature with the help of the heated bed, ensuring that the printed object adheres to the surface appropriately.

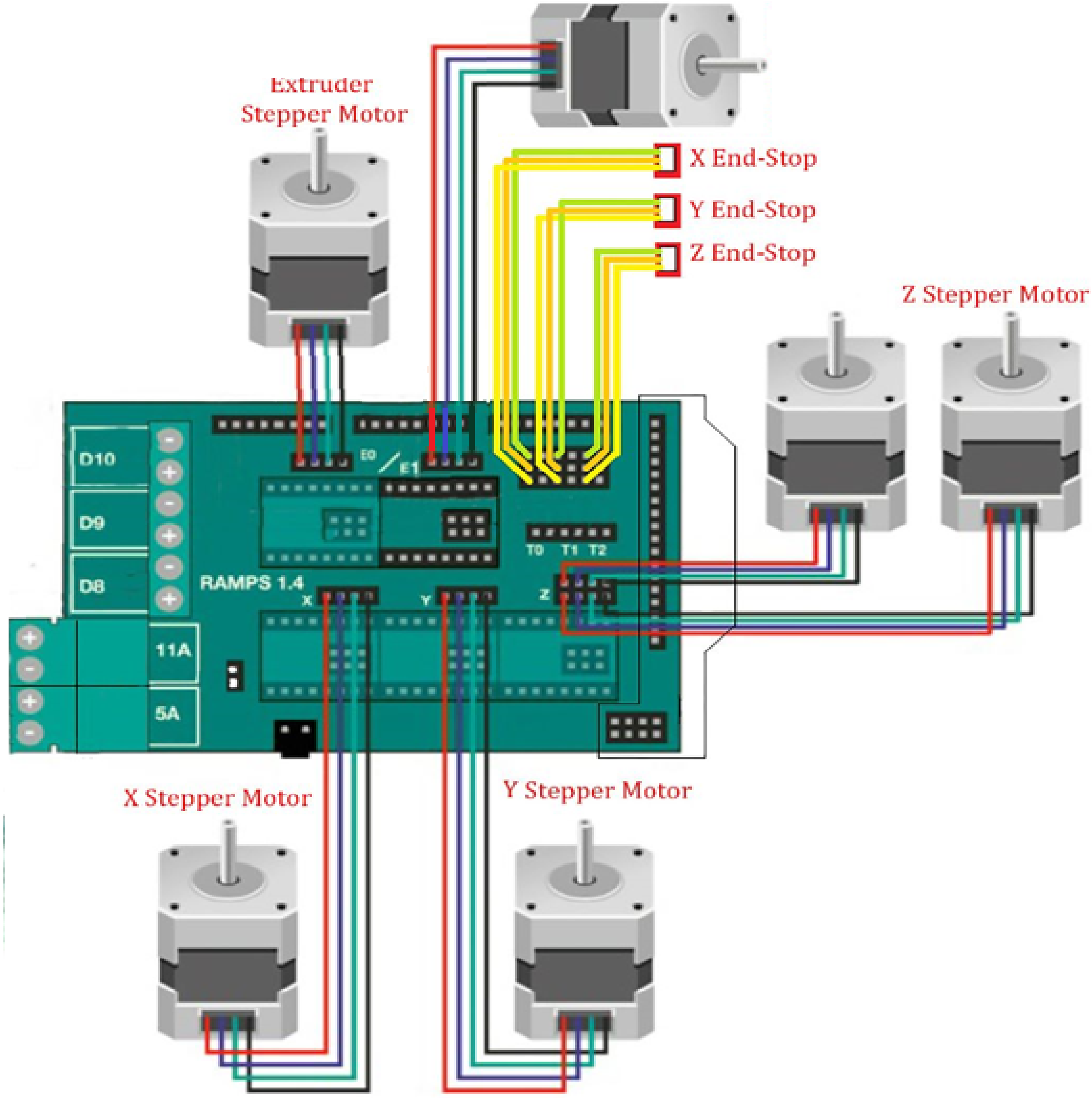


Figure 6.4: End Stop Switches

We can see a thermistor, a cooling fan, and an extruder attached to the board in addition to the components shown in the previous image. The hotend, the component of the printer that melts and extrudes the filament, is where the thermistor, a temperature sensor, measures the temperature. After it is extruded, the printed object is cooled using the cooling fan, which reduces warping and assures superior print quality. The filament must be pushed into the hotend by the extruder in order for it to be melted and extruded there.

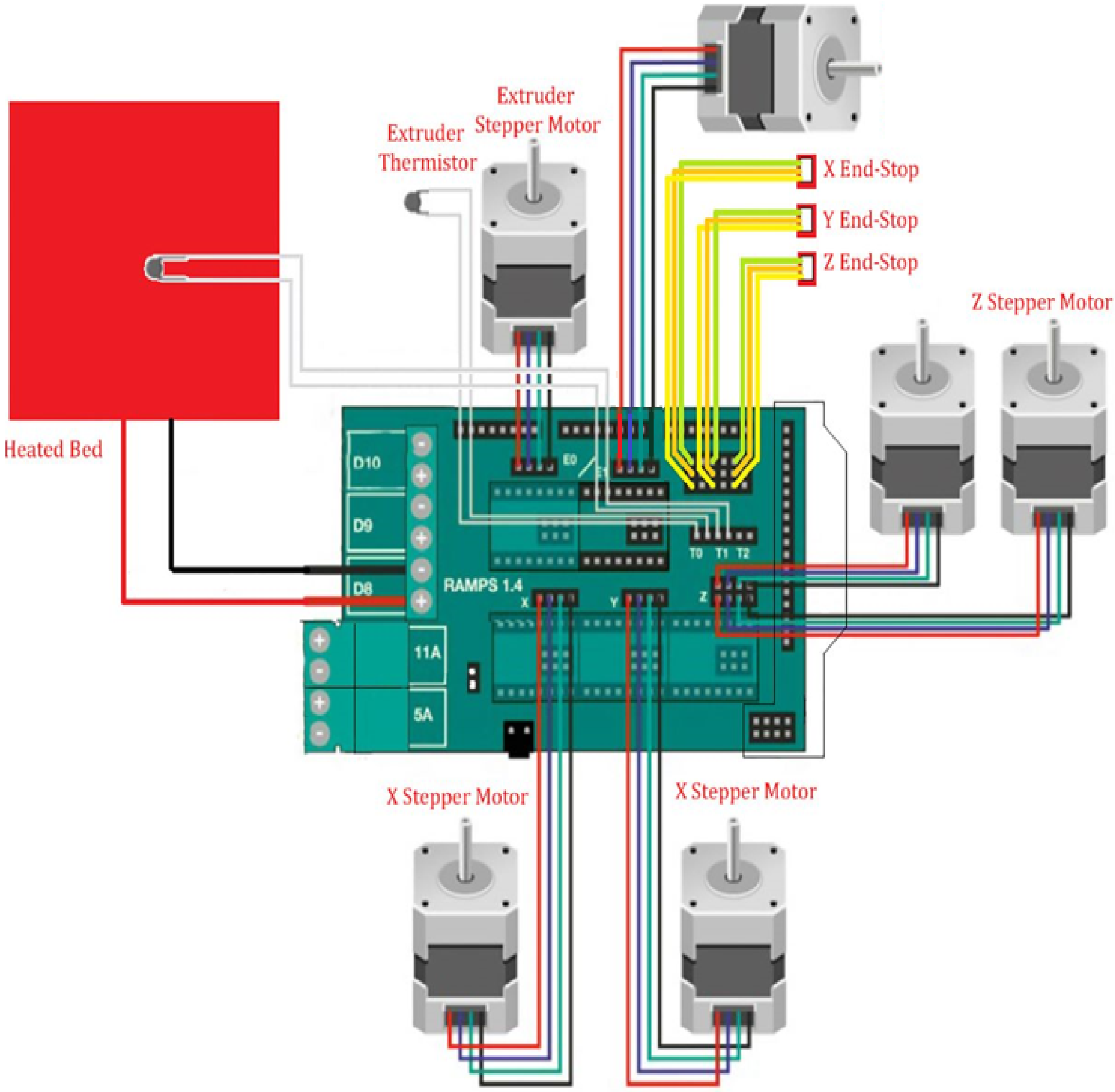


Figure 6.5: Hotend Bed

A stepper motor, a hobbed gear, a filament guide tube, and a hotend make up the extruder assembly. The hobbed gear, which the stepper motor drives, grabs and forces the filament into the hotend. The filament guide tube makes sure that it enters the hotend smoothly. A heating element, a thermistor, and a nozzle make up the hotend, which is the component of the printer that melts and extrudes the filament.

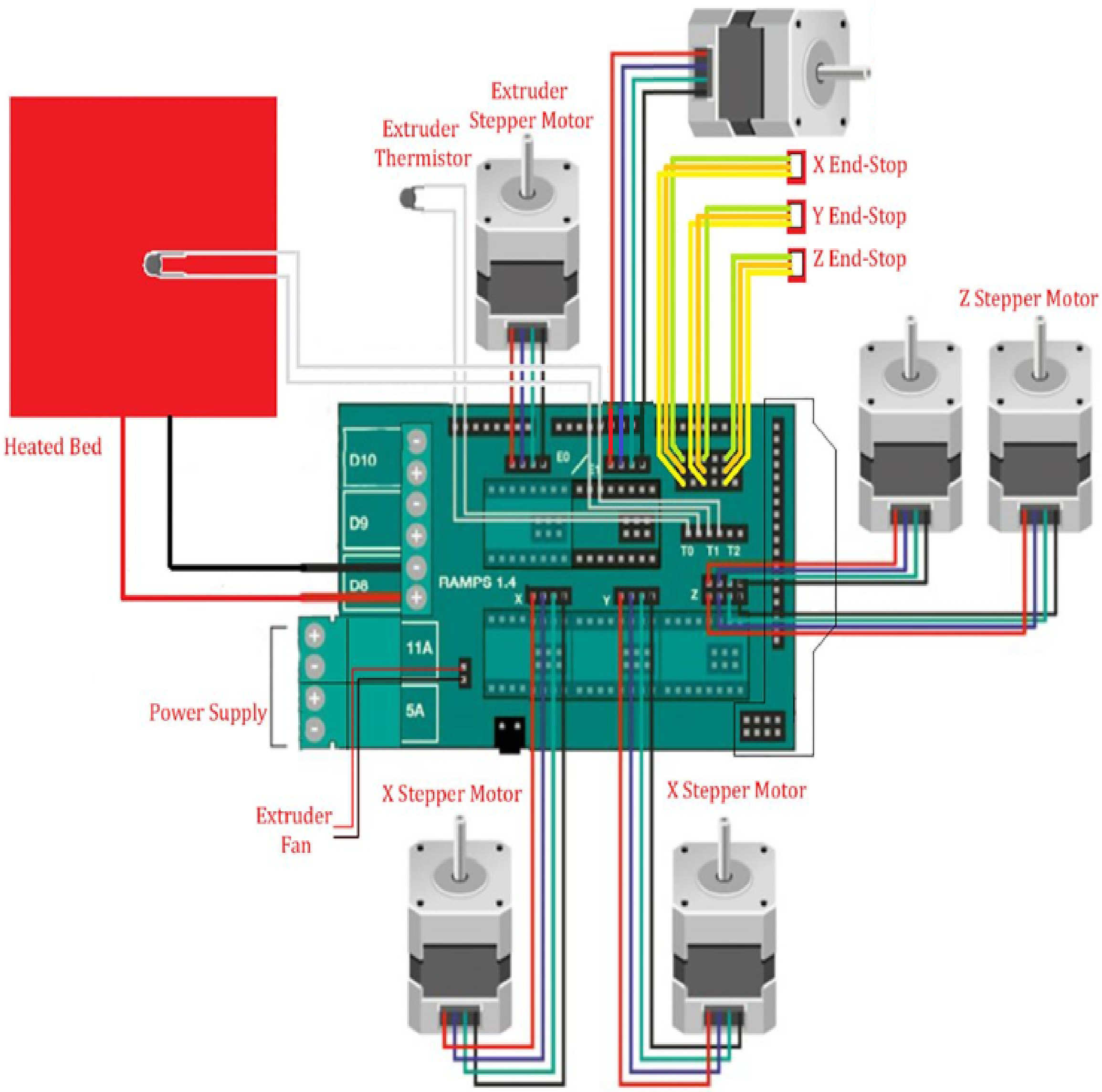


Figure 6.6: Schematics of Printer

A PCB heating element, a thermistor, and an aluminum build plate make up the heated bed assembly. A thin layer of copper traces that produce heat when a current is carried through them make up the PCB heating element. The build plate temperature is measured by the thermistor, allowing the printer to run at a constant temperature. The PCB heating element's heat and the aluminium build plate's flat, strong surface help the printed object cling to the surface effectively.

Chapter 7
AutoCAD Modelling

7.1 AutoCAD Design

These days, 3-D printers are commonly employed. The most popular type of 3D printer is additive manufactured (AM), which refers to the process of building 3D models layer by layer.They can develop 3D items and its prototype may be useful for industrial applications, among other advantages. Large-scale industrial projects are initially prototyped into 3D models for the display, which aids in comprehension. Any project that is created, as we can see, is modelled in three dimensions so that visitors may preview how their project will seem after it is made public. Our idea is a 3D printer using fused deposition modelling (FDM) that has a significant social and environmental impact. First of all, it's a portable 3D printer, which means that one can use it for office work as well as at home and in the educational sector. One of our goals was to make this printer affordable for the average person, therefore it is priced reasonably compared to other 3D printers. Additionally, FDM 3D printers are more environmentally friendly because they employ plastic polymers to create 3D models. This plastic is biodegradable, meaning it can be recycled, and it has a lower melting point than other polymers made from fossil fuels. Due to its low price and low melting point, PLA, a plastic polymer, has a 45 percent market share and is one of the two most popular plastics for 3D printing.[4][5] To confirm the correct compatibility of the parts and enclosure, we used a combination of CAD and designing tools to create the aforementioned prototype. [11] To create models for our prototype design, we used AutoCAD.

