C.T.E.S.'s
N. G. Acharya & D. K. Marathe College
Of Arts, Science & Commerce
NAAC ACCREDITED "A" GRADE

PROJECT REPORT ON

"COMPUTER NUMERICAL CONTROL MACHINE"

SUBMMITTED BY

Mr. DNYANESHWAR VITTHAL SHEKADE

UNDER THE GUIDANCE OF

Ms. SHRADDHA D. DAWAN

TO

UNIVERSITY OF MUMBAI
IN PARTIAL FULLFILMENT OF
T.Y.B. Sc.(COMPUTER SCIENCE)

SEM-V
ACADEMIC YEAR 2020-2021

C.T.E.S.'s

N. G. Acharya & D. K. Marathe College
Of Arts, Science & Commerce

NAAC ACCREDITED "A" GRADE

CERTIFICATE

This is to certify that ""**Mr. DNYANESHWAR VITTHAL SHEKADE**" Examination Seat No. _____ Student of N. G. ACHARYA & D. K. MARATHE COLLEGE has successfully completed the project work entitled **"COMPUTER NUMERICAL CONTROL MACHINE"** during the academic year 2020-2021. In partial fulfilment of the requirement for the award of Degree in Bachelor of Computer Science under University Of Mumbai.

It is further certified that student completed all required phases of project.

PLACE: Mumbai
DATE: _____

_____
PROJECT GUIDE
(Ms. Shraddha D. Dawan)

_____
HEAD OF DEPARTMENT
(Ms. Archana Jadhav)

_____
INTERNAL EXAMINAR

_____
EXTERNAL EXAMINAR

ACKNOWLEDGEMENT

A project is always a result of the amalgamation of various ideas and support from countless people. I would like to acknowledge the contribution made by them. I am glad to represent my project on **"COMPUTER NUMERICAL CONTROL MACHINE"** which help to humans even companies within the same industry may find different applications for CNC machining in their needs.

- Medical Industry
- **Aerospace Industry**
- **Transportation Industry**.
- **Oil and Gas Industry.**
- **Military and Défense Industry.**

This project has been a learning and challenging process for me.

I express my sincere thanks to the principal **Dr. VIDHYAGAURI LELE** also express my thanks to **Ms. PRADNYA CHAVAN** and staff of the Computer Science Section/Department. I would like to thank my Project Guide **Ms. SHRADDHA D. DAWAN** and Head Of Department **Ms. ARCHANA JADHAV** for her encouragement and guidance, which helped me in completing the project.

I immensely grateful to all those because without their inspiration, constant prompting and useful suggestions.

I sincerely appreciate the help provided by those in the careful preparation of the project.

Sincere thanks from,
Mr. Dnyaneshwar Vitthal Shekade.

Tables of Contents

Introduction

1.1 Overview of the Project

Description:

CNC machining is a manufacturing process in which programmable computer software codes dictates the movement of factory tools and machinery. The process can be used to control a range of complex machinery, from grinders and lathes to mills and routers. With CNC machining, three-dimensional cutting/writing/3d parts making tasks can be accomplished in a single set of prompts.

Short for "computer numerical control," the CNC process runs in contrast to — and thereby supersedes — the limitations of manual control, where live operators are needed to prompt and guide the commands of machining tools via levers, buttons and wheels. To the onlooker, a CNC system might resemble a regular set of computer components, but the software programs and consoles employed in CNC machining distinguish it from all other forms of computation

In CNC, machines are operated via numerical control, wherein a software program is designated to control an object. The language behind CNC machining is alternately referred to as G-code, and it's written to control the various behaviour's of a corresponding machine, such as the speed, feed rate and coordination.

Basically, CNC machining makes it possible to pre-program the speed and position of machine tool functions and run them via software in repetitive, predictable cycles, all with little involvement from human operators. Due to these capabilities, the process has been adopted across all corners of the manufacturing sector and is especially vital in the areas of metal and plastic production.

For starters, a 2D or 3D CAD drawing is conceived, which is then translated to computer code for the CNC system to execute. After the program is inputted, the operator gives it a trial run to ensure no mistakes are present in the coding.

The CNC machine make work easy and accurate in less time as compare to humans and this a one Step towards the machine automation.

