International Journal of Research in Engineering and Science (IJRES)
ISSN (Online): 2320-9364, ISSN (Print): 2320-9356
www.ijres.org Volume 10 Issue 6 ǁ 2022 ǁ PP. 1311-1332

CNC X-Y PLOTTER

MR. AMAR H. PANDULE
MR. GANESH N. BANDGAR
MR. SHUBHAM B. KALE
MR. CHAITANYA A. WAMBORIKAR
Department Of Mechanical Engineering,
S B Patil Engineering Vangali, Indapur

ABSTRACT
The CNC XY Plotter is required to plot two dimensional continuous or discontinuous data on a rectangular coordinate system. This project is done to fabricate a XY plotter using independent motion along X-direction and Y-direction, and microcontroller system (Arduino) to control those motions. Implementation of this project is carried out through the computer linked with the Arduino software, to which the G-code is transmitted by the Processing software.

Date of Submission: 06-06-2022 Date of acceptance: 21-06-2022

I. INTRODUCTION

1.1 Introduction to CNC XY Plotter

A CNC XY Plotter Machine is a 2D plotting machine which uses a pen to draw text or image, continuous or discontinuous drawings on any given solid surface.

CNC plotting is one of the most common plotting processes used in various applications. Since the introduction of Computer Numerical Control in the 1960s, CNC machines have become an integral part of industrial & commercial manufacturing processes.

Computer Numerical Control is a process that uses computers to control tools to create different types of products. These machines require specialized software and consoles for operation. A computer program is customized for an object and the machines are programmed with CNC machining language (G-code) that essentially controls all features of the final product like coordination, location & speeds.

1.2 Problem Statement

Nowadays CNC XY plotters are being made all over the world using the CD drive for the scanner mechanism which provides the requires motion along X-direction & Y-direction. So, this manufacturing process is not suitable for mass production of plotters. Again, the price of the plotters in the current market is quite high too.

So, in order to solve these limitations, we designed a CNC XY Plotter suitable for mass productions which is cheap as well.

1.3 Objectives

Objectives of this project are:
1 To build and design an electromechanical device able to create continuous or discontinuous vector graphics on a solid surface.
2 To minimize the cost of manufacturing.
3 To design a model suitable for mass production

II. OVERVIEW OF THE PROJECT

2.1 Key Features

I. Ability to draw any continuous or discontinuous figure
II. Can draw any geometrical shape like circle, triangle, square etc.
III. Range of plotting is 180mm x180mm.
IV. Maximum error can be 1mm
V. Drawing speed can be controlled.
VI. Can draw multiple intersecting shape.
VII. Mechanical parts like gear, cam etc. can be drawn with utmost accuracy.

VIII. Can produce drawing from any image format available.
IX. Rigid structure to absorb vibration and give stability.
X. Water proof, portable, cheap.
XI. Mass producible.
XII. Low power consumption.

2.2 Advantages

I. Cheap: Cheapest XY Plotter available in the current market.

II. Price Comparison*:* The lowest price of an XY plotter in the current market is $132 whereas our model costs only $55

III. Mass Production: This model was not made in the most popular and conventional way in which the motion along the X-axis & Y-axis is provided by recycled CD drives of Old computers. Two stepper motors were used to provide motion along the X-axis & Y-axis. It is comprehendible that mass production of this product cannot be done through recycling process of CD drives. So, this model is suitable for mass production and marketing at a cheap rate.

IV. Improved Mechanical Properties*:* The model is light weight as plastic wood was used to build the structure. Plastic wood is also Water resistive, so there are less likely to get affected by corrosion or rusting.

V. Portable: Due to its easy-to-carry structure and being lightweight, the setup is easily portable.

VI. Precision Drawing Application: This model can be used for applications where precision drawing is needed to be done.

2.3 Future Prospects

I. Same Mechanism can be used for 3D printing by adding Z-axis mechanism.

II. Drilling, PCB making, Laser engraving can be done with this model by changing the pen holder component.

2.4 Limitations and Disadvantages

I. This project cannot perform precision motion if the distance between previous point & the destination point is less than 1 mm.
II. The base is a little wobbly which may induce some error.
III. The setup was initially to be made a wireless device, which could not be implemented due to slow transmission of data by the wireless device.
IV. Pen holder component could be improved for easy handling and changing pens.
V. Belt- pulley system was installed manually in this model, due to which a little slip occurs, thus error occurs.
VI. If speed is increased, after a certain limit of speed, accuracy drastically decreases.