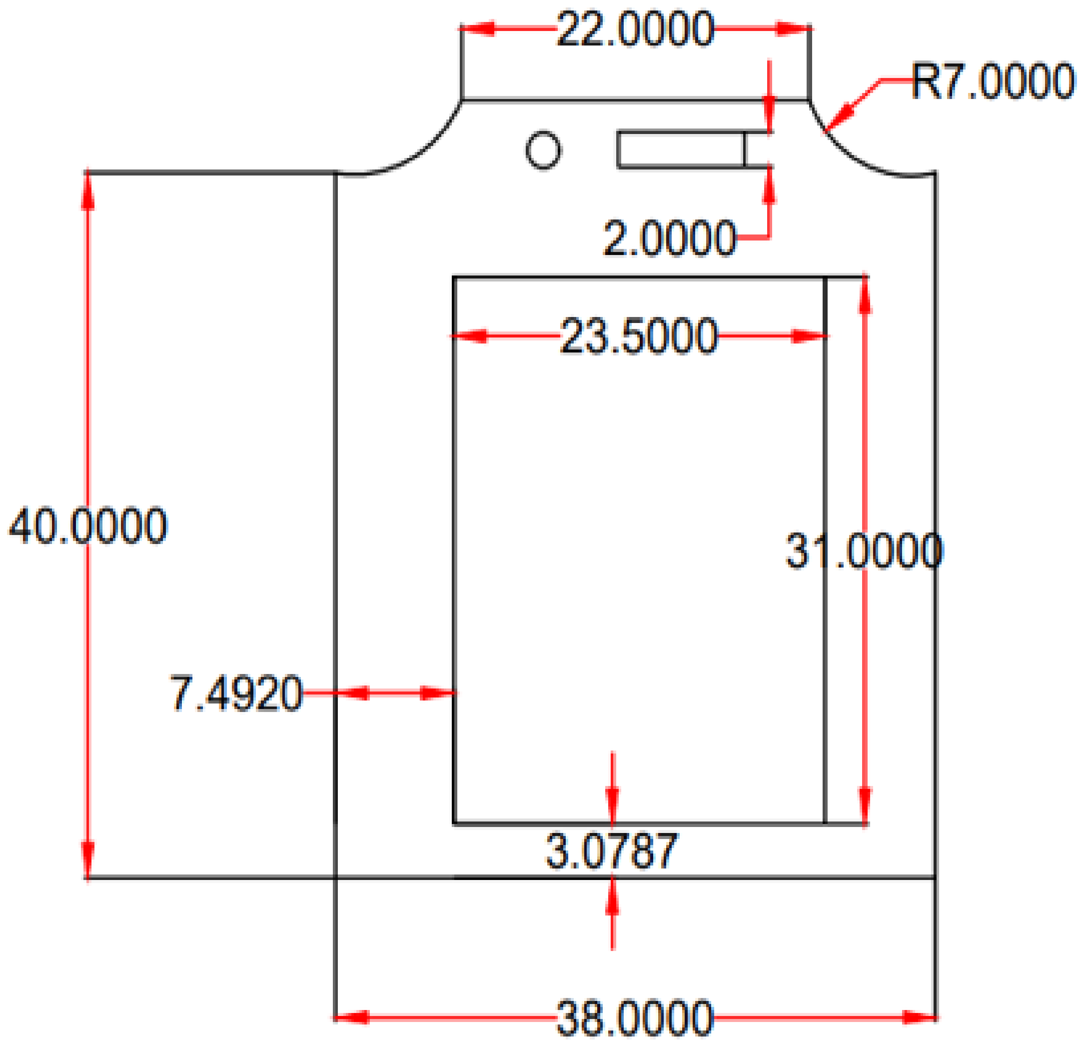


Figure 7.1: AutoCAD Drawing of frame

The front of a 3D printer is seen in this AutoCAD 2D model. The printer's frame, extruder assembly, and other parts mounted to the frame are all shown. The print bed's location and motion along the printer's Z-axis are also depicted in the model. The model also shows where the control panel and any other parts that make up the printer's front view are located. Overall, the model offers a thorough portrayal of the front-view design and construction of the printer.

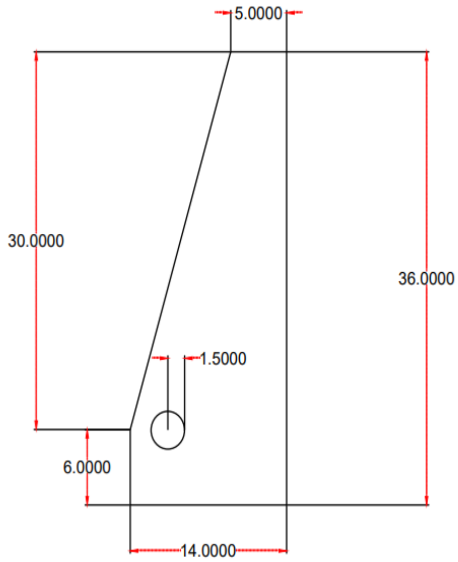


Figure 7.2: Right Side View

A 3D printer is seen in this AutoCAD 2D model from the right side. The model also demonstrates the construction of the frame and the mounting of the various components.

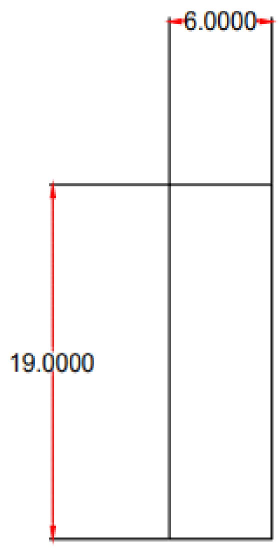


Figure 7.3: Y axis Carriage Back view

The Y axis carriage of a 3D printer is shown from the back in this AutoCAD 2D model. The carriage parts, such as the linear bearings, rods, and belt tensioners, are depicted in detail in the model. The carriage's attachment to the printer's frame and its interactions with other parts are also depicted in the model.

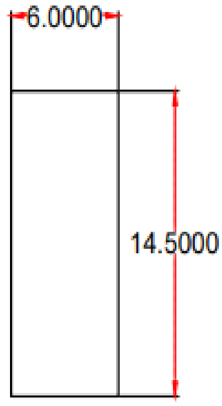


Figure 7.4: Y axis Carriage Front view

The Y-axis carriage of a 3D printer is shown in front perspective in this AutoCAD 2D model. The carriage's attachment to the printer's frame and its interactions with other parts are also depicted in the model. Overall, the model offers an accurate and thorough portrayal of the front-view design and construction of the Y-axis carriage.

Figure 7.5 is the AutoCAD 2D model displays the top view of a 3D printer's Z-axis assembly. The Z-axis's constituent parts, including the lead screw, stepper motor, and linear bearings. The Z-axis assembly's attachment to the printer's frame and its interactions with other parts are also depicted in the model. Overall, the model offers an accurate and complete representation of the top view design and construction of the Z-axis assembly.

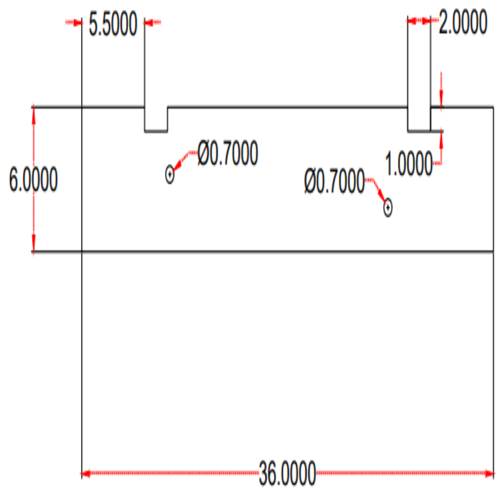


Figure 7.5: Top View For Z axis

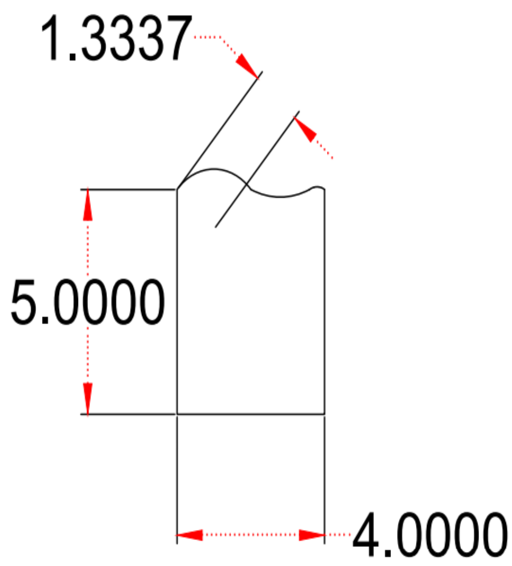


Figure 7.6: AutoCAD Drawing of Front

The AutoCAD 2D model displays the top view of a 3D printer's Z-axis assembly. The Z-axis's constituent parts, including the lead screw, stepper motor, and linear bearings. The

Z-axis assembly's attachment to the printer's frame and its interactions with other parts are also depicted in the model. Overall, the model offers an accurate and complete representation of the top view design and construction of the Z-axis assembly.

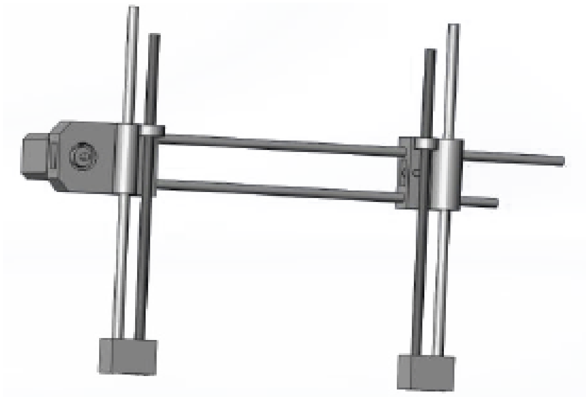


Figure 7.7: Frame Structure

The frame of a 3D printer is seen in this AutoCAD 3D model. The model has intricate drawings of the corner brackets, extruded aluminium profiles, and other parts of the printer's frame. The model demonstrates the connection and assembly of the various frame parts to create the frame structure of the printer. The model offers a comprehensive and accurate picture of the 3D printer's frame design and construction.

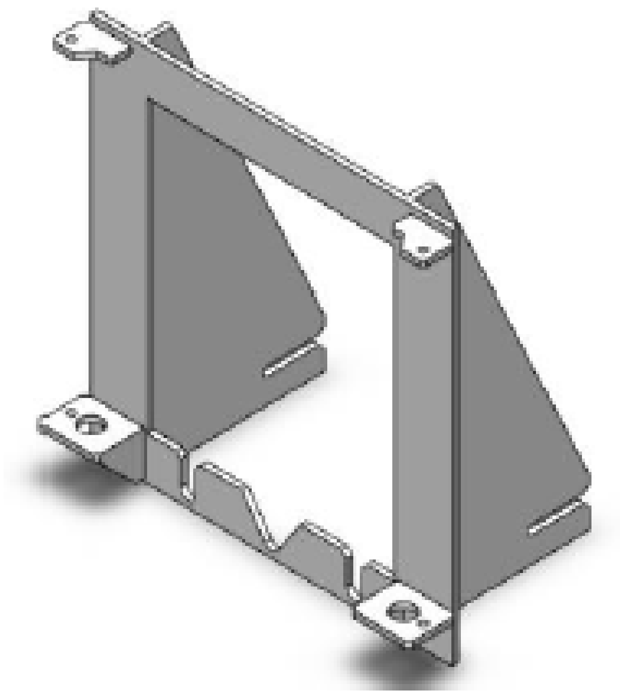


Figure 7.8: Acrylic structure

This is the Acrylic structure of a 3D printer. It depicts how the Z axis motors are positioned on the sides and the y axis is positioned horizontally.

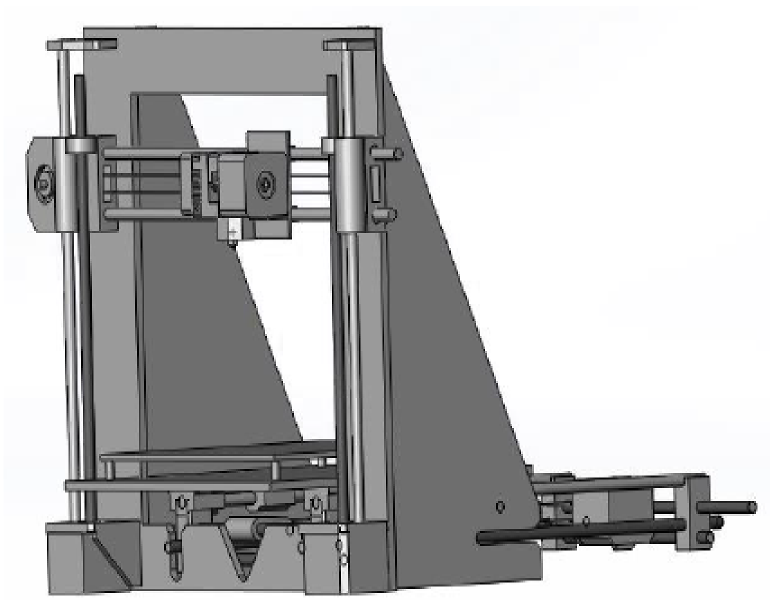


Figure 7.9: Assembled

This is the hotend and nozzle of a 3D printer's extruder assembly. The part of the extruder that melts the filament is called the hotend. Since the extruder is a Bowden extruder, the filament is powered by a motor that is situated distant from the hotend. It has a print bed with a glass plate on top of the bed in particular. The glass plate acts as a flat and smooth surface for the 3D-printed object to adhere to during printing. Clips in the corners secure the glass plate to the printer's bed.

Chapter 8
Marlin Firmware

A common open-source firmware used in 3D printers, CNC machines, and other automated systems is called Marlin Firmware. It offers functions including temperature control, motor control, and connectivity with a host computer. It is designed to regulate the movement and operation of these devices.

