**VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELGAUM 59014**

****

Internet of Things Project

Report on

**“CNC Drawing Machine with Cloud Computation”**

By

**Deepthi Bhat (1BM16CS003) Aditi Awasthi(1BM16CS008)**

**Medhini Oak (1BM16CS047)**

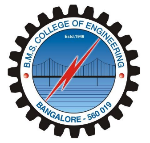
Under the Guidance of

**Dr. Jyothi S Nayak**

Associate Professor, Department of CSE

BMS College of Engineering

IoT Application Development carried out

****

Department of Computer Science and Engineering

BMS College of Engineering

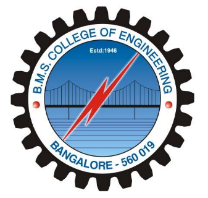
(Autonomous college under VTU)

P.O. Box No.: 1908, Bull Temple Road, Bangalore-560 019

2018-2019

**BMS COLLEGE OF ENGINEERING**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

****

***CERTIFICATE***

This is to certify that the Internet of Things project titled “**CNC Drawing Machine with Cloud Computation**” has been carried out by Deepthi Bhat (1BM16CS003), Aditi Awasthi(1BM16CS008) and Medhini Oak (1BM16CS047) during the academic year 2018-2019.

Signature of the guide

**Dr. Jyothi S Nayak**

Associate Professor

Department of Computer Science and Engineering

BMS College of Engineering, Bangalore

**Examiners**

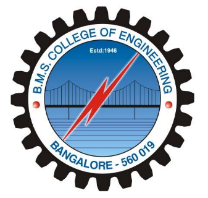
**Name Signature**

**1.**

**2.**

**BMS COLLEGE OF ENGINEERING**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

****

***DECLARATION***

We, Deepthi Bhat(1BM16CS003), Aditi Awasthi(1BM16CS008) and Medhini Oak (1BM16CS047) students of 5th Semester, B.E, Department of Computer Science and Engineering, BMS College of Engineering, Bangalore, hereby declare that, this IoT Application development work entitled "**CNC Drawing Machine with Cloud Computation**" has been carried out by us under the guidance of Jyothi S Nayak, Professor, Department of CSE, BMS College of Engineering, Bangalore during the academic semester Aug-Dec 2017

We also declare that to the best of our knowledge and belief, the development reported here is not from part of any other report by any other students.

Signature

Deepthi Bhat (1BM16CS003)

Aditi Awasthi(1BM16CS008)

Medhini Oak(1BM16CS047)

**Introduction**

**Objective of the project**

Creation of a Computer Numerical Control (CNC) Writing device to draw the basic form of a picture uploaded by the user to a hosted website, on paper using the concept of Internet of Things. This device will work using Arduino Uno microcontroller.

**Abstract**

As one who started using computers in the last decade, one would find the current resurgence of pen plotters somewhat nostalgic. The difference, of course, is that in this decade it is easier to make your own writing machine, which is what is accomplished in this project.

3D printers, laser cutters, water jet cutters, robot arms, delta robots, stewart platforms: all of these are examples of CNC machines. CNC machines need to move accurately and on command. Stepper motors are a great way to move accurately – they move a predictable amount and then stay where you put them. To command the stepper motors we need a way to easily turn our human desires into machine instructions into stepper motor steps. In effect we need our robot brain to be an interpreter and create the drawing on paper.

**Description**

CNC stands for Computer Numeric Control and typically refers to a machine whose operation is controlled by a computer. The most common usage of CNC, and the one relevant to us, is the name given to devices that, under computer control are able to cut, etch, mill, engrave, build, turn and otherwise perform manufacturing operations on various materials. By controlling a CNC machine through a PC it is possible for the user to design a product on-screen, convert it to CNC-readable code and then send that data to the CNC machine for it to produce a physical copy of the item designed.

A 2D CNC plotter offers a way to efficiently produce very large drawings. Pen plotters will be able to print by moving a pen or other writing device across the surface of a piece of paper. This means that plotters are vector graphics devices, rather than raster graphics. Pen plotters can draw complex line art, including text, but do so slowly because of the mechanical movement of the writing device such as pen.