Solid works Design vs Main Design

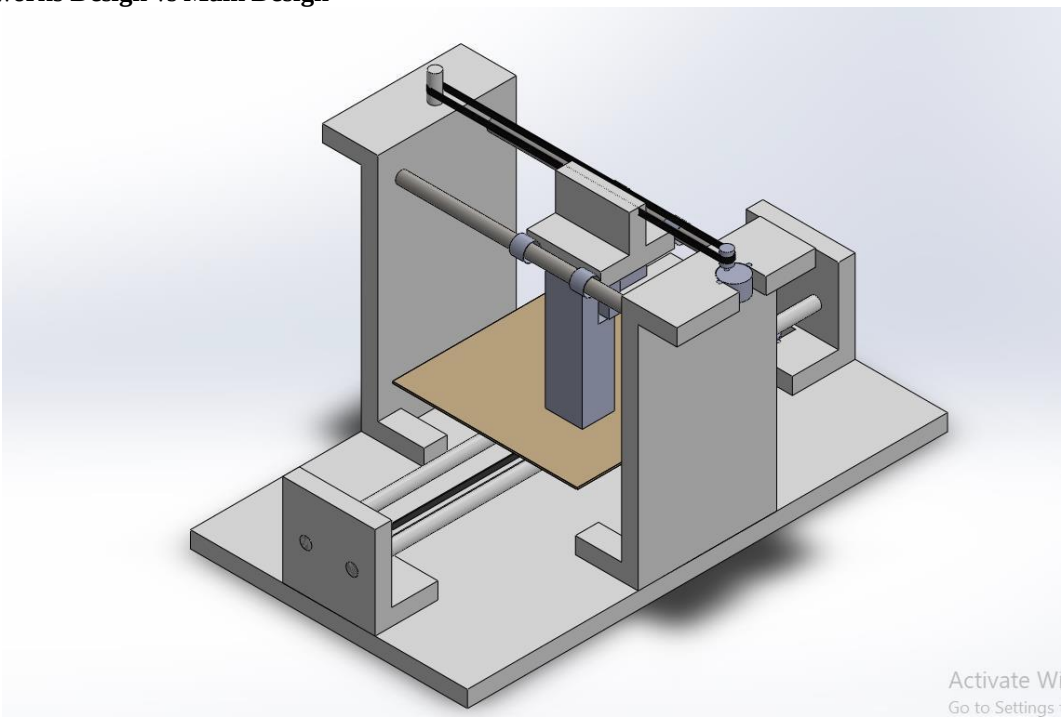

Fig 1 solid works design vs main design

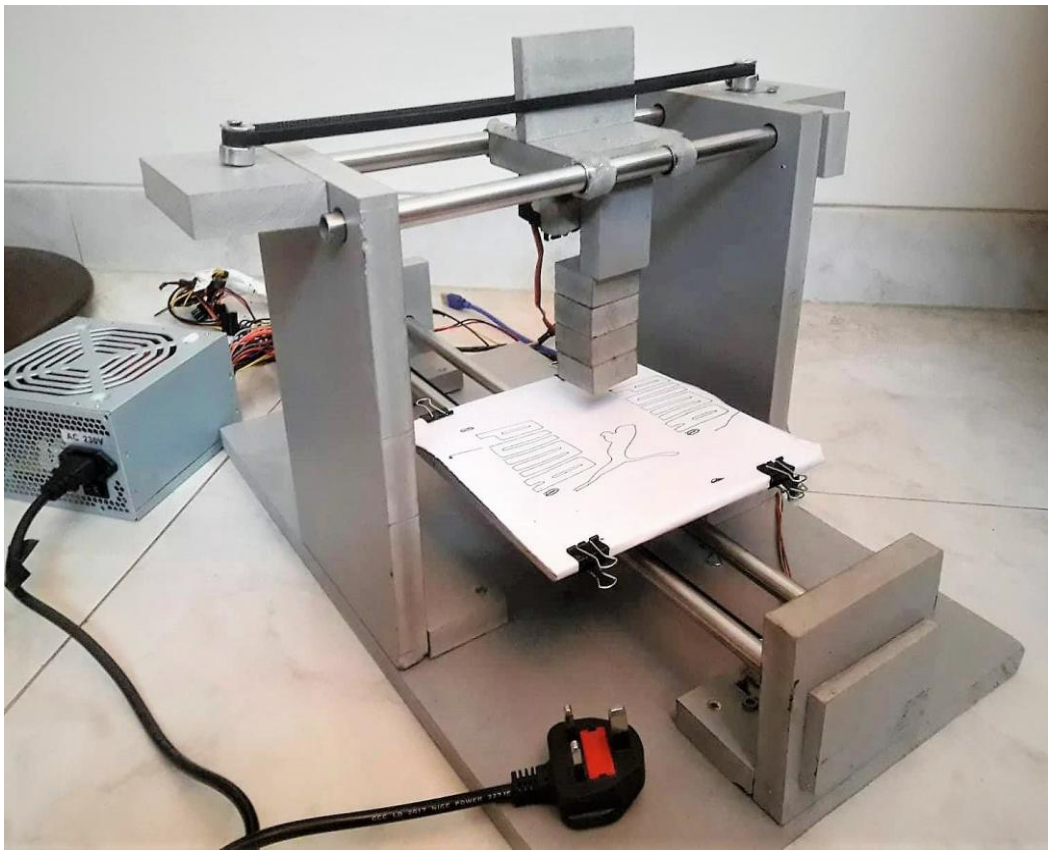

Fig 2 Structural Component

1. Plastic Wood:

Plastic wood was used to build the structure of the model. Thickness: 18 mm
• Water resistant
• Moderate tensile & compressive strength
• Cheap & available

2. Stainless Steel Pipes:

Stainless steel pipes were used to provide axes along which motions will be directed. A pipe of slightly larger diameter was used to make holders which can smoothly slide over the smaller-diameter stainless steel pipe.
• ¾" and ½" Diameter
• High strength
• Non-corrosive

3.Belt:

Belts were used to link an idler pulley and the rotating shaft of the stepper motor working under no slip condition.
• Material: Rubber
• Non-plastic deformable

4.Pulley:

Pulleys were used to transmit power through the belt via teeth. They were used to move the platform carrying the paper and the pen holder component by transmitting power from the stepper motors.
• Material: Stainless steel

HARDWARE

1. Arduino Uno:

• Microcontroller: ATmega328
• Operating Voltage: 5V
• Flash memory:32KB
• Clock speed: 16 MHz