Marlin Firmware has a great degree of adaptability and can be set up to operate with a variety of hardware layouts and configurations. With several features that may be activated or disabled as needed, it is made to be adaptable and modular. [12]

Marlin Firmware has a number of important features, including:

- Support for a wide range of 3D printers and CNC machines
- Automatic bed leveling and calibration
- Support for multiple extruders and hotends
- Temperature control and monitoring for extruders and heated beds
- Support for G-code commands and other industry-standard protocols
- Real-time communication with a host computer via USB or other interfaces
- Advanced motion control and acceleration algorithms for smooth and accurate movement

8.1 Marlin Configuration.h code


```
/**
 * Marlin 3D Printer Firmware
 * Copyright (c) 2020 MarlinFirmware [https://github.com/MarlinFirmware/Marlin]
 *
 * Based on Sprinter and grbl.
 * Copyright (c) 2011 Camiel Gubbels / Erik van der Zalm
 *
 * This program is free software: you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation, either version 3 of the License, or
 * (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program.  If not, see <https://www.gnu.org/licenses/>.
 *
 */
#pragma once

/**
 * Configuration.h
 *
 * Basic settings such as:
 *
 * - Type of electronics
 * - Type of temperature sensor
 * - Printer geometry
 * - Endstop configuration
```

Marlin | Configuration.h | Configuration_adv.h | Version.h

```
 */
#define CONFIGURATION_H_VERSION 02010300

//===========================================================================
//============================= Getting Started =============================
//===========================================================================

/**
 * Here are some useful links to help get your machine configured and calibrated:
 *
 * Example Configs:     https://github.com/MarlinFirmware/Configurations/branches/all
 *
 * Průša Calculator:    https://blog.prusaprinters.org/calculator_3416/
 *
 * Calibration Guides:  https://reprap.org/wiki/Calibration
 *                      https://reprap.org/wiki/Triffid_Hunter%27s_Calibration_Guide
 *                      https://sites.google.com/site/repraplogphase/calibration-of-your-reprap
 *                      https://youtu.be/wAL9d7FgInk
 *
 * Calibration Objects: https://www.thingiverse.com/thing:5573
 *                      https://www.thingiverse.com/thing:1278865
 */

// @section info

// Author info of this build printed to the host during boot and M115
#define STRING_CONFIG_H_AUTHOR "(none, default config)" // Who made the changes.
//#define CUSTOM_VERSION_FILE Version.h // Path from the root directory (no quotes)

// @section machine

// Choose the name from boards.h that matches your setup
```

Marlin | Configuration.h | Configuration_adv.h | Version.h

```
// Choose the name from boards.h that matches your setup
#ifndef MOTHERBOARD
  #define MOTHERBOARD BOARD_RAMPS_14_EFB
#endif

/**
 * Select the serial port on the board to use for communication with the host.
 * This allows the connection of wireless adapters (for instance) to non-default port pins.
 * Serial port -1 is the USB emulated serial port, if available.
 * Note: The first serial port (-1 or 0) will always be used by the Arduino bootloader.
 *
 * :[-1, 0, 1, 2, 3, 4, 5, 6, 7]
 */
#define SERIAL_PORT 0

/**
 * Serial Port Baud Rate
 * This is the default communication speed for all serial ports.
 * Set the baud rate defaults for additional serial ports below.
 *
 * 250000 works in most cases, but you might try a lower speed if
 * you commonly experience drop-outs during host printing.
 * You may try up to 1000000 to speed up SD file transfer.
 *
 * :[2400, 9600, 19200, 38400, 57600, 115200, 250000, 500000, 1000000]
 */
#define BAUDRATE 115200

//#define BAUD_RATE_GCODE     // Enable G-code M575 to set the baud rate

/**
```

Marlin | Configuration.h | Configuration_adv.h | Version.h

```
// Name displayed in the LCD "Ready" message and Info menu
#define CUSTOM_MACHINE_NAME "AU FDM 3D Printer"

// Printer's unique ID, used by some programs to differentiate between machines.
// Choose your own or use a service like https://www.uuidgenerator.net/version4
//#define MACHINE_UUID "00000000-0000-0000-0000-000000000000"

// @section stepper drivers

/**
 * Stepper Drivers
 *
 * These settings allow Marlin to tune stepper driver timing and enable advanced options for
 * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
 *
 * Use TMC2208/TMC2208_STANDALONE for TMC2225 drivers and TMC2209/TMC2209_STANDALONE for TMC2226 drivers
 *
 * Options: A4988, A5984, DRV8825, LV8729, TB6560, TB6600, TMC2100,
 *          TMC2130, TMC2130_STANDALONE, TMC2160, TMC2160_STANDALONE,
 *          TMC2208, TMC2208_STANDALONE, TMC2209, TMC2209_STANDALONE,
 *          TMC26X,  TMC26X_STANDALONE,  TMC2660, TMC2660_STANDALONE,
 *          TMC5130, TMC5130_STANDALONE, TMC5160, TMC5160_STANDALONE
 * :['A4988', 'A5984', 'DRV8825', 'LV8729', 'TB6560', 'TB6600', 'TMC2100', 'TMC2130', 'TMC2130_STANDALON
 */
#define X_DRIVER_TYPE  A4988
#define Y_DRIVER_TYPE  A4988
#define Z_DRIVER_TYPE  A4988
//#define X2_DRIVER_TYPE A4988
//#define Y2_DRIVER_TYPE A4988
//#define Z2_DRIVER_TYPE A4988
//#define Z3_DRIVER_TYPE A4988
```

```
//#define Z4_DRIVER_TYPE A4988
//#define I_DRIVER_TYPE  A4988
//#define J_DRIVER_TYPE  A4988
//#define K_DRIVER_TYPE  A4988
//#define U_DRIVER_TYPE  A4988
//#define V_DRIVER_TYPE  A4988
//#define W_DRIVER_TYPE  A4988
#define E0_DRIVER_TYPE A4988
#define E1_DRIVER_TYPE A4988
//#define E2_DRIVER_TYPE A4988
//#define E3_DRIVER_TYPE A4988
//#define E4_DRIVER_TYPE A4988
//#define E5_DRIVER_TYPE A4988
//#define E6_DRIVER_TYPE A4988
//#define E7_DRIVER_TYPE A4988

/**
 * Additional Axis Settings
 *
 * Define AXISn_ROTATES for all axes that rotate or pivot.
 * Rotational axis coordinates are expressed in degrees.
 *
 * AXISn_NAME defines the letter used to refer to the axis in (most) G-code commands.
 * By convention the names and roles are typically:
 *   'A' : Rotational axis parallel to X
 *   'B' : Rotational axis parallel to Y
 *   'C' : Rotational axis parallel to Z
 *   'U' : Secondary linear axis parallel to X
 *   'V' : Secondary linear axis parallel to Y
 *   'W' : Secondary linear axis parallel to Z
 *
 * Regardless of these settings the axes are internally named I, J, K, U, V, W.
```

Marlin | Configuration.h | Configuration_adv.h | Version.h

```
#ifdef I_DRIVER_TYPE
  #define AXIS4_NAME 'A' // :['A', 'B', 'C', 'U', 'V', 'W']
  #define AXIS4_ROTATES
#endif
#ifdef J_DRIVER_TYPE
  #define AXIS5_NAME 'B' // :['B', 'C', 'U', 'V', 'W']
  #define AXIS5_ROTATES
#endif
#ifdef K_DRIVER_TYPE
  #define AXIS6_NAME 'C' // :['C', 'U', 'V', 'W']
  #define AXIS6_ROTATES
#endif
#ifdef U_DRIVER_TYPE
  #define AXIS7_NAME 'U' // :['U', 'V', 'W']
  //#define AXIS7_ROTATES
#endif
#ifdef V_DRIVER_TYPE
  #define AXIS8_NAME 'V' // :['V', 'W']
  //#define AXIS8_ROTATES
#endif
#ifdef W_DRIVER_TYPE
  #define AXIS9_NAME 'W' // :['W']
  //#define AXIS9_ROTATES
#endif

// @section extruder

// This defines the number of extruders
// :[0, 1, 2, 3, 4, 5, 6, 7, 8]
#define EXTRUDERS 2

// Generally expected filament diameter (1.75, 2.85, 3.0, ...). Used for Volumetric,
```

Marlin | Configuration.h | Configuration_adv.h | Version.h

```
// Generally expected filament diameter (1.75, 2.85, 3.0, ...). Used f
#define DEFAULT_NOMINAL_FILAMENT_DIA 1.75

// For Cyclops or any "multi-extruder" that shares a single nozzle.
#define SINGLENOZZLE

// Save and restore temperature and fan speed on tool-change.
// Set standby for the unselected tool with M104/106/109 T...
#if ENABLED(SINGLENOZZLE)
  #define SINGLENOZZLE_STANDBY_TEMP
  #define SINGLENOZZLE_STANDBY_FAN
#endif

// @section multi-material

/**
 * Multi-Material Unit
 * Set to one of these predefined models:
 *
 *   PRUSA_MMU1             : Průša MMU1 (The "multiplexer" version)
 *   PRUSA_MMU2             : Průša MMU2
 *   PRUSA_MMU2S            : Průša MMU2S (Requires MK3S extruder with m
 *   EXTENDABLE_EMU_MMU2    : MMU with configurable number of filaments
 *   EXTENDABLE_EMU_MMU2S   : MMUS with configurable number of filaments
 *
 * Requires NOZZLE_PARK_FEATURE to park print head in case MMU unit fa
 * See additional options in Configuration_adv.h.
 * :["PRUSA_MMU1", "PRUSA_MMU2", "PRUSA_MMU2S", "EXTENDABLE_EMU_MMU2",
 */
//#define MMU_MODEL PRUSA_MMU2

// A dual extruder that uses a single stepper motor
```

Marlin | Configuration.h | Configuration_adv.h | Version.h

```
//#define SWITCHING_EXTRUDER
#if ENABLED(SWITCHING_EXTRUDER)
  #define SWITCHING_EXTRUDER_SERVO_NR 0
  #define SWITCHING_EXTRUDER_SERVO_ANGLES { 0, 90 } // Angles for E0, E1[, E2, E3]
  #if EXTRUDERS > 3
    #define SWITCHING_EXTRUDER_E23_SERVO_NR 1
  #endif
#endif

// Switch extruders by bumping the toolhead. Requires EVENT_GCODE_TOOLCHANGE_#.
//#define MECHANICAL_SWITCHING_EXTRUDER

/**
 * A dual-nozzle that uses a servomotor to raise/lower one (or both) of the nozzles.
 * Can be combined with SWITCHING_EXTRUDER.
 */
//#define SWITCHING_NOZZLE
#if ENABLED(SWITCHING_NOZZLE)
  #define SWITCHING_NOZZLE_SERVO_NR 0
  //#define SWITCHING_NOZZLE_E1_SERVO_NR 1          // If two servos are used, the index of the seco
  #define SWITCHING_NOZZLE_SERVO_ANGLES { 0, 90 }   // Angles for E0, E1 (single servo) or lowered/r
  #define SWITCHING_NOZZLE_SERVO_DWELL 2500         // Dwell time to wait for servo to make physical
#endif

// Switch nozzles by bumping the toolhead. Requires EVENT_GCODE_TOOLCHANGE_#.
//#define MECHANICAL_SWITCHING_NOZZLE

/**
 * Two separate X-carriages with extruders that connect to a moving part
 * via a solenoid docking mechanism. Requires SOL1_PIN and SOL2_PIN.
 */
//#define PARKING_EXTRUDER
```

```
 * Supports more than 2 Toolheads. See https://youtu.be/JolbsAKTKf4
 */
//#define ELECTROMAGNETIC_SWITCHING_TOOLHEAD

#if ANY(SWITCHING_TOOLHEAD, MAGNETIC_SWITCHING_TOOLHEAD, ELECTROMAGNETIC_SWITCHING_TOOLHEAD)
  #define SWITCHING_TOOLHEAD_Y_POS          235         // (mm) Y position of the toolhead dock
  #define SWITCHING_TOOLHEAD_Y_SECURITY      10         // (mm) Security distance Y axis
  #define SWITCHING_TOOLHEAD_Y_CLEAR         60         // (mm) Minimum distance from dock for unobstructed X axis
  #define SWITCHING_TOOLHEAD_X_POS          { 215, 0 }  // (mm) X positions for parking the extruders
  #if ENABLED(SWITCHING_TOOLHEAD)
    #define SWITCHING_TOOLHEAD_SERVO_NR       2         // Index of the servo connector
    #define SWITCHING_TOOLHEAD_SERVO_ANGLES { 0, 180 }  // (degrees) Angles for Lock, Unlock
  #elif ENABLED(MAGNETIC_SWITCHING_TOOLHEAD)
    #define SWITCHING_TOOLHEAD_Y_RELEASE      5         // (mm) Security distance Y axis
    #define SWITCHING_TOOLHEAD_X_SECURITY   { 90, 150 } // (mm) Security distance X axis (T0,T1)
    //#define PRIME_BEFORE_REMOVE                       // Prime the nozzle before release from the dock
    #if ENABLED(PRIME_BEFORE_REMOVE)
      #define SWITCHING_TOOLHEAD_PRIME_MM           20  // (mm)   Extruder prime length
      #define SWITCHING_TOOLHEAD_RETRACT_MM         10  // (mm)   Retract after priming length
      #define SWITCHING_TOOLHEAD_PRIME_FEEDRATE    300  // (mm/min) Extruder prime feedrate
      #define SWITCHING_TOOLHEAD_RETRACT_FEEDRATE 2400  // (mm/min) Extruder retract feedrate
    #endif
  #elif ENABLED(ELECTROMAGNETIC_SWITCHING_TOOLHEAD)
    #define SWITCHING_TOOLHEAD_Z_HOP          2         // (mm) Z raise for switching
  #endif
#endif

/**
 * "Mixing Extruder"
 *   - Adds G-codes M163 and M164 to set and "commit" the current mix factors.
 *   - Extends the stepping routines to move multiple steppers in proportion to the mix.
 *   - Optional support for Repetier Firmware's 'M164 S<index>' supporting virtual tools.
```

