Custom CNC Milling Machine

Supervised by
Dr. Md. Khalilur Rahman
Associate Professor
Department of Computer Science & Engineering
BRAC University, Dhaka.

<u>SUBMITTED BY</u>	<u>ID</u>
MASRUR KHAN	14121030
MERAJ KHAWJA AHMED	14110004
NEAZ AHMED ABUL ULAY	13110018
MD.NOOR ISLAM SARWAR	13221009

A Thesis report submitted to the Department of Electrical & Electronic Engineering, BRAC University, in partial fulfillment of the requirements for the Bachelor of Science degree in Electrical & Electronic Engineering

Summer 2018

DECLARATION

We hereby declare that the thesis work titled, “Custom CNC Milling Machine” is our own original work. This work has not been submitted elsewhere for the honour of any degree or any other publication nor has it been presented anywhere else for assessment. Any material that has been used from other sources is properly acknowledged.

Signature of Thesis Supervisor

Dr. Md. Khalilur Rahman

Signature of Authors

Masrur Khan

Meraj Khawja Ahmed

Neaz Ahmed Abul Ulay

Md. Noor Islam Sarwar

ACKNOWLEDGEMENTS

Firstly, we are extremely thankful to our supervisor Dr. Md. Khalilur Rahman- Associate Professor of the Department of Computer Science & Engineering (CSE), BRAC University; for his continuous support, guidance and feedback to complete this thesis. Secondly, we would like to express our gratitude towards BRAC University for financing this project.

ABSTRACT

Computer Numerical Control or CNC Machines are integral in shaping complex layouts into physical products. The plotting mechanism of such machines comprise of very fine accuracy that is never humanly possible to create. In Bangladesh, various types of products from raw materials like wood, copper board or fabric involve the use of CNC machines for industrial production. However, like the problems in most developing countries, the use of such machines require technical expertise, especially for repairs or modifications and this country has lack of such engineers and technicians. Also, these are mainly imported from abroad and the required software to operate such machines is proprietary which adds more to the cost. These issues have been the country's long term problem in the industrial sector. For an innovative solution, this paper discusses about the concept of building a CNC machine with a more cost-effective approach and available resources. This includes the use of open source software as an alternative to proprietary software in order to make the CNC machine operations easier to understand for the average engineer. Initially, a prototype and then a final structure has been built, based on that prototype design for printing and milling applications, where the final structure printed with about 60% accuracy.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

NC	Numerical Control
CNC	Computer Numerical Control
GCODE	Gerber/Geometry Code
GCTRL	Gerber Control (file name)
IDE	Integrated Development Environment
PCB	Printed Circuit Board
COM port	Communication port
SVG	Scalable Vector Graphics
PVC	Polyvinyl Chloride
I/O	Input/Output
PWM	Pulse Width Modulation
DC	Direct Current
VCC	Voltage Source
GND	Ground

CHAPTER 1: INTRODUCTION

1.1 MOTIVATION

To build an affordable and efficient CNC milling machine with industrial properties. The aim is to reduce the cost of custom PCBs which may be manufactured for future electronic inventions and provide more flexibility on Electronic research.

1.2 INTRODUCTION TO CNC MACHINES AND THEIR HISTORY

When the operations and movements of a power-driven machine are manipulated by means of a setup program consisting coded alphanumeric data, it is called, ”Computer Numerical Control”(CNC).The CNC is a technology that uses microcomputers to develop, parse and execute the chronological control that describes the end effectors’ behavior. It can control the mobility of the equipment, the input specifications such as feed, intensity of cut, pace and functions such as twisting the pivot on/off, turning the coolant on/off.

In the manufacturing and production industries, computer numerical control may be a new concept but the origin of a CNC goes back to the elemental idea of NC, or numerical control. Basically, when particular concepts of programmable logic were infused into the automation of machine instruments, the perception of numerical control began. At first, the early NC appliances were constructed back in the 1940’s.In 1950; a bit more high-tech machines came along. These building machines were made based on the existing apparatus that were adjusted with the motors laid out to move the controls of the machine. The precise points that were incorporated into the machine on punched tape were followed by these controls. With both analog and digital computers these primitive mechanisms were shortly upgraded.

Around 1960-1970 the first CNC machines were developed. The initiation of computer technology into the concept of numerical control brought about what we now know as computer numerical control.

Fig. 1.1: CNC machine in 1940

Fig. 1.2: CNC machine in 1950

1.3 IMPACT OF CNC MACHINES ON THE ECONOMY OF BANGLADESH

Modern day CNC machines have a large impact on the economy of Bangladesh. From textiles, leather and wood to plastic and paper, from electronics and metal to stone and agriculture, it is being used in almost every type of industries. They are also utilized by and familiar among engineers, research institutions, high schools, universities, small to medium businesses and hobbyists. The range of applications include fabric-roll cutter, PCB drill machine, wood carving machine, diamond cutting machine and many more etc. Furthermore, the uses of CNC consist of both machine tool as well as non machine tool regions. Machine Tool Category includes Lathe, Drill Press, Milling Machine, Grinding Unit, Laser, Sheet-Metal Press Working Machine and Tube Bending Machine etc. Non Machine Tool Category involves Welding Machines (arc and resistance), Coordinate Measuring Machine, Electronic Assembly and Tape Laying & Filament Winding Machines for composites etc.

In today's era, CNC machines are an important factor for our industrial and non-industrial production system as their utilization led to great benefits. The benefits are as follows : great preciseness in constructing, less yielding time, more production flexibility, easier repairing, contour machining(2 to 5 axis machining), decreased human flaw, less manpower and hence less cost. Without CNC machines, the condition of many major companies, organizations and corporations might go back to the days before 1950 or 1940 when labor force was required to mass produce products.

1.4 CATEGORIES OF CNC MACHINES

1. CNC MILLING MACHINE - A machine which uses computer controlled rotary cutters to drill and cut hard objects like metal. It can be categorized into vertical mill (cutter in the z-axis), with two other planes working in the x and y axes or into horizontal mill (cutter in the x-axis or y-axis), with the two other plane working in the rest of the axes [10, 11].
2. CNC LATHE MACHINE - Like mills, lathes [12] are also computer operated machines which are used to remove material from a spinning hard object like metal by means of different cutting appliances, such as tool bits and drill bits.
3. CNC ROUTER MACHINE - A computer manipulated machine which is used to cut relatively softer objects like wood, plastic and foams via a rotary cutter which can move in all directions(x, y, z and other axes). It can also perform faster than a mill but is less precise [11].
4. CNC PLASMA CUTTER MACHINE - A CNC plasma cutter [13] is very comparable to a CNC router in magnitude and structure but it uses a plasma torch to cut things like sheet metals, metal plates, straps, bolts and pipes etc.
5. CNC LASER CUTTER MACHINE - A CNC laser cutter [14] serves the identical objectives just as the plasma cutter but uses a laser instead of a plasma cutter.
6. 3D PRINTER - It uses CNC technology to construct 3-dimensional item layer by layer
7. PICK AND PLACE CNC MACHINE - To raise electrical constituents for electronic devices and allocate them in the wanted position, pick and place CNC machines are put to use which comprise of several nozzles. These are very frequently applied in the creation of cell phones, computers, tablets, etc.

1.5 CATEGORIES OF A CNC MILL (IN TERMS OF SIZE)

1. Regular CNC mill - Very big CNC mill, usually covers a floor or even larger spaces. Weighs between 30,000 kg-28 T. Price ranges from 80, 000, 00 Tk. to 10, 000, 000, 0 Tk. Works with very large objects.

2. Mini CNC mill - Medium sized CNC mill, usually covers a room. Weighs between 900 kg-17, 000 kg. Price ranges from 10, 00000 Tk. to 65, 00000 Tk. Works with medium size objects.

3. Micro CNC mill -Very small CNC mill, usually kept on a table or desk. Weighs between 16 kg-600 kg. Price ranges from 40, 000 Tk. to 70, 0000 Tk. Works on very small and precise regions of the work piece.

1.6 PROBLEMS WITH THE TRADITIONAL CNC MACHINES

Trouble with conventional CNC machines are as follows: they are expensive, they need preservation or upkeep and an expert part programmer is needed to operate them. There are no companies and industries in Bangladesh which produce any kind of CNC machines but only a few of them assemble CNC machines, whose parts are imported from abroad (like China and India etc.).The customers normally have to pay import duty or tariff for the components of the CNC machine or the whole CNC machine that is transported from another country. The operating software for the CNC machine is often provided by the manufacturing company and is not taken from open sources, which is not free of cost either. Furthermore, the delivery of the machine through shipping, cargo plane or by any other means adds extra cost.

1.7 PROJECT OVERVIEW

The key point of this research is to show that anyone with a good knowledge of Arduino programming and many other open source software can build an inexpensive and labor-saving CNC machine. Though the purpose for constructing a CNC milling machine is to decrease the price of custom PCBs, they can use it for other objectives as well.

In the initial phase, some information was gathered about the CNC machines. The fundamental elements of a CNC had been determined:-

1. Part programmed
2. Machine Control Unit(MCU)
3. Machine Tool(Lathe, Drill Press and Milling Machine etc)

Their applications and operating system were studied.
It was decided to build a CNC milling machine which would fall under the category of a micro CNC mill due to its light weight and low price.

In the middle phase a prototype 3 axes and 2 dimensional, CNC printing machine with two used DVD drivers was developed. Each driver had a stepper motor attached and were placed in the X axis and Y axis. A pen fixed with a servo motor was also placed in the Z axis. After that, for the machine to function, five freely available softwares were used: one for circuit schematic, one for Arduino and the rest were CNC softwares.

In the last phase, an effort was made to set up 3 axes and 2 dimensional, CNC milling machine with a metal body, glass motors and a drill bit. However, for the following reasons, the drill bit was removed - the CNC machine's weight is already 19.87 kgs; that is almost 20 kgs and was difficult to carry around during trial and error. Had the drill been set, the weight would have been compromised and it would have made interfacing with the computers more tedious at least for the time being. The Y-axis on the machine was a touch unstable and imbalanced, so had the drill been adjusted, it would have given very inaccurate outputs, although proper precision was set on the CNC Machine code with feedback. For most major projects, hardware and software must always work in synchronized form. To run X axis and Y axis, a glass motor was used for each. To run the Z axis, a pen affixed to a servo motor was used like the prototype. More attributes were added, such as feedback system and manual control function. The control softwares that had been used in building the prototype were also used in creating the actual machine.

CHAPTER 2: REFERENCE OF AN EXISTING CNC MILLING MODEL

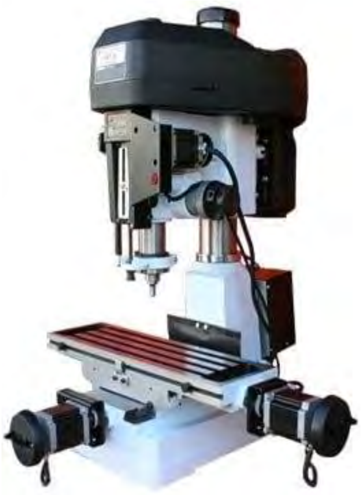


Fig. 2.1: CNC Jr. Table Top Milling machine

2.1 INTRODUCTION/ OVERVIEW OF AN EXISTING MODEL

The optimal model that has been referred to in comparison of the CNC milling machine in this project is CNC Jr. Table Top milling machine, produced by CNC MASTERS at Irwin dale, California, USA. It is 3 axes, 3 dimensional table top CNC milling machine used for milling metals like copper and aluminum etc. It costs $5,573 or470138.28 Tk. (import tariff and transport price separate). It operated with MASTER MX software, created by CNC MASTERS, which can run on windows 7, 8 or 10, touch screen, desktop or laptop PC by standard USB port connection. Additionally, from other CAD-CAM softwares such as MasterCAM, SmartCam, SurfCam, etc., the MX can also run standard FANUC G-code files. If any sort of problem appears, the CNC MASTER's team of technical assistance and instructors can be approached online or by phone.

The CNC Jr. Table Top milling machine is currently being used by, "Jet Propulsion Laboratory" of, "NASA", at Pasadena, in California [8].

2.2 WHY THIS MODEL WAS CHOSEN?

The main objective of choosing a reference model was to look for a CNC mill which would be same, similar or very close to the model and features compared to the CNC mill, created in this project. Thus, at first a primary research was conducted, by looking into factories and workshops. However, when that did not turn out to be fruitful, a secondary research was conducted by browsing the internet. This was the closest that could be found in comparison to the CNC mill of this project. It is 3 axes, 3 dimensional table top CNC milling machine, with a manual control function and also a feedback system. The Tech Specs and features of this model are specified in reference index [8].

CHAPTER 3: BUILDING THE PROTOTYPE

3.1 HARDWARE APPARATUS

Before proceeding on building the actual machine, steps were first taken in constructing a prototype CNC structure [1] which would later work as a guide and determine the shape and structure of our final design. For this prototype the following components were gathered using personal funding:

Components	**Price (tk.)**
Arduino Uno	475
L293D Motor Driver Shield	300
Used DVD Drivers (two quantities)	400
3mm PVC Board	90
Wires	40
Screws, Washers & ‘L’ Brackets	37
Total	**1342/=**

Table 3.1: Cost of components (for prototype)

3.2 CONTROL SOFTWARES

To avoid the trouble of using expensive and complex softwares, open source softwares were downloaded from the internet, installed and used [6]. They are easy to use, free of cost and each one has their own unique function. These include,

1. Electronic PCB Software (Example:- Proteus & Eagle PCB etc.):

There are numerous PCB design softwares that can be downloaded from the internet but for now, Proteus Design Suite was chosen as the go-to Electronics software which was created by Labcenter Electronics Ltd. in 1988. Proteus is a proprietary software for electronic design automation. It has a user friendly interface and its design system is segmented in three different tabs- the ISIS schematic capture tab, where we selected the components and created the connections for our desired circuit; the ARES PCB layout tab, where the desired PCB was shaped according to the required dimensions, the components were placed on desired positions and connection routes were set; and finally the 3D Visualizer, which will provide a rendered 3D view of the total electronic board design. For the time being, the ISIS and ARES tabs were used primarily in pre-thesis.

2. Inkscape:

Inkscape is a free and open source vector graphics software first released in 2003. Its primary file format is Scalable Vector Graphics or SVG [7], but it can still import and export various other file formats. This software is the most critical segment of the thesis work as this will determine whether the print design is readable or not. It has plenty of extensions and formats, but for this task, the requirement is Inkscape's G-code format [9].

3. Arduino (CNC code):

Arduino microcontroller board is used as the brains of the operation for the CNC machine, so Arduino IDE software is also used. This software is used as cross platform application written in programming language. The file 'CNC code' is uploaded to the Arduino board which activate the moving axes of CNC machine.

4. CAMotics:

CAMotics is a CAM (Computer Aided Machine) software used for simulating a CNC machine prints or plot processes. This software is useful for machine learning [6] as it verifies how the hardware will operate. The G-code file from Inkscape [9] will be opened on CAMotics. This process does not guarantee the accuracy of the machine, but it does indicate whether it can print (or plot) properly or not. In short, it gives an overview of how the final print design will look like as well as the coordinate points of the machine. Simulation is a vital step recommended in any engineering field.