• 14 Digital I/O pins,6 Analog input pins

2.Stepper Motor (Unipolar):

• Model name: 28BYJ-48
• Rated voltage: 12V DC
• Stride Angle: 5.625°/64
• Pull in torque: 300 gf.cm

3.Stepper Motor Driver:

• Model Name: ULN2003
• Voltage input:7-12 V

4.Servo motor:

• Model name: MG995
• Voltage: 5V
• Stall torque: 8.5 kgf·cm (4.8V),
10 kgf·cm (6 V)
• Temperature range: 0 ºC – 55 ºC

5.Power Supply:

• Model Name: DX-550W
• AC input: 230 V
• DC output: 5V, -12V, 12V, +3.3V

6.Jumper Wires:

• Male to male
• Male to female

SOFTWARE
1.Arduino:

• Version: 1.8.5
• Arduino is a cross-platform IDE that works with an Arduino controller for writing, compiling and uploading code to the Arduino board.
• The software provides support for a wide array of Arduino boards, including Arduino Uno, Nano, Mega, Ethernet, Frio, Pro or Pro Mini etc.
• The universal languages for Arduino are C and C++, thus the software is fit for professionals who are familiar with these two. Features such as syntax highlighting, automatic indentation and brace matching makes it a modern alternative to other IDEs.
• This software was used to control the motion of two stepper motors required to provide motion along X-direction & Y-direction.