Marlin | Configuration.h | Configuration_adv.h | Version.h

```
 *   - Optional support for Repetier Firmware's 'M164 S<index>' supporting virtual tools.
 *   - This implementation supports up to two mixing extruders.
 *   - Enable DIRECT_MIXING_IN_G1 for M165 and mixing in G1 (from Pia Taubert's reference implementation).
 */
//#define MIXING_EXTRUDER
#if ENABLED(MIXING_EXTRUDER)
  #define MIXING_STEPPERS 2        // Number of steppers in your mixing extruder
  #define MIXING_VIRTUAL_TOOLS 16  // Use the Virtual Tool method with M163 and M164
  #define DIRECT_MIXING_IN_G1      // Allow ABCDHI mix factors in G1 movement commands
  #define GRADIENT_MIX             // Support for gradient mixing with M166 and LCD
  #define MIXING_PRESETS           // Assign 8 default V-tool presets for 2 or 3 MIXING_STEPPERS
  #if ENABLED(GRADIENT_MIX)
    #define GRADIENT_VTOOL         // Add M166 T to use a V-tool index as a Gradient alias
  #endif
#endif

// Offset of the extruders (uncomment if using more than one and relying on firmware to position when changing).
// The offset has to be X=0, Y=0 for the extruder 0 hotend (default extruder).
// For the other hotends it is their distance from the extruder 0 hotend.
//#define HOTEND_OFFSET_X { 0.0, 20.00 } // (mm) relative X-offset for each nozzle
//#define HOTEND_OFFSET_Y { 0.0, 5.00 }  // (mm) relative Y-offset for each nozzle
//#define HOTEND_OFFSET_Z { 0.0, 0.00 }  // (mm) relative Z-offset for each nozzle

// @section psu control

/**
 * Power Supply Control
 *
 * Enable and connect the power supply to the PS_ON_PIN.
 * Specify whether the power supply is active HIGH or active LOW.
 */
//#define PSU_CONTROL
```