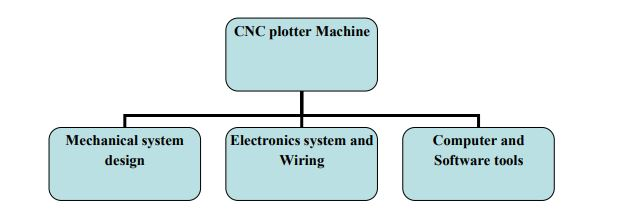
**Literature Review**

* Many techniques for the acceleration and deceleration of industrial robots and computer numerical control (CNC) machine tools have been proposed in order to make industrial robots and CNC machine tools perform given tasks efficiently. Although the techniques selecting polynomial functions can generate various acceleration and deceleration characteristics, the major problem is the computational load. Given a moving distance, and acceleration and deceleration intervals, a velocity pro le having the desired characteristics of acceleration and deceleration can be efficiently generated by using these coefficients. Several velocity profiles generated by the proposed technique will be applied to one single-axis control system. [1]
* Computer numerical control (CNC) machines are used to shape metal parts by milling, boring, cutting, drilling, and grinding. A CNC machine generally consists of a computer-controlled servo-ampli er, servo-motors, spindle motor, and various tooling. The machine can be programmed to shape a part by use of a front control panel. Finally, a recommended powering and grounding practice is presented to help eliminate power quality related operating problems with CNC machines while maintaining the safety requirements of electrical codes. [2]
* The stability, steady-state error analysis, damping factor, and setting time of discrete data drives for computer numerical control (CNC) machine tools are analyzed to obtain the necessary information for the design of a practical system. The stability of the drive is reviewed using Jury's test and the Mitrovic criterion. The variation of damping factor and settling time with respect to system parameters are presented based on the Mitrovic criterion. [3]

**Existing similar commercial and non-commercial products**

| **Sl.No** | **Name of the Project or Product (Existing)** | **Commercial or Non-Commercial** | **Features** |
| --- | --- | --- | --- |
| 1 | Mill | Commercial | Translates programs consisting of specific numbers and letters to move the spindle (or workpiece) to various locations and depths. Functions include: face milling, shoulder milling, tapping, drilling and some even offer turning.  <https://en.wikipedia.org/wiki/Mill> |
| 2 | 3-D printer | Commercial | Allows the creation of a physical object from a three-dimensional digital model, typically by laying down many thin layers of a material in succession.  <https://en.wikipedia.org/wiki/3D_printing> |
| 3 | Laser Cutter | Commercial | Uses a laser to cut materials, typically used for industrial manufacturing application. Laser cutting works by directing the output of a high-power laser most commonly through optics. The laser optics and CNC are used to direct the material or the laser beam generated.  <https://en.wikipedia.org/wiki/Laser_cutting> |

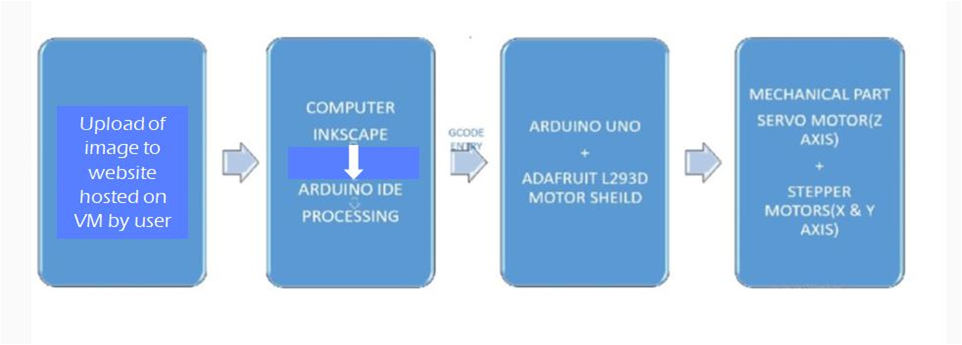
**Proposed Project**

****

The 2D Pen Plotter works based on the principle of Computer Numerical Control. It works with two stepper motors and a servo motor, wherein it plots the input given from the computer on the drawing board using ATMEGA 328p microcontroller. The plotter has a two axis control using two stepper motors and a special mechanism to raise and lower the pen, using a servo motor.