Motion is controlled along multiple axes, normally at least two (X and Y) and a tool spindle that moves in the Z (depth). The position of the tool is driven by direct-drive stepper motor or servo motors in order to provide highly accurate movements, or in older designs, motors through a series of step down gears. Open-loop control works as long as the forces are kept small enough and speeds are not too great. On commercial metalworking machines, closed loop controls are standard and required in order to provide the accuracy, speed, and repeatability demanded

1.2 Project Synopsis

1.2.1 Title of Project

Title : Computer Numerical control Machine

1.2.2 Category of the Project

Category: Internet of Things open source

1.2.3 Tools/Platform Requirements (Runtime Environment Specification)

Hardware

- Arduino nano : 1
- Stepper motor from old DVD rom : 2
- Servo motor : 1
- IC L293D Motor Driver : 2
- 2X DVD Drive : 2
- 5v 700 mA Power Supply : 1
- Acrylic sheet 3mm : 15x15cm
- Blank PCB : 5x5 cm
- Jumper wires : 8 pieces
- Shouldering machine : 1

Laptop

- Processor : i5 2GHz speed
- Ram : 4GB
- HDD : 128 GB SSD

Software

- Operating System : Linux, Windows, MacOS X
- Front End : Arduino ide, Processing 3
- Coding language : C, Java
- Backend : Gcode

1.3

Problem Definitions

I have faced two major problems while working on this project:

1. As I are giving power supply by SMPS (Switch Mode Power Supply) I am getting power supply of 12 volts. It's caused us major losses of the parts.

The parts that had been short-circuited by giving false voltage are as follows:

   a. 2x stepper motor drives
   b. 1x Arduino
   c. 2x L293D Motor driver shield

2. Then biggest challenge is making shield from blank pcb.
3. The second major problem was to identify the pair of rights wires of the stepper motor to connect L293D MOTOR driver

There were many other problems occurred during the project assembly which consists of dismantling the cd-drives casing from selection and connecting the wires required for this project as well.

1.4

Objectives

1. The main objective of this project is to provide plotting/drafting operation as an alternative for traditional manual operations done by manually operated mechanism.

2. The project is used as a principle which resents the plotting operation which can be replaced by any other cutting tools such as Drilling machine, Milling cutters, even Laser cutter, etc.

3. Computer Numerical control plotter is a cheaper CNC plotter made up by old DVD ROMS of and can be easily made by scrapped/used hardware parts.

4. The plotting can also be used for drafting purposes.

1.5

Working Principle

"It works on the G-codes generated by open source and free software's.

First we have to write code in Arduino Ide and upload to Arduino Board For correct inputs.

and then we have to write Java code for executing Gcode input to draw the output by the CNC Plotter.