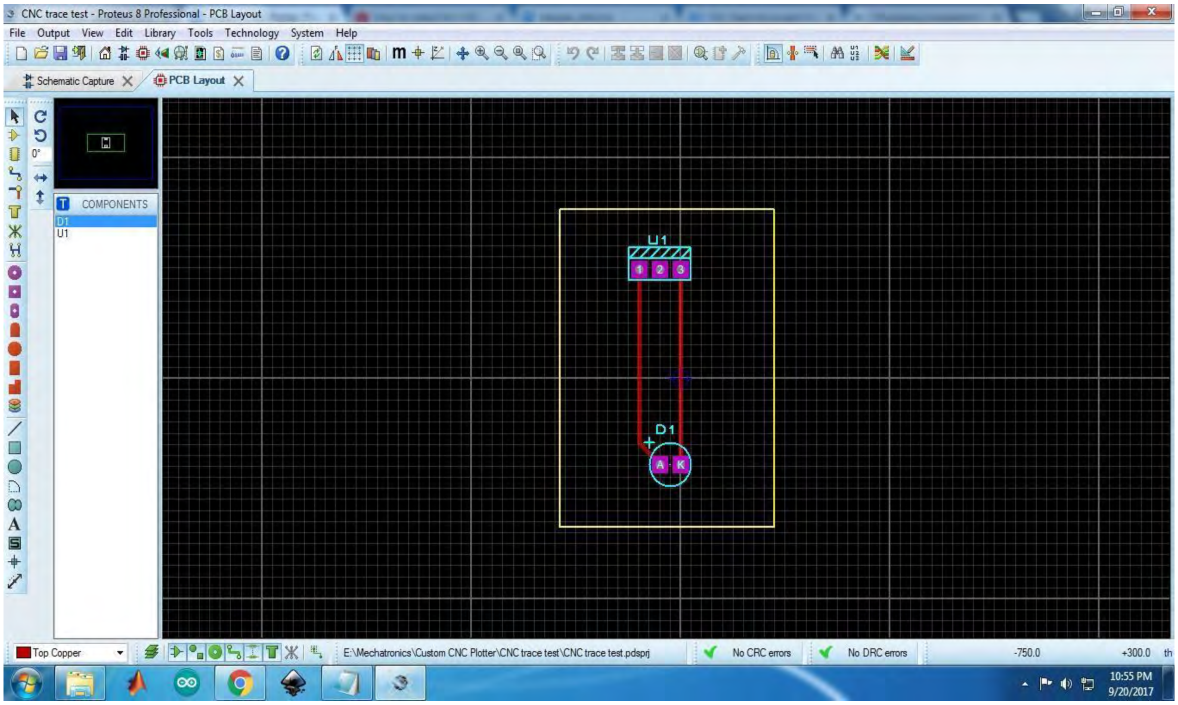
5. Processing (GCTRL code):

Processing IDE is another open source computer programming software written in Java & JavaScript initially released in 2001. The CNC code file of Arduino will activate the CNC machine but the GCTRL code file of Processing will execute the G-code file, meaning, this will activate the print of the circuit design to be done by the CNC machine.

3.3 SOFTWARE SETUP

To understand how a CNC machine works, research was done on a CNC simulation process. For this simulation Test, three softwares were used:

1. Proteus 8.5 Design Suite (minimum version 8.0)
2. Inkscape 0.92.2 (or version 0.48)
3. CAMotics 1.1.1


CNC trace test - Proteus 8 Professional - PCB Layout
File Output View Edit Library Tools Technology System Help
Schematic Capture
PCB Layout
0°
COMPONENTS
D1
U1
U1
1 2 3
D1
A K
Top Copper
-750.0
+300.0
th
10:55 PM
9/20/2017

E:\Mechatronics\Custom CNC Plotter\CNC trace test\CNC trace test.pdsprj

No CRC errors

No DRC errors

Fig. 3.1: Circuit design PCB Layout on Proteus

First, Proteus was opened and a very simple circuit was designed on the ISIS schematic.

After opening the circuit to the ARES PCB Layout (Fig. 3.1) and the routing was set, the circuit design was exported to a bitmap image. The Bitmap image is shown on the right figure (Fig. 3.2)

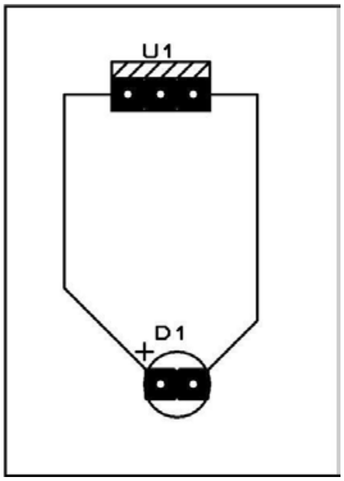
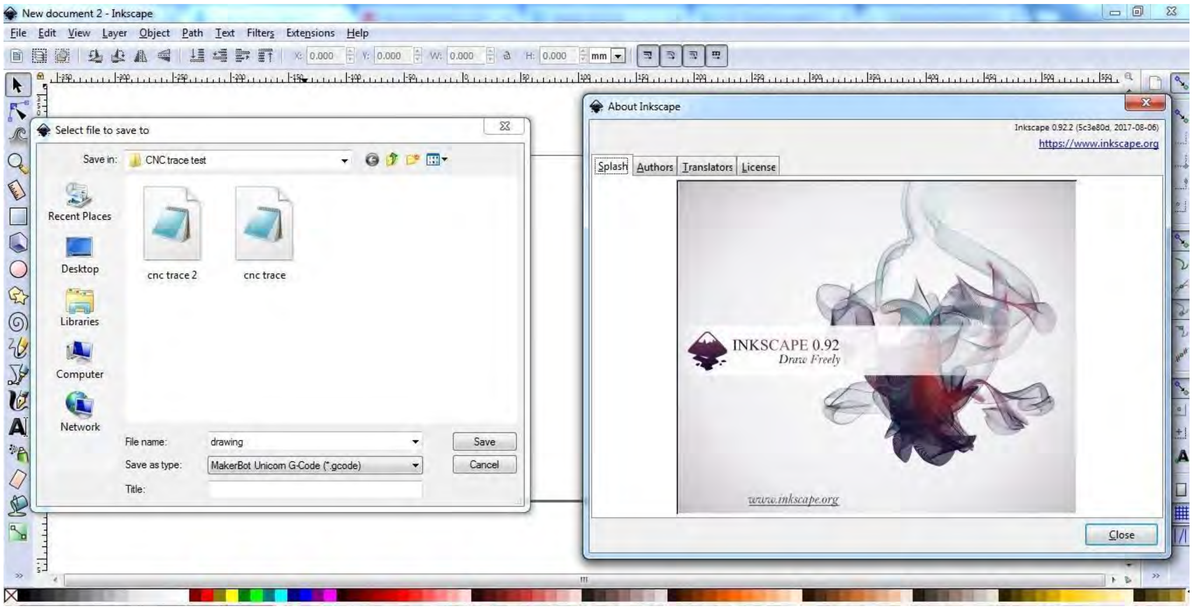


Fig. 3.2: Circuit Bitmap image



Save in: CNC trace test

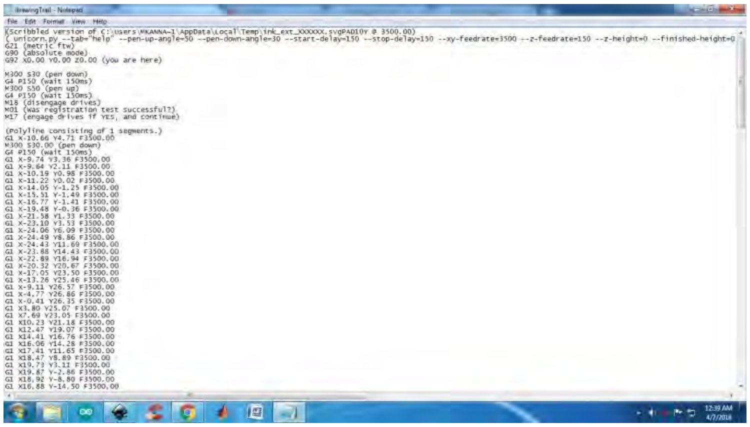
File name: drawing

Save as type: MakerBot Unicorn G-Code (*.gcode)

Title:

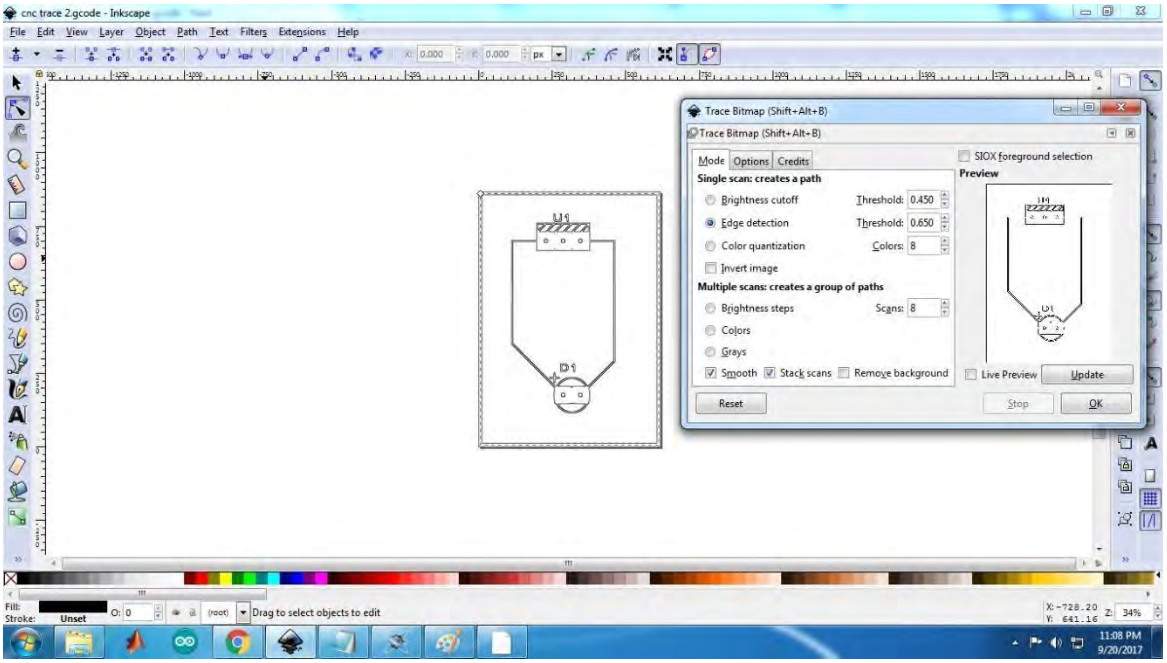
Fig. 3.3: Inkscape with G-code extension

Next, the latest version of Inkscape was downloaded. CNC machines require a programming language called G-code (also called Geometry Code or Gerber Code) which tells a machine "how" to make something. During research, it was discovered that Inkscape has this operation, but initially when it's downloaded, it does not have the G-code extension in its file formats [9]. That extension needs to be added manually. The G-code extension file was downloaded from a given link on the Github repository website and extracted. All the required files from the 'src' folder were copied to the 'extensions' folder of Inkscape's installed directory. This enabled Inkscape to save certain files under the MakerBot Unicorn G-code format (Fig. 3.3). When a design file from Inkscape is saved as g-code, it generates a notepad file which calculates all the plotting coordinates for the CNC machine to set the tool position and then plot or print (Fig. 3.4).



```
(Scribbled version of C:\Users\MKANNA~1\AppData\Local\Temp\ink_ext_XXXXXX.svgPAD10Y @ 3500.00)
( unicorn.py --tab="help" --pen-up-angle=50 --pen-down-angle=30 --start-delay=150 --stop-delay=150 --xy-feedrate=3500 --z-feedrate=150 --z-height=0 --finished-height=0
G21 (metric ftw)
G90 (absolute mode)
G92 X0.00 Y0.00 Z0.00 (you are here)

M300 S30 (pen down)
G4 P150 (wait 150ms)
M300 S50 (pen up)
G4 P150 (wait 150ms)
M18 (disengage drives)
M01 (Was registration test successful?)
M17 (engage drives if YES, and continue)

(Polyline consisting of 1 segments.)
G1 X-10.66 Y4.71 F3500.00
M300 S30.00 (pen down)
G4 P150 (wait 150ms)
G1 X-9.74 Y3.36 F3500.00
G1 X-9.64 Y2.11 F3500.00
G1 X-10.19 Y0.98 F3500.00
G1 X-11.22 Y0.02 F3500.00
G1 X-14.05 Y-1.25 F3500.00
G1 X-15.51 Y-1.49 F3500.00
G1 X-16.77 Y-1.41 F3500.00
G1 X-19.48 Y-0.36 F3500.00
G1 X-21.58 Y1.33 F3500.00
G1 X-23.10 Y3.53 F3500.00
G1 X-24.06 Y6.09 F3500.00
G1 X-24.49 Y8.86 F3500.00
G1 X-24.43 Y11.69 F3500.00
G1 X-23.88 Y14.43 F3500.00
G1 X-22.89 Y16.94 F3500.00
G1 X-20.32 Y20.67 F3500.00
G1 X-17.05 Y23.50 F3500.00
G1 X-13.26 Y25.46 F3500.00
G1 X-9.11 Y26.57 F3500.00
G1 X-4.77 Y26.86 F3500.00
G1 X-0.41 Y26.35 F3500.00
G1 X3.80 Y25.07 F3500.00
G1 X7.69 Y23.05 F3500.00
G1 X10.23 Y21.18 F3500.00
G1 X12.47 Y19.07 F3500.00
G1 X14.41 Y16.76 F3500.00
G1 X16.06 Y14.28 F3500.00
G1 X17.41 Y11.65 F3500.00
G1 X18.47 Y8.89 F3500.00
G1 X19.73 Y3.11 F3500.00
G1 X19.87 Y-2.86 F3500.00
G1 X18.92 Y-8.80 F3500.00
G1 X16.88 Y-14.50 F3500.00
```

Fig. 3.4: Sample G-code file


cnc trace 2.gcode - Inkscape
File Edit View Layer Object Path Text Filters Extensions Help
U1
D1
Drag to select objects to edit
11:08 PM
9/20/2017

Trace Bitmap (Shift+Alt+B)

Mode | Options | Credits

□ SIOX foreground selection

Preview

Single scan: creates a path

○ Brightness cutoff Threshold: 0.450

● Edge detection Threshold: 0.650

○ Color quantization Colors: 8

□ Invert image

Multiple scans: creates a group of paths

○ Brightness steps Scans: 8

○ Colors

○ Grays

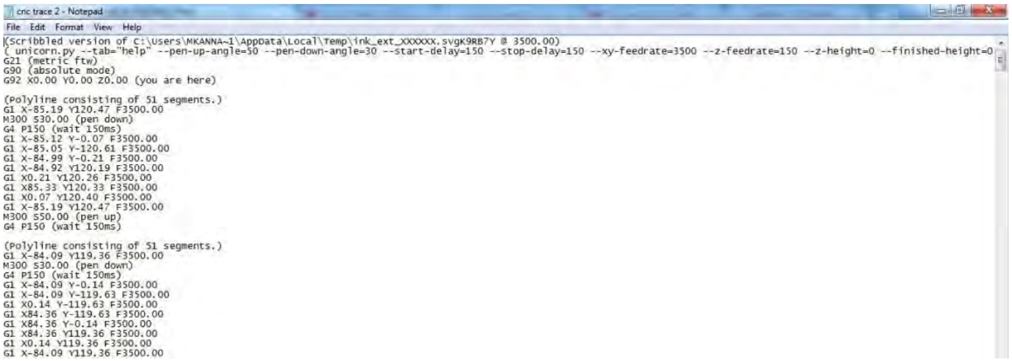
☑ Smooth ☑ Stack scans □ Remove background

□ Live Preview | Update

Reset | Stop | OK

Fig. 3.5: Trace Bitmap

Focusing back to the circuit design, the bitmap image was opened on Inkscape (no import settings were changed; clicked OK to open). The image was selected and the Inkscape dimensional platform was adjusted on the Document Properties option (going to File>Document Properties) with respect to the dimensions of the design platform. The design was given the file name ‘cnc trace 2’. Next, the image was selected and on the Path>Trace Bitmap option, the edge of the design was traced using the Edge detection option. The edged design or design outline was dragged out and the original design was deleted. The design outline was placed on the dimensional platform (Fig. 3.5). The design objects were converted to paths on the Path>Object to Path and then Path>Dynamic Offset options. Then finally, the design was saved as G-code format (Fig. 3.6).



```
(Scribbled version of C:\Users\MKANNA~1\AppData\Local\Temp\ink_ext_XXXXXX.svgK9RB7Y @ 3500.00)
( unicorn.py --tab="help" --pen-up-angle=50 --pen-down-angle=30 --start-delay=150 --stop-delay=150 --xy-feedrate=3500 --z-feedrate=150 --z-height=0 --finished-height=0
G21 (metric ftw)
G90 (absolute mode)
G92 X0.00 Y0.00 Z0.00 (you are here)

(Polyline consisting of 51 segments.)
G1 X-85.19 Y120.47 F3500.00
M300 S30.00 (pen down)
G4 P150 (wait 150ms)
G1 X-85.12 Y-0.07 F3500.00
G1 X-85.05 Y-120.61 F3500.00
G1 X-84.99 Y-0.21 F3500.00
G1 X-84.92 Y120.19 F3500.00
G1 X0.21 Y120.26 F3500.00
G1 X85.33 Y120.33 F3500.00
G1 X0.07 Y120.40 F3500.00
G1 X-85.19 Y120.47 F3500.00
M300 S50.00 (pen up)
G4 P150 (wait 150ms)

(Polyline consisting of 51 segments.)
G1 X-84.09 Y119.36 F3500.00
M300 S30.00 (pen down)
G4 P150 (wait 150ms)
G1 X-84.09 Y-0.14 F3500.00
G1 X-84.09 Y-119.63 F3500.00
G1 X0.14 Y-119.63 F3500.00
G1 X84.36 Y-119.63 F3500.00
G1 X84.36 Y-0.14 F3500.00
G1 X84.36 Y119.36 F3500.00
G1 X0.14 Y119.36 F3500.00
G1 X-84.09 Y119.36 F3500.00
```