2.Processing:

• Version:3.2.4
• Processing is a flexible software sketchbook and a language for learning how to code within the context of the visual arts.
• The code required for processing G-code was written in JAVA language.
• This is an open source software and free to download

3.Inkscape Unicorn Master:

• Version: 0.48.5
• A powerful, free design tool
• Broad file format compatibility, can convert any type image files into G-code.
• Powerful text tool, can generate texts to create a G-code.
• This is an open source software and Free to download software, can be used for specialized purposes by adding extensions.

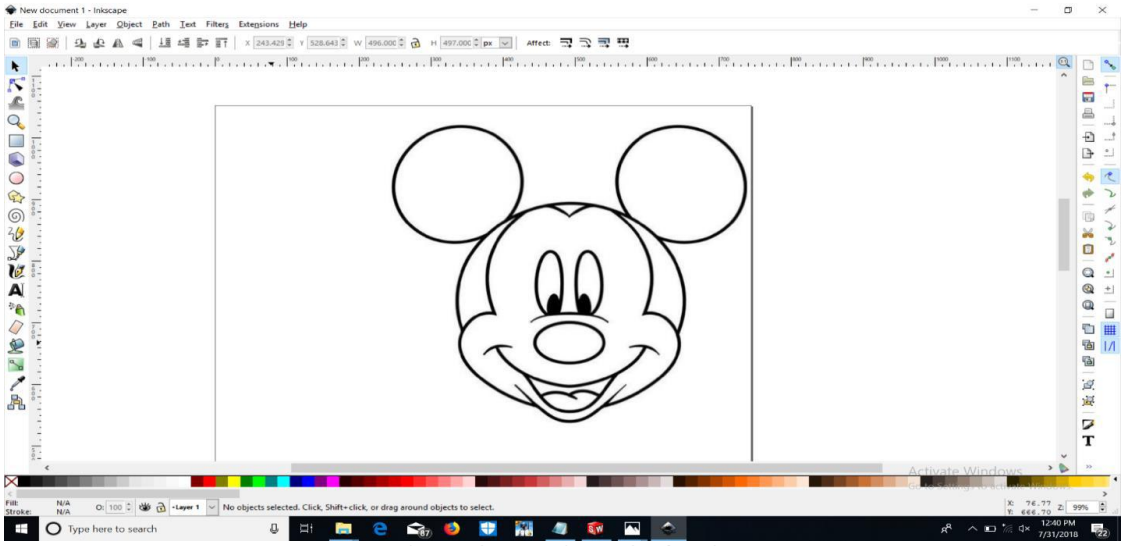

FIG 4.WORKING MECHANISM

1 CIRCUIT

All the wiring in the above circuit have been done by jumper wires. Power to the stepper motor and servo motor was given through power supply.

Figure 16: Circuit

1 and 5- Stepper motor

2 and 4- Motor driver

3- Arduino Uno

4- Servo motor

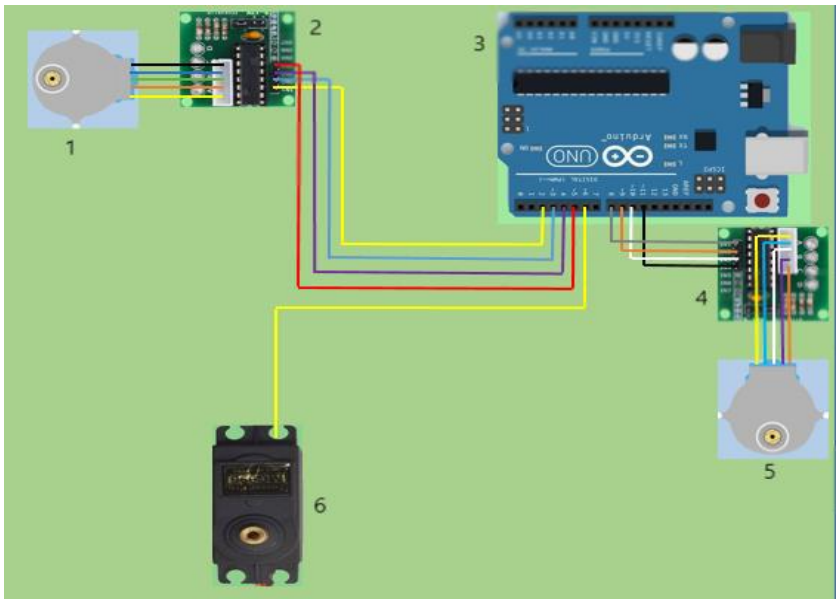

FIG 5. CIRCUIT DIAGRAM

2 X axis motion mechanism

The pen holder with servo motor moves right along the X-axis when the stepper motor rotates counter-clockwise