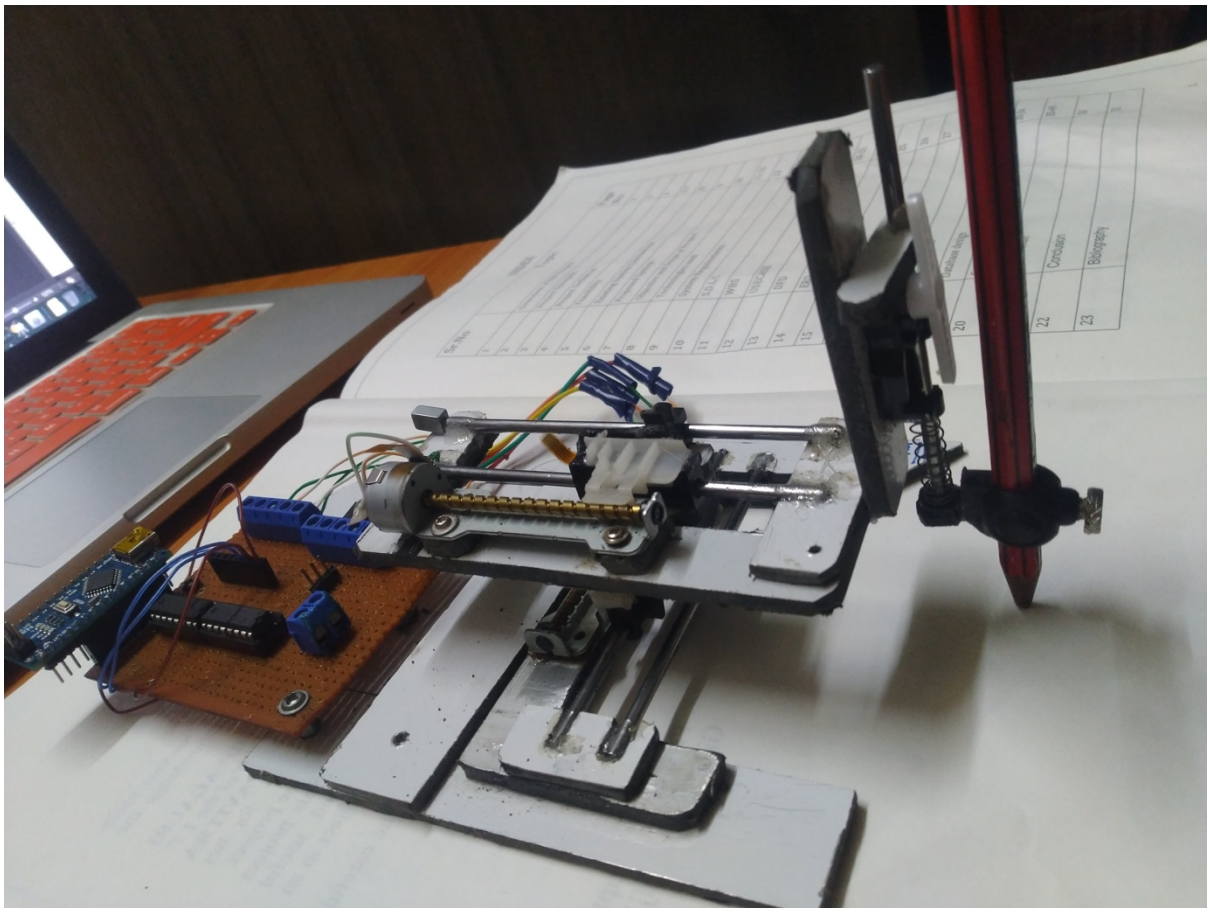


Fig.1: Working principle of CNC PLOTTER

1.6

Advantages

1. It’s an open source program and thus a free one, it supports various kinds of hardware and configurations and it’s very easy to use.
2. The stepper motor never misses any step. Hence the whole operation is reliable.
3. Project is Small in size so we can put CNC machine easily where we want to draw or plot something.
4. The whole project is cheaper and can be easily constructed through used parts of hardware form computers.

1.7 Limitations

Apart from the great advantages, there are major limitations too for CNC PLOTTER.

1. As per the size and capacity of the project, it can't operate in larger load applications and cutting forces as well. It cannot withstand higher cutting forces while cutting and the higher weight of cutting tools as well.

2. Because of the circuit used the shapes that can be draft are limited. Arduino Uno capacity is limited up-to limited shapes considering of Grbl software used as well.

3. The maximum height of the cd-drive carriage is limited up to 50-60 mm. Hence 3D projects can be small dimensions only.

1.8:

Applications

1. 3D printing: By replacing the pen with 3d extrusion pen, we can perform 3D modelling which is the fastest growing technology in present.

2. Laser engraving: By replacing the pen with laser machine we can perform laser engraving operation for engraving different logo, trademarks, bench marks, etc.

3. Laser cutting: Instead of pen we can use laser cutters for high precision machining operations.

4. Milling operations: We can even perform different milling operations such as slotting, grooving, etc. provided with overarm and arbor.

2

Market Survey

2.1: For Product

"Computer Numerical Control Machine " is nothing but the principle project based on 3D printers and mini CNC cutting machines as well. 3D printers or mini CNCs are available at various prices in market.
Following are some lists of product available in market:

Sr. No.	Product name	Price
1.	Precision 3D printer	₹ 60,000
2.	Prusa i3 3d printer DIY	₹ 18,000
3.	Prusa Rework 3D printer kit	₹ 34,000
4.	Shapeoko 2 DIY Desktop CNC kit	₹ 40,000
5.	Laser Engraver Engraving Cutting Machine DIY CNC Kit	₹ 31,000

There are number of CNC 3D printers/ mini CNC machines which cost above ₹ 30,000. Hence our "Computer Numerical Control Machine" which is the principle project cost just ₹ 1000 -₹ 2000 approximately.

2.1 :

For various parts used

Different parts used in the project is selected depending upon its requirement and its market pricing by visiting different places in Grant Road and online search as well.