Fig 3.6: G-code file of ‘cnc trace 2’

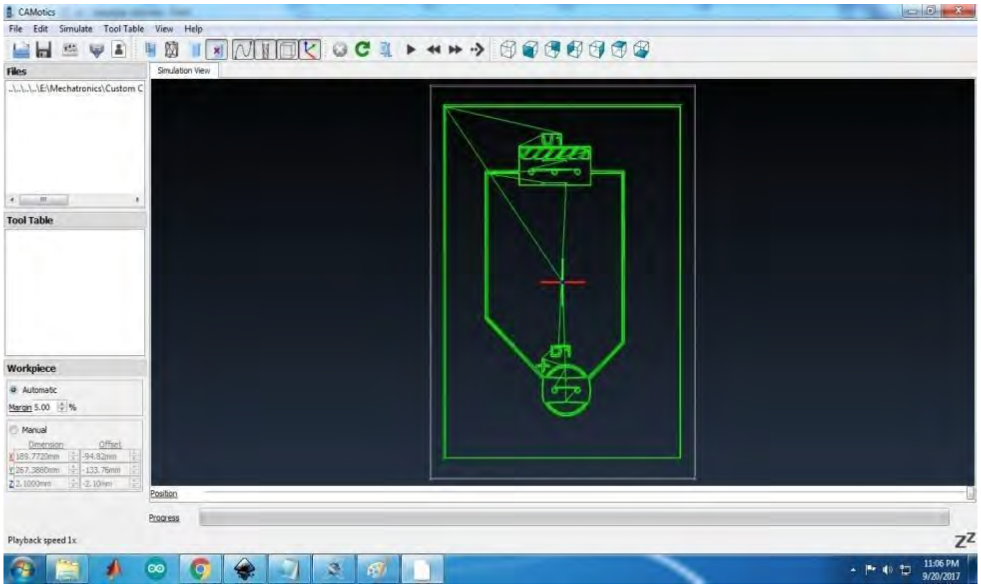
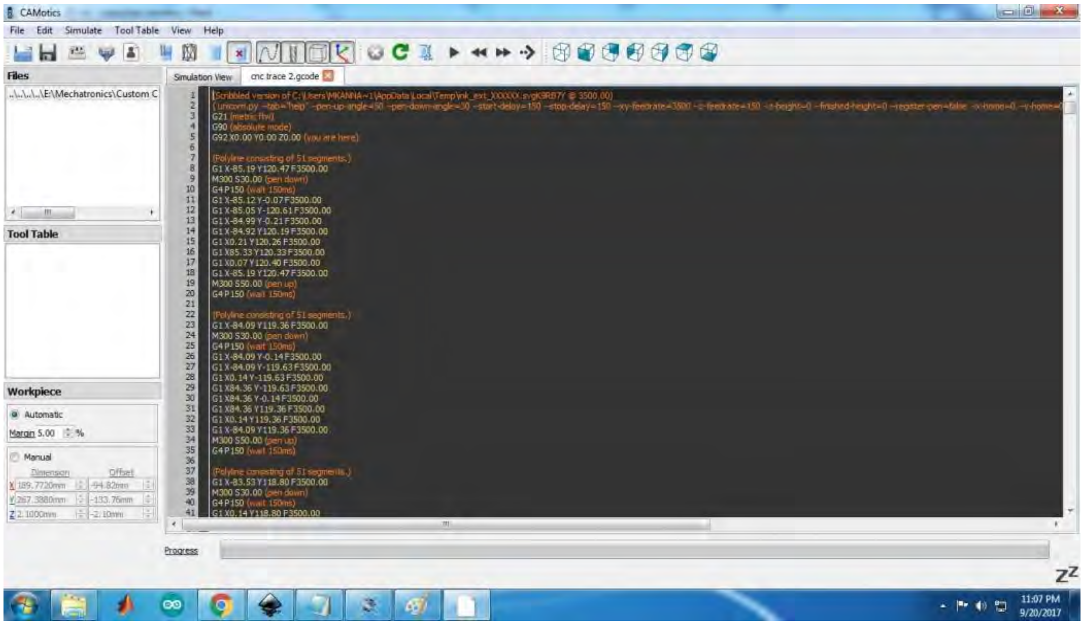


Fig. 3.7: CAMotics simulation

Finally, CAMotics was downloaded and opened, which is an open source simulator and CAM (Computer Aided Machining) software. The saved G-code file was opened on CAMotics as shown in Fig. 3.7. By clicking on the 'Play' button, the file was simulated to examine the CNC plot tool directory and accuracy. Double clicking on the file from ‘Files’ on the left panel opened the G-code file in another tab beside the Simulation view with all the G-code coordinates (Fig. 3.8). This software gave assurance whether the designs can print or not.



```
(Scribbled version of C:\Users\MKANNA~1\AppData\Local\Temp\ink_ext_XXXXXX.svgK9RB7Y @ 3500.00)
( unicorn.py --tab="help" --pen-up-angle=50 --pen-down-angle=30 --start-delay=150 --stop-delay=150 --xy-feedrate=3500 --z-feedrate=150 --z-height=0 --finished-height=0 --register-pen=false --x-home=0 --y-home=0
G21 (metric ftw)
G90 (absolute mode)
G92 X0.00 Y0.00 Z0.00 (you are here)

(Polyline consisting of 51 segments.)
G1 X-85.19 Y120.47 F3500.00
M300 S30.00 (pen down)
G4 P150 (wait 150ms)
G1 X-85.12 Y-0.07 F3500.00
G1 X-85.05 Y-120.61 F3500.00
G1 X-84.99 Y-0.21 F3500.00
G1 X-84.92 Y120.19 F3500.00
G1 X0.21 Y120.26 F3500.00
G1 X85.33 Y120.33 F3500.00
G1 X0.07 Y120.40 F3500.00
G1 X-85.19 Y120.47 F3500.00
M300 S50.00 (pen up)
G4 P150 (wait 150ms)

(Polyline consisting of 51 segments.)
G1 X-84.09 Y119.36 F3500.00
M300 S30.00 (pen down)
G4 P150 (wait 150ms)
G1 X-84.09 Y-0.14 F3500.00
G1 X-84.09 Y-119.63 F3500.00
G1 X0.14 Y-119.63 F3500.00
G1 X84.36 Y-119.63 F3500.00
G1 X84.36 Y-0.14 F3500.00
G1 X84.36 Y119.36 F3500.00
G1 X0.14 Y119.36 F3500.00
G1 X-84.09 Y119.36 F3500.00
M300 S50.00 (pen up)
G4 P150 (wait 150ms)

(Polyline consisting of 51 segments.)
G1 X-83.53 Y118.80 F3500.00
M300 S30.00 (pen down)
G4 P150 (wait 150ms)
G1 X0.14 Y118.80 F3500.00
```

Fig. 3.8: CAMotics G-code reading

3.4 HARDWARE SETUP

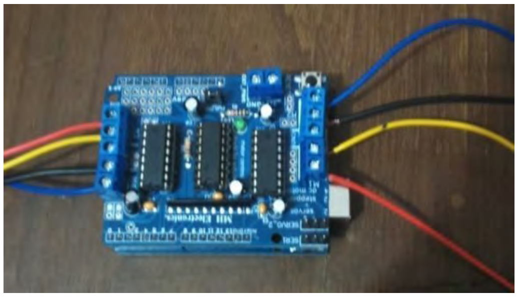


Fig. 3.9: Motor Driver on Arduino Uno

Fig. 3.10: Testing the DVD drives

Firstly, each of the DVD drivers was tested via a Stepper Motor Test. Each of the two DVD drives had a small Stepper motor built-in. Four wires were soldered to four terminals on the stepper motor of each DVD drive. Unlike a unipolar DC motor, a stepper motor has two wires determining one winding, meaning bipolar. There were two windings on each stepper, stretched by four wires. Setting a multimeter to continuity function, it was determined which two wires indicated one winding (Lesser value on multimeter means higher continuity). After determining the two windings of each stepper, the wires were connected to the Motor Shield. After downloading the Arduino code of the Stepper Motor Test, it was uploaded to the Arduino Uno board on which the Motor Shield was connected. Both the DVD drives were tested individually. Both DVDs operated, which means both drives passed the individual Stepper Motor Test.

Next, the Arduino CNC code was downloaded, the main code to operate the machine using both DVD drives in sync. This time, both DVD drives were connected to the Motor Shield (Fig. 3.9). The CNC code was uploaded to the Arduino Uno and GCTRL code file opened in Processing Software, was used to run the code for both drives using a sample G-code file. Both drives operated, which meant that both drives will work if a design is uploaded.

Fig 3.11: Building Prototype

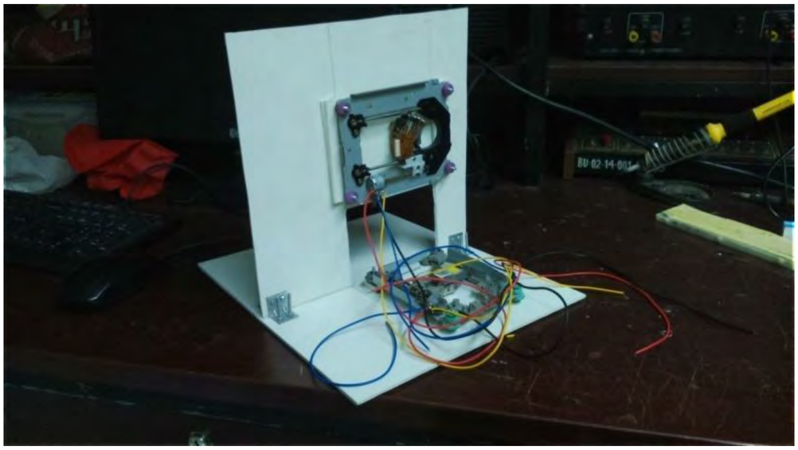


Fig 3.12: Prototype design

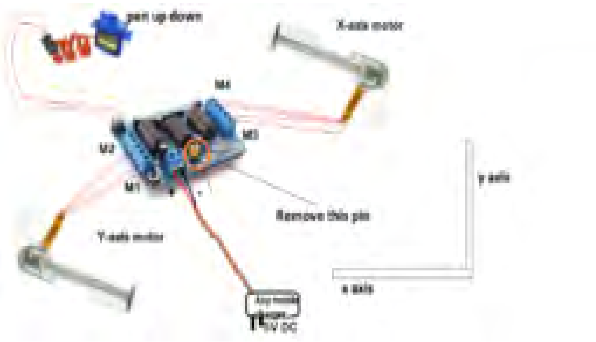
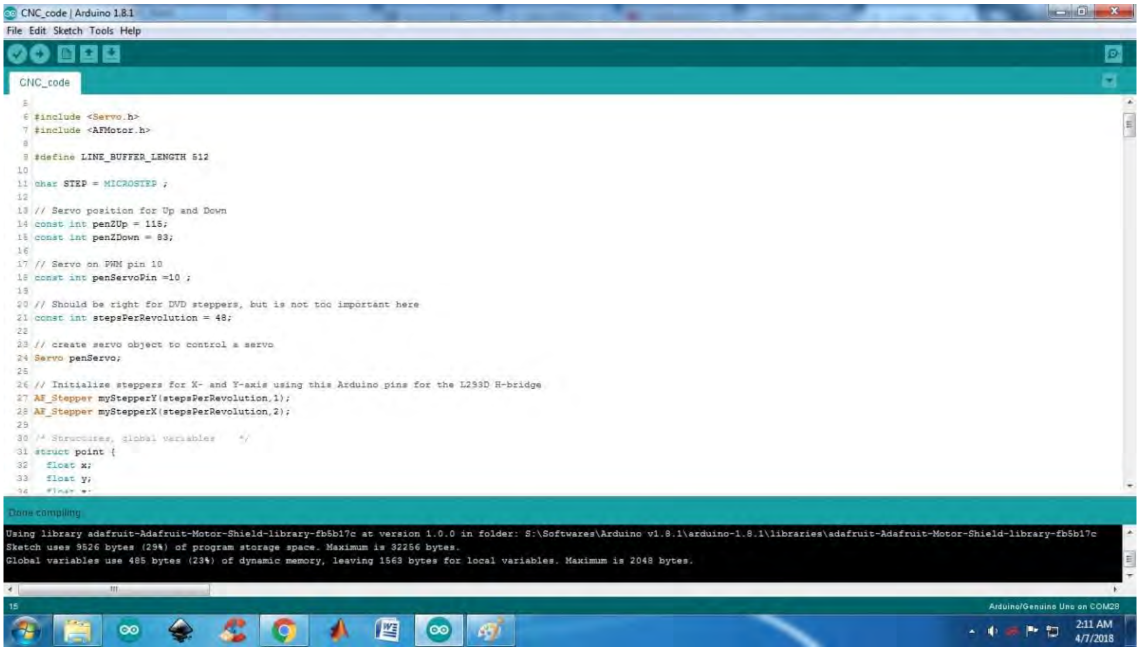


Fig. 3.13: Circuit connection

The prototype was later given a shape by cutting a large 3mm PVC board to four squares. The two DVD drivers will work as the two axes for a 2D CNC plotting machine. It was planned to set one DVD drive at the base of a PVC board as the X-axis and the other drive set as Y-axis on a vertically mounted PVC board (Fig. 3.12). No use of a Servo motor for Z-axis because for the moment, only the X and Y axes were needed to be tested before proceeding.

3.5 WORKING METHOD

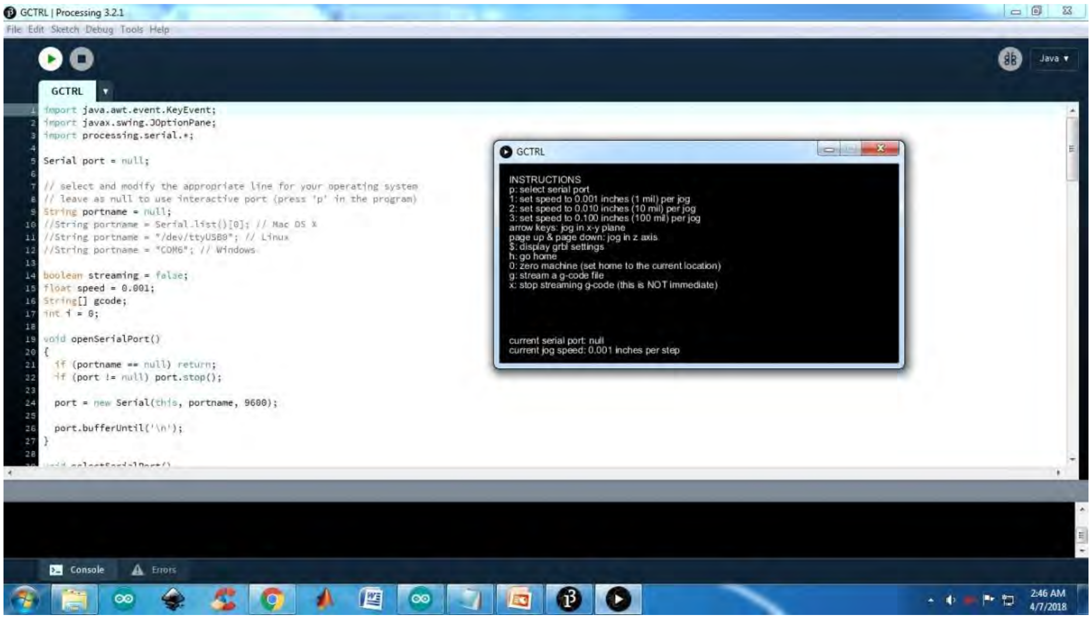


```

#include <Servo.h>
#include <AFMotor.h>

#define LINE_BUFFER_LENGTH 512

char STEP = MICROSTEP ;

// Servo position for Up and Down
const int penZUp = 115;
const int penZDown = 83;

// Servo on PWM pin 10
const int penServoPin =10 ;

// Should be right for DVD steppers, but is not too important here
const int stepsPerRevolution = 48;

// create servo object to control a servo
Servo penServo;

// Initialize steppers for X- and Y-axis using this Arduino pins for the L293D H-bridge
AF_Stepper myStepperY(stepsPerRevolution,1);
AF_Stepper myStepperX(stepsPerRevolution,2);

/* Structures, global variables    */
struct point {
  float x;
  float y;
  float z;
```

```
Using library adafruit-Adafruit-Motor-Shield-library-fb5b17c at version 1.0.0 in folder: S:\Softwares\Arduino v1.8.1\arduino-1.8.1\libraries\adafruit-Adafruit-Motor-Shield-library-fb5b17c
Sketch uses 9526 bytes (29%) of program storage space. Maximum is 32256 bytes.
Global variables use 485 bytes (23%) of dynamic memory, leaving 1563 bytes for local variables. Maximum is 2048 bytes.
```

