

Design and Evaluation of a Basic 3D Printer

Qusai Suwan, E2, Mohamad Fares Al Jajeh, E2, and Samer Abuthaher, E1

Electrical and Computer Engineering
American University of Beirut
Beirut, Lebanon

qzs00@mail.aub.edu, mma162@mail.aub.edu, sma90@mail.aub.edu

Abstract-We designed a simple 3D printer that uses PLA plastic to construct prototypes of computerized models. This document outlines the methodology of constructing this machine, the constraints that were faced in the design process, and an evaluation of the final product.

I. INTRODUCTION

In the world of industrial automation, the need to precisely produce goods has long been a driving force for technology. A form of Numeric Controls appeared as early as the 18th century, when knitting machines were developed to use punched cards to produce cloth patterns. With the advent of electronic computers in the 20th century, Numeric Controls became the de facto standard for machine tool manufacturers, with Computer Numeric Controls (CNCs) taking over in the 1970s with more programmable power [1].

CNC is a wide category that includes a variety of very different machines, with the unifying feature of using computer instructions to control the movement of tools, in order to manufacture a certain object. CNCs come in many shapes and sizes, probably the most commonly well known of which is the CNC mill, which uses a moving drill in 3 axes to clear away material from, most commonly, wood or plastic, curving it out into the desired model. Another common type of CNCs is the lathe, or turning machine, which are more advanced versions that have rotary axes and are generally used for more complex machining requirements [2].

3D printers are a relatively recent development to the world of CNC machines. They were proposed as means for Rapid Prototyping, as a quick way to create a physical model of CAD-designed objects. By laying down successive layers of a material, usually plastic, a 3D printer is able to construct physical objects to a very high degree of precision. Nowadays, 3D printing is becoming cheaper and more prevalent, which lead to the boom in the availability of personal 3D printers, and the inclusion of this technology to many new fields.

The technology most commonly used in 3D printers nowadays is called Fused Deposition Modeling (FDM). This process works by heating a filament of thermoplastic until it softens, then arranging that plastic material onto the printing surface by means of an extruder. The printing then goes on layer by layer, until the 3D model is completed (Fig 1). While this process is somewhat slow, it is faster than setting up a

production line, and is used in instances where a prototype is wanted on a short notice [3].

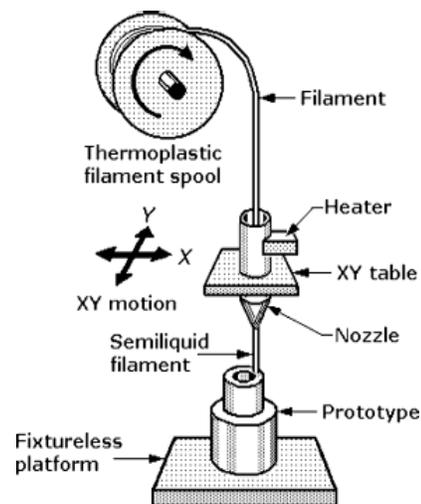


Figure 1 Fusion Deposition Modelling Extruder, Source: [3]

In this paper, we explain the design and manufacture of a simple 3D printer, constrained with limited budget and minimal professional experience on our part. The design is to be based on FDM technology, using an extruder to push out PLA plastic. The motion is controlled by a Cartesian robot, made of stepper motors for controlling three linear axes.

This paper will outline the strategies we used in designing a 3d printer machine using a controller with limited resources, and provide an overview of the various components of our final system. Finally, we will evaluate our end product, and see how well it performs in various cases, as well as provide some possible areas of improvement.

II. DESIGN PROCESS

After referring to the existing literature on CNC machines and 3D printers, we had a preliminary idea of the different ways the system is to be constituted. The different design concepts were then evaluated according to practicality and value, and at the end a concept that balances those two variables was selected.

We decided to build a rapid prototyping machine that uses FDM technology because of its relative simplicity and common

usage. This machine would be the natural extension of a CNC mill, working by depositing material instead of carving it away.

This 3D printer machine is based on the idea of a “Cartesian robot” often seen in CNC mills, and nowadays finding way into 3D printers. It is defined by 3 simple linear axes that can move independently. This type of motion comes naturally for 3D printers and CNC routers alike, as dealing with rotary axes adds significant complexities to the design if the machine was to maintain preciseness of operation. Moreover, and specifically for 3D printers, FDM works best when the material is extruded against gravity.