Following are the list of parts with their pricing:

Sr. No.	Part Name	Price
1.	Arduino Nano Online and at shops	₹ 400 - ₹ 500
2.	L293D Motor Driver Ic	₹ 220 - ₹ 500
3.	Stepper motor drive from DVD-rom	₹ 130 - ₹ 165
4.	Acrylic Sheet	₹ 130 - ₹ 165
5.	Servo motor	₹ 150- ₹ 250
6.	5v power Supply	₹ 80 - ₹ 90
7.	Blanck PCB	₹ 100- ₹ 150

3

Technical Specification of project

3.1 Arduino Nano

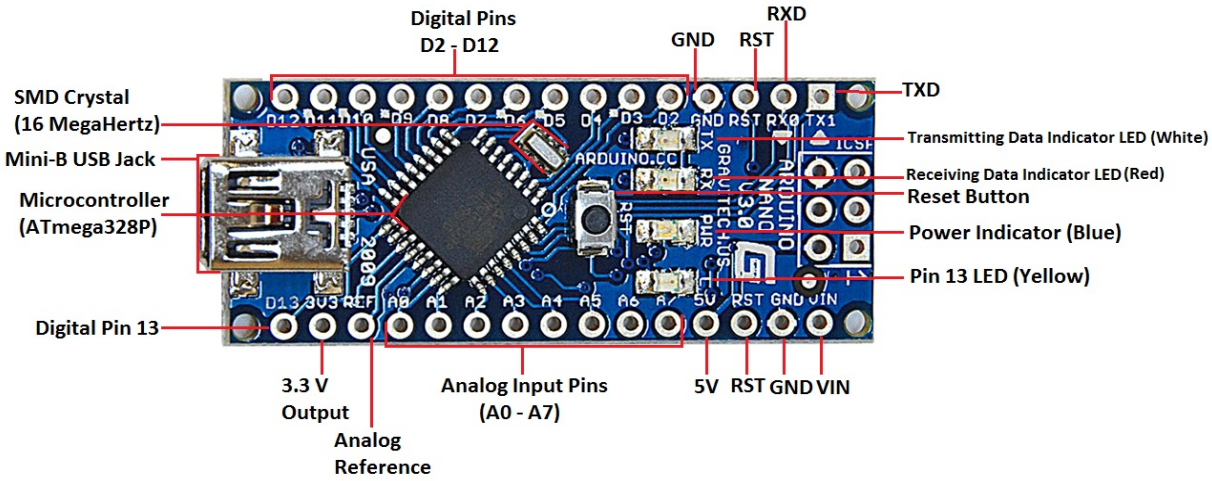


Fig.3.1 Arduino Nano Pinout

Arduino is the heart of the project, hence it should be selected by proper selection. In our project we have selected Arduino Nano which run on C,C# language. The programming of an Arduino nano can be done using the Arduino software. Click the Tools option and select the nano board. Microcontroller ATmega328 over the Nano board comes with pre-programmed with a boot loader. This boot loader lets to upload new code without using an exterior hardware programmer. The communication of this can be done with the STK500 protocol. Here the boot loader can also be avoided & the microcontroller program can be done using the header of in-circuit serial programming or ICSP with an Arduino ISP.

Microcontroller	ATmega328P – 8 bit AVR family microcontroller
Operating Voltage	5V
Recommended Input Voltage for Vin pin	7-12V
Analog Input Pins	6 (A0 – A5)
Digital I/O Pins	14 (Out of which 6 provide PWM output)

DC Current on I/O Pins	40 mA
DC Current on 3.3V Pin	50 mA
Flash Memory	32 KB (2 KB is used for Bootloader)
SRAM	2 KB
EEPROM	1 KB
Frequency (Clock Speed)	16 MHz
Communication	IIC, SPI, USART