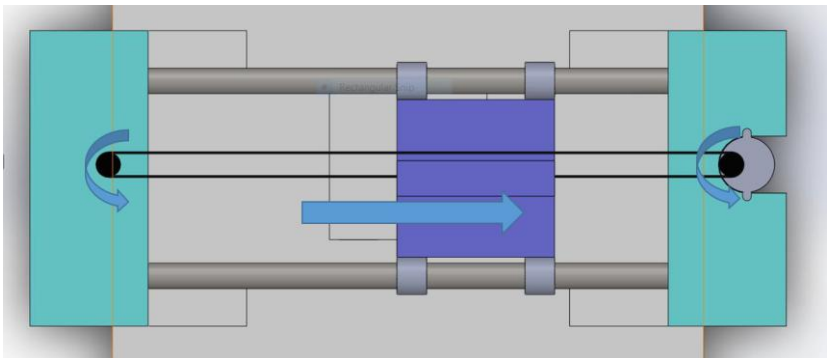

FIG 6 x AXIS MOTION

3. Y axis motion mechanism

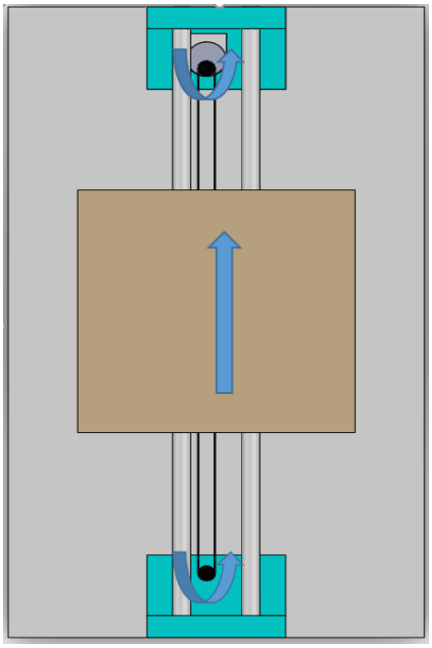

Fig no 7.Y axis motion

The drawing base along with the paper moves down along positive Y-axis as the stepper rotates clockwise
The drawing base along with the paper moves up along positive Y-axis as the stepper rotates counter-clockwise

4.pen up-down mechanism

The pen holder mechanism we used is not conventional yet effective. We used servo motor to push up the block that holds the pen. To make the pen go down we simply used the effect of gravity. When the servo does not push the pen holder, the pen simply goes down due to gravity. We also used a sign pen so that the weight of the pen is enough to write on the paper.