Marlin | Configuration.h | Configuration_adv.h | Version.h

```
 */
#define TEMP_SENSOR_0 1
#define TEMP_SENSOR_1 0
#define TEMP_SENSOR_2 0
#define TEMP_SENSOR_3 0
#define TEMP_SENSOR_4 0
#define TEMP_SENSOR_5 0
#define TEMP_SENSOR_6 0
#define TEMP_SENSOR_7 0
#define TEMP_SENSOR_BED 11
#define TEMP_SENSOR_PROBE 0
#define TEMP_SENSOR_CHAMBER 0
#define TEMP_SENSOR_COOLER 0
#define TEMP_SENSOR_BOARD 0
#define TEMP_SENSOR_REDUNDANT 0

// Dummy thermistor constant temperature readings, for use with 998 and 999
#define DUMMY_THERMISTOR_998_VALUE  25
#define DUMMY_THERMISTOR_999_VALUE 100

// Resistor values when using MAX31865 sensors (-5) on TEMP_SENSOR_0 / 1
#if TEMP_SENSOR_IS_MAX_TC(0)
  #define MAX31865_SENSOR_OHMS_0      100 // (Ω) Typically 100 or 1000 (PT100 or PT1000)
  #define MAX31865_CALIBRATION_OHMS_0 430 // (Ω) Typically 430 for Adafruit PT100; 4300 for Adafruit PT1000
#endif
#if TEMP_SENSOR_IS_MAX_TC(1)
  #define MAX31865_SENSOR_OHMS_1      100
  #define MAX31865_CALIBRATION_OHMS_1 430
#endif
#if TEMP_SENSOR_IS_MAX_TC(2)
  #define MAX31865_SENSOR_OHMS_2      100
  #define MAX31865_CALIBRATION_OHMS_2 430
```

```
#endif
#if TEMP_SENSOR_IS_MAX_TC(1)
  #define MAX31865_SENSOR_OHMS_1      100
  #define MAX31865_CALIBRATION_OHMS_1 430
#endif
#if TEMP_SENSOR_IS_MAX_TC(2)
  #define MAX31865_SENSOR_OHMS_2      100
  #define MAX31865_CALIBRATION_OHMS_2 430
#endif

#if HAS_E_TEMP_SENSOR
  #define TEMP_RESIDENCY_TIME         10  // (seconds) Time to wait for hotend to "settle" in M109
  #define TEMP_WINDOW                  1  // (°C) Temperature proximity for the "temperature reached" timer
  #define TEMP_HYSTERESIS              3  // (°C) Temperature proximity considered "close enough" to the target
#endif

#if TEMP_SENSOR_BED
  #define TEMP_BED_RESIDENCY_TIME     10  // (seconds) Time to wait for bed to "settle" in M190
  #define TEMP_BED_WINDOW              1  // (°C) Temperature proximity for the "temperature reached" timer
  #define TEMP_BED_HYSTERESIS          3  // (°C) Temperature proximity considered "close enough" to the target
#endif

#if TEMP_SENSOR_CHAMBER
  #define TEMP_CHAMBER_RESIDENCY_TIME 10  // (seconds) Time to wait for chamber to "settle" in M191
  #define TEMP_CHAMBER_WINDOW          1  // (°C) Temperature proximity for the "temperature reached" timer
  #define TEMP_CHAMBER_HYSTERESIS      3  // (°C) Temperature proximity considered "close enough" to the target
#endif

/**
 * Redundant Temperature Sensor (TEMP_SENSOR_REDUNDANT)
 *
 * Use a temp sensor as a redundant sensor for another reading. Select an unused temperature sensor, and another
```

```
#if TEMP_SENSOR_REDUNDANT
  #define TEMP_SENSOR_REDUNDANT_SOURCE    E1  // The sensor that will provide the redundant reading.
  #define TEMP_SENSOR_REDUNDANT_TARGET    E0  // The sensor that we are providing a redundant reading for.
  #define TEMP_SENSOR_REDUNDANT_MAX_DIFF  10  // (°C) Temperature difference that will trigger a print abort.
#endif

// Below this temperature the heater will be switched off
// because it probably indicates a broken thermistor wire.
#define HEATER_0_MINTEMP   5
#define HEATER_1_MINTEMP   5
#define HEATER_2_MINTEMP   5
#define HEATER_3_MINTEMP   5
#define HEATER_4_MINTEMP   5
#define HEATER_5_MINTEMP   5
#define HEATER_6_MINTEMP   5
#define HEATER_7_MINTEMP   5
#define BED_MINTEMP        5
#define CHAMBER_MINTEMP    5

// Above this temperature the heater will be switched off.
// This can protect components from overheating, but NOT from shorts and failures.
// (Use MINTEMP for thermistor short/failure protection.)
#define HEATER_0_MAXTEMP 255
#define HEATER_1_MAXTEMP 275
#define HEATER_2_MAXTEMP 275
#define HEATER_3_MAXTEMP 275
#define HEATER_4_MAXTEMP 275
#define HEATER_5_MAXTEMP 275
#define HEATER_6_MAXTEMP 275
#define HEATER_7_MAXTEMP 275
#define BED_MAXTEMP      60
#define CHAMBER_MAXTEMP  60
```

```
 */
#define HOTEND_OVERSHOOT 15   // (°C) Forbid temperatures over MAXTEMP - OVERSHOOT
#define BED_OVERSHOOT    10   // (°C) Forbid temperatures over MAXTEMP - OVERSHOOT
#define COOLER_OVERSHOOT  2   // (°C) Forbid temperatures closer than OVERSHOOT

//===========================================================================
//============================= PID Settings ================================
//===========================================================================

// @section hotend temp

// Enable PIDTEMP for PID control or MPCTEMP for Predictive Model.
// temperature control. Disable both for bang-bang heating.
#define PIDTEMP          // See the PID Tuning Guide at https://reprap.org/wiki/PID_Tuning
//#define MPCTEMP        // ** EXPERIMENTAL **

#define BANG_MAX 255     // Limits current to nozzle while in bang-bang mode; 255=full current
#define PID_MAX BANG_MAX // Limits current to nozzle while PID is active (see PID_FUNCTIONAL_RANGE below); 255=full current
#define PID_K1 0.95      // Smoothing factor within any PID loop

#if ENABLED(PIDTEMP)
  //#define PID_DEBUG             // Print PID debug data to the serial port. Use 'M303 D' to toggle activation.
 // #define PID_PARAMS_PER_HOTEND // Use separate PID parameters for each extruder (useful for mismatched extruders)
                                  // Set/get with G-code: M301 E[extruder number, 0-2]

  #if ENABLED(PID_PARAMS_PER_HOTEND)
    // Specify up to one value per hotend here, according to your setup.
    // If there are fewer values, the last one applies to the remaining hotends.
    #define DEFAULT_Kp_LIST {  22.20,  22.20 }
    #define DEFAULT_Ki_LIST {   1.08,   1.08 }
    #define DEFAULT_Kd_LIST { 114.00, 114.00 }
  #else
    #define DEFAULT_Kp  26.75
    #define DEFAULT_Ki   1.25
    #define DEFAULT_Kd 142.98
  #endif
#endif
```

```
#if ENABLED(MPCTEMP)
  //#define MPC_AUTOTUNE                          // Include a method to do MPC auto-tuning (~5664-5882 bytes of flash)
  //#define MPC_EDIT_MENU                         // Add MPC editing to the "Advanced Settings" menu. (~1300 bytes of flash)
  //#define MPC_AUTOTUNE_MENU                     // Add MPC auto-tuning to the "Advanced Settings" menu. (~350 bytes of flash)

  #define MPC_MAX BANG_MAX                        // (0..255) Current to nozzle while MPC is active.
  #define MPC_HEATER_POWER { 40.0f }              // (W) Heat cartridge powers.

  #define MPC_INCLUDE_FAN                         // Model the fan speed?

  // Measured physical constants from M306
  #define MPC_BLOCK_HEAT_CAPACITY { 16.7f }       // (J/K) Heat block heat capacities.
  #define MPC_SENSOR_RESPONSIVENESS { 0.22f }     // (K/s per ∆K) Rate of change of sensor temperature from heat block.
  #define MPC_AMBIENT_XFER_COEFF { 0.068f }       // (W/K) Heat transfer coefficients from heat block to room air with fan off.
  #if ENABLED(MPC_INCLUDE_FAN)
    #define MPC_AMBIENT_XFER_COEFF_FAN255 { 0.097f } // (W/K) Heat transfer coefficients from heat block to room air with fan on full.
  #endif

  // For one fan and multiple hotends MPC needs to know how to apply the fan cooling effect.
  #if ENABLED(MPC_INCLUDE_FAN)
    //#define MPC_FAN_0_ALL_HOTENDS
    //#define MPC_FAN_0_ACTIVE_HOTEND
  #endif

  #define FILAMENT_HEAT_CAPACITY_PERMM { 5.6e-3f }    // 0.0056 J/K/mm for 1.75mm PLA (0.0149 J/K/mm for 2.85mm PLA).
  //#define FILAMENT_HEAT_CAPACITY_PERMM { 3.6e-3f }  // 0.0036 J/K/mm for 1.75mm PETG (0.0094 J/K/mm for 2.85mm PETG).

  // Advanced options
  #define MPC_SMOOTHING_FACTOR 0.5f               // (0.0...1.0) Noisy temperature sensors may need a lower value for stabilization.
  #define MPC_MIN_AMBIENT_CHANGE 1.0f             // (K/s) Modeled ambient temperature rate of change, when correcting model inaccuracies.
  #define MPC_STEADYSTATE 0.5f                    // (K/s) Temperature change rate for steady state logic to be enforced.

  #define MPC_TUNING_POS { X_CENTER, Y_CENTER, 1.0f } // (mm) M306 Autotuning position, ideally bed center at first layer height.
  #define MPC_TUNING_END_Z 10.0f                  // (mm) M306 Autotuning final Z position.
#endif

//===========================================================================
```

```
 * heater. If your configuration is significantly different than this and you don't understand
 * the issues involved, don't use bed PID until someone else verifies that your hardware works.
 * @section bed temp
 */
#define PIDTEMPBED

//#define BED_LIMIT_SWITCHING

/**
 * Max Bed Power
 * Applies to all forms of bed control (PID, bang-bang, and bang-bang with hysteresis).
 * When set to any value below 255, enables a form of PWM to the bed that acts like a divider
 * so don't use it unless you are OK with PWM on your bed. (See the comment on enabling PIDTEMPBED)
 */
#define MAX_BED_POWER 255 // limits duty cycle to bed; 255=full current

#if ENABLED(PIDTEMPBED)
  //#define MIN_BED_POWER 0
  //#define PID_BED_DEBUG // Print Bed PID debug data to the serial port.

  // 120V 250W silicone heater into 4mm borosilicate (MendelMax 1.5+)
  // from FOPDT model - kp=.39 Tp=405 Tdead=66, Tc set to 79.2, aggressive factor of .15 (vs .1, 1, 10)
  #define DEFAULT_bedKp 10.00
  #define DEFAULT_bedKi .023
  #define DEFAULT_bedKd 305.4

  // FIND YOUR OWN: "M303 E-1 C8 S90" to run autotune on the bed at 90 degreesC for 8 cycles.
#endif // PIDTEMPBED

//===========================================================================
//==================== PID > Chamber Temperature Control ====================
//===========================================================================

/**
 * PID Chamber Heating
 *
 * If this option is enabled set PID constants below.
```

```
 */
#define PREVENT_COLD_EXTRUSION
#define EXTRUDE_MINTEMP 170

/**
 * Prevent a single extrusion longer than EXTRUDE_MAXLENGTH.
 * Note: For Bowden Extruders make this large enough to allow load/unload.
 */
#define PREVENT_LENGTHY_EXTRUDE
#define EXTRUDE_MAXLENGTH 200

//===========================================================================
//======================== Thermal Runaway Protection =======================
//===========================================================================

/**
 * Thermal Protection provides additional protection to your printer from damage
 * and fire. Marlin always includes safe min and max temperature ranges which
 * protect against a broken or disconnected thermistor wire.
 *
 * The issue: If a thermistor falls out, it will report the much lower
 * temperature of the air in the room, and the the firmware will keep
 * the heater on.
 *
 * If you get "Thermal Runaway" or "Heating failed" errors the
 * details can be tuned in Configuration_adv.h
 */

#define THERMAL_PROTECTION_HOTENDS // Enable thermal protection for all extruders
#define THERMAL_PROTECTION_BED     // Enable thermal protection for the heated bed
#define THERMAL_PROTECTION_CHAMBER // Enable thermal protection for the heated chamber
#define THERMAL_PROTECTION_COOLER  // Enable thermal protection for the laser cooling

//===========================================================================
//============================= Mechanical Settings =========================
//===========================================================================
```

```
// Enable one of the options below for CoreXY, CoreXZ, or CoreYZ kinematics,
// either in the usual order or reversed
//#define COREXY
//#define COREXZ
//#define COREYZ
//#define COREYX
//#define COREZX
//#define COREZY
//#define MARKFORGED_XY  // MarkForged. See https://reprap.org/forum/read.php?152,504042
//#define MARKFORGED_YX

// Enable for a belt style printer with endless "Z" motion
//#define BELTPRINTER

// Enable for Polargraph Kinematics
//#define POLARGRAPH
#if ENABLED(POLARGRAPH)
  #define POLARGRAPH_MAX_BELT_LEN  1035.0 // (mm) Belt length at full extension. Override with M665 H
  #define DEFAULT_SEGMENTS_PER_SECOND 5   // Move segmentation based on duration
  #define PEN_UP_DOWN_MENU                // Add "Pen Up" and "Pen Down" to the MarlinUI menu
#endif

// @section delta

// Enable for DELTA kinematics and configure below
//#define DELTA
#if ENABLED(DELTA)

  // Make delta curves from many straight lines (linear interpolation).
  // This is a trade-off between visible corners (not enough segments)
  // and processor overload (too many expensive sqrt calls).
  #define DEFAULT_SEGMENTS_PER_SECOND 200

  // After homing move down to a height where XY movement is unconstrained
  //#define DELTA_HOME_TO_SAFE_ZONE
```

```
 */
//#define MORGAN_SCARA
//#define MP_SCARA
#if EITHER(MORGAN_SCARA, MP_SCARA)
  // If movement is choppy try lowering this value
  #define DEFAULT_SEGMENTS_PER_SECOND 200

  // Length of inner and outer support arms. Measure arm lengths precisely.
  #define SCARA_LINKAGE_1 150       // (mm)
  #define SCARA_LINKAGE_2 150       // (mm)

  // SCARA tower offset (position of Tower relative to bed zero position)
  // This needs to be reasonably accurate as it defines the printbed position in the SCARA space.
  #define SCARA_OFFSET_X  100       // (mm)
  #define SCARA_OFFSET_Y  -56       // (mm)

  #if ENABLED(MORGAN_SCARA)

    //#define DEBUG_SCARA_KINEMATICS
    #define FEEDRATE_SCALING        // Convert XY feedrate from mm/s to degrees/s on the fly

    // Radius around the center where the arm cannot reach
    #define MIDDLE_DEAD_ZONE_R   0  // (mm)

    #define THETA_HOMING_OFFSET  0  // Calculated from Calibration Guide and M360 / M114. See http://reprap.harleystudio.co.za/?page_id=1073
    #define PSI_HOMING_OFFSET    0  // Calculated from Calibration Guide and M364 / M114. See http://reprap.harleystudio.co.za/?page_id=1073

  #elif ENABLED(MP_SCARA)

    #define SCARA_OFFSET_THETA1  12 // degrees
    #define SCARA_OFFSET_THETA2 131 // degrees

  #endif

#endif
```

```
//===========================================================================
//============================== Endstop Settings ===========================
//===========================================================================

// @section endstops

// Specify here all the endstop connectors that are connected to any endstop or probe.
// Almost all printers will be using one per axis. Probes will use one or more of the
// extra connectors. Leave undefined any used for non-endstop and non-probe purposes.
#define USE_XMIN_PLUG
#define USE_YMIN_PLUG
#define USE_ZMIN_PLUG
//#define USE_IMIN_PLUG
//#define USE_JMIN_PLUG
//#define USE_KMIN_PLUG
//#define USE_UMIN_PLUG
//#define USE_VMIN_PLUG
//#define USE_WMIN_PLUG
//#define USE_XMAX_PLUG
//#define USE_YMAX_PLUG
//#define USE_ZMAX_PLUG
//#define USE_IMAX_PLUG
//#define USE_JMAX_PLUG
//#define USE_KMAX_PLUG
//#define USE_UMAX_PLUG
//#define USE_VMAX_PLUG
//#define USE_WMAX_PLUG

// Enable pullup for all endstops to prevent a floating state
#define ENDSTOPPULLUPS
#if DISABLED(ENDSTOPPULLUPS)
  // Disable ENDSTOPPULLUPS to set pullups individually
  //#define ENDSTOPPULLUP_XMIN
  //#define ENDSTOPPULLUP_YMIN
  //#define ENDSTOPPULLUP_ZMIN
  //#define ENDSTOPPULLUP_IMIN
```

```
Marlin   Configuration.h   Configuration_adv.h   Version.h
#endif

// Enable pulldown for all endstops to prevent a floating state
//#define ENDSTOPPULLDOWNS
#if DISABLED(ENDSTOPPULLDOWNS)
  // Disable ENDSTOPPULLDOWNS to set pulldowns individually
  //#define ENDSTOPPULLDOWN_XMIN
  //#define ENDSTOPPULLDOWN_YMIN
  //#define ENDSTOPPULLDOWN_ZMIN
  //#define ENDSTOPPULLDOWN_IMIN
  //#define ENDSTOPPULLDOWN_JMIN
  //#define ENDSTOPPULLDOWN_KMIN
  //#define ENDSTOPPULLDOWN_UMIN
  //#define ENDSTOPPULLDOWN_VMIN
  //#define ENDSTOPPULLDOWN_WMIN
  //#define ENDSTOPPULLDOWN_XMAX
  //#define ENDSTOPPULLDOWN_YMAX
  //#define ENDSTOPPULLDOWN_ZMAX
  //#define ENDSTOPPULLDOWN_IMAX
  //#define ENDSTOPPULLDOWN_JMAX
  //#define ENDSTOPPULLDOWN_KMAX
  //#define ENDSTOPPULLDOWN_UMAX
  //#define ENDSTOPPULLDOWN_VMAX
  //#define ENDSTOPPULLDOWN_WMAX
  //#define ENDSTOPPULLDOWN_ZMIN_PROBE
#endif

/**
 * Endstop "Hit" State
 * Set to the state (HIGH or LOW) that applies to each endstop.
 */
#define X_MIN_ENDSTOP_HIT_STATE LOW
#define X_MAX_ENDSTOP_HIT_STATE LOW
#define Y_MIN_ENDSTOP_HIT_STATE LOW
#define Y_MAX_ENDSTOP_HIT_STATE LOW
#define Z_MIN_ENDSTOP_HIT_STATE LOW
#define Z_MAX_ENDSTOP_HIT_STATE LOW
```

```
//===========================================================================
//============================= Movement Settings ============================
//===========================================================================
// @section motion

/**
 * Default Settings
 *
 * These settings can be reset by M502
 *
 * Note that if EEPROM is enabled, saved values will override these.
 */

/**
 * With this option each E stepper can have its own factors for the
 * following movement settings. If fewer factors are given than the
 * total number of extruders, the last value applies to the rest.
 */
#define DISTINCT_E_FACTORS

/**
 * Default Axis Steps Per Unit (linear=steps/mm, rotational=steps/°)
 * Override with M92
 *                                      X, Y, Z [, I [, J [, K...]]], E0 [, E1[, E2...]]
 */

#define DEFAULT_AXIS_STEPS_PER_UNIT {80,80,400,170,9.45}

/**
 * Default Max Feed Rate (linear=mm/s, rotational=°/s)
 * Override with M203
 *                                      X, Y, Z [, I [, J [, K...]]], E0 [, E1[, E2...]]
 */
#define DEFAULT_MAX_FEEDRATE          { 400, 400, 5, 25 }

//#define LIMITED_MAX_FR_EDITING        // Limit edit via M203 or LCD to DEFAULT_MAX_FEEDRATE * 2
#if ENABLED(LIMITED_MAX_FR_EDITING)
```

```
#if ENABLED(LIMITED_MAX_FR_EDITING)
  #define MAX_FEEDRATE_EDIT_VALUES    { 600, 600, 10, 50 } // ...or, set your own edit limits
#endif

/**
 * Default Max Acceleration (speed change with time) (linear=mm/(s^2), rotational=°/(s^2))
 * (Maximum start speed for accelerated moves)
 * Override with M201
 *                                      X, Y, Z [, I [, J [, K...]]], E0 [, E1[, E2...]]
 */
#define DEFAULT_MAX_ACCELERATION      { 3000, 3000, 100, 10000 }

//#define LIMITED_MAX_ACCEL_EDITING     // Limit edit via M201 or LCD to DEFAULT_MAX_ACCELERATION * 2
#if ENABLED(LIMITED_MAX_ACCEL_EDITING)
  #define MAX_ACCEL_EDIT_VALUES       { 9000, 9000, 100, 20000 } // ...or, set your own edit limits
#endif

/**
 * Default Acceleration (speed change with time) (linear=mm/(s^2), rotational=°/(s^2))
 * Override with M204
 *
 *   M204 P    Acceleration
 *   M204 R    Retract Acceleration
 *   M204 T    Travel Acceleration
 */
#define DEFAULT_ACCELERATION          3000    // X, Y, Z and E acceleration for printing moves
#define DEFAULT_RETRACT_ACCELERATION  3000    // E acceleration for retracts
#define DEFAULT_TRAVEL_ACCELERATION   3000    // X, Y, Z acceleration for travel (non printing) moves

/**
 * Default Jerk limits (mm/s)
 * Override with M205 X Y Z . . . E
 *
 * "Jerk" specifies the minimum speed change that requires acceleration.
 * When changing speed and direction, if the difference is less than the
 * value set here, it may happen instantaneously.
 */
```

```
 */
#define CLASSIC_JERK
#if ENABLED(CLASSIC_JERK)
  #define DEFAULT_XJERK 20.0
  #define DEFAULT_YJERK 20.0
  #define DEFAULT_ZJERK  0.3
  //#define DEFAULT_IJERK  0.3
  //#define DEFAULT_JJERK  0.3
  //#define DEFAULT_KJERK  0.3
  //#define DEFAULT_UJERK  0.3
  //#define DEFAULT_VJERK  0.3
  //#define DEFAULT_WJERK  0.3

  //#define TRAVEL_EXTRA_XYJERK 0.0     // Additional jerk allowance for all travel moves

  //#define LIMITED_JERK_EDITING        // Limit edit via M205 or LCD to DEFAULT_aJERK * 2
  #if ENABLED(LIMITED_JERK_EDITING)
    #define MAX_JERK_EDIT_VALUES { 20, 20, 0.6, 10 } // ...or, set your own edit limits
  #endif
#endif

#define DEFAULT_EJERK    5.0  // May be used by Linear Advance

/**
 * Junction Deviation Factor
 *
 * See:
 *   https://reprap.org/forum/read.php?1,739819
 *   https://blog.kyneticcnc.com/2018/10/computing-junction-deviation-for-marlin.html
 */
#if DISABLED(CLASSIC_JERK)
  #define JUNCTION_DEVIATION_MM 0.013 // (mm) Distance from real junction edge
  #define JD_HANDLE_SMALL_SEGMENTS    // Use curvature estimation instead of just the junction angle
                                      // for small segments (< 1mm) with large junction angles (> 135°).
#endif
```

```
// A sled-mounted probe like those designed by Charles Bell.
//#define Z_PROBE_SLED
//#define SLED_DOCKING_OFFSET 5  // The extra distance the X axis must travel to pickup the sled. 0 should be fine but you can push it further if you'd like

// A probe deployed by moving the x-axis, such as the Wilson II's rack-and-pinion probe designed by Marty Rice.
//#define RACK_AND_PINION_PROBE
#if ENABLED(RACK_AND_PINION_PROBE)
  #define Z_PROBE_DEPLOY_X  X_MIN_POS
  #define Z_PROBE_RETRACT_X X_MAX_POS
#endif

/**
 * Magnetically Mounted Probe
 * For probes such as Euclid, Klicky, Klackender, etc.
 */
//#define MAG_MOUNTED_PROBE
#if ENABLED(MAG_MOUNTED_PROBE)
  #define PROBE_DEPLOY_FEEDRATE (133*60)  // (mm/min) Probe deploy speed
  #define PROBE_STOW_FEEDRATE   (133*60)  // (mm/min) Probe stow speed

  #define MAG_MOUNTED_DEPLOY_1 { PROBE_DEPLOY_FEEDRATE, { 245, 114, 30 } }  // Move to side Dock & Attach probe
  #define MAG_MOUNTED_DEPLOY_2 { PROBE_DEPLOY_FEEDRATE, { 210, 114, 30 } }  // Move probe off dock
  #define MAG_MOUNTED_DEPLOY_3 { PROBE_DEPLOY_FEEDRATE, {   0,   0,  0 } }  // Extra move if needed
  #define MAG_MOUNTED_DEPLOY_4 { PROBE_DEPLOY_FEEDRATE, {   0,   0,  0 } }  // Extra move if needed
  #define MAG_MOUNTED_DEPLOY_5 { PROBE_DEPLOY_FEEDRATE, {   0,   0,  0 } }  // Extra move if needed
  #define MAG_MOUNTED_STOW_1   { PROBE_STOW_FEEDRATE,   { 245, 114, 20 } }  // Move to dock
  #define MAG_MOUNTED_STOW_2   { PROBE_STOW_FEEDRATE,   { 245, 114,  0 } }  // Place probe beside remover
  #define MAG_MOUNTED_STOW_3   { PROBE_STOW_FEEDRATE,   { 230, 114,  0 } }  // Side move to remove probe
  #define MAG_MOUNTED_STOW_4   { PROBE_STOW_FEEDRATE,   { 210, 114, 20 } }  // Side move to remove probe
  #define MAG_MOUNTED_STOW_5   { PROBE_STOW_FEEDRATE,   {   0,   0,  0 } }  // Extra move if needed
#endif

// Duet Smart Effector (for delta printers) - https://bit.ly/2ul5U7J
// When the pin is defined you can use M672 to set/reset the probe sensitivity.
//#define DUET_SMART_EFFECTOR
#if ENABLED(DUET_SMART_EFFECTOR)
  #define SMART_EFFECTOR_MOD_PIN  -1  // Connect a GPIO pin to the Smart Effector MOD pin
```

```
/**
 * Allen key retractable z-probe as seen on many Kossel delta printers - https://reprap.org/wiki/Kossel#Automatic_bed_leveling_probe
 * Deploys by touching z-axis belt. Retracts by pushing the probe down.
 */
//#define Z_PROBE_ALLEN_KEY
#if ENABLED(Z_PROBE_ALLEN_KEY)
  // 2 or 3 sets of coordinates for deploying and retracting the spring loaded touch probe on G29,
  // if servo actuated touch probe is not defined. Uncomment as appropriate for your printer/probe.

  #define Z_PROBE_ALLEN_KEY_DEPLOY_1 { 30.0, PRINTABLE_RADIUS, 100.0 }
  #define Z_PROBE_ALLEN_KEY_DEPLOY_1_FEEDRATE XY_PROBE_FEEDRATE

  #define Z_PROBE_ALLEN_KEY_DEPLOY_2 { 0.0, PRINTABLE_RADIUS, 100.0 }
  #define Z_PROBE_ALLEN_KEY_DEPLOY_2_FEEDRATE (XY_PROBE_FEEDRATE)/10

  #define Z_PROBE_ALLEN_KEY_DEPLOY_3 { 0.0, (PRINTABLE_RADIUS) * 0.75, 100.0 }
  #define Z_PROBE_ALLEN_KEY_DEPLOY_3_FEEDRATE XY_PROBE_FEEDRATE

  #define Z_PROBE_ALLEN_KEY_STOW_1 { -64.0, 56.0, 23.0 } // Move the probe into position
  #define Z_PROBE_ALLEN_KEY_STOW_1_FEEDRATE XY_PROBE_FEEDRATE

  #define Z_PROBE_ALLEN_KEY_STOW_2 { -64.0, 56.0, 3.0 } // Push it down
  #define Z_PROBE_ALLEN_KEY_STOW_2_FEEDRATE (XY_PROBE_FEEDRATE)/10

  #define Z_PROBE_ALLEN_KEY_STOW_3 { -64.0, 56.0, 50.0 } // Move it up to clear
  #define Z_PROBE_ALLEN_KEY_STOW_3_FEEDRATE XY_PROBE_FEEDRATE

  #define Z_PROBE_ALLEN_KEY_STOW_4 { 0.0, 0.0, 50.0 }
  #define Z_PROBE_ALLEN_KEY_STOW_4_FEEDRATE XY_PROBE_FEEDRATE

#endif // Z_PROBE_ALLEN_KEY

/**
 * Nozzle-to-Probe offsets { X, Y, Z }
 *
 * X and Y offset
 *   Use a caliper or ruler to measure the distance from the tip of
```

```
 */
#define NOZZLE_TO_PROBE_OFFSET { 10, 10, 0 }

// Enable and set to use a specific tool for probing. Disable to allow any tool.
#define PROBING_TOOL 0
#ifdef PROBING_TOOL
  //#define PROBE_TOOLCHANGE_NO_MOVE  // Suppress motion on probe tool-change
#endif

// Most probes should stay away from the edges of the bed, but
// with NOZZLE_AS_PROBE this can be negative for a wider probing area.
#define PROBING_MARGIN 10

// X and Y axis travel speed (mm/min) between probes
#define XY_PROBE_FEEDRATE (133*60)

// Feedrate (mm/min) for the first approach when double-probing (MULTIPLE_PROBING == 2)
#define Z_PROBE_FEEDRATE_FAST (4*60)

// Feedrate (mm/min) for the "accurate" probe of each point
#define Z_PROBE_FEEDRATE_SLOW (Z_PROBE_FEEDRATE_FAST / 2)

/**
 * Probe Activation Switch
 * A switch indicating proper deployment, or an optical
 * switch triggered when the carriage is near the bed.
 */
//#define PROBE_ACTIVATION_SWITCH
#if ENABLED(PROBE_ACTIVATION_SWITCH)
  #define PROBE_ACTIVATION_SWITCH_STATE LOW // State indicating probe is active
  //#define PROBE_ACTIVATION_SWITCH_PIN PC6 // Override default pin
#endif

/**
 * Tare Probe (determine zero-point) prior to each probe.
 * Useful for a strain gauge or piezo sensor that needs to factor out
 * elements such as cables pulling on the carriage.
```

```
// Require minimum nozzle and/or bed temperature for probing
//#define PREHEAT_BEFORE_PROBING
#if ENABLED(PREHEAT_BEFORE_PROBING)
  #define PROBING_NOZZLE_TEMP 120   // (°C) Only applies to E0 at this time
  #define PROBING_BED_TEMP     50
#endif

// For Inverting Stepper Enable Pins (Active Low) use 0, Non Inverting (Active High) use 1
// :{ 0:'Low', 1:'High' }
#define X_ENABLE_ON 0
#define Y_ENABLE_ON 0
#define Z_ENABLE_ON 0
#define E_ENABLE_ON 0 // For all extruders
//#define I_ENABLE_ON 0
//#define J_ENABLE_ON 0
//#define K_ENABLE_ON 0
//#define U_ENABLE_ON 0
//#define V_ENABLE_ON 0
//#define W_ENABLE_ON 0

// Disable axis steppers immediately when they're not being stepped.
// WARNING: When motors turn off there is a chance of losing position accuracy!
//#define DISABLE_X
//#define DISABLE_Y
//#define DISABLE_Z
//#define DISABLE_I
//#define DISABLE_J
//#define DISABLE_K
//#define DISABLE_U
//#define DISABLE_V
//#define DISABLE_W

// Turn off the display blinking that warns about possible accuracy reduction
//#define DISABLE_REDUCED_ACCURACY_WARNING

// @section extruder
```

```
//#define DISABLE_E               // Disable the extruder when not stepping
#define DISABLE_OTHER_EXTRUDERS   // Keep only the active extruder enabled

// @section motion

// Invert the stepper direction. Change (or reverse the motor connector) if an axis goes the wrong way.
#define INVERT_X_DIR true
#define INVERT_Y_DIR true
#define INVERT_Z_DIR false
//#define INVERT_I_DIR false
//#define INVERT_J_DIR false
//#define INVERT_K_DIR false
//#define INVERT_U_DIR false
//#define INVERT_V_DIR false
//#define INVERT_W_DIR false

// @section extruder

// For direct drive extruder v9 set to true, for geared extruder set to false.
#define INVERT_E0_DIR false
#define INVERT_E1_DIR false
#define INVERT_E2_DIR false
#define INVERT_E3_DIR false
#define INVERT_E4_DIR false
#define INVERT_E5_DIR false
#define INVERT_E6_DIR false
#define INVERT_E7_DIR false

// @section homing

//#define NO_MOTION_BEFORE_HOMING // Inhibit movement until all axes have been homed. Also enable HOME_AFTER_DEACTIVATE for e
//#define HOME_AFTER_DEACTIVATE   // Require rehoming after steppers are deactivated. Also enable NO_MOTION_BEFORE_HOMING for

/**
 * Set Z_IDLE_HEIGHT if the Z-Axis moves on its own when steppers are disabled.
 *  - Use a low value (i.e., Z_MIN_POS) if the nozzle falls down to the bed.
```

```
// Direction of endstops when homing; 1=MAX, -1=MIN
// :[-1,1]
#define X_HOME_DIR -1
#define Y_HOME_DIR -1
#define Z_HOME_DIR -1
//#define I_HOME_DIR -1
//#define J_HOME_DIR -1
//#define K_HOME_DIR -1
//#define U_HOME_DIR -1
//#define V_HOME_DIR -1
//#define W_HOME_DIR -1

// @section geometry

// The size of the printable area
#define X_BED_SIZE 200
#define Y_BED_SIZE 200

// Travel limits (linear=mm, rotational=°) after homing, corresponding to endstop positions.
#define X_MIN_POS 0
#define Y_MIN_POS 0
#define Z_MIN_POS 0
#define X_MAX_POS X_BED_SIZE
#define Y_MAX_POS Y_BED_SIZE
#define Z_MAX_POS 200
//#define I_MIN_POS 0
//#define I_MAX_POS 50
//#define J_MIN_POS 0
//#define J_MAX_POS 50
//#define K_MIN_POS 0
//#define K_MAX_POS 50
//#define U_MIN_POS 0
//#define U_MAX_POS 50
//#define V_MIN_POS 0
//#define V_MAX_POS 50
//#define W_MIN_POS 0
```

```
// Min software endstops constrain movement within minimum coordinate bounds
#define MIN_SOFTWARE_ENDSTOPS
#if ENABLED(MIN_SOFTWARE_ENDSTOPS)
  #define MIN_SOFTWARE_ENDSTOP_X
  #define MIN_SOFTWARE_ENDSTOP_Y
  #define MIN_SOFTWARE_ENDSTOP_Z
  #define MIN_SOFTWARE_ENDSTOP_I
  #define MIN_SOFTWARE_ENDSTOP_J
  #define MIN_SOFTWARE_ENDSTOP_K
  #define MIN_SOFTWARE_ENDSTOP_U
  #define MIN_SOFTWARE_ENDSTOP_V
  #define MIN_SOFTWARE_ENDSTOP_W
#endif

// Max software endstops constrain movement within maximum coordinate bounds
#define MAX_SOFTWARE_ENDSTOPS
#if ENABLED(MAX_SOFTWARE_ENDSTOPS)
  #define MAX_SOFTWARE_ENDSTOP_X
  #define MAX_SOFTWARE_ENDSTOP_Y
  #define MAX_SOFTWARE_ENDSTOP_Z
  #define MAX_SOFTWARE_ENDSTOP_I
  #define MAX_SOFTWARE_ENDSTOP_J
  #define MAX_SOFTWARE_ENDSTOP_K
  #define MAX_SOFTWARE_ENDSTOP_U
  #define MAX_SOFTWARE_ENDSTOP_V
  #define MAX_SOFTWARE_ENDSTOP_W
#endif

#if EITHER(MIN_SOFTWARE_ENDSTOPS, MAX_SOFTWARE_ENDSTOPS)
  //#define SOFT_ENDSTOPS_MENU_ITEM  // Enable/Disable software endstops from the LCD
#endif

/**
 * Filament Runout Sensors
```

```
   * contours of the bed more closely than edge-to-edge straight moves.
   */
  #define SEGMENT_LEVELED_MOVES
  #define LEVELED_SEGMENT_LENGTH 5.0 // (mm) Length of all segments (except the last one)

  /**
   * Enable the G26 Mesh Validation Pattern tool.
   */
  //#define G26_MESH_VALIDATION
  #if ENABLED(G26_MESH_VALIDATION)
    #define MESH_TEST_NOZZLE_SIZE    0.4  // (mm) Diameter of primary nozzle.
    #define MESH_TEST_LAYER_HEIGHT   0.2  // (mm) Default layer height for G26.
    #define MESH_TEST_HOTEND_TEMP  205    // (°C) Default nozzle temperature for G26.
    #define MESH_TEST_BED_TEMP      60    // (°C) Default bed temperature for G26.
    #define G26_XY_FEEDRATE         20    // (mm/s) Feedrate for G26 XY moves.
    #define G26_XY_FEEDRATE_TRAVEL 100    // (mm/s) Feedrate for G26 XY travel moves.
    #define G26_RETRACT_MULTIPLIER   1.0  // G26 Q (retraction) used by default between mesh test elements.
  #endif

#endif

#if EITHER(AUTO_BED_LEVELING_LINEAR, AUTO_BED_LEVELING_BILINEAR)

  // Set the number of grid points per dimension.
  #define GRID_MAX_POINTS_X 3
  #define GRID_MAX_POINTS_Y GRID_MAX_POINTS_X

  // Probe along the Y axis, advancing X after each column
  //#define PROBE_Y_FIRST

  #if ENABLED(AUTO_BED_LEVELING_BILINEAR)

    // Beyond the probed grid, continue the implied tilt?
    // Default is to maintain the height of the nearest edge.
    //#define EXTRAPOLATE_BEYOND_GRID

    //



// Manually set the home position. Leave these undefined for automatic settings.
// For DELTA this is the top-center of the Cartesian print volume.
//#define MANUAL_X_HOME_POS 0
//#define MANUAL_Y_HOME_POS 0
//#define MANUAL_Z_HOME_POS 0
//#define MANUAL_I_HOME_POS 0
//#define MANUAL_J_HOME_POS 0
//#define MANUAL_K_HOME_POS 0
//#define MANUAL_U_HOME_POS 0
//#define MANUAL_V_HOME_POS 0
//#define MANUAL_W_HOME_POS 0

/**
 * Use "Z Safe Homing" to avoid homing with a Z probe outside the bed area.
 *
 * - Moves the Z probe (or nozzle) to a defined XY point before Z homing.
 * - Allows Z homing only when XY positions are known and trusted.
 * - If stepper drivers sleep, XY homing may be required again before Z homing.
 */
//#define Z_SAFE_HOMING

#if ENABLED(Z_SAFE_HOMING)
  #define Z_SAFE_HOMING_X_POINT X_CENTER  // X point for Z homing
  #define Z_SAFE_HOMING_Y_POINT Y_CENTER  // Y point for Z homing
  //#define Z_SAFE_HOMING_POINT_ABSOLUTE  // Ignore home offsets (M206) for Z homing position
#endif

// Homing speeds (linear=mm/min, rotational=°/min)
#define HOMING_FEEDRATE_MM_M { (50*60), (50*60), (4*60) }

// Validate that endstops are triggered on homing moves
#define VALIDATE_HOMING_ENDSTOPS

// @section calibrate
```

```
 *    |  A-------D         |  A-------D         |  A-------D
 *    +-------------->X    +-------------->X    +-------------->Y
 *     XY_SKEW_FACTOR       XZ_SKEW_FACTOR       YZ_SKEW_FACTOR
 */
//#define SKEW_CORRECTION

#if ENABLED(SKEW_CORRECTION)
  // Input all length measurements here:
  #define XY_DIAG_AC 282.8427124746
  #define XY_DIAG_BD 282.8427124746
  #define XY_SIDE_AD 200

  // Or, set the XY skew factor directly:
  //#define XY_SKEW_FACTOR 0.0

  //#define SKEW_CORRECTION_FOR_Z
  #if ENABLED(SKEW_CORRECTION_FOR_Z)
    #define XZ_DIAG_AC 282.8427124746
    #define XZ_DIAG_BD 282.8427124746
    #define YZ_DIAG_AC 282.8427124746
    #define YZ_DIAG_BD 282.8427124746
    #define YZ_SIDE_AD 200

    // Or, set the Z skew factors directly:
    //#define XZ_SKEW_FACTOR 0.0
    //#define YZ_SKEW_FACTOR 0.0
  #endif

  // Enable this option for M852 to set skew at runtime
  //#define SKEW_CORRECTION_GCODE
#endif

//=============================================================================
//============================= Additional Features ===========================
//=============================================================================

// @section eeprom
```

```
//
#define PREHEAT_1_LABEL       "PLA"
#define PREHEAT_1_TEMP_HOTEND 240
#define PREHEAT_1_TEMP_BED     60
#define PREHEAT_1_TEMP_CHAMBER 35
#define PREHEAT_1_FAN_SPEED     0 // Value from 0 to 255

#define PREHEAT_2_LABEL       "ABS"
#define PREHEAT_2_TEMP_HOTEND 240
#define PREHEAT_2_TEMP_BED    110
#define PREHEAT_2_TEMP_CHAMBER 35
#define PREHEAT_2_FAN_SPEED     0 // Value from 0 to 255

// @section motion

/**
 * Nozzle Park
 *
 * Park the nozzle at the given XYZ position on idle or G27.
 *
 * The "P" parameter controls the action applied to the Z axis:
 *
 *    P0  (Default) If Z is below park Z raise the nozzle.
 *    P1  Raise the nozzle always to Z-park height.
 *    P2  Raise the nozzle by Z-park amount, limited to Z_MAX_POS.
 */
//#define NOZZLE_PARK_FEATURE

#if ENABLED(NOZZLE_PARK_FEATURE)
  // Specify a park position as { X, Y, Z_raise }
  #define NOZZLE_PARK_POINT { (X_MIN_POS + 10), (Y_MAX_POS - 10), 20 }
  #define NOZZLE_PARK_MOVE          0   // Park motion: 0 = XY Move, 1 = X Only, 2 = Y Only, 3 = X before Y, 4 = Y before X
  #define NOZZLE_PARK_Z_RAISE_MIN   2   // (mm) Always raise Z by at least this distance
  #define NOZZLE_PARK_XY_FEEDRATE 100   // (mm/s) X and Y axes feedrate (also used for delta Z axis)
  #define NOZZLE_PARK_Z_FEEDRATE    5   // (mm/s) Z axis feedrate (not used for delta printers)
#endif
```

```
 *   Caveats: The ending Z should be the same as starting Z.
 */
//#define NOZZLE_CLEAN_FEATURE

#if ENABLED(NOZZLE_CLEAN_FEATURE)
  // Default number of pattern repetitions
  #define NOZZLE_CLEAN_STROKES  12

  // Default number of triangles
  #define NOZZLE_CLEAN_TRIANGLES  3

  // Specify positions for each tool as { { X, Y, Z }, { X, Y, Z } }
  // Dual hotend system may use { {  -20, (Y_BED_SIZE / 2), (Z_MIN_POS + 1) },  {  420, (Y_BED_SIZE / 2), (Z_MIN_POS + 1) }}
  #define NOZZLE_CLEAN_START_POINT { {  30, 30, (Z_MIN_POS + 1) } }
  #define NOZZLE_CLEAN_END_POINT   { { 100, 60, (Z_MIN_POS + 1) } }

  // Circular pattern radius
  #define NOZZLE_CLEAN_CIRCLE_RADIUS 6.5
  // Circular pattern circle fragments number
  #define NOZZLE_CLEAN_CIRCLE_FN 10
  // Middle point of circle
  #define NOZZLE_CLEAN_CIRCLE_MIDDLE NOZZLE_CLEAN_START_POINT

  // Move the nozzle to the initial position after cleaning
  #define NOZZLE_CLEAN_GOBACK

  // For a purge/clean station that's always at the gantry height (thus no Z move)
  //#define NOZZLE_CLEAN_NO_Z

  // For a purge/clean station mounted on the X axis
  //#define NOZZLE_CLEAN_NO_Y

  // Require a minimum hotend temperature for cleaning
  #define NOZZLE_CLEAN_MIN_TEMP 170
  //#define NOZZLE_CLEAN_HEATUP       // Heat up the nozzle instead of skipping wipe

  // Explicit wipe G-code script applies to a G12 with no arguments.
```

```
//=============================================================================
//======================== LCD / Controller Selection =========================
//========================   (Character-based LCDs)   =========================
//=============================================================================
// @section lcd

//
// RepRapDiscount Smart Controller.
// https://reprap.org/wiki/RepRapDiscount_Smart_Controller
//
// Note: Usually sold with a white PCB.
//
#define REPRAP_DISCOUNT_SMART_CONTROLLER

//
// GT2560 (YHCB2004) LCD Display
//
// Requires Testato, Koepel softwarewire library and
// Andriy Golovnya's LiquidCrystal_AIP31068 library.
//
//#define YHCB2004

//
// Original RADDS LCD Display+Encoder+SDCardReader
// http://doku.radds.org/dokumentation/lcd-display/
//
//#define RADDS_DISPLAY

//
// ULTIMAKER Controller.
//
//#define ULTIMAKERCONTROLLER

//
// ULTIPANEL as seen on Thingiverse.
```

```
// Set number of user-controlled fans. Disable to use all board-defined fans.
// :[1,2,3,4,5,6,7,8]
//#define NUM_M106_FANS 1

// Use software PWM to drive the fan, as for the heaters. This uses a very low frequency
// which is not as annoying as with the hardware PWM. On the other hand, if this frequency
// is too low, you should also increment SOFT_PWM_SCALE.
//#define FAN_SOFT_PWM

// Incrementing this by 1 will double the software PWM frequency,
// affecting heaters, and the fan if FAN_SOFT_PWM is enabled.
// However, control resolution will be halved for each increment;
// at zero value, there are 128 effective control positions.
// :[0,1,2,3,4,5,6,7]
#define SOFT_PWM_SCALE 0

// If SOFT_PWM_SCALE is set to a value higher than 0, dithering can
// be used to mitigate the associated resolution loss. If enabled,
// some of the PWM cycles are stretched so on average the desired
// duty cycle is attained.
//#define SOFT_PWM_DITHER

// @section extras

// Support for the BariCUDA Paste Extruder
//#define BARICUDA

// @section lights

// Temperature status LEDs that display the hotend and bed temperature.
// If all hotends, bed temperature, and target temperature are under 54C
// then the BLUE led is on. Otherwise the RED led is on. (1C hysteresis)
//#define TEMP_STAT_LEDS

// Support for BlinkM/CyzRgb
//#define BLINKM
```