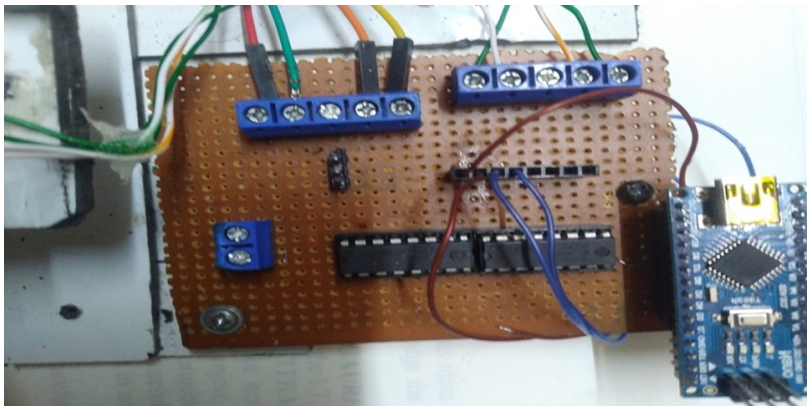
Pinout chart of Arduino nano

Pin Category	**Pin Name**	**Details**
Power	**Vin, 3.3V, 5V, GND**	**Vin:** Input voltage to Arduino when using an external power source (6-12V). **5V:** Regulated power supply used to power microcontroller and other components on the board. **3.3V:** 3.3V supply generated by on-board voltage regulator. Maximum current draw is 50mA. **GND:** Ground pins.
Reset	**Reset**	Resets the microcontroller.
Analog Pins	**A0 – A7**	Used to measure analog voltage in the range of 0-5V
Input/Output Pins	**Digital Pins D0 - D13**	Can be used as input or output pins. 0V (low) and 5V (high)
Serial	Rx, **Tx**	Used to receive and transmit TTL serial data.
External Interrupts	2, 3	To trigger an interrupt.

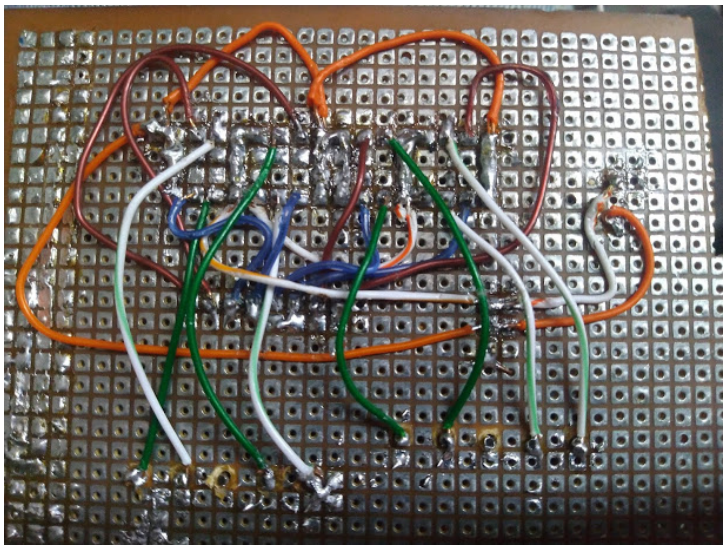
PWM	3, 5, 6, 9, 11	Provides 8-bit PWM output.
SPI	10 (SS), 11 (MOSI), 12 (MISO) and 13 (SCK)	Used for SPI communication.
Inbuilt LED	**13**	To turn on the inbuilt LED.
IIC	A4 (SDA), A5 (SCA)	Used for TWI communication.
AREF	**AREF**	To provide reference voltage for input voltage.