Using the Microsoft Azure Portal, a Windows 2012 R2 Server Virtual Machine is installed on the cloud. The virtual machine is configured with Internet Information Services(IIS) and FastCGI to facilitate online access to the project website.

The software used for programming the Arduino board are namely Inkscape(0.48.5), Processing (3.0.2), Arduino IDE. The correct and eﬃcient arrangement and proper use of the programs along with the circuit makes up an eﬃcient 2D CNC Plotter.

****

**Features**

* This system is designed to draw a simple image in digital format onto physical paper.
* A website is hosted on a cloud-based Windows Server 2012 Virtual Machine (using Microsoft Azure Portal) using IIS and FastCGI. This website can be accessed from any system by using the following IP address : “104.211.209.166”
* An SVG image is fed into the website that needs to be drawn
* The image is processed on the Virtual Machine to produce a G-code file.
* This G-code file can be sent to the arduino and the image will be drawn on the physical paper.
* This project implements a client-server model that can be used as a commercial service or for educational purposes in colleges or schools
* This system is suitable for drawing simple, small images.

**Advantages**

1. Low cost hardware as compared to mill/3-D printer/laser cutter.
2. DIY project with easy design made from household materials.
3. The processing of the image takes place on cloud.
4. Can be scaled for drawing very large images.
5. Client-server model can be used to commercialise this service.
6. Can be used in educational institutions to learn Computer Aided Design.

**Hardware and Software Requirements**

**Hardware Requirements**

The XY-plotter consists of two axes operating orthogonally to each other. Each axis includes a shaft that is rotated using a stepper motor, that is driven by an appropriate means. Additionally, a third axis, with limited motion capability is used to actuate the write head.

**Electronic Components:**

1. 2 PCs Nema 17 Stepper motor
2. Arduino Uno
3. L293D Motor Driver (Motor Driver IC)

The Device is a monolithic integrated high voltage, high current four channel driver designed to accept standard DTL or TTL logic levels and drive inductive loads (such as relays solenoids, DC and stepping motors) and switching power transistors. To simplify use as two bridges each pair of channels is equipped with an enable input. A separate supply input is provided for the logic, allowing operation at a lower voltage and internal clamp diodes are included. This device is suitable for use in switching applications at frequencies up to 5 kHz.

1. Mg 90 S Metal Gear Servo
2. 2-9v DC Cell or battery

**Mechanical Components:**

Part A: X-axis and Y-axis gantries

1. Wood for frame
2. Knitting Sticks
3. Sleeve(To hold the motor and the shart together)
4. Shaft
5. Pen body
6. Screws + Tie Ups

Part B: FLEXIBLE NIB-PEN

1. Ball ben
2. Spring
3. Thread
4. Compass grip

Tools Required:

1. Drill machine
2. Cutting Machine
3. Glue gun

**Software Requirements**

**1. Inkscape (Version 0.48.5).**

Inkscape is a free and open-source vector graphics editor which used to create or edit vector graphics such as illustrations, diagrams, line arts, charts, logos and complex paintings. Inkscape's primary vector graphics format is Scalable Vector Graphics (SVG);

SVG stands for Scalable Vector Graphics. Scalable refers to the notion that a drawing can be scaled to an arbitrary size without losing detail. The SVG standard is directed toward a complete description of two-dimensional graphics, including animation in an XML (eXtensible Markup Language) format.

Inkscape along with plugins is used to convert these SVGs into paths and paths into G-code. G-code is a language in which people tell computerized machine tools how to make something. The g-code provides instructions to a machine controller that tells the motors where to move, how fast to move, and what path to follow.