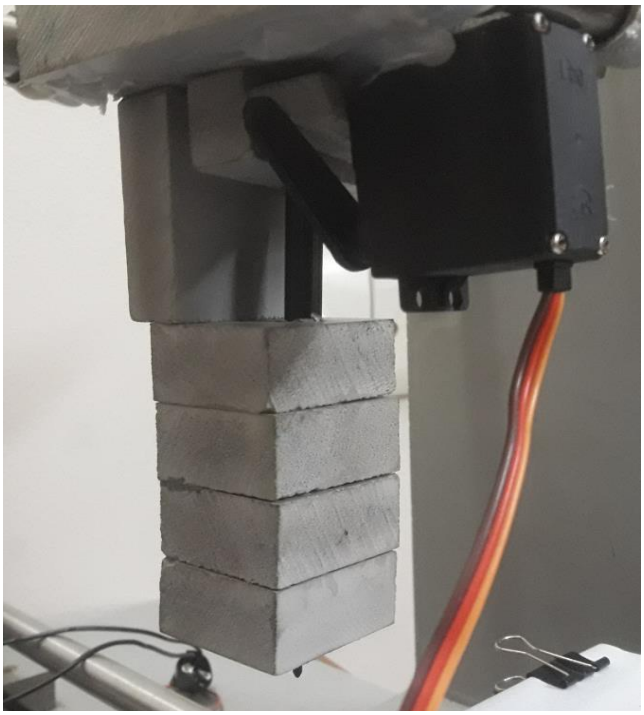

Fig no 8.pen up down mechanism

range | Arduino 1.8.5

File Edit Sketch Tools Help

range

```
#include <Servo.h>
#include <Stepper.h>

#define LINE_BUFFER_LENGTH 512

const int penZUp = 80;
const int penZDown = 40;

const int penServoPin = 6;

const int stepsPerRevolution = 64;

Servo penServo;

Stepper myStepperY(stepsPerRevolution, 5,3,4,2);
Stepper myStepperX(stepsPerRevolution, 11,9,10,8);

struct point {
  float x;
  float y;
  float z;
};

struct point actuatorPos;

float StepInc = 1;
int StepDelay = 0;
int LineDelay = 50;
int penDelay = 50;

float StepsPerMillimeterX = 52;
float StepsPerMillimeterY = 52;

float Xmin = -500;
float Xmax = 500;
```

7

ARDUINO CODE

gcode_executer | Processing 3.2.4

File Edit Sketch Debug Tools Help

gcode_executer

```
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()
{
  String result = (String) JOptionPane.showInputDialog(frame,
    "Select the serial port that corresponds to your Arduino board.",
    "Select serial port",
    JOptionPane.QUESTION_MESSAGE,
    null,
    Serial.list(),
    0);

  if (result != null) {
    portname = result;
    openSerialPort();
  }
}

void setup()
```

gcode_executer | Processing 3.2.4

File Edit Sketch Debug Tools Help

gcode_executer

```
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()
{
  String result = (String) JOptionPane.showInputDialog(frame,
    "Select the serial port that corresponds to your Arduino board.",
    "Select serial port",
    JOptionPane.QUESTION_MESSAGE,
    null,
    Serial.list(),
    0);

  if (result != null) {
    portname = result;
    openSerialPort();
  }
}

void setup()
```