3.2

CNC Shield Circuit



Img. 3.2 CNC Shield which created for project



Img. 3.3

Img. Project connection between L293D and motor

3.2.1

CNC Shield Circuit Diagram

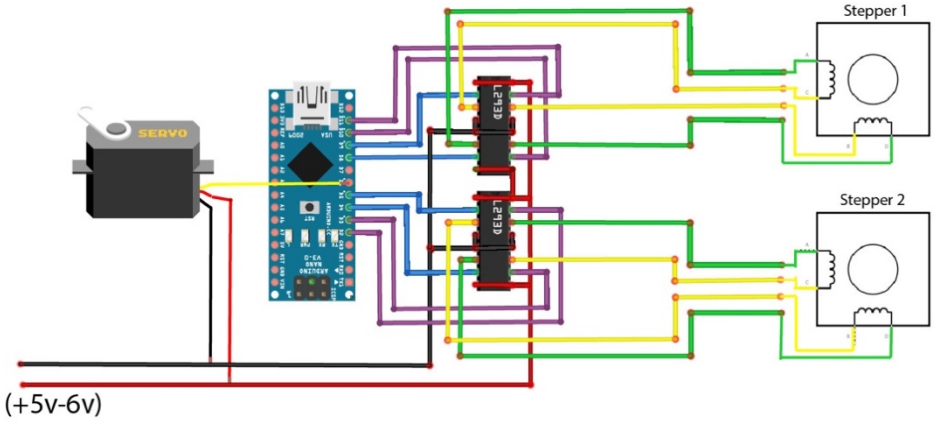


Fig. circuit diagram form CNC Shield

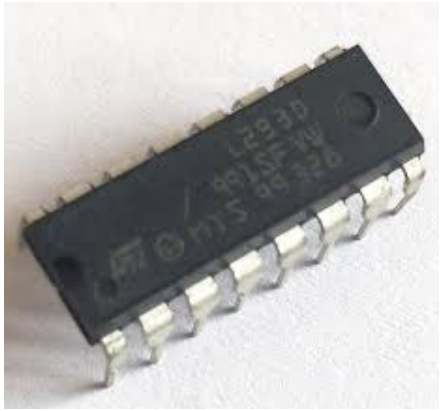
This is the complete circuit diagram, I connect everything according to the circuit diagram.
For the connection I use 2x L293D ICs.
2x female sockets for ICs.
2x sockets for connecting motor wires .
1x socket for connecting power supply.
1x socket for connecting Arduino Nano on Board
4x purple wires for connect Arduino to L293D ICs
4x blue wires for connect Arduino nano to L293D
1x black wire for negative Power supply
1x red wire for 5V positive Power supply
4x green wire for connect stepper motor
4x yellow wire for connect stpper motor

3.3

L293D Motor Driver

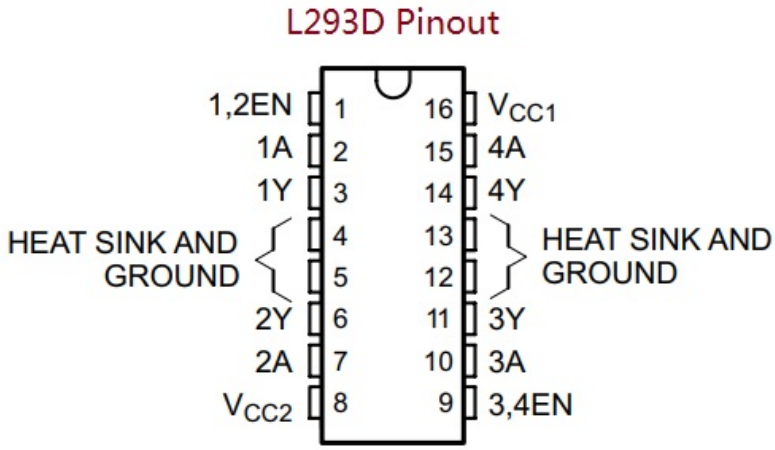
I use 2x L293D for giving inputs and output to the stepper Motor As well as servo motor.

The L293D is designed to provide bidirectional drive currents of up to 600-mA at voltages from 4.5 V to 36 V. Both devices are designed to drive inductive loads such as relays, solenoids, DC and bipolar stepping motors, as well as other high-current/high-voltage loads in positive-supply applications.



3.2.1

L293D Motor Driver Circuit Diagram



3.4

Servo Motor

A **servomotor** is a rotary actuator or linear actuator that allows for precise control of angular or linear position, velocity and acceleration. It consists of a suitable **motor** coupled to a sensor for position feedback.