The Cartesian robot moves a part, called the *extruder*, in space. The extruder consists of two parts; a motor assembly called the *coldend* that pulls plastic filament into the *hotend*, which is a heating element that melts that plastic and forces it out through a thin nozzle. The temperature of the hotend is controlled so as to keep the plastic in a semi-liquid state. The plastic is pushed onto the surface below, called the *printbed*. The printbed needs to be of a material that sticks with the plastic just enough so the model stays stable when printing, but not too much to make removing a hassle.

The material to be 3D printed with was chosen to be PLA plastic. ABS plastic was considered, and while it provides a more durable final product, it requires higher temperatures in the hotend and thus would increase potential complexities. Moreover, PLA plastic can be used without heating the printbed thus further simplifying our design [4].

We sketched various configuration for the different components of the system, weighing options and alternatives, until a suitable design was reached. In the next section, the chosen design will be described.

III. SYSTEM DESCRIPTION

A. System Overview

To ease design, our 3D printing system was broken into smaller components (Fig. 2). On a high level, the system consists of a computer part that includes the software controlling the printer, and the printer itself. The software toolchain takes in the 3D model to be printed, and outputs a series of instructions to the machine. The machine itself is further divided into an electronic controller providing the logic and computer interaction, an electronics part that interfaces with the physical hardware, and the mechanism of the machine itself. In the coming sections, each component of the system will be examined.

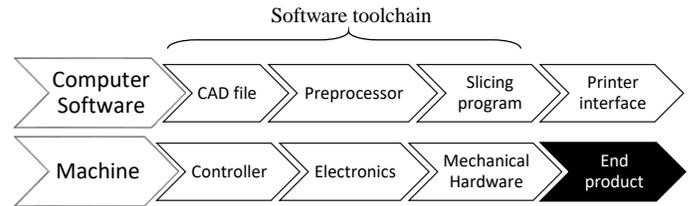


Figure 2 3D printer system components

A. Software Toolchain

A computer program first analyses a CAD model of the object to be printed. That model can either be generated by our controlling program, for simple shapes or testing, or imported from any common CAD design tool such as Autodesk® AutoCAD®. Then, a preprocessor would examine that model, and generate an approximate representation that can be machined within the constraints of our 3D printer.

Afterwards, a toolpath would be generated by a slicing program, to calculate the exact movements and extrusions needed to model the object. Inspired by the raster machining strategy for 3D milling, the program lays down successive layers of plastic, progressing through each layer in width-wise passes (Fig. 2). This toolpath is then expressed in a variant of G-code, a commonly used CNC programming language [5].

Finally, the printer driver transmits the G-codes to the printer. The transmission takes place over a USB serial interface, and the program file is transferred in one go to the controller. After the transfer is completed, the program gets executed, and the serial link provides information on job execution to the connected computer.

Throughout the program, a session stays open with the computer, and the operator is able to send commands to control certain parameters of the program execution, and to provide an additional safety mechanism to the machine.

B. Electronic Controller

The 3D printer uses an Arduino Pro Micro¹ for the controller. The Arduino interfaces serially via USB and receives commands from the software toolchain.

While we used the existing G-code standard as a guideline, we ended up implementing our own protocol, which vaguely resembles the standard but does not comply fully with it, due to limitations in our machine. In our implementation, commands are categorized into two categories: M-codes, which are codes that change and read the various states of the machine, and G-codes, which controls and affects mechanical movement. M-codes are non-blocking, in that the controller can execute them while a G-code is in progress, while a G-code can be executed only after the previous one has finished. A reference of implemented codes, and their syntax, is included at the end of this paper (Fig. 6 and Table 1).

A challenge was faced in managing the limited resources in the Arduino, and being able to receive commands while movements are being executed. We used the timer interrupts

¹ <https://www.sparkfun.com/products/12640>

built in the Arduino microcontroller to run a routine task that does the actual driving, which needed to run often enough while not blocking the execution of other parts of the program. Moreover, dynamic memory was also severely limited, leading us to adopt an approach using global states and compile-time constants whenever possible, to reduce unnecessary memory usage.

The movement of the axes employs open loop control of the stepper motors. The internal state of the machine maintains the current position of the motors for use in absolute positioning, but no feedback link exists to verify the exact movements took place. The precision of the steps in the stepper motor allows us to forgo positional feedback without significant loss of accuracy in machining [6].

In addition to that, the temperature of the hot end needs to be controlled so it stays within a given range. A simple PID controller was implemented for that purpose, with the parameters determined experimentally. One set of parameters was devised, that performs reasonably well both when pushing out plastic and when the filament is stationary. The PID controller is activated by another timer interrupt, so as to not block the flow of the program.