Fig. 3.14: Arduino CNC code for Stepper Motors

The logic of this project’s workflow is not that difficult to understand as it appears to be. Fig. 3.14 shows the Arduino code or algorithm for a CNC machine having Stepper motors. This code is compiled and uploaded onto the Arduino Uno Board. Next Processing IDE software was opened and the GCTRL code file was opened on Processing (Fig. 3.15).



```
import java.awt.event.KeyEvent;
import javax.swing.JOptionPane;
import processing.serial.*;

Serial port = null;

// select and modify the appropriate line for your operating system
// leave as null to use interactive port (press 'p' in the program)
String portname = null;
//String portname = Serial.list()[0]; // Mac OS X
//String portname = "/dev/ttyUSB0"; // Linux
//String portname = "COM6"; // Windows

boolean streaming = false;
float speed = 0.001;
String[] gcode;
int i = 0;

void openSerialPort()
{
  if (portname == null) return;
  if (port != null) port.stop();

  port = new Serial(this, portname, 9600);

  port.bufferUntil('\n');
}

void selectSerialPort()
```

```
GCTRL
INSTRUCTIONS
p: select serial port
1: set speed to 0.001 inches (1 mil) per jog
2: set speed to 0.010 inches (10 mil) per jog
3: set speed to 0.100 inches (100 mil) per jog
arrow keys: jog in x-y plane
page up & page down: jog in z axis
$: display grbl settings
h: go home
0: zero machine (set home to the current location)
g: stream a g-code file
x: stop streaming g-code (this is NOT immediate)



current serial port: null
current jog speed: 0.001 inches per step
```

Fig. 3.15: GCTRL code from Processing

When the GCTRL code was run, it opens a tab to select the Arduino COM port (port on which Arduino is connected) and then stream a G-code file. A sample G-code file is selected and then this code executes the G-code file, sending commands to the Arduino, thus activating the machine to print. The print follows code based on machine learning format. The intended workflow or working method is summarized in the following flow chart:

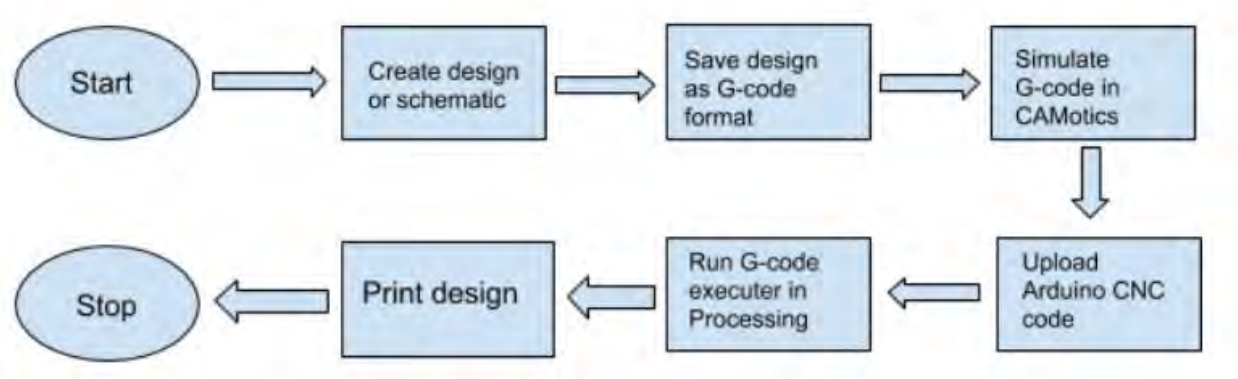

Start
Create design or schematic
Save design as G-code format
Simulate G-code in CAMotics
Upload Arduino CNC code
Run G-code executer in Processing
Print design
Stop

Fig 3.16: Flowchart of CNC operation

CHAPTER 4: RESULT PRESENTATION, ANALYSIS & CONCLUSION (FOR PROTOTYPE)

4.1 RESULT

The result did not work out the way it was expected, at least for operating the CNC axes. The X-axis DVD worked well but the Y-axis DVD was not working due to load. After multiple attempts in trying to solve the problem, the capacitor C7 (max 16 V) near the LED1 of the Motor Shield exploded when voltage supply accidently exceeded to 17.1 V, so unfortunately, the full system for our prototype could not be fulfilled. Still, it gave a better overview of how a CNC machine requires to be adjusted and some test runs were recorded by camera to verify all the trial and error.

4.2 ANALYSIS

There were several problems dealt with. From the hardware segment, the Y-axis stepper motor did not move at all due to load since the vertical PVC board was not mounted with enough support, in fact, not rigid enough. From the software segment, Inkscape had trouble reading complex designs while saving in G-code format and was showing error most of the time, so that too needed to be noted.

CHAPTER 5: BUILDING THE ACTUAL MACHINE

5.1 HARDWARE APPARATUS

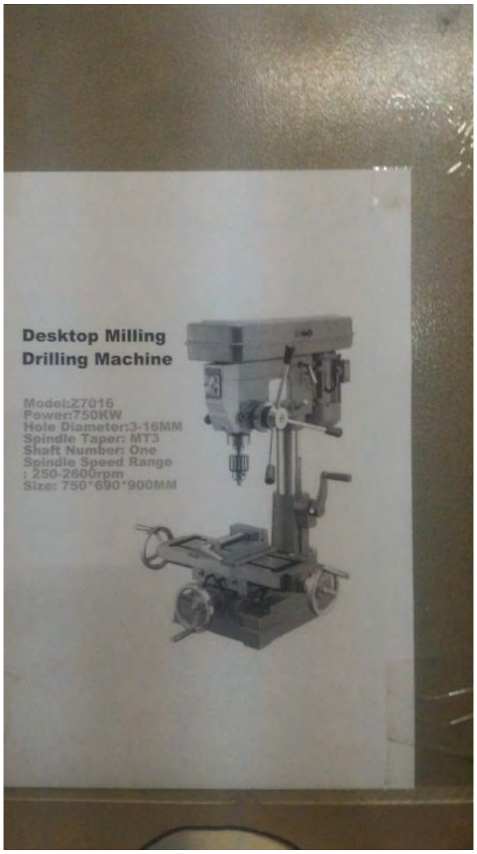


Fig. 5.1: Estimated design

Fig. 5.2: Types of Nuts & Bolts

Fig. 5.3: Threaded Nut and Bolt

During Fall 2017, plans were made to build the respective hardware for the final design. The estimated design is shown in Fig 5.1, whereas Fig 5.2 and 5.3 shows the type of materials that was bought by personal funding. One key element that was on the shop was the threaded Stainless Steel (SS) Rod of 72 inches in length and 10mm in diameter with six threaded nuts of 10mm diameter, altogether costing up to Tk. 390. To provide the CNC machine a skeletal frame, the rod was cut into three pieces- 32 inches for X-axis, 32 inches for Y-axis and 8 inches for Z-axis; estimated. Such measurements were taken so that there will be room for any mechanical modification. The final design came to the fore in February 2018 which is shown in Fig 5.4. Here is a list of all the components gathered for the final design using personal funding considering estimated costs.

Components	**Quantity**	**Unit Price (tk.)**	**Subtotal (tk.)**
CNC Metal Structure	1	12,000	12000
72" SS Rod with nuts & bolts	1	390	390
Arduino Mega 2560 R3	1	850	850
BTS7960 Motor Driver	2	1200	2400
SG90 Servo Motor	1	160	160
Li-po Batteries 11.1V 1500mAh	2	1500	3000
PVC Board 5mm	2	150	300
HC-SR04 Sonar Sensor	2	105	210
HC-05 Bluetooth Module	1	270	270
XT60 Connector Pairs	4	70	280
T Plug M-F Connector Pairs	2	35	70
4 pin F-F Single row Flex Cable	4	8	32
4 pin F-F Dual row Flex Cable	2	15	30
CNC Control PCB	1	455	455
M-M Jumper wire (40 pcs)	1	55	55
40 pin single row M header 2.54mm	1	5	5
40 pin double row M header 2.54mm	1	25	25
40 pin single row F header 2.54mm	1	5	5
Total (approx.)			**20537/=**

Table 5.1: Cost of components (for final design)

5.2 HARDWARE SETUP

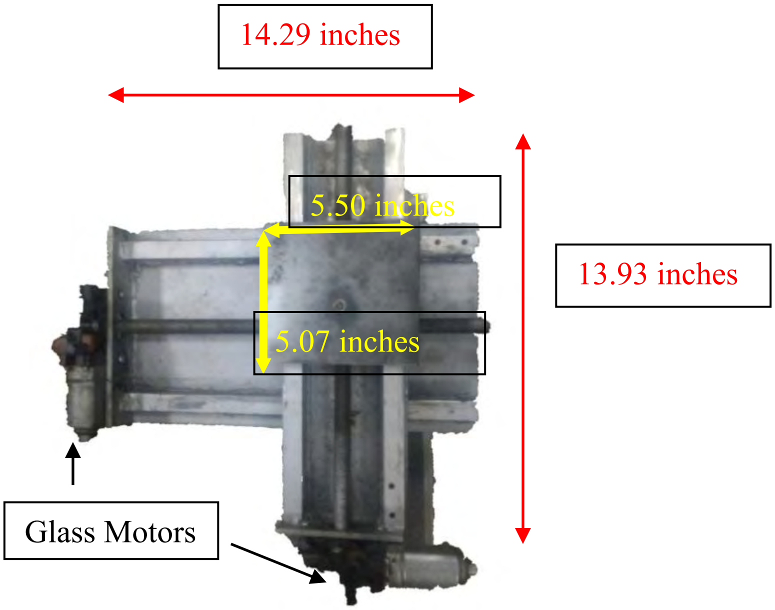


Fig. 5.4: CNC Machine Final design

The final design had a solid structure made out of Aluminum based alloy metal with the Y-axis structure on top of the movable X-axis. Both ends of the rods were equipped with Glass motors; the same motors which are used in the windows of a car. Each motor had a specification of 12V voltage and about 1.6A of current; unipolar, meaning two wires, so that they can be used as an equivalent to DC motors as opposed to the bipolar stepper motors with four wires; plus a print platform where we were to obtain the output of our design or schematic created in software. After doing some adjustments, the length of the X-axis rod is 14.29 inches (362.96 mm) and that of the Y-axis rod is 13.93 inches (353.82 mm). The print platform was 5.50 inches (139 mm) across X-axis and 5.07 inches (129 mm) across Y-axis. The entire structure weighed about 19.87 kgs. The measurements were converted to millimeter (mm) because using this as a reference unit in the Arduino codes provide sheer accuracy for the modules and sensors to compute.

The innovation behind this project is to prove that Glass motors, having good torque, can work as an alternative to stepper motors [2], since stepper motors are not available everywhere when it comes to developing and under-developed countries. The description of electronic components are in reference index [3, 4, 5, 15, 16, 17].

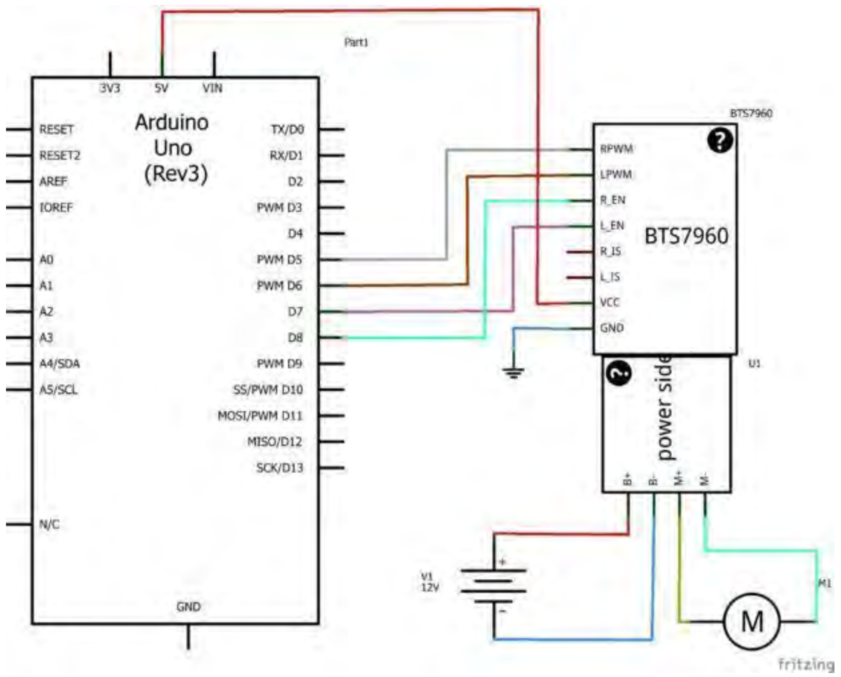
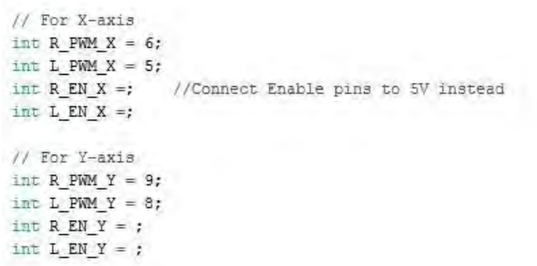


Fig. 5.5: Circuit Connection of BTS7960 motor driver with respect to Arduino

Understanding the circuit connection of BTS7960 motor driver with respect to an Arduino was a tricky one. Although, in Fig. 5.5, the connection shows with respect to an Arduino Uno, this connection mainly applies to any Arduino version, especially for the Mega. The BTS7960 motor driver has two blocks on the rear power side, one for a motor and another for a battery; on the pinout side two Enable pins, two PWM pins, a VCC and GND pin. The Enable pins may be connected to the digital I/O pins of Arduino, but in this case, they were connected in common with the VCC to be generated with 5V. The PWM pins of BTS7960 go to Arduino's PWM pins; Pin 6, 5 for the X-axis driver and pins 9, 8 for the Y-axis driver. Fig. 5.6 shows the pinout initialization during Arduino coding of the motor driver. Fig. 5.7 shows the XT and T-plug connectors soldered on a Li-po battery in sequence; the T-plug and then XT connectors were then strengthened with heat shrink tubes to fortify the connections.



```
// For X-axis
int R_PWM_X = 6;
int L_PWM_X = 5;
int R_EN_X =;     //Connect Enable pins to 5V instead
int L_EN_X =;

// For Y-axis
int R_PWM_Y = 9;
int L_PWM_Y = 8;
int R_EN_Y = ;
int L_EN_Y = ;
```

Fig. 5.6: BTS7960 pinout code

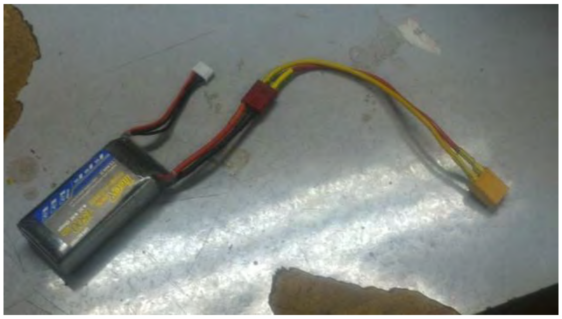


Fig. 5.7: Li-po battery with connectors

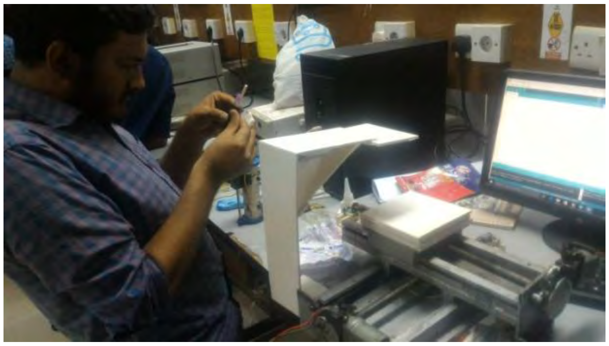


Fig. 5.8: Constructing Z-axis

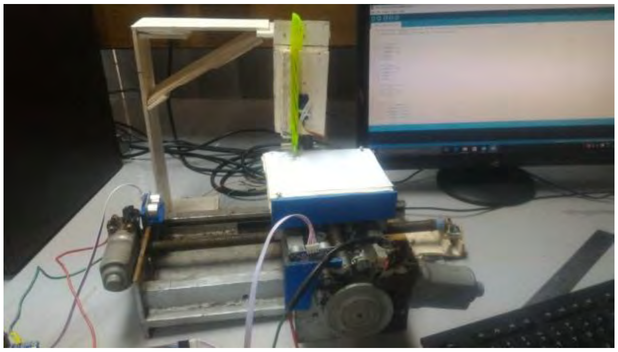


Fig. 5.9: X-Y-Z axis structure

A Z-axis was created cutting out PVC boards (Fig. 5.8) with a Servo motor and a Pen fixated (Fig. 5.9). The custom Z-axis was not rigid but it was good enough to perform some experimental printouts. In order to bring simplicity among the connectivity of the modules with respect to the Arduino Mega and avoid the use of excessive jumper wires and wires from Ethernet cables, a custom CNC Control PCB was designed on software and printed from an electronics shop. After obtaining the Control PCB, the respective headers of the modules and sensors were soldered onto the PCB (Fig. 5.10). Additional footprints of components were used in the PCB design in case of adding any backup modules. A further description about the Control PCB design in software is discussed in Section 5.3.