**2. Arduino IDE**

The Arduino project provides the Arduino integrated development environment (IDE), which is a cross-platform application written in the programming language Java. It originated from the IDE for the languages Processing and Wiring. It is designed to introduce programming to artists and other newcomers unfamiliar with software development. It includes a code editor with features such as syntax highlighting, brace matching, and automatic indentation, and provides simple one-click mechanism to compile and load programs to an Arduino board.

The Arduino IDE is used to write the program that will always sit on the Arduino(firmware) and will receive incoming gcode file and interpret each line. The motors are driven according to the gcode command received.

**3. Processing 3.0.2**

Processing is a simple programming environment that was created to make it easier to develop visually oriented applications with an emphasis on animation and providing users with instant feedback through interaction.

Processing consists of:

• The Processing Development Environment (PDE). This is the software that runs when you double-click the Processing icon. The PDE is an Integrated Development Environment (IDE) with a minimalist set of features designed as a simple introduction to programming or for testing one-oﬀ ideas.

• A collection of functions that support more advanced features such as sending data over a network, reading live images from a webcam, and saving complex imagery in PDF format.

• A language syntax, identical to Java but with a few modiﬁcations.

The Processing IDE is used to in this project to interact with the Arduino instead of the Serial monitor since we need to send the gcode file to the arduino for interpretation.

**4. Django 2.1 with IIS and FastCGI**

A Windows Server 2012 is setup in the Microsoft Azure Cloud Service Provider. The server is configured with Internet Information Services,FastCGI and Django.

Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Using django, a basic file-uploading website is created and hosted on a Windows Server 2012 using IIS and FastCGI.

Internet Information Services (IIS) for Windows Servers is a flexible, secure and manageable Web server for hosting anything on the Web. From media streaming to web applications, IIS's scalable and open architecture is ready to handle the most demanding tasks.

FastCGI is a [binary protocol](https://en.wikipedia.org/wiki/Binary_protocol) for interfacing interactive programs with a [web server](https://en.wikipedia.org/wiki/Web_server). FastCGI's main aim is to reduce the overhead associated with interfacing the web server and [CGI](https://en.wikipedia.org/wiki/Common_Gateway_Interface) programs, allowing a server to handle more web page requests per same amount of time.

The default Website of the server is configured to display the aforementioned django website, hence this website can be accessed using the address “104.211.209.166”, which is nothing but the IP address of the server.

**Cost analysis**

2 PCs Nema 17 Stepper motor Rs. 953.26

Arduino Uno R3 Rs. 445.00

Mg 90 S Metal Gear Servo Rs. 319.00

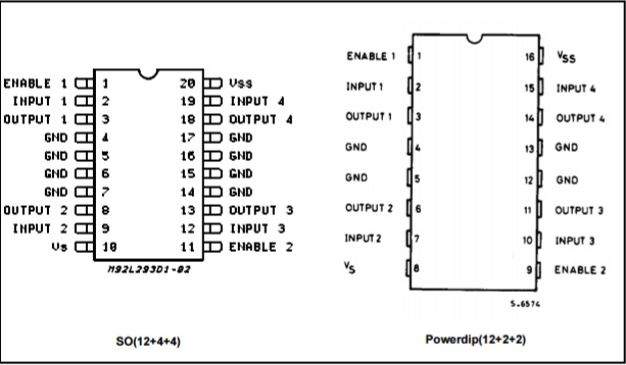
L293D Motor Driver Rs. 180.00

Total cost for mechanical parts Rs. 350.0

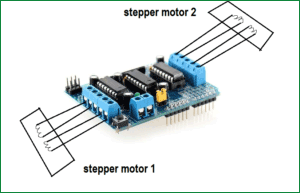
Cost of Azure VM Free for a limited time

**Total cost of the Project Rs. 2,247.26 + Cost of VM**

**Design**

**Architectural diagram** 

**Pin Configuration for L293D Shield**

****

**Setup for the project**

**Stage 1-Mechanical Setup**

1. Two gantries for each axis is constructed with a rotating shaft connected to the stepper motor and a stationary support rod.