gcode_executer | Processing 3.2.4

File Edit Sketch Debug Tools Help

gcode_executer

```
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()
{
  String result = (String) JOptionPane.showInputDialog(frame,
    "Select the serial port that corresponds to your Arduino board.",
    "Select serial port",
    JOptionPane.QUESTION_MESSAGE,
    null,
    Serial.list(),
    0);

  if (result != null) {
    portname = result;
    openSerialPort();
  }
}

void setup()
```

```
{
  delay(200);
  char line[ LINE_BUFFER_LENGTH ];
  char c;
  int lineIndex;
  bool lineIsComment, lineSemiColon;

  lineIndex = 0;
  lineSemiColon = false;
  lineIsComment = false;

  while (1) {

    while ( Serial.available()>0 ) {
      c = Serial.read();
      if (( c == '\n') || (c == '\r') ) {
        if ( lineIndex > 0 ) {
          line[ lineIndex ] = '\0';
          if (verbose) {
            Serial.print( "Received : ");
            Serial.println( line );
          }
          processIncomingLine( line, lineIndex );
          lineIndex = 0;
        }
        else {

        }
        lineIsComment = false;
        lineSemiColon = false;
        Serial.println("ok");
      }
      else {
        if ( (lineIsComment) || (lineSemiColon) ) {
          if ( c == ')' )  lineIsComment = false;
```

```
        }
        else {
          if ( c <= ' ' ) {                              // Throw
          }
          else if ( c == '/' ) {                         // Block de
          }
          else if ( c == '(' ) {                         // Enable c
            lineIsComment = true;
          }
          else if ( c == ';' ) {
            lineSemiColon = true;
          }
          else if ( lineIndex >= LINE_BUFFER_LENGTH-1 ) {
            Serial.println( "ERROR - lineBuffer overflow" );
            lineIsComment = false;
            lineSemiColon = false;
          }
          else if ( c >= 'a' && c <= 'z' ) {             // Upcase l
            line[ lineIndex++ ] = c-'a'+'A';
          }
          else {
            line[ lineIndex++ ] = c;
          }
        }
      }
    }
  }
}

void processIncomingLine( char* line, int charNB ) {
  int currentIndex = 0;
  char buffer[ 64 ];                                     // Hope th
  struct point newPos;

  newPos.x = 0.0;
```

```
newPos.y = 0.0;

//  Needs to interpret
//  G1 for moving
//  G4 P300 (wait 150ms)
//  G1 X60 Y30
//  G1 X30 Y50
//  M300 S30 (pen down)
//  M300 S50 (pen up)
//  Discard anything with a (
//  Discard any other command!

while( currentIndex < charNB ) {
  switch ( line[ currentIndex++ ] ) {                 // Sel
  case 'U':
    penUp();
    break;
  case 'D':
    penDown();
    break;
  case 'G':
    buffer[0] = line[ currentIndex++ ];           // /!\ D
    //      buffer[1] = line[ currentIndex++ ];
    //      buffer[2] = '\0';
    buffer[1] = '\0';

    switch ( atoi( buffer ) ){                    // Selec
    case 0:                                     // G00 & G0
    case 1:
      // /!\ Dirty - Suppose that X is before Y
      char* indexX = strchr( line+currentIndex, 'X' );  /
      char* indexY = strchr( line+currentIndex, 'Y' );
      if ( indexY <= 0 ) {
        newPos.x = atof( indexX + 1);
        newPos.y = actuatorPos.y;
```

```
      }
      else if ( indexX <= 0 ) {
        newPos.y = atof( indexY + 1);
        newPos.x = actuatorPos.x;
      }
      else {
        newPos.y = atof( indexY + 1);
        indexY = '\0';
        newPos.x = atof( indexX + 1);
      }
      drawLine(newPos.x, newPos.y );
      //        Serial.println("ok");
      actuatorPos.x = newPos.x;
      actuatorPos.y = newPos.y;
      break;
    }
    break;
  case 'M':
    buffer[0] = line[ currentIndex++ ];        // /!\ Dirt
    buffer[1] = line[ currentIndex++ ];
    buffer[2] = line[ currentIndex++ ];
    buffer[3] = '\0';
    switch ( atoi( buffer ) ){
    case 300:
      {
        char* indexS = strchr( line+currentIndex, 'S' );
        float Spos = atof( indexS + 1);
        //          Serial.println("ok");
        if (Spos == 30) {
          penDown();
        }
        if (Spos == 50) {
          penUp();
        }
        break;
```

```
        }
      case 114:                                   // M114
        Serial.print( "Absolute position : X = " );
        Serial.print( actuatorPos.x );
        Serial.print( "  -  Y = " );
        Serial.println( actuatorPos.y );
        break;
      default:
        Serial.print( "Command not recognized : M");
        Serial.println( buffer );
      }
    }
  }



}



void drawLine(float x1, float y1) {

  if (verbose)
  {
    Serial.print("fx1, fy1: ");
    Serial.print(x1);
    Serial.print(",");
    Serial.print(y1);
    Serial.println("");
  }


  if (x1 >= Xmax) {
    x1 = Xmax;
  }
```

```
if (x1 <= Xmin) {
  x1 = Xmin;
}
if (y1 >= Ymax) {
  y1 = Ymax;
}
if (y1 <= Ymin) {
  y1 = Ymin;
}

if (verbose)
{
  Serial.print("Xpos, Ypos: ");
  Serial.print(Xpos);
  Serial.print(",");
  Serial.print(Ypos);
  Serial.println("");
}

if (verbose)
{
  Serial.print("x1, y1: ");
  Serial.print(x1);
  Serial.print(",");
  Serial.print(y1);
  Serial.println("");
}

//  Convert coordinates to steps
x1 = (int)(x1*StepsPerMillimeterX);
y1 = (int)(y1*StepsPerMillimeterY);
float x0 = Xpos;
float y0 = Ypos;
```

```
long dx = abs(x1-x0);
long dy = abs(y1-y0);
int sx = x0<x1 ? StepInc : -StepInc;
int sy = y0<y1 ? StepInc : -StepInc;

long i;
long over = 0;

if (dx > dy) {
  for (i=0; i<dx; ++i) {
    myStepperX.step(sx);
    over+=dy;
    if (over>=dx) {
      over-=dx;
      myStepperY.step(sy);
    }
    delay(StepDelay);
  }
}
else {
  for (i=0; i<dy; ++i) {
    myStepperY.step(sy);
    over+=dx;
    if (over>=dy) {
      over-=dy;
      myStepperX.step(sx);
    }
    delay(StepDelay);
  }
}

if (verbose)
{
  Serial.print("dx, dy:");
  Serial.print(dx);
```

```
  }

  if (verbose)
  {
    Serial.print("Going to (");
    Serial.print(x0);
    Serial.print(",");
    Serial.print(y0);
    Serial.println(")");
  }


  delay(LineDelay);

  Xpos = x1;
  Ypos = y1;
}


void penUp() {
  penServo.write(penZUp);
  delay(LineDelay);
  Zpos=Zmax;
  if (verbose) {
    Serial.println("Pen up!");
  }
}
void penDown() {
  penServo.write(penZDown);
  delay(LineDelay);
  Zpos=Zmin;
  if (verbose) {
    Serial.println("Pen down.");
  }
}
```