I am using servo Motor for pen Up And Down Moments

Img . Servo motor

3.4.1

Circuit Diagram Servo Motor

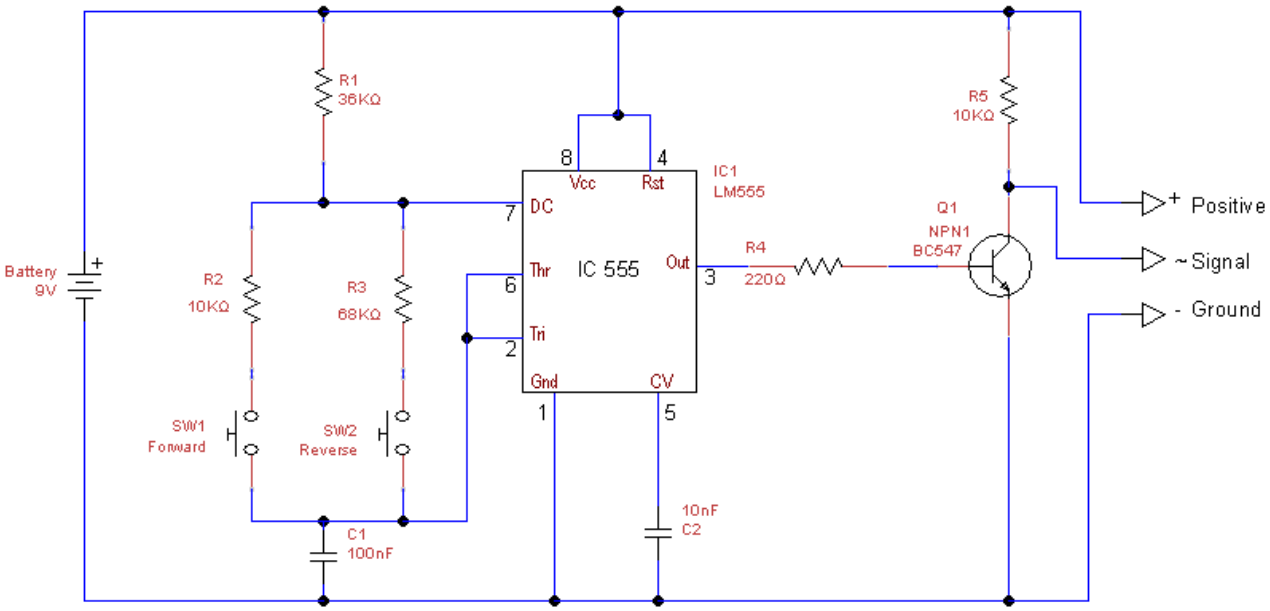


Fig. Circuit Diagram for Servo motor

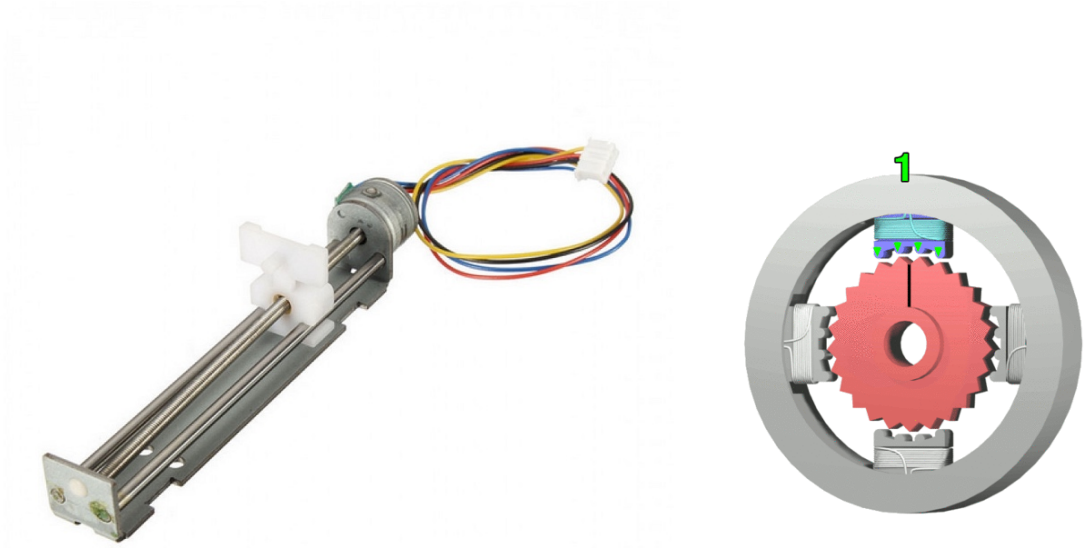
3.5

Stepper Motor

A stepper motor, also known as step motor or stepping motor, is a brushless DC electric motor that divides a full rotation into a number of equal steps. The motor's position can then be commanded to move and hold at one of these steps without any position sensor for feedback (an open-loop controller), as long as the motor is carefully sized to the application in respect to torque and speed

A stepper motor is used to achieve precise positioning via digital control. The motor operates by accurately synchronizing with the pulse signal output from the controller to the driver. Stepper motors, with their ability to produce high torque at a low speed while minimizing vibration, are ideal for applications requiring quick positioning over a short distance.