With optimizations and C-style stateful programming, we were able to make a well-behaved 3D printer firmware that works with the limited RAM and program space of the Arduino.

C. Electrical Components

Two switching mode power supplies provide power for the Arduino, the motors, and the extruder assembly. Those power source has low ripple voltage, giving us stable operation for the machine, while providing enough power for our printer.

We are using bipolar stepper motors to control the movement of the three axes. The stepper motors, requiring a high driving currents, are controlled by driver circuits. The design of a dedicated controlling circuit allows the current through the stepper to reach its final value more rapidly, compared to using a naïve transistor-switched control. Moreover, the driver employs micro-stepping to provide smoother output and finer control of the stepper motor. In addition to that, it protects the controller and the power supply from voltage surges associated with the inductive load [6].

Due to the relatively high currents involved, care needed to be taken when considering ways of connecting the various components. A large wire cross section was essential in avoiding heating issues because of the high current. Moreover, the driver circuit generates a relatively large amount of heat, and thus care had to be taken in mechanical design to avoid any issues.

Moreover, some physical safety mechanisms are provided via the hardware; in case of a malfunction, a physical emergency switch can immediately disable all the stepper motors.

D. Hardware Design

Mechanical driving wise, the transformation of rotary motion into linear motion is used. We used threaded rods mounted on

the stepper motor shafts, and ball screws mounted on the threaded rods. Hence, when the motor shaft rotates, the ball screw moves along the threaded rod. This mechanism is basically used for the 3 axes. On to the printer floor, there is the printed bed which moves along the y-axis. This print bed is mounted on the ball screw of the y-axis stepper motor.

Next, two supports from two sides of the printer extends upward. On one of the supportters, a stepper motor is mounted, which is responsible for the x-axis movement. On the x-axis ball screw, a support is used to lift the z-axis mechanical parts. At the top of the latter support, the z-axis stepper motor is mounted. On the Z axis ball screw, the extruder is mounted. The design is 3D sketched using Google SketchUp 2015, Refer to Figures below.

The maximum attainable object dimensions are 15x30x20 cm. Extra space was given to increase manufacturing tolerances, but the firmware does not allow it to be reached by the extruder.

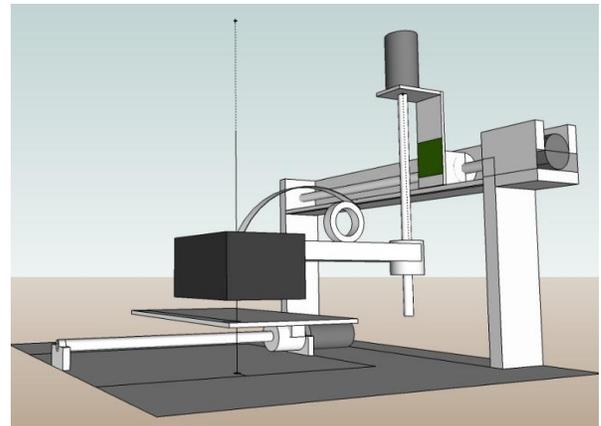


Figure 3 Sketch of the 3D printer

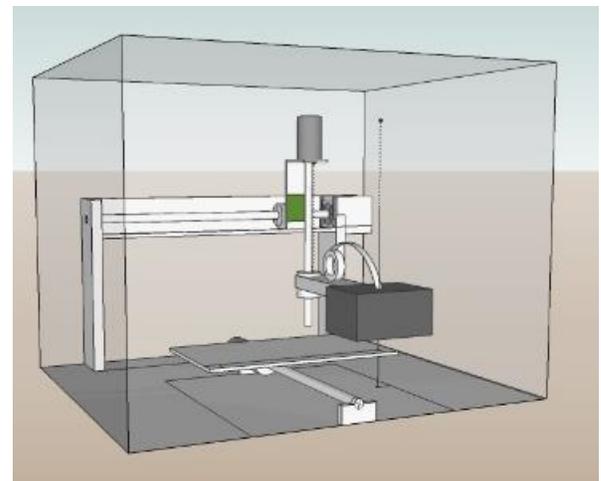


Figure 4 Sketch of 3D printer, another perspective

IV. EVALUATION

We were successful in implementing a controller with the limited resources available to the Arduino, and it was able to

process commands and accurately drive the stepper motors at the maximum speed allowed by the motor. Furthermore, using ¼ step microstepping, we reduced mechanical oscillations in the motion of the stepper motor, making for a precision of movement on the order of tenths to hundreds of millimeters.