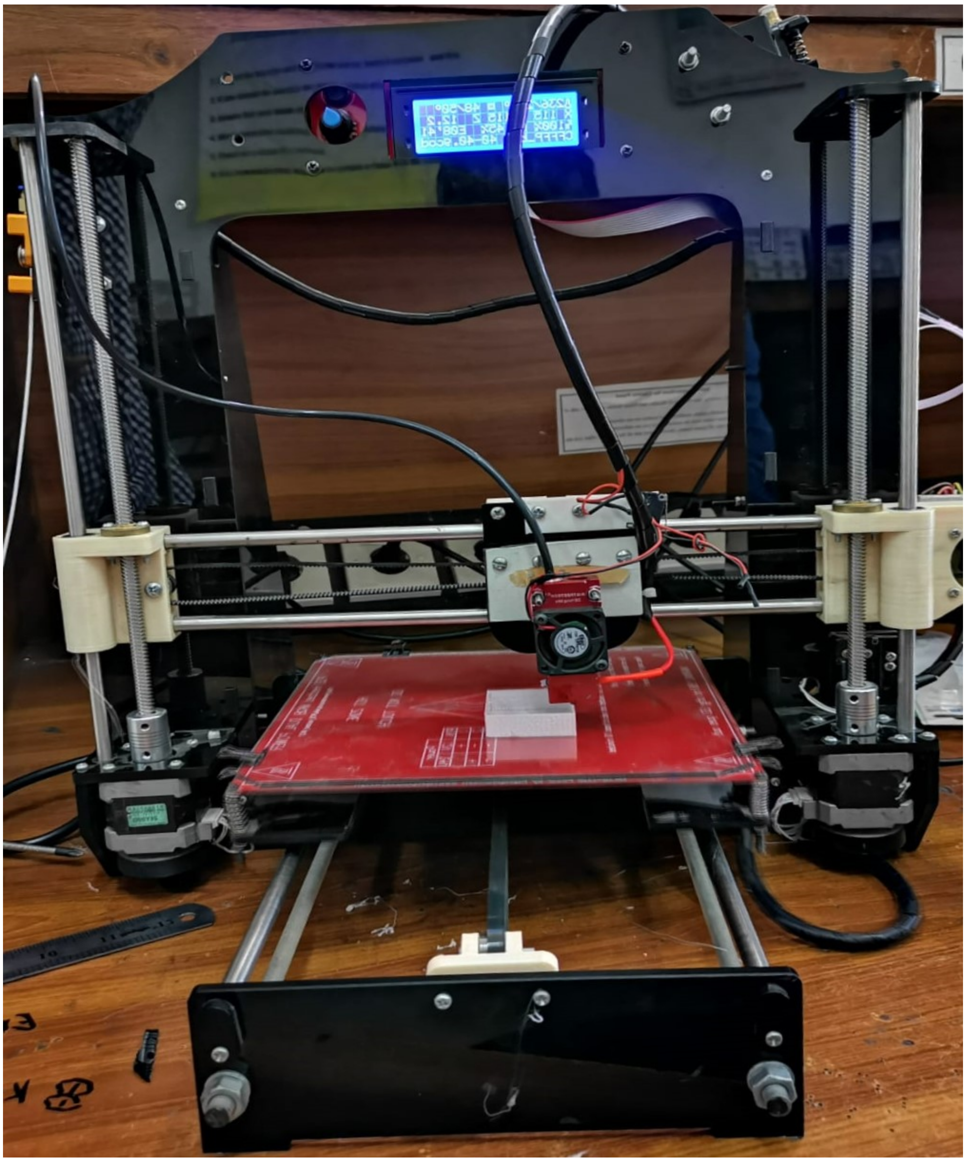
```
// Support for Adafruit NeoPixel LED driver
//#define NEOPIXEL_LED
#if ENABLED(NEOPIXEL_LED)
  #define NEOPIXEL_TYPE          NEO_GRBW // NEO_GRBW, NEO_RGBW, NEO_GRB, NEO_RBG, etc.
                                          // See https://github.com/adafruit/Adafruit_NeoPixel/blob/master/Adafruit_NeoPixel.h
  //#define NEOPIXEL_PIN                4 // LED driving pin
  //#define NEOPIXEL2_TYPE  NEOPIXEL_TYPE
  //#define NEOPIXEL2_PIN               5
  #define NEOPIXEL_PIXELS              30 // Number of LEDs in the strip. (Longest strip when NEOPIXEL2_SEPARATE is disabled.)
  #define NEOPIXEL_IS_SEQUENTIAL          // Sequential display for temperature change - LED by LED. Disable to change all LEDs at once.
  #define NEOPIXEL_BRIGHTNESS         127 // Initial brightness (0-255)
  //#define NEOPIXEL_STARTUP_TEST         // Cycle through colors at startup

  // Support for second Adafruit NeoPixel LED driver controlled with M150 S1 ...
  //#define NEOPIXEL2_SEPARATE
  #if ENABLED(NEOPIXEL2_SEPARATE)
    #define NEOPIXEL2_PIXELS           15 // Number of LEDs in the second strip
    #define NEOPIXEL2_BRIGHTNESS      127 // Initial brightness (0-255)
    #define NEOPIXEL2_STARTUP_TEST        // Cycle through colors at startup
    #define NEOPIXEL_M150_DEFAULT      -1 // Default strip for M150 without 'S'. Use -1 to set all by default.
  #else
    //#define NEOPIXEL2_INSERIES          // Default behavior is NeoPixel 2 in parallel
  #endif

  // Use some of the NeoPixel LEDs for static (background) lighting
  //#define NEOPIXEL_BKGD_INDEX_FIRST   0 // Index of the first background LED
  //#define NEOPIXEL_BKGD_INDEX_LAST    5 // Index of the last background LED
  //#define NEOPIXEL_BKGD_COLOR         { 255, 255, 255, 0 }  // R, G, B, W
  //#define NEOPIXEL_BKGD_TIMEOUT_COLOR {  25,  25,  25, 0 }  // R, G, B, W
  //#define NEOPIXEL_BKGD_ALWAYS_ON       // Keep the backlight on when other NeoPixels are off
#endif
```