Fig. 5.10: Soldering headers

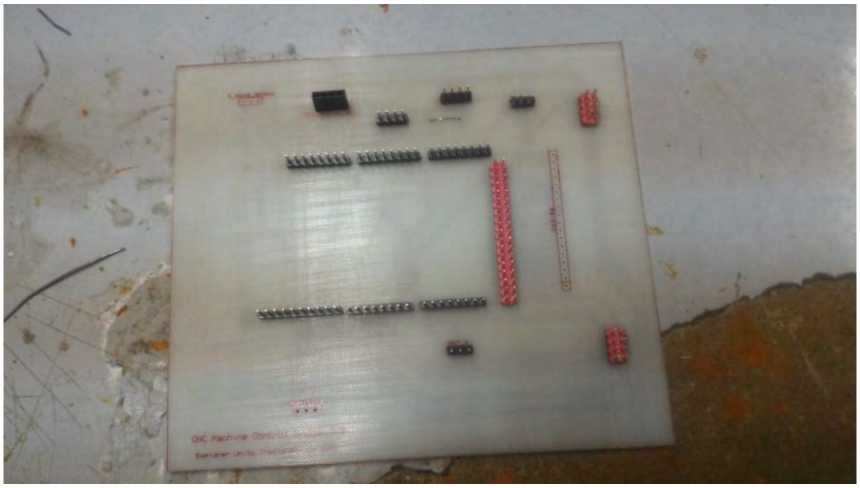


Fig. 5.11: Complete PCB board

5.3 CONTROL SOFTWARES

Reviewing the list of Control Softwares from Section 3.2- Proteus, Inkscape, CAMotics, Arduino and Processing are to be utilized. The design of the Control PCB is described as follows.

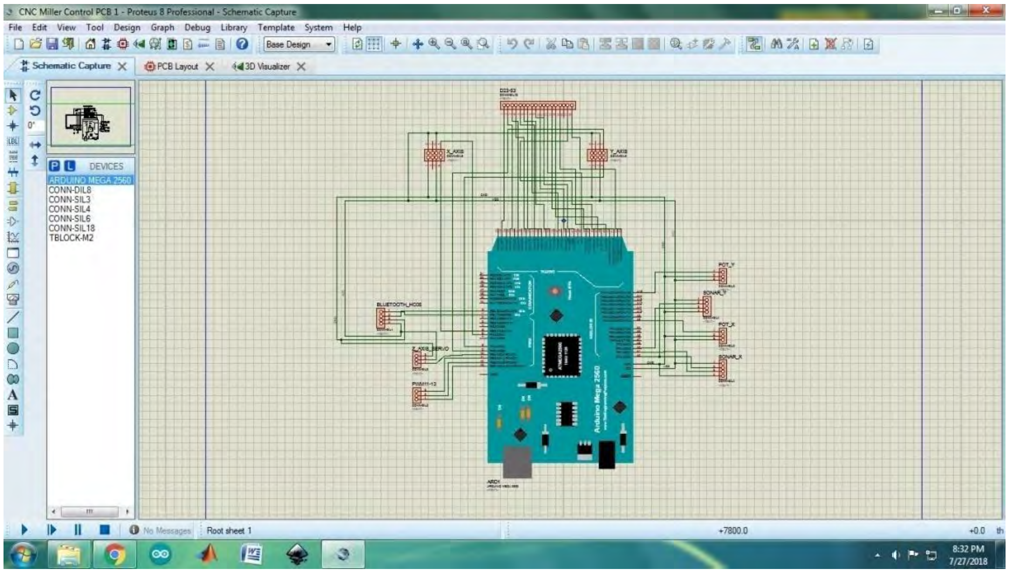


Fig. 5.12: CNC Control PCB circuit on Proteus 8.0 ISIS

The CNC Control PCB was designed on Proteus Software version 8.0. The circuit on Fig. 5.12 shows the ISIS schematic. But designing this circuit came with some difficulties. There were plenty of Arduino library files available on the internet like for example, TheEngineeringProjects.com [18] website had an array of five different Arduino libraries, but none of the libraries had a proper PCB footprint for the Proteus ARES tab, the tab where the components are placed onto the PCB layer and the track routes are set manually. After rigorous searching, an Arduino PCB package was obtained from a website named nhbelectronic.blogspot.com; which was open-source for software developers. The routed design on ARES tab is shown in Fig. 5.13, prioritized to be a single layer PCB.

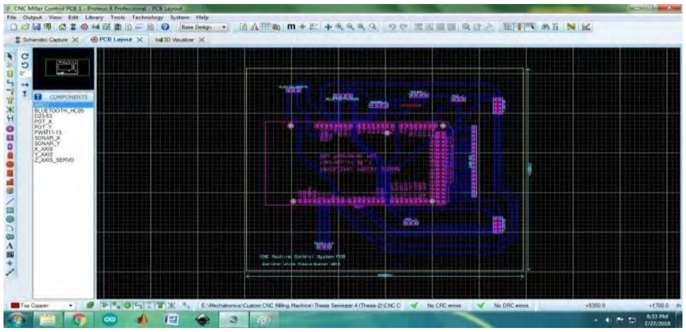


Fig. 5.13: Circuit Design on Proteus ARES

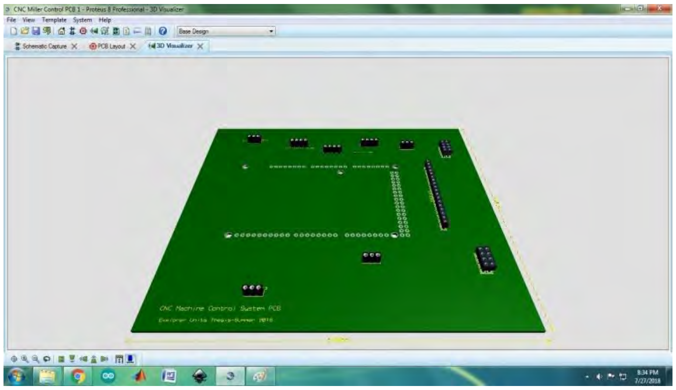

CNC Miller Control PCB 1 - Proteus 8 Professional - 3D Visualizer
File View Template System Help
Base Design
Schematic Capture
PCB Layout
3D Visualizer
CNC Machine Control System PCB
Ewclpner Units Thesis-Summer 2018
8:34 PM
7/27/2018

Fig. 5.14: PCB 3D Top View

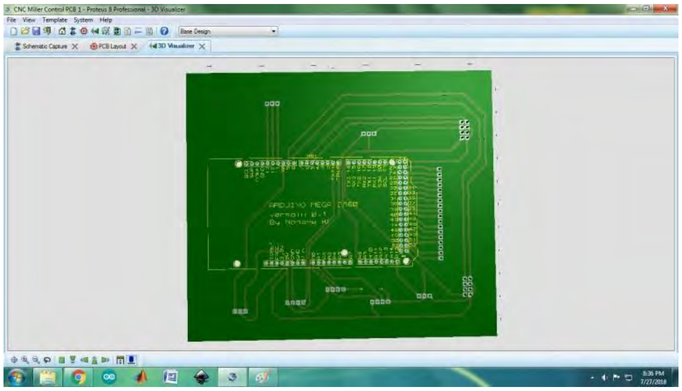

CNC Miller Control PCB 1 - Proteus 8 Professional - 3D Visualizer
File View Template System Help
Base Design
Schematic Capture
PCB Layout
3D Visualizer
ARDUINO MEGA 2560
8:36 PM
7/27/2018

Fig. 5.15: PCB 3D Bottom View

The PCB connections on ARES were routed using Bottom Copper (blue colored route) and vias; any routes requiring to be set on top layer, were routed using Top Copper layer option (red colored route). The dimensions of the PCB were measured from the ARES tab, which showed 5.90 inches for length and 5.10 inches for breadth. The PCB design was later rendered in the Proteus 3D Visualizer, to display a 3D view of both the top (Fig. 5.14) and bottom (Fig. 5.15) layers; an overview of how the final product will appear to be like. After the PCB was printed from the shop and delivered, the electronic components were all connected to the PCB using the connectors and they were tested out for trial and error. A Physical view of the PCB connectivity is shown in Fig. 5.16.

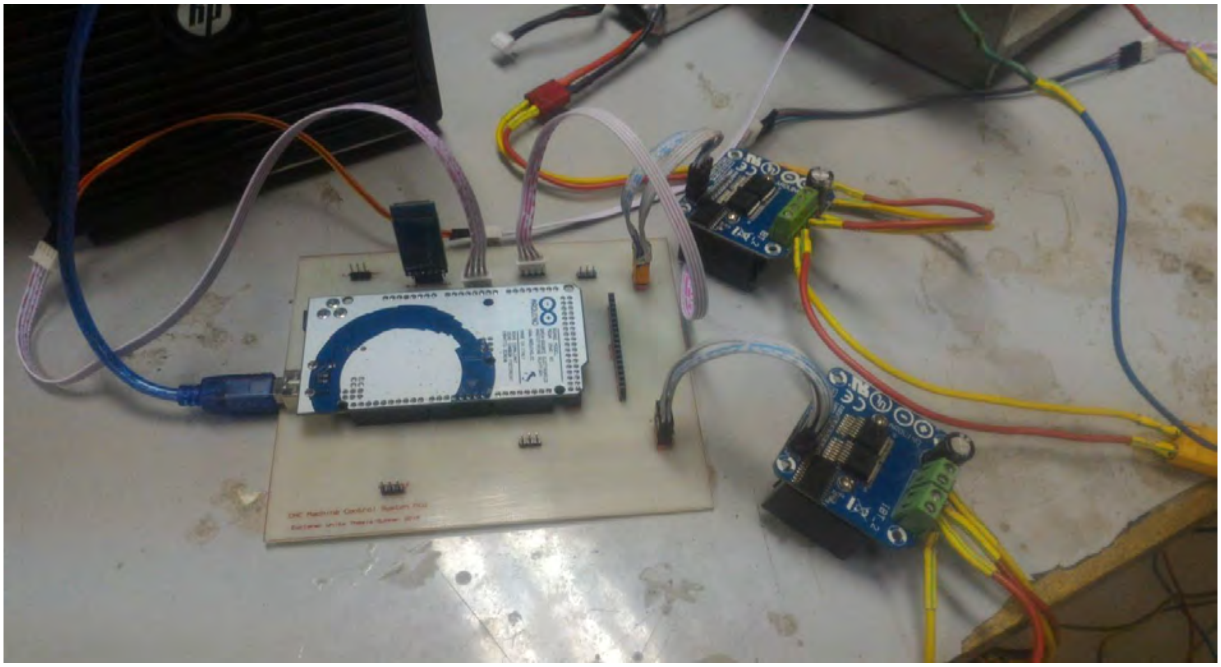
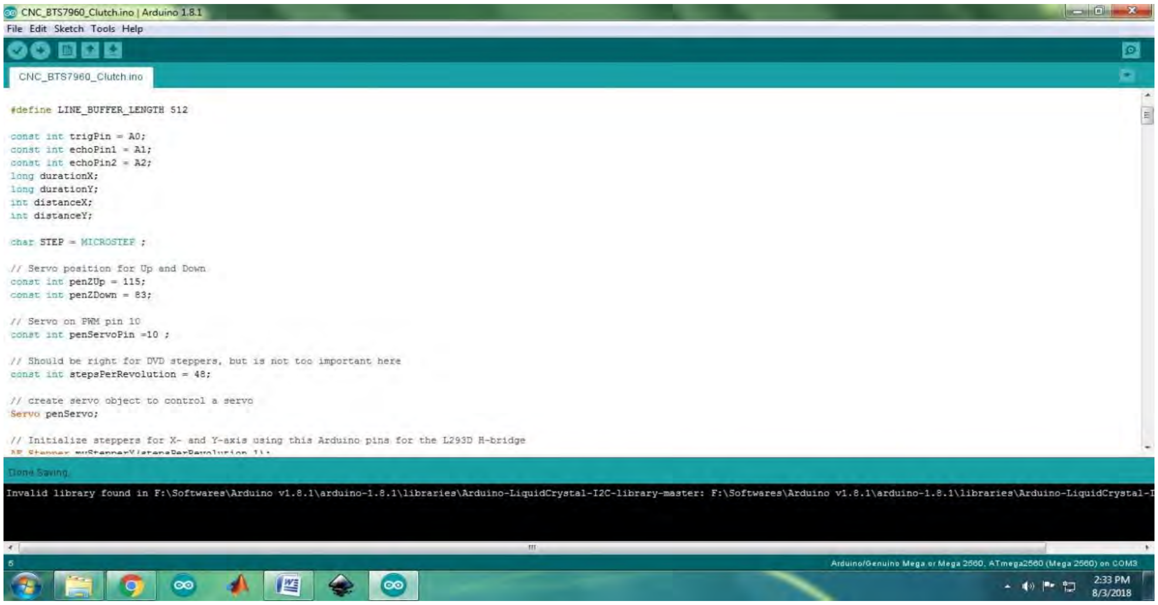


Fig. 5.16: Connections of modules with the PCB

5.4 SOFTWARE SETUP

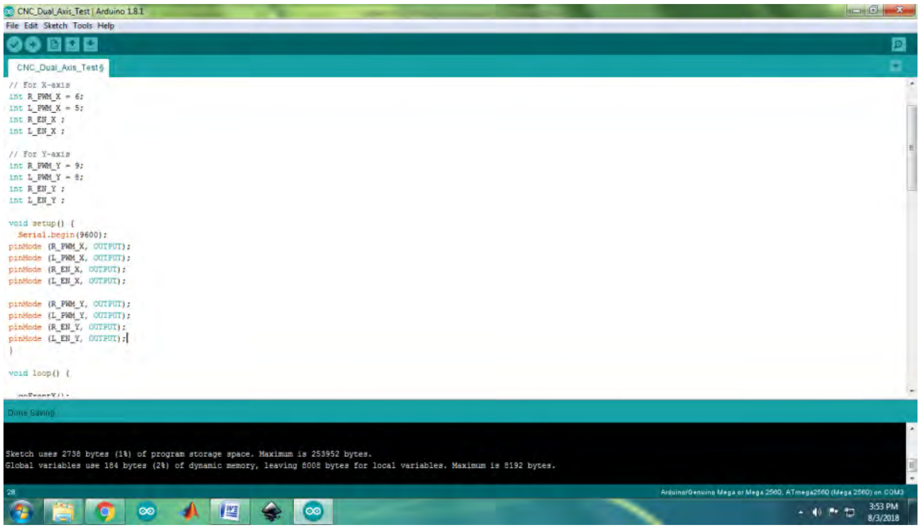

CNC_BTS7960_Clutch.ino | Arduino 1.8.1
File Edit Sketch Tools Help
CNC_BTS7960_Clutch.ino
Done Saving.
6
Arduino/Genuino Mega or Mega 2560, ATmega2560 (Mega 2560) on COM3
2:33 PM
8/3/2018

```
#define LINE_BUFFER_LENGTH 512

const int trigPin = A0;
const int echoPin1 = A1;
const int echoPin2 = A2;
long durationX;
long durationY;
int distanceX;
int distanceY;

char STEP = MICROSTEP ;

// Servo position for Up and Down
const int penZUp = 115;
const int penZDown = 83;

// Servo on PWM pin 10
const int penServoPin =10 ;

// Should be right for DVD steppers, but is not too important here
const int stepsPerRevolution = 48;

// create servo object to control a servo
Servo penServo;

// Initialize steppers for X- and Y-axis using this Arduino pins for the L293D H-bridge
AF_Stepper myStepperY(stepsPerRevolution,1);
```

```
Invalid library found in F:\Softwares\Arduino v1.8.1\arduino-1.8.1\libraries\Arduino-LiquidCrystal-I2C-library-master: F:\Softwares\Arduino v1.8.1\arduino-1.8.1\libraries\Arduino-LiquidCrystal-I
```

Fig. 5.17: Arduino CNC code for BTS7960 motor drivers

An Arduino CNC machine code (Fig. 5.17) was created using the Arduino IDE software. The code needed to be compiled to operate the BTS7960 motor drivers and the Sonar Sensor feedback system in synchronized manner. The code logic is, after uploading a design, the feedback system will help the motor driver determine the coordinates for movement of the axes. Before the machine was tried and tested, an experimental Arduino code was created for a dual axis motor test (Fig. 5.18), a test run to observe how the glass motors operated with the BTS7960 motor drivers.