Stepper motors enable accurate positioning with ease. They are used in various types of equipment for accurate rotation angle and speed control using pulse signals. Stepper motors generate high torque with a compact body, and are ideal for quick acceleration and response. Stepper motors also hold their position at stop, due to their mechanical design. Stepper motor solutions consist of a driver



3.5.1

Circuit Diagram Stepper Motor

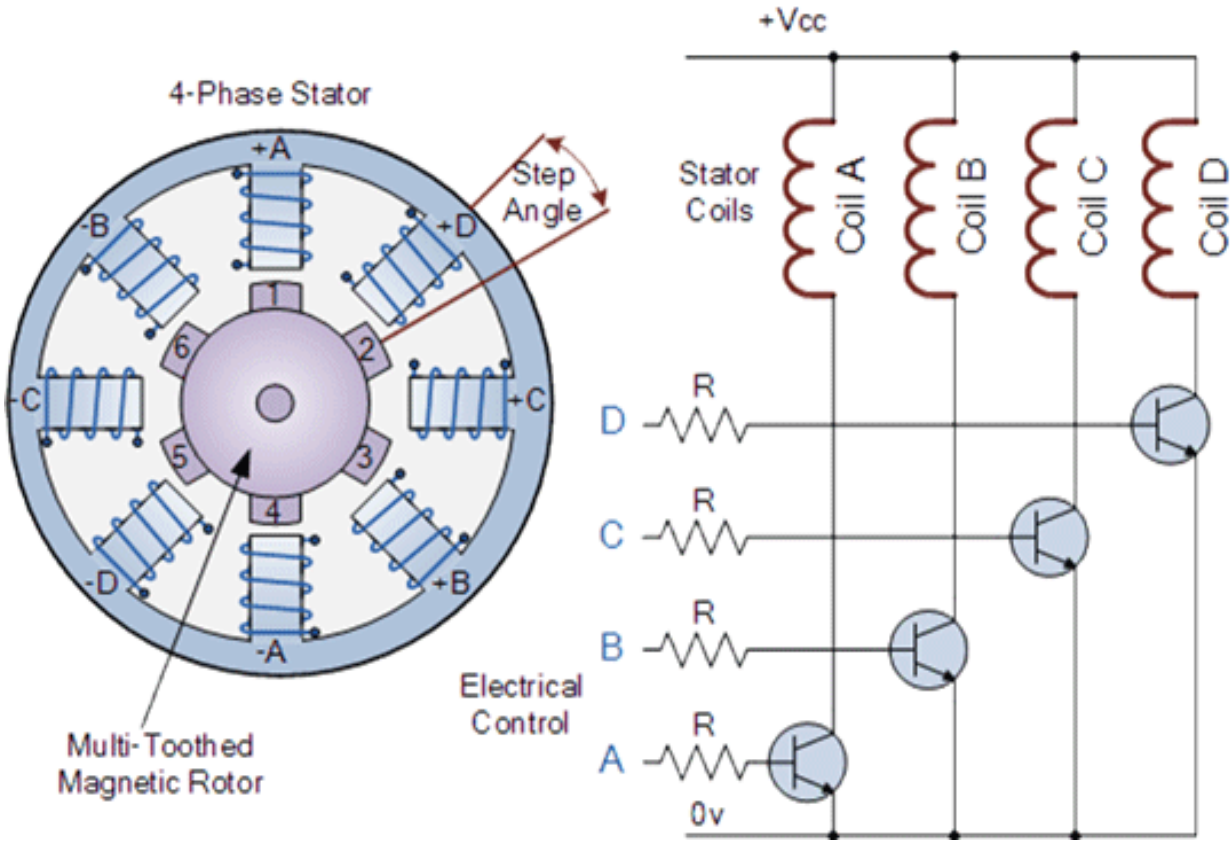


Fig.3.4.1 Stepper motor circuit

For better understanding of step rotation of stepper motor we are considering a four stage stepper motor as shown in figure 3.4.1.

Now consider, for an example, all coils are magnetized at a time. The rotor experiences forces of equal magnitude from all around it and so it does not move. Because all are of equal magnitude and are expressing opposite direction. Now if the coil D only magnetized, the teeth 1 on rotor experiences an attractive force towards +D and teeth 5 of rotor experiences a repulsive force opposing the –D, theses two forces are represents an additive force clock wise. So the rotor moves to complete a step. After that it stops for the next coil to energize to complete next step. This goes on until the four steps are complete. For the rotor to rotate this cycle of pulsing must be going on.

As explained before, the preset is set to a value for a certain frequency of pulses. This clock is fed to the decade counter to get regular outputs from it. The outputs from decade counter are given to transistors to drive the high power coils of stepper motor in sequential order. The tricky part is, once a sequence is complete