Chapter 9

Result and Discussion



9.1 40x40x40 mm Box without Tuning

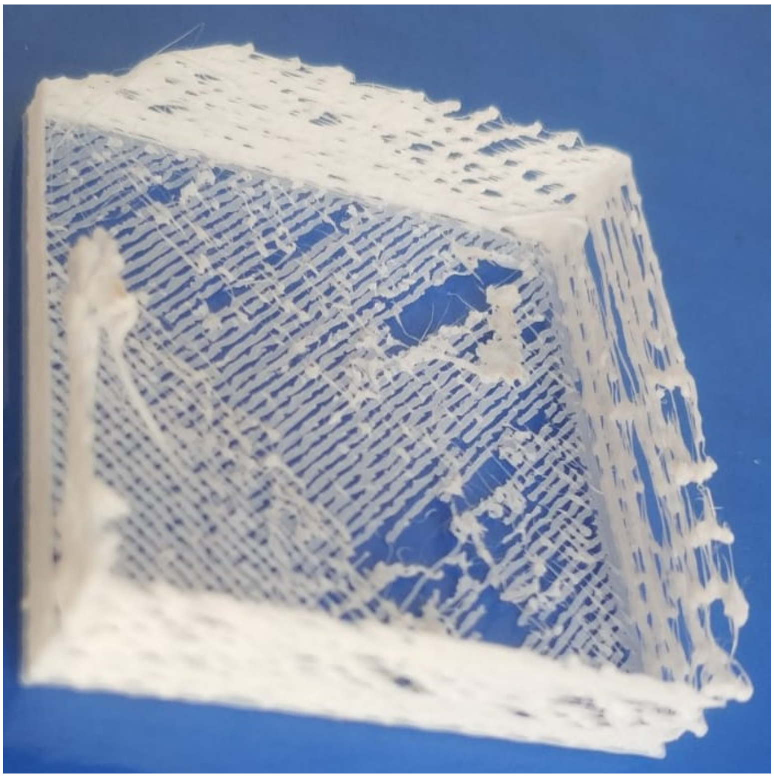


Figure 9.1: 40x40x40 box without tuning the printer

The structure in the photograph resembles a cube and has a rough surface texture. The object was manufactured using the FDM (Fused Deposition Modelling) method of 3D printing because the print lines can be seen on its surface. The object's surface seems a little crooked, which raises the possibility that the printer wasn't calibrated or adjusted properly when it was printed. The object looks to be a basic geometric form that was used to gauge the dual extruder 3D printer's capabilities before it was fine-tuned.