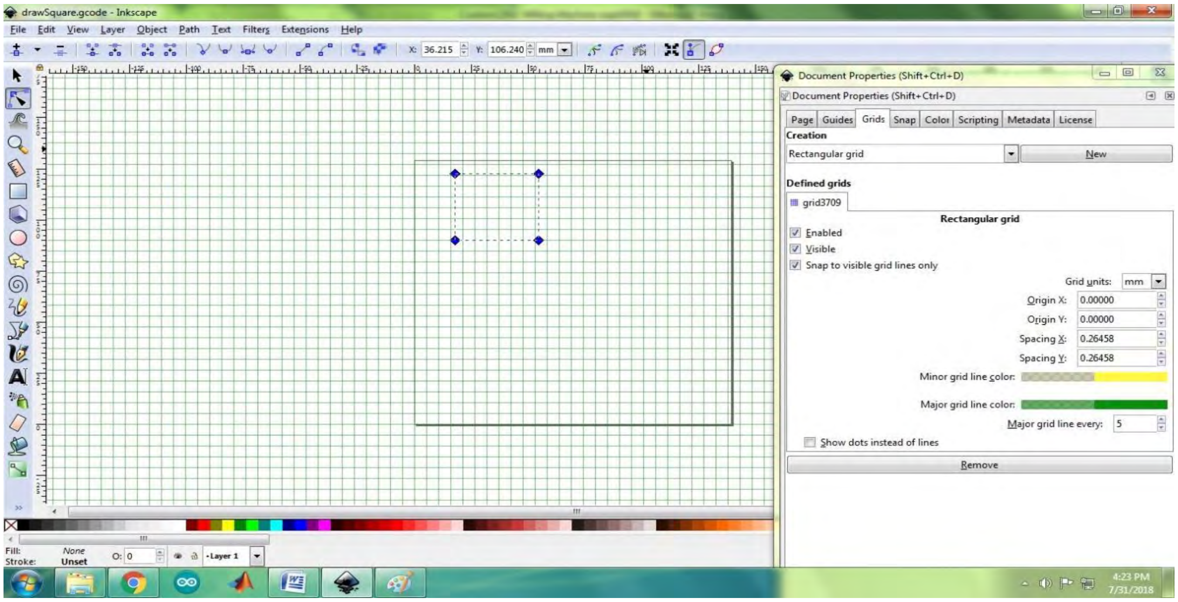

CNC_Dual_Axis_Test | Arduino 1.8.1
File Edit Sketch Tools Help
CNC_Dual_Axis_Test §
Done Saving.
28
Arduino/Genuino Mega or Mega 2560, ATmega2560 (Mega 2560) on COM3
3:53 PM
8/3/2018

```
// For X-axis
int R_PWM_X = 6;
int L_PWM_X = 5;
int R_EN_X ;
int L_EN_X ;

// For Y-axis
int R_PWM_Y = 9;
int L_PWM_Y = 8;
int R_EN_Y ;
int L_EN_Y ;

void setup() {
  Serial.begin(9600);
pinMode (R_PWM_X, OUTPUT);
pinMode (L_PWM_X, OUTPUT);
pinMode (R_EN_X, OUTPUT);
pinMode (L_EN_X, OUTPUT);

pinMode (R_PWM_Y, OUTPUT);
pinMode (L_PWM_Y, OUTPUT);
pinMode (R_EN_Y, OUTPUT);
pinMode (L_EN_Y, OUTPUT);
}

void loop() {

  goFrontY();
```

```
Sketch uses 2738 bytes (1%) of program storage space. Maximum is 253952 bytes.
Global variables use 184 bytes (2%) of dynamic memory, leaving 8008 bytes for local variables. Maximum is 8192 bytes.
```

Fig. 5.18: Test code for dual axis motor drivers

5.5 WORKING METHOD


drawSquare.gcode - Inkscape
File Edit View Layer Object Path Text Filters Extensions Help
Document Properties (Shift+Ctrl+D)
Page Guides Grids Snap Color Scripting Metadata License
New
Remove
Fill: None
Stroke: Unset
-Layer 1
4:23 PM
7/31/2018

X: 36.215 Y: 106.240 mm

Creation

Rectangular grid

Defined grids

grid3709

Rectangular grid

☑ Enabled
☑ Visible
☑ Snap to visible grid lines only

Grid units: mm
Origin X: 0.00000
Origin Y: 0.00000
Spacing X: 0.26458
Spacing Y: 0.26458
Minor grid line color:
Major grid line color:
Major grid line every: 5
☐ Show dots instead of lines

Fig. 5.19: Inkscape Grid

When building the prototype, initial plans were to gather components, interface with softwares and then test out a machine with stepper motors. There was a change in approach when it came to the final design with glass motors; choice of hardware was crucial and all the softwares and logistics were configured based on those components. But that is exactly why the prototype was needed; it laid a foundation for the choice of softwares to interface with. The work operation is pretty much the same as shown in the flowchart in Fig. 3.16; however the software operations will be slightly different for this machine. Inkscape was opened and from the Document Properties>Page option, the size of the center page was adjusted with respect to the dimensions of the print platform, i.e., 5.50in x 5.07in; then a small square was drawn on the page as a sample design to be printed. Again from the Document Properties>Grid option, a grid (in green color) was setup in the layout and the grid units were set in millimeter (mm); because that is the unit set in the feedback system portion of the Arduino code. Key 'F2' was pressed to edit path by nodes for the selected design for which the nodes were highlighted in blue as shown in Fig. 5.19. Then, the design with filename 'drawSquare' was saved as .gcode extension file under the MakerBot Unicorn gcode format type; which generated a notepad file with the design coordinates. The design file on Inkscape can also be saved as .svg format, Inkscape's standard file format, for backup use and modifications.

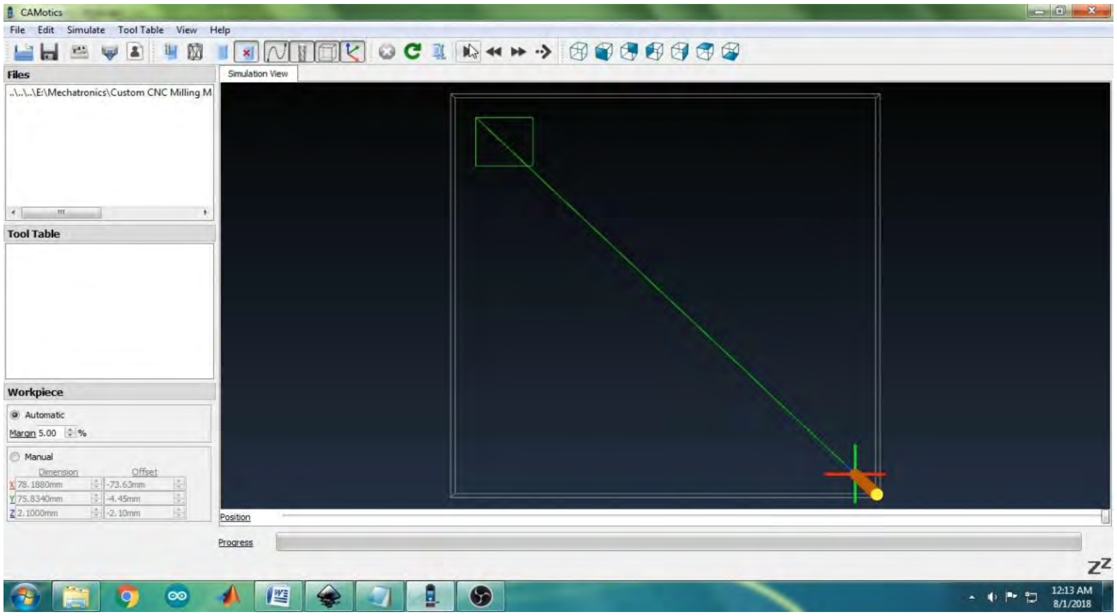
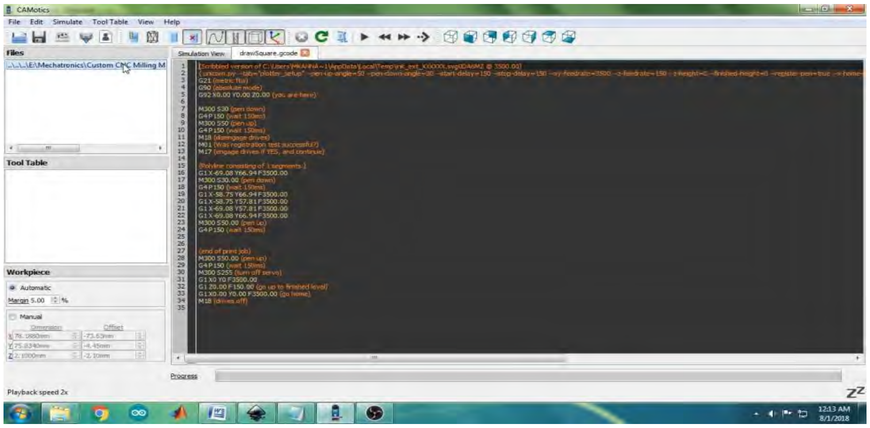


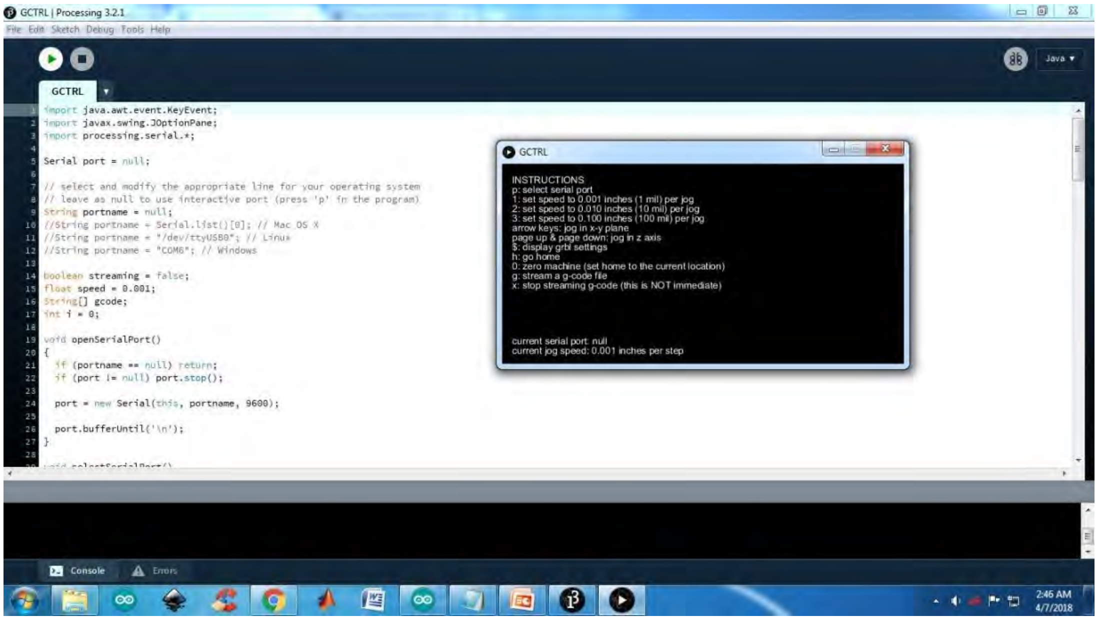
Fig. 5.20: CAMotics simulation test

Next step is simulation before printing the design. After opening the CAMotics software, the 'drawSquare.gcode' file was opened from File>Open Project option with the Dialog settings kept default and units set in millimeter (mm). The simulation is run by pressing the play button; tool animation start-stop button as pointed by the cursor in Fig. 5.20; to observe the direction of the tool path (green line) of the tool (yellow stick) in the software. The tool resembles the Z-axis pen in the machine's case. The simulation is not always accurate, but the Z-axis position can be readjusted in accordance to the main CNC machine's suitability. By clicking on the file directory on the 'Files' tab located on the left, the design gcode file opens with all the coordinates that were on the notepad file (Fig. 5.21). If the simulation test is satisfactory, then the work operation can proceed to the next step.



```
(Scribbled version of C:\Users\MKAHNA~1\AppData\Local\Temp\ink_ext_XXXXXX.svg0DA6MZ @ 3500.00)
( unicorn.py --tab="plotter_setup" --pen-up-angle=50 --pen-down-angle=30 --start-delay=150 --stop-delay=150 --xy-feedrate=3500 --z-feedrate=150 --z-height=0 --finished-height=0 --register-pen=true --x-home=
G21 (metric ftw)
G90 (absolute mode)
G92 X0.00 Y0.00 Z0.00 (you are here)

M300 S30 (pen down)
G4 P150 (wait 150ms)
M300 S50 (pen up)
G4 P150 (wait 150ms)
M18 (disengage drives)
M01 (Was registration test successful?)
M17 (engage drives if YES, and continue)

(Polyline consisting of 1 segments.)
G1 X-69.08 Y66.94 F3500.00
M300 S30.00 (pen down)
G4 P150 (wait 150ms)
G1 X-58.75 Y66.94 F3500.00
G1 X-58.75 Y57.81 F3500.00
G1 X-69.08 Y57.81 F3500.00
G1 X-69.08 Y66.94 F3500.00
M300 S50.00 (pen up)
G4 P150 (wait 150ms)


(end of print job)
M300 S50.00 (pen up)
G4 P150 (wait 150ms)
M300 S255 (turn off servo)
G1 X0 Y0 F3500.00
G1 Z0.00 F150.00 (go up to finished level)
G1 X0.00 Y0.00 F3500.00 (go home)
M18 (drives off)
```

Fig. 5.21: CAMotics gcode File tab



```
import java.awt.event.KeyEvent;
import javax.swing.JOptionPane;
import processing.serial.*;

Serial port = null;

// select and modify the appropriate line for your operating system
// leave as null to use interactive port (press 'p' in the program)
String portname = null;
//String portname = Serial.list()[0]; // Mac OS X
//String portname = "/dev/ttyUSB0"; // Linux
//String portname = "COM6"; // Windows

boolean streaming = false;
float speed = 0.001;
String[] gcode;
int i = 0;

void openSerialPort()
{
  if (portname == null) return;
  if (port != null) port.stop();

  port = new Serial(this, portname, 9600);

  port.bufferUntil('\n');
}
```

```
GCTRL

INSTRUCTIONS
p: select serial port
1: set speed to 0.001 inches (1 mil) per jog
2: set speed to 0.010 inches (10 mil) per jog
3: set speed to 0.100 inches (100 mil) per jog
arrow keys: jog in x-y plane
page up & page down: jog in z axis
$: display grbl settings
h: go home
0: zero machine (set home to the current location)
g: stream a g-code file
x: stop streaming g-code (this is NOT immediate)



current serial port: null
current jog speed: 0.001 inches per step
```

Fig. 5.22: GCODE file

The next and final step of the work operation is same as that of the prototype design. First the modified Arduino CNC code for BTS7960 motor drivers (Fig. 5.17) is uploaded onto the Arduino Mega plugged onto the PCB board and then the unchanged GCODE file opened from Processing software is clicked to run (Fig. 5.22). COM port is selected, then Key 'G' is pressed to stream the option for selecting a .gcode file, 'drawSquare.gcode' in this case and after pressing 'OK', the machine is set to run the print. A complete picture of the entire project is shown in Fig. 5.23.