As of the date this paper is submitted, the system is not 100% complete, so a comprehensive evaluation cannot be provided until we complete our rounds of testing. The full evaluation will be provided on the date of the conference.

For further development, one can see improve that can be done upon our work. Primarily, one can optimize the mechanical design for more precision, faster manufacturing, or a larger work area. Moreover, the controller can be extended as to support more commands, and making it able to process standard G-codes that control commercial 3D printers. A storage buffer can be added, so as to make the 3D printer able to work independently of a computer. Those are some basic ways one can improve on our work outlined here.

V. CONCLUSION

Here, we have shown the components that constitute an FDM 3D printer, and a basic design for such a machine. The rapid prototyping process outlined in the paper starts with a CAD model, which gets processed into a format for machining, which is then turned into a plastic model by injecting successive layers of PLA plastic to constitute the object.

During the making of the machine, we faced obstacles that included limited time, the constraints of the hardware, and our limited experience in the field being E2 and E1 students. However, we were able to overcome those challenges and work on producing the machine.

This project was a personal experience to us, in that it gave us an insight into the workings of 3D printers and CNC machines. Studying the design is a first step towards contributing research in this field, and this project gave us a firsthand account of the problems that can be tackled within this area.

VI. ACKNOWLEDGMENT

We would like to thank Dr Youssef Nasser and Dr Imad Elhadj for their help and advice in the development of the project. We further acknowledge Mr Khaled Joujou and Mr Salam Abyad for their invaluable help in providing us with a workplace, helping us with the funding, and their technical advice. We are also very grateful to the ECE department of AUB, for covering a large part of our expenses.

VII. REFERENCES

- [1] S. F. Krar, *Computer Numerical Control Simplified*, New York: Industrial Press Inc., 2001.
- [2] H. B. Kief and H. A. Roschiwal, *CNC Handbook*, New York: McGraw-Hill Education, 2012.

- [3] N. Sclater, *Mechanisms and Mechanical Devices Sourcebook*, Fifth Edition, New York: McGraw-Hill Professional, 2011.
- [4] B. Evans, "Practical 3D printers: the science and art of 3D printing," Apress, New York, 2012.
- [5] J. R. C. Bralla, "Computer numerical control," McGraw-Hill Education, 2014.
- [6] R. Condit and D. W. Jones, "Stepping Motors Fundamentals," *Microchip*, 2004. [Online]. Available: <http://homepage.cs.uiowa.edu/~jones/step/an907a.pdf>. [Accessed 22 March 2015].
- [7] L. G. Richards and K. P. White Jr., *Computer-aided design and manufacturing*, McGraw-Hill Education, 2014.

A. Reference of implemented G-codes

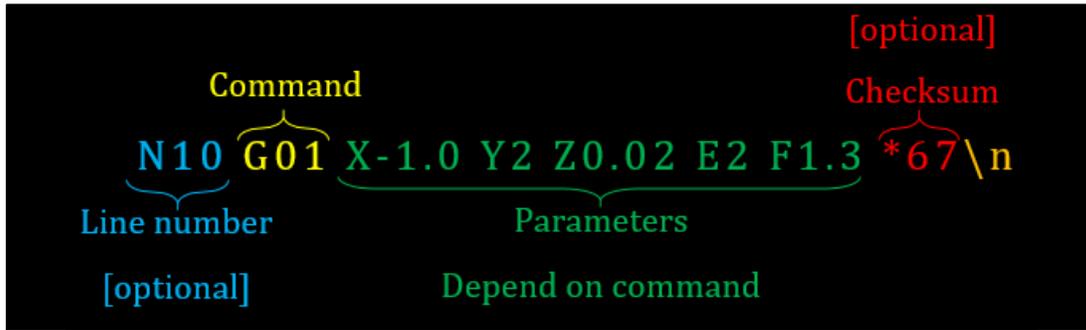


Figure 5 G-code structure

Table 1 Implemented commands

Code	Description	Parameters*
M70	Get axis flags	None
G00	Rapid positioning	X, Y, Z
G01	Linear Interpolation	X, Y, Z, E, F
M77	Echo (Repeats back the values)	X, Y, Z, E, F
M00	HALT – stops all movement of steppers immediately	None
M97	Resume after HALT, resumes moves that were in progress	None
M96	Recover after HALT, removing moves that were in progress	None
M114	Get current position	None
G90	Enter absolute mode	None
G91	Enter incremental mode	None

*Axes: X, Y, Z, E are in millimeters, F is in mm/sec
E is the length of extrusion, F is the speed of the XYZ movement