gcode_executer | Processing 3.2.4

File Edit Sketch Debug Tools Help

gcode_executer

```
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()
{
  String result = (String) JOptionPane.showInputDialog(frame,
    "Select the serial port that corresponds to your Arduino board.",
    "Select serial port",
    JOptionPane.QUESTION_MESSAGE,
    null,
    Serial.list(),
    0);

  if (result != null) {
    portname = result;
    openSerialPort();
  }
}

void setup()
```

Complete CNC Plotter After all those mounting of motors with the printing mechanism unit and axis with each other also with basement implementation become completed. The Fig. 12 shows complete CNC plotter.

Calibration After finishing the complete build-up of the machine, it is necessary to calibrate the movement of the axes. Test code for stepper motor mentioned in the section IV (D) is used where stepper motors steps per revolution was 20 and 160 steps were allowed for the motor to travel. On running the code, it is found that the motor has travelled 26.5 mm which indicates that the motor takes 6 steps to cover 1mm. this is how calibration is done for both X and Y axes.

APPLICATIONS

A CNC plotter machine is a 3D controlled 2D plotting machines which uses a pen to draw text or image on any given solid surface. It can be used for the purposes such as **PCB Design, logo design**, etc.
. This technique is based on CNC plotter machine.

- CNC Machining Can Be Repeated. The best benefit of CNC punching is that once you set the design, you can make as many copies of it as your materials will allow. ...
- Dental Equipment. ...
- Weapons. ...
- Construction. ...
- Transportation. ...
- Research And Development. ...
- Niche Manufacturing

COST ANALYSIS

Components	**Price (BDT)**
Arduino Uno	350
Stepper motor x 2	200 x 2
Motor driver x 2	100 x 2
Servo motor	450
Plastic wood	600
Belt	100
Pulley x 2	100 x 2
Steel rod	150
Structure building cost	900
Power Supply	600
Miscellaneous	450
Total Cost	**4400**

III. CONCLUSION

The primary goal of this project was to find & build a cheaper and mass producible design of a CNC XY Plotter maintaining the accuracy as well.
This project is further extendable. Modification of pen holder component can make this project suitable for a lot of applications.
We tried our best in designing and implementing this model. Our lack of expertise & experience, manual fabrication of various parts sometimes hindered our progress and affected the perfection to be attained. Overcoming these obstacles and reaching the fruition was a wonderful experience for us.

REFERENCES

[1]. http://www.bdeinc.com/blogs/understanding-benefits-applications-cnc-machining/
[2]. https://inkscape.org/en/
[3]. https://www.amazon.com/s/ref=nb_sb_noss_2?url=search-alias%3Doffice-products&field-keywords=XY+plotter
[4]. https://www.arduino.cc
[5]. https://components101.com/motors/28byj-48-stepper-motor
[6]. https://servodatabase.com/servo/towerpro/mg995