2. The y-axis gantry is mounted on top of the x-axis gantry

3. Z- axis setup is achieved by attaching one end of a string to the pen refill which is connected to the spring, and the other end to a servo motor, to achieve an up/down motion.

**Stage 2-Electronic Setup**

1. The L293D motor driver shield compatible with the Arduino board is mounted on it.
2. The X-axis stepper motor is connected to Stepper 2 and Y-axis Stepper motor to Stepper 1 on the motor Driver Shield.
3. The Servo motor is connected to Servo 2 on the motor Driver Shield.
4. Square batteries are connected as the power supply to the setup.
5. The entire setup is plugged into a computer to send gcodes.

**Stage 3-Server setup**

1. Install Windows Server 2012 DataCenter on the Microsoft Azure Cloud Service Provider.
2. Connect from the physical machine to the VM using Remote Desktop Connection
3. Install Python, Wfastcgi and Django applications on the VM.
4. Enable the Web Server(IIS) service in the server-manager.
5. Create the Django website application in the “inetpub\wwwroot” directory
6. Configure the FastCGI settings for the Default website of the server in the IIS manager. Set the default path equal to the path of the project folder.
7. Type “104.211.209.166” in a browser window (in or out of the server) to access the website.

**Implementation**

**Generation of gcode-**

1. The user uploads an svg image file through the live django website set up on an VM hosted on Microsoft Azure Cloud Service Provider.
2. The image is fed into inkscape software that is loaded on the cloud, with the help of which the image can be moved/resized.
3. The image is converted into paths and the gcode of the image is generated. This gcode file is stored on the cloud until the drawing machine is ready to run.

**Interpretation of gcode-**

1. The Arduino Code(firmware) to interpret the gcode file is burned onto the microcontroller. Now the arduino is ready to receive gcode files to interpret.
2. The gcode file is copied onto the local system from the Virtual Machine.
3. Run the file sender code on the Processing IDE. A window appears. Keypress ‘p’ to select the port and ‘g’ to browse files. Click on the appropriate gcode file. Gcode interpretation begins.
4. The arduino processes each line of gcode and drives the motors to work appropriately.

**Source code**

**Arduino Code to interpret G-Code and drive the Motors**

#include <Servo.h>

#include <AFMotor.h>

#define LINE\_BUFFER\_LENGTH 512

char STEP = SINGLE ;

char BACK = BACKWARD;

char FRONT = FORWARD;

const int penZUp = 0;

const int penZDown = 80;

const int penServoPin =9 ;

const int stepsPerRevolution = 48;

Servo penServo;

AF\_Stepper myStepperY(stepsPerRevolution,1);

AF\_Stepper myStepperX(stepsPerRevolution,2);

struct point {

float x;

float y;

float z;

};

struct point actuatorPos;

float StepsPerMillimeterX = 208.0;

float StepsPerMillimeterY = 208.0;

float Xmin = 0;

float Xmax = 60;

float Ymin = 0;

float Ymax = 80;

float Zmin = 0;

float Zmax = 1;

float Xpos = Xmin;

float Ypos = Ymin;

float Zpos = Zmax;

void setup() {

Serial.begin( 9600 );

penServo.attach(penServoPin);

penServo.write(penZUp);

delay(100);

myStepperX.setSpeed(200);

myStepperY.setSpeed(200);

**}**

void loop()

{

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';

processIncomingLine( line, lineIndex );

lineIndex = 0;

}

lineIsComment = false;

lineSemiColon = false;

Serial.println("ok");

}

else {

if ( (lineIsComment) || (lineSemiColon) ) {

if ( c == ')' ) lineIsComment = false;

}

else if ( c == '(' ) {

lineIsComment = true;

}

else if ( c == ';' ) {

lineSemiColon = true;

}

else {

line[ lineIndex++ ] = c;

}}}}}}

void processIncomingLine( char\* line, int charNB ) {

int currentIndex = 0;

char buffer[ 64 ];

struct point newPos;

newPos.x = 0.0;

newPos.y = 0.0;

while( currentIndex < charNB ) {

switch ( line[ currentIndex++ ] ) {

case 'G':

buffer[0] = line[ currentIndex++ ];

buffer[1] = '\0';