- The thing has rounded corners and edges and looks to be a cube. The cube has a uniformly thick wall thickness and is hollow on the inside.
- The object's surface seems to be relatively uneven and rough, with obvious layer lines and other abnormalities. This is a typical trait of FDM 3D prints, particularly

when the printer is not calibrated or tuned properly. It's possible to change the printer's settings to get prints that are smoother and more reliable.

- The object most likely came from a dual extruder 3D printer, which has two print heads that may simultaneously deposit various materials or colours. This enables the creation of intricate prints using various colours or materials.

- One explanation for some of the obvious flaws could be because the object was printed with a low-quality or low-resolution setting. Alternately, the printer might have been printing quickly, which can likewise produce a surface finish with more grit.

9.2 40x40x40 mm Box with Tuning

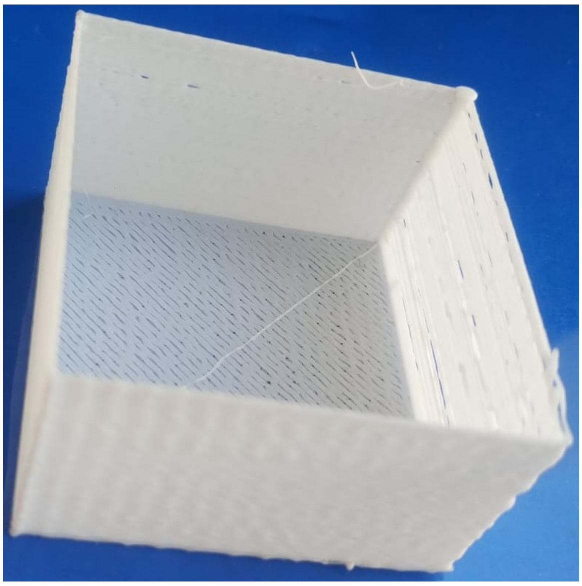


Figure 9.2: 40x40x40 box with tuning the printer

- A small structure that resembles a number of connected pieces makes up the object. It features an intricate geometric form with numerous acute angles and curves. The wall thickness is constant throughout, and the object is hollow on the inside.

- The object's surface seems to be considerably more even and consistent than the previous one. The overall finish is more uniform, and the layer lines are considerably less obvious. This means that in order to generate a high-quality print, the printer has been correctly calibrated and tweaked.
- The print appears to have been produced at a high resolution based on its level of complexity and detail.

Overall, compared to the previous 3D-printed object, this one seems to be more intricate and aesthetically pleasing. When calibrated and set appropriately, the twin extruder 3D printer can produce objects with a variety of colours and detailed patterns on their surface.

Chapter 10
Conclusion

The development of an FDM-based 3D printer for advanced additive manufacturing represents a significant stride in the realm of manufacturing technology. This engineering project has provided a comprehensive exploration of the principles and capabilities of 3D printing, highlighting the transformative potential it holds across various industries. By harnessing the power of additive manufacturing, our 3D printer demonstrates its ability to fabricate intricate objects with unprecedented precision and efficiency. The utilization of digital blueprints created through Computer-Aided Design (CAD) software allows for the seamless translation of ideas into tangible, customized products. The controlled material extrusion process, facilitated by a heated nozzle, ensures the precise layer-by-layer deposition of materials, paving the way for the creation of complex geometries and innovative designs.

The advantages of 3D printing extend beyond traditional manufacturing methods, offering enhanced flexibility, reduced waste, and cost-effectiveness. This technology enables rapid prototyping, product development, and even medical advancements, revolutionizing the way objects are conceptualized, designed, and manufactured. Moreover, the ability to produce customized, one-of-a-kind objects on-demand presents opportunities for unprecedented levels of personalization and efficiency.

Through this project, we have made significant progress in developing a functional prototype that showcases the immense potential of FDM-based 3D printers. Our work serves as a testament to the versatility, efficiency, and innovation that this technology brings to the manufacturing industry and beyond. As 3D printing continues to advance, it holds the promise of reshaping traditional manufacturing practices and unlocking new possibilities in design and production.

Looking ahead, further research and development in 3D printing technology will undoubtedly lead to even more groundbreaking applications and advancements. As en-

gineers and innovators continue to push the boundaries of additive manufacturing, we anticipate witnessing a future where 3D printers play an increasingly integral role in various industries, facilitating unparalleled creativity, efficiency, and customization.
The FDM-based 3D printer serves as a testament to the transformative power of additive manufacturing. The possibilities are vast, and the impact on industries and society as a whole is profound. By embracing this technology and continuously pushing its boundaries, we embark on a journey of manufacturing innovation that has the potential to reshape our world.

10.1 Achievement


Ref No: PEC/PPDC/22/ 105

PAKISTAN ENGINEERING COUNCIL

REGULATING THE ENGINEERING PROFESSION

"Final Year Design Project 2022-23"

Proposal Acceptance Certificate

It is to certify you that Final Year Design Project

3D Printer Based on FDM

(Electrical Department, Air University, Islamabad)

of your university has been approved for financial assistance by PEC

CONGRATULATIONS

It is required to complete this Design Project in accordance with the PEC standards so it can be included in the Capstone Expo-2023 and have it uploaded to the PEC E-library.

ENGR. MIR MASOOD RASHID
Convener, PPDC, PEC

ENGR. DR. NASIR M. KHAN
Registrar, PEC

Figure 10.1: PEC CERTIFICATE

10.2 Future Aspects

- **O**ne of the most important benefits of dual extruder 3D printers is the capacity to print with a variety of materials. In the future, other materials—including ones with special features like conductive filaments, magnetic filaments, and others—might be made available for use in twin extruder printers. [13]

- **Improved Software:** We might see more sophisticated features as the software powering 3D printers advances, like automated support generation, enhanced slicing algorithms, and more exact control over the two extruders. Due to this, a wider spectrum of customers will find twin extruder 3D printing to be simpler, more effective, and more affordable.
- **Advanced Color Capabilities:** Multi-color printing is now possible with dual extruder 3D printers, but in the future, we might see more sophisticated colour blending and mixing capabilities. This will make it possible for printed goods to have more intricate and realistic colour gradients.
- **Smaller Nozzles:** Smaller nozzles that provide finer detail in printed things may become more common as nozzle technology advances.
- **Faster Printing:** Even though FDM 3D printing has considerably increased in speed recently, there is still opportunity for growth. Dual extruder 3D printing may become even more effective and useful for large-scale production in the future as printing speeds increase.

References

[1] M. Attaran, “The rise of 3-d printing: The advantages of additive manufacturing over traditional manufacturing,” *Business horizons*, vol. 60, no. 5, pp. 677–688, 2017.

[2] C. M. Thakar, S. S. Parkhe, A. Jain, K. Phasinam, G. Murugesan, and R. J. M. Ventayen, “3d printing: Basic principles and applications,” *Materials Today: Proceedings*, vol. 51, pp. 842–849, 2022.

[3] C. K. Chua and K. F. Leong, *3D Printing and additive manufacturing: Principles and applications (with companion media pack)-of rapid prototyping*. World Scientific Publishing Company, 2014.

[4] S. Stopp, T. Wolff, F. Irlinger, and T. Lueth, “A new method for printer calibration and contour accuracy manufacturing with 3d-print technology,” *Rapid Prototyping Journal*, 2008.

[5] S. Olivera, H. Muralidhara, K. Venkatesh, and K. Gopalakrishna, “Evaluation of surface integrity and strength characteristics of electroplated abs plastics developed using fdm process,” 01 2016.

[6] K. Kun, “Reconstruction and Development of a 3d Printer Using FDM Technology,” *Procedia Engineering*, vol. 149, pp. 203–211, 2016.

[7] P. L. QIN and X. D. LI, “Hardware Control Circuit Design of FDM 3d Printer,” *DEStech Transactions on Engineering and Technology Research*, feb 14 2018.

[8] A. Altınkaynak, M. Gupta, M. A. Spalding, and S. L. Crabtree, “Melting in a Single Screw Extruder: Experiments and 3d Finite Element Simulations,” *International Polymer Processing*, vol. 26, no. 2, pp. 182–196, may 1 2011.

[9] C. T. Yu and U. HobbyPRESS, *A Practical Guide to Operating and Maintaining a Cartesian FDM 3D Printer: For Beginners*, jun 16 2021.

[10] “Calibration - RepRap,” https://reprap.org/wiki/Calibration.

[11] J. MOLINA SALAZAR, S. LOZANO LOZANO, M. RODRÍGUEZ MEDINA, and E. R. POBLANO OJINAGA, “Dual EXTRUDER 3d PRINTER PRINTING PROCESS USING FUSED DEPOSITION MODELING (FDM),” *DYNA*, vol. 97, no. 3, pp. 230–230, may 1 2022.

[12] J. Průša, “Reprap Calculator - Original Prusa 3d Printers,” https://blog.prusa3d.com/calculator_3416/.

[13] M. R. Khosravani, F. Berto, M. R. Ayatollahi, and T. Reinicke, “Characterization of 3d-printed PLA parts with different raster orientations and printing speeds - Scientific Reports,” https://www.nature.com/articles/s41598-022-05005-4, jan 19 2022.

Sustainable Development Goals (SDGs)

- QUALITY EDUCATION

- AFFORDABILITY AND CLEAN ENERGY

- INNOVATION AND INFRASTRUCTURE

- CLIMATE ACTION

- PARTNERSHIP FOR THE GOALS

3D printer

ORIGINALITY REPORT

11%	10%	4%	6%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Higher Education Commission Pakistan Student Paper	1%
2	circuit.pk Internet Source	1%
3	docs.com Internet Source	1%
4	www.scientific.net Internet Source	<1%
5	Submitted to University of Derby Student Paper	<1%
6	create.arduino.cc Internet Source	<1%
7	V. Manikandan, S. Anantha Babu. "A Survey on Cryptocurrencies Utilizing the Blockchain", 2022 11th International Conference on System Modeling & Advancement in Research Trends (SMART), 2022 Publication	<1%

8	pure.tue.nl Internet Source	<1 %
9	Submitted to University of Edinburgh Student Paper	<1 %
10	www.coursehero.com Internet Source	<1 %
11	dokumen.pub Internet Source	<1 %
12	link.springer.com Internet Source	<1 %
13	fics.nust.edu.pk Internet Source	<1 %
14	apessay.elementfx.com Internet Source	<1 %
15	hdl.handle.net Internet Source	<1 %
16	Submitted to Engineers Australia Student Paper	<1 %
17	Submitted to German University of Technology in Oman Student Paper	<1 %
18	downloads.hindawi.com Internet Source	<1 %
19	archive.org	

Internet Source

<1%

20	pastel.archives-ouvertes.fr	Internet Source	<1%
21	openaccess.city.ac.uk	Internet Source	<1%
22	www.collectionscanada.ca	Internet Source	<1%
23	Submitted to The Hong Kong Institute of Education	Student Paper	<1%
24	www.cchit.org	Internet Source	<1%
25	Submitted to The University of Manchester	Student Paper	<1%
26	www.semanticscholar.org	Internet Source	<1%
27	Submitted to University of Hertfordshire	Student Paper	<1%
28	Enrique Soriano Heras, Fernando Blaya Haro, José María de Agustín del Burgo, Manuel Enrique Islán Marcos. "Plate auto-level system for fused deposition modelling (FDM) 3D printers", Rapid Prototyping Journal, 2017	Publication	<1%

29	os.zhdk.cloud.switch.ch Internet Source	<1%
30	Submitted to Arab Academy for Science, Technology & Maritime Transport CAIRO Student Paper	<1%
31	Submitted to Singapore Institute of Technology Student Paper	<1%
32	dora.dmu.ac.uk Internet Source	<1%
33	etqan.sa Internet Source	<1%
34	repository.lib.ncsu.edu Internet Source	<1%
35	researchcommons.waikato.ac.nz Internet Source	<1%
36	vdocument.in Internet Source	<1%
37	www.youscribe.com Internet Source	<1%
38	5.imimg.com Internet Source	<1%
39	Mehdi Jahandardoost, Abbas S. Milani. "Multiphysics Modeling and Experimental Investigation of the Deposition Process in	<1%

Fused Filament Fabrication Method, under High-Viscosity and Non-Newtonian Material Flow", Journal of Materials Engineering and Performance, 2021
Publication

40	Thomas Ottnad, Franz Irlinger, Tim C. Lueth. "Design, construction, and verification of a printhead - tolerant towards bubbles - dosing liquid wax using rapid prototyping techniques", 2011 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), 2011 Publication	<1 %
41	erepository.uonbi.ac.ke Internet Source	<1 %
42	research-repository.griffith.edu.au Internet Source	<1 %
43	ses.library.usyd.edu.au Internet Source	<1 %
44	www.filastruder.com Internet Source	<1 %
45	www.instructables.com Internet Source	<1 %

Exclude quotes Off

Exclude matches Off