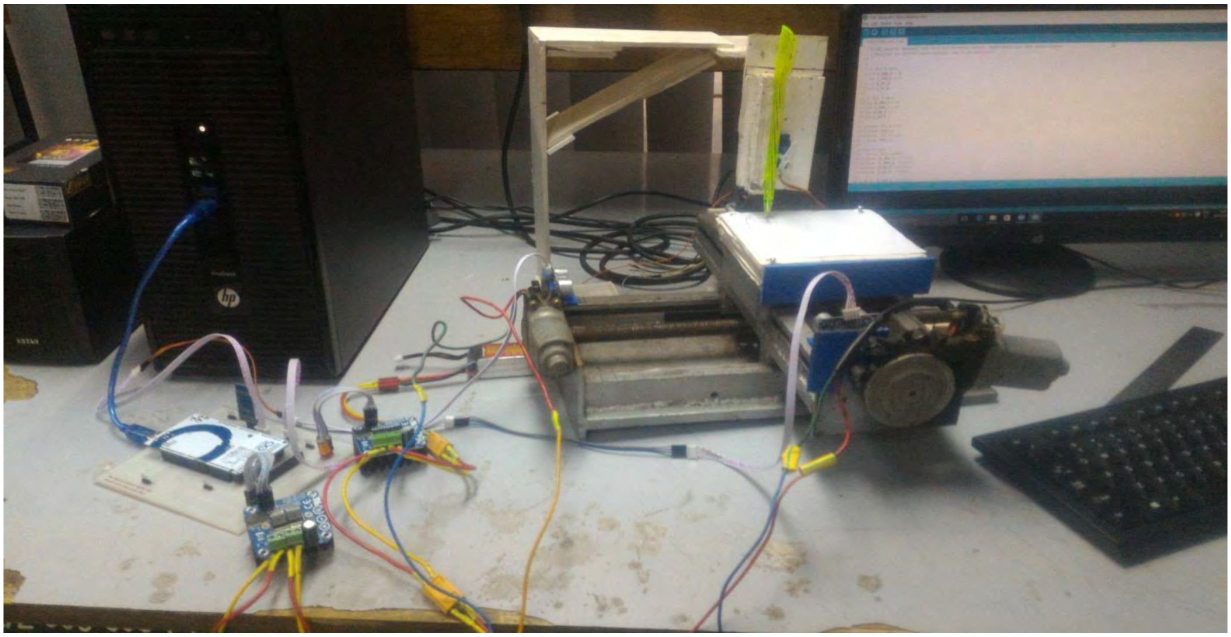


Fig. 5.23: Complete setup of the CNC machine

CHAPTER 6: RESULT PRESENTATION & ANALYSIS FOR ACTUAL MACHINE

6.1 RESULT

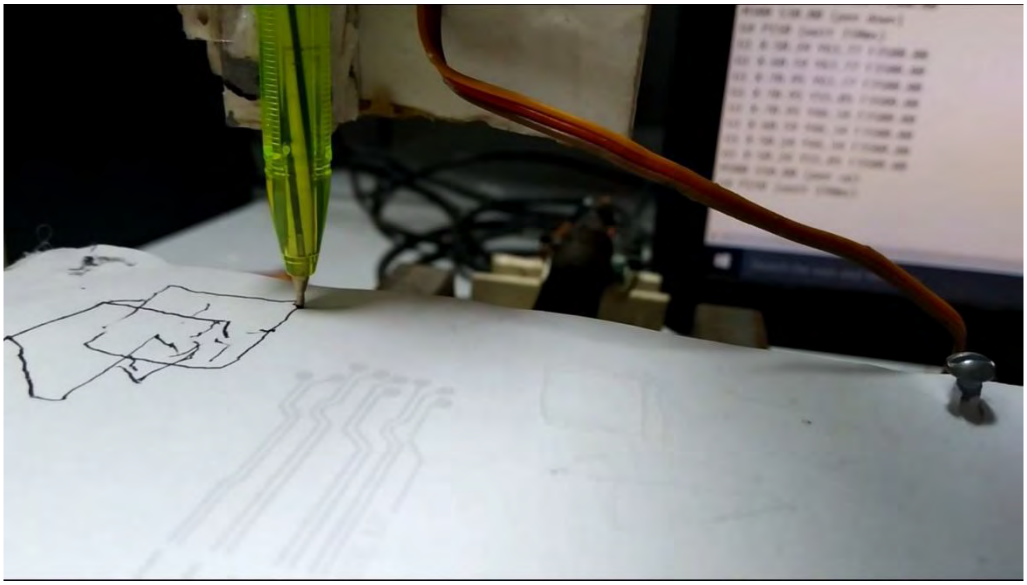


Fig. 6.1: Design print output

The output of the Square shape design is shown in Fig. 6.1. As it appears, considering the other zigzag lines on the print platform, this experiment took multiple attempts to get that perfect shape. At least compared to the prototype, the final CNC machine structure fared better.

6.2 ANALYSIS

There were numerous problems dealt with in the process of building up to this machine. Prior to using the various connectors, the use of thin Ethernet wires was not a good option as there occurred several shorted circuits and overflow of current considering the use of 11.1V Li-po batteries. In short, the connections were fiddly. Batteries should never be directly connected to the Motor drivers without the Arduino plugged in as they can sometimes create excessive charge, causing the capacitor of BTS7960 to explode which did happen during experiment. Installing the connector and planning on the Control PCB solved this circuit connection problem. Sonar sensors were aligned in a proper position to take readings of the distances of the axes during print execution and the surfaces of the axes were ensured to be smooth, even with PVC boards, so that the Sonar sensors can transmit smooth deflection of ultrasonic waves

CHAPTER 7: ADDITIONAL FEATURES

7.1 FEEDBACK SYSTEM

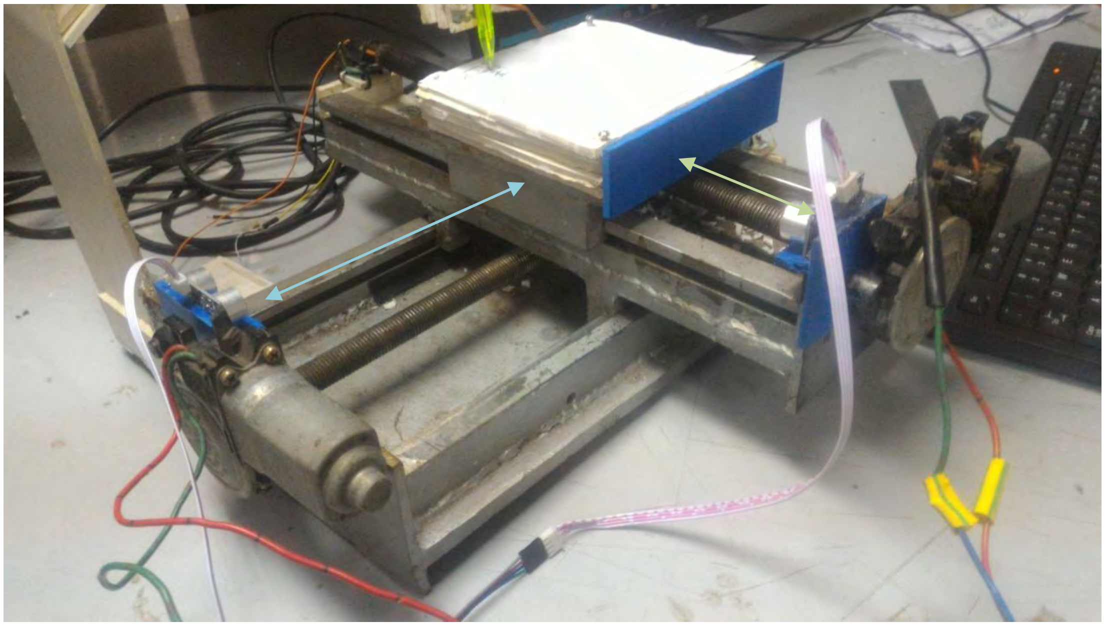
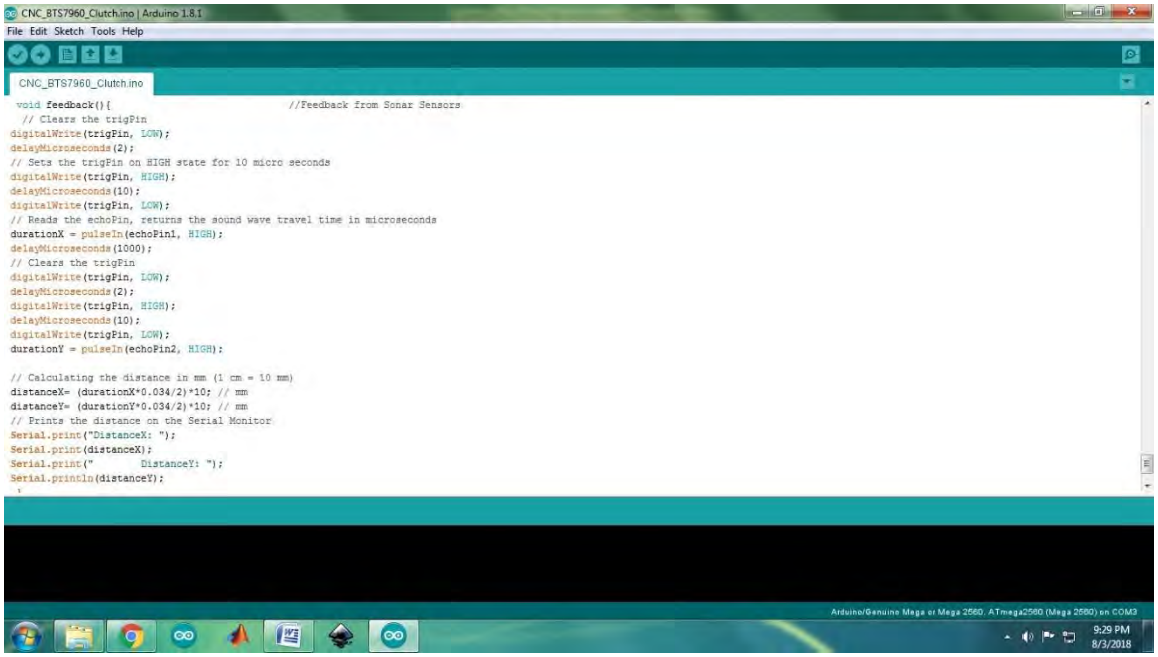


Fig. 7.1: Feedback system using Sonar Sensors

A Feedback System is defined as a system where all or part of the output signal [19] either positive or negative is fed back to the input. It is basically a system in a loop. Like most engineering machines, this CNC machine required a separate feedback system as well. Stepper motors comprise of a function called Micro stepping, which works as a feedback for determining the positional coordinates of traditional CNC machine axes. This micro stepping works because these stepper motors can have four or six magnetic poles, making the rotor move at fine acute angles of less than 1.8° with precision. This is not the case with glass motors; considering as an equivalent to DC motors, they comprise of only two magnetic poles inside their structure and can only be defined for speed in revolutions per minute (rpm). These DC motors cannot interpret or compute a position, even if the software logic or coding states accordingly, so a good option is bring in an additional sensor to support the glass motors and motor drivers for feedback. There are many methods, but for this project, Sonar Sensors were the best option [3,4]. These sensors were placed on the position of the glass motors using PVC boards and were connected to the main control PCB using female header flex cables (Fig. 7.1).

7.1.1 OVERVIEW OF OUR FEEDBACK SYSTEM


CNC_BTS7960_Clutch.ino | Arduino 1.8.1
File Edit Sketch Tools Help
CNC_BTS7960_Clutch.ino
9:29 PM
8/3/2018

```
 void feedback(){                             //Feedback from Sonar Sensors
  // Clears the trigPin
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
// Sets the trigPin on HIGH state for 10 micro seconds
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
// Reads the echoPin, returns the sound wave travel time in microseconds
durationX = pulseIn(echoPin1, HIGH);
delayMicroseconds(1000);
// Clears the trigPin
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
durationY = pulseIn(echoPin2, HIGH);

// Calculating the distance in mm (1 cm = 10 mm)
distanceX= (durationX*0.034/2)*10; // mm
distanceY= (durationY*0.034/2)*10; // mm
// Prints the distance on the Serial Monitor
Serial.print("DistanceX: ");
Serial.print(distanceX);
Serial.print("        DistanceY: ");
Serial.println(distanceY);
 }
```

Arduino/Genuine Mega or Mega 2560, ATmega2560 (Mega 2560) on COM3

Fig. 7.2: Feedback segment of the Arduino CNC code

The hardware is shown in Fig. 7.1 and the feedback logic of the main Arduino CNC code for glass motors is shown in Fig. 7.2. A separate void method was called for collecting feedback from the Sonar sensors in the form of distance readings in millimeter (mm). The distance of the axes are calculated from the following error calculation equations set in the code,

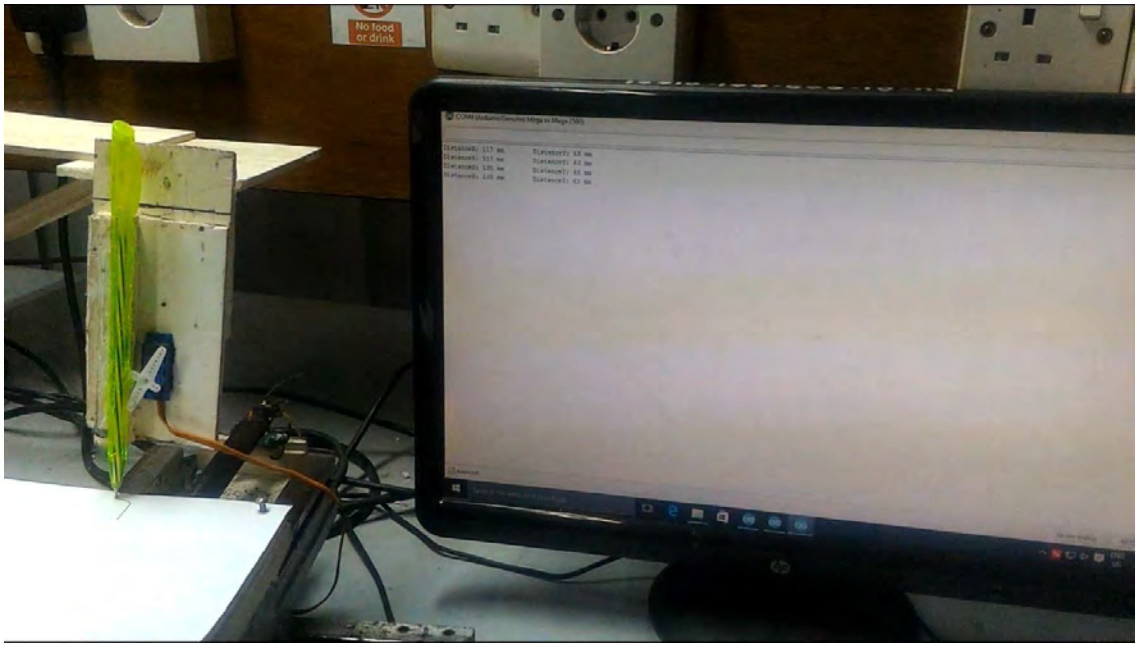
```
// Calculating the distance in mm (1 cm = 10 mm)
distanceX= (durationX*0.034/2)*10; // mm
distanceY= (durationY*0.034/2)*10; // mm
```

Here, the standard equation for error calculation is, $distance = (\frac{duration*0.034}{2}) * 10$

A Sonar sensor triggers ultrasonic waves as input from the trigger Sonar. These waves get deflected from the surface of the CNC axes and are received on the echo Sonar as output. From the given equation, the term 'duration' is a pulse input received by the echo, which calculates time taken for the ultrasonic sound wave to travel one complete cycle in microseconds. The value of 'duration' obtained is then placed in the equation to determine the 'distance' in mm.

In the code, distanceX and durationX are parameters for the X-axis whereas distanceY and durationY are parameters for the Y-axis. The distance and duration parameters in the equation are directly proportional to each other. So, to summarize, the sonar sensors compute the distance and the motor drivers use it as feedback to determine the coordinates of a design with necessary measurements set in the code.