Serial.print(atoi(buffer));

switch ( atoi( buffer ) ){

case 0:

case 1:

char\* indexX = strchr( line+currentIndex, 'X' );

char\* indexY = strchr( line+currentIndex, 'Y' );

drawLine(newPos.x, newPos.y );

actuatorPos.x = newPos.x;

actuatorPos.y = newPos.y;

break;

}

break;

case 'M':

buffer[0] = line[ currentIndex++ ];

buffer[1] = line[ currentIndex++ ];

buffer[2] = line[ currentIndex++ ];

buffer[3] = '\0';

switch ( atoi( buffer ) ){

case 300:

**{**

char\* indexS = strchr( line+currentIndex, 'S' );

int Spos = atoi( indexS + 1);

if (Spos == 30) {

penDown();

Serial.println("Pen Down");

}

if (Spos == 50) {

penUp();

Serial.println("Pen Up");

}

break;

}

default:

Serial.print( "Command not recognized : M");

Serial.println( buffer );

}}}}

void drawLine(float x1, float y1) {

if (x1 >= Xmax) {

x1 = Xmax;

}

if (x1 <= Xmin) {

x1 = Xmin;

}

if (y1 >= Ymax) {

y1 = Ymax;

}

if (y1 <= Ymin) {

y1 = Ymin;

}

x1 = (int)(x1\*StepsPerMillimeterX);

y1 = (int)(y1\*StepsPerMillimeterY);

float x0 = Xpos;

float y0 = Ypos;

long dx = abs(x1-x0);

long dy = abs(y1-y0);

char sx = x0<x1 ? FRONT : BACK;

char sy = y0<y1 ? FRONT : BACK;

long i,over = 0;

if (dx > dy) {

for (i=0; i<dx; ++i) {

myStepperX.step(1,sx,STEP);

over+=dy;

if (over>=dx) {

over-=dx;

myStepperY.step(1,sy,STEP);

}}}

else {

for (i=0; i<dy; ++i) {

myStepperY.step(1,sy,STEP);

over+=dx;

if (over>=dy) {

over-=dy;

myStepperX.step(1,sx,STEP);

}}}

Xpos = x1;

Ypos = y1;

}

void penUp() {

penServo.write(penZUp);

Zpos=Zmax;

}

void penDown() {

penServo.write(penZDown);

Zpos=Zmin;

}

**Conclusion**

In this project, it was attempted to develop X-Y plotter that accurately synchronizes with the Arduino software system for better response on the movement of X and Y axis. Different IDE and different languages have been tried and used to complete this project to meet the objectives. With a lot of new technologies being developed nowadays, this project serves to provide a good platform for future development for XY plotter system and even other systems.

**Future Enhancements**

**PCB Mill**

A PCB Mill is a device that etches out a pattern on a copper clad board such that it makes a Printed Circuit Board (PCB). When prototyping, the delay and setup costs associated with sending a layout to a manufacturer can often mean days of downtime. While this may not seem costly at ﬁrst, it can prove to be a signiﬁcant nuisance since most boards contain a wiring bug that was overlooked or misunderstood and must then be remade.

This project can be extended and made more accurate to produce an efficient Printed Circuit Board Milling Machine.

**References**

[1] Jae Wook Jeon and Young Youl Ha, A Generalized Approach for the Acceleration and Deceleration of Industrial Robots and CNC Machine Tools, IEEE Transactions on Indus-trial Electronics, Vol. 47, No. 1, February 2000, pp. 133-139

[2] Allen G. Morinec, Power Quality Considerations for CNC Machines: Grounding, IEEE Transactions on Industrial Electronics, Vol. 38, No. 1, January/February 2002, pp. 3-11.

[3] Venkatram Ramachandran, Evaluation of Performance Criteria of CNC Machine Tool Drive System, IEEE Transactions on Industrial Electronics, Vol. 45, No. 3, June 1998, pp. 462-468