7.1.2 PROBLEMS FACED AND SOLUTIONS



COM4 (Arduino/Genuino Mega or Mega 2560)

```
Distance8: 117 mm        Distance7: 63 mm
Distance8: 117 mm        Distance7: 63 mm
Distance8: 135 mm        Distance7: 63 mm
Distance8: 135 mm        Distance7: 63 mm
```

Fig. 7.3: Sonar Sensor Feedback simulation on screen

Fig. 7.3 shows a simulation test on screen of how the Sonar sensors actually calculate the distances of the X and Y-axes during an actual design print. From the simulation test, fluctuations were detected in the readings. For example, when the X-axis with respect to the platform, is to move further from 160mm, the value of its distance is suppose increase, to be greater than 160mm, but instead the readings dropped to 158mm or 159mm at certain stages as it moved away. This was termed as fluctuation. Two reasons behind this is that, one, the surface wall of the platform X-axis is uneven under microscopic view, even though it appears smooth from the naked eye; and two, ultrasonic sound waves travel through air, a non-material matter that does create obstruction at times, depending on the density and room temperature. Not even the error calculation equations can compute that out of the algorithm. Basically, this is not a hardware or software problem, it is just an error that has to be taken into account. There is no clear solution regarding this issue but this criterion should add to the list of future research scopes and works.

7.2 MANUAL CONTROL FUNCTION

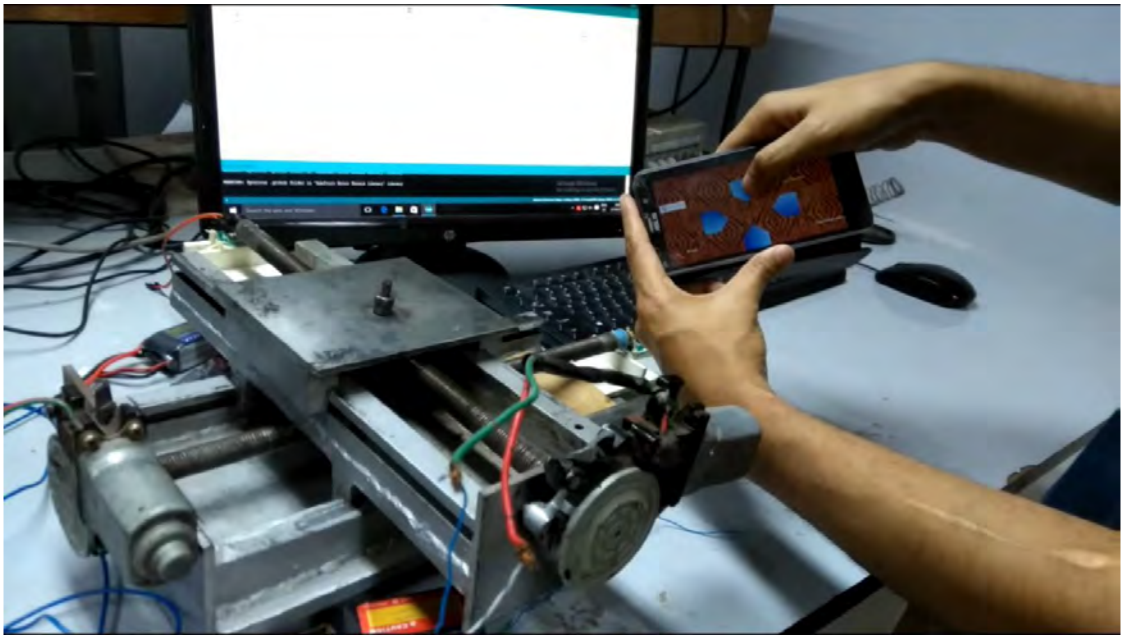


Fig. 7.4: Bluetooth Android App control for positioning the platform

Prior to printing any design, it was necessary to position the print platform. This operation can be set by a manual control system. A concept was brought to plan on manually controlling the platform using a Smartphone Application with Bluetooth synchronization. The idea behind this feature comes from the concept of building Android controlled Arduino projects with interfacing modules, making the Smartphone work as a remote control [5].

7.2.1 OVERVIEW OF OUR MANUAL CONTROL FUNCTION

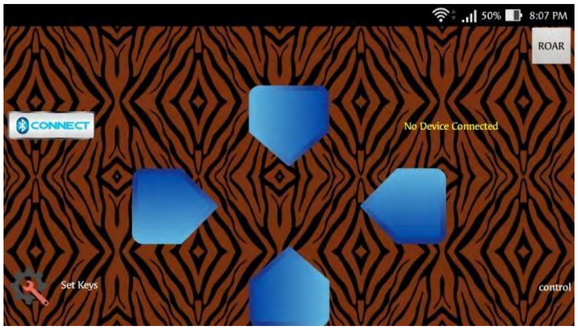
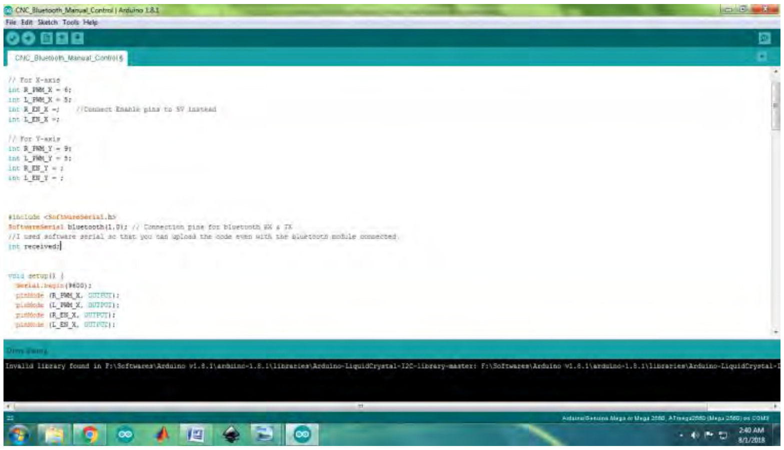


Fig. 7.5: Smartphone App

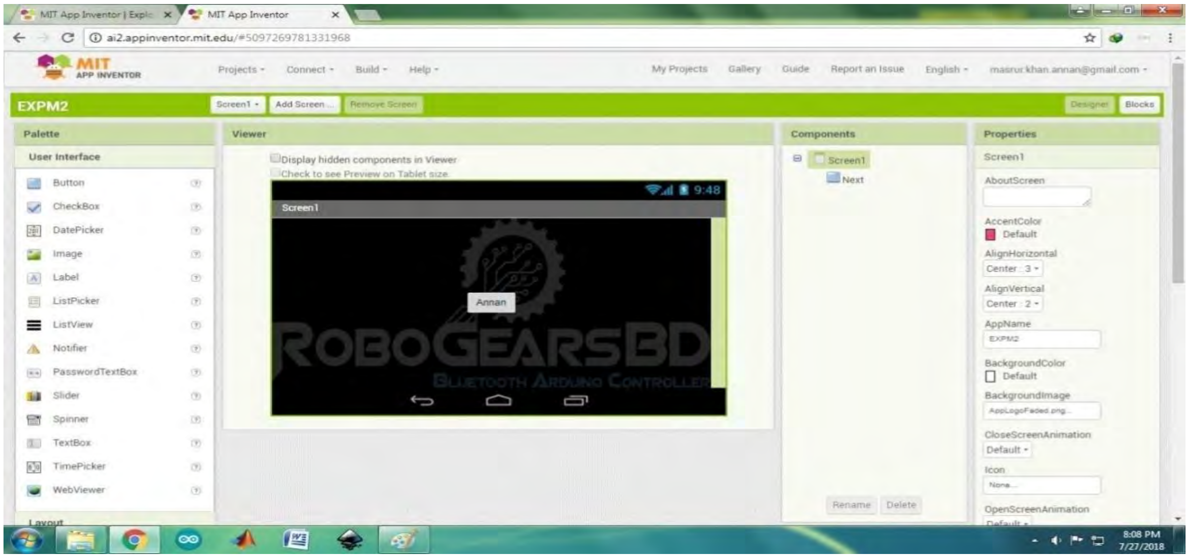
This system of the CNC machine mainly comprised of two elements- an HC-05 Bluetooth Module and a Smartphone with a Bluetooth application named "Arduino_Bluetooth_Controller.apk" (Fig. 7.5). An Arduino code for Bluetooth is compiled on the Arduino software with necessary pinouts defined. The code is uploaded onto the Arduino Mega and then the Bluetooth module is connected to its PCB slot. The Smartphone is paired with the Bluetooth Module and then the Application (or App) is opened. The App is connected to the Bluetooth module via pairing and from the 'Set Keys' option; four characters are set on the four touch screen buttons to execute Bluetooth commands sent to the Arduino. These characters can be letters or numbers or whatever is set in the code. The four touch buttons or keys send commands to the Motor Drivers to move the platform left, right, forward and back. With respect to the machine and the app's orientation, the up and down arrow keys control the X-axis forward and backward and the left and right keys command the Y-axis to go back and front respectively. But these commands can be changed again from the code with respect to a user's suitable control. After positioning the platform, the Bluetooth pairing can be terminated by exiting the App.


CNC_Bluetooth_Manual_Control | Arduino 1.8.1
File Edit Sketch Tools Help
CNC_Bluetooth_Manual_Control §
Arduino/Genuino Mega or Mega 2560, ATmega2560 (Mega 2560) on COM3
2:40 AM
8/1/2018

```
// For X-axis
int R_PWM_X = 6;
int L_PWM_X = 5;
int R_EN_X =;     //Connect Enable pins to 5V instead
int L_EN_X =;

// For Y-axis
int R_PWM_Y = 9;
int L_PWM_Y = 5;
int R_EN_Y = ;
int L_EN_Y = ;



#include <SoftwareSerial.h>
SoftwareSerial bluetooth(1,0); // Connection pins for bluetooth RX & TX
//I used software serial so that you can upload the code even with the bluetooth module connected
int received;


void setup() {
  Serial.begin(9600);
  pinMode (R_PWM_X, OUTPUT);
  pinMode (L_PWM_X, OUTPUT);
  pinMode (R_EN_X, OUTPUT);
  pinMode (L_EN_X, OUTPUT);
```

```
Done Saving.
Invalid library found in F:\Softwares\Arduino v1.8.1\arduino-1.8.1\libraries\Arduino-LiquidCrystal-I2C-library-master: F:\Softwares\Arduino v1.8.1\arduino-1.8.1\libraries\Arduino-LiquidCrystal-I
```

Fig. 7.6: Arduino Bluetooth Control code



AboutScreen
AccentColor: Default
AlignHorizontal: Center : 3
AlignVertical: Center : 2
AppName: EXPM2
BackgroundColor: Default
BackgroundImage: AppLogoFaded.png...
CloseScreenAnimation: Default
Icon: None...
OpenScreenAnimation: Default

Fig. 7.7: MIT App Inventor 2

The Smartphone App was customized in MIT App Inventor version 2, which is an App design based website. The Phone user interface and the app command operations were installed using the Graphics and design options as well as the Node terminal blocks to make the interface appear like a remote control keypad to execute commands (Fig. 7.7).

7.2.2 PROBLEMS FACED AND SOLUTIONS

During the first trial run of the manual control system, the Bluetooth module did not pair with the Smartphone app nor was executing commands properly even if it was paired. The reason was the RX and TX pins of Bluetooth module was connected to pins 10 and 11 of the Arduino Mega and it turns out that Arduino Mega has its own set of designated pins for Serial interfacing using TX and RX. According to Arduino Mega's datasheet, pins 10 and 11 were not a category of those designated pins. Using *Softwareserial.h* function command, the datasheet was reviewed and during the PCB design, the proper pinouts were routed for the module.

CHAPTER 8: CONCLUSION

8.1 SUMMARY

Finally, some data was collected about CNC machines and examined. Later, a prototype, 3 axes and 2 dimensional CNC printing machine was designed with stepper motors. At the end, an attempt was made to construct a 3 axes and 2 dimensional CNC milling machine with metal body, glass motors and a drill bit, with additional features such as manual control function and feedback system. However, the drill bit could not be connected due to the heavyweight CNC body and slightly unstable y plane. Hence, a pen secured with a servo motor was used instead of the drill bit.

.

8.2 FUTURE RESEARCH & WORKS

There are many scopes of improvement in this project-High powered DC motors can be used instead of glass motors for faster operation or proximity sensors like laser sensors can be used for more accurate readings. Furthermore, the CNC machine body can be made with a lighter metal such as Aluminum or the Y axis can be made to be a bit stable for the installation of the drill bit into the machine. As for Research scopes- Machine can be modified to do Laser Cutting by adjusting the requisite Spindle, can be modified to work on Carpentry in 2D scale, like milling a design on wooden doors or cutting fine jewellery. From a Software perspective, softwares like Gerber Controller (Grbl Controller) can be utilized, making it compatible with the machine; the software provides a map of the 2D design on its interface during printing or milling. The CNC machine can also be modified to work as a 3D printer using PLA Filament and print 3D models designed via any 3D software like Handicrafts. Now, most of these types of CNC machines exist, but they are mainly prominent for First world countries. The concept for this machine is built mainly for developing and underdeveloped countries, where many nations, including Bangladesh; either cannot afford or cannot create the raw materials or just do not have the expansive knowledge. In short, this concept is a start to initiate a CNC machine production using this country's very own technical laboring workforce.

REFERENCES

1. Da Rocha, P. A. S., de Silva e Souza, R. D., & de Lima Tostes, M. E., "Prototype CNC machine design", 9th IEEE/IAS International Conference on Industry Applications - INDUSCON, Sao Paulo, Brazil, 2010.
2. Wang, S., Cao, H., Wu, W., Cao, H., Huang, Z., & Liu, Y., "Design of a large DDR motor for high-end CNC machine tools applications", International Conference on Electrical Machines and Systems, Beijing, China, 2011.
3. Carullo, A., & Parvis, M., "An ultrasonic sensor for distance measurement in automotive applications" IEEE Sensors Journal, 1(2), 143, August 2001.
4. Chu, B., "Mobile robot position control algorithm based on multiple ultrasonic distance sensors", 15th International Conference on Control, Automation and Systems (ICCAS), Busan, South Korea, 2015.
5. Cai, J., Wu, J., Wu, M., & Huo, M., "A Bluetooth toy car control realization by android equipment", Proceedings 2011 International Conference on Transportation, Mechanical, and Electrical Engineering (TMEE), Changchun, China, 2011.
6. Shukun, C., Heng, Z., Xiangbo, Z., Qiujuan, Y., & Changsheng, A., "Software and Hardware Platform Design for Open-CNC System", IEEE International Symposium on Knowledge Acquisition and Modeling Workshop, Wuhan, China, 2008.
7. https://www.youtube.com/channel/UCmsyo-QzVRmuzKl3l4--Xqg/playlists, Inkscape Tutorials by Amitabha Dey, Last access at August 31, 2018;
8. https://www.cncmasters.com/product/cnc-milling-cnc-jr-table-top-mill/, CNC JR. Table Top Mill, Tech Specs & Features, Last access at August 31, 2018;
9. http://electricdiylab.com/how-to-generate-g-code-in-inkscape-for-diy-cnc-machines/, How to add G-code extension in Inkscape, Last access at August 31, 2018;
10. https://www.thomasnet.com/about/cnc-milling-51276103.html, Description on CNC milling, Last access at August 31, 2018;
11. https://www.camaster.com/what-is-the-difference-between-milling-machines-and-a-cnc-router/, Difference between CNC Milling and CNC Router machine, Last access at August 31, 2018;
12. https://lathemachine.co/?gclid=Cj0KCQjwqPDaBRC6ARIsACAf4hC2VAudoAr9135_KwbPiil0t3tLuo74iPatUiFq-BCrzYsh_upg47saAhkxEALw_wcB, Topic on Lathe Machines, Last access at August 31, 2018;
13. http://www.esabna.com/us/en/education/blog/what-is-a-cnc-plasma-cutter.cfm, Topic on Plasma Cutter, Last access at August 31, 2018;
14. http://www.apa21.org/2013/09/what-is-cnc-laser-cutter/, Topic on Laser Cutter, Last access at August 31, 2018;
15. https://www.arrow.com/en/research-and-events/articles/arduino-mega-2560-overview, Arduino Mega 2560 Overview, Last access at August 31, 2018;
16. https://servodatabase.com/servo/towerpro/sg90, TowerPro SG90 Servo Database, Last access at August 31, 2018;
17. https://www.makerlab-electronics.com/product/motor-driver-43a-bts7960/, BTS7960 Motor Driver specification, Last access at August 31, 2018;
18. https://www.theengineeringprojects.com/2015/12/arduino-library-proteus-simulation.html, Arduino Library for Proteus software, Last access at August 31, 2018;

19. https://www.electronics-tutorials.ws/systems/feedback-systems.html, Feedback Systems, Last access at August 31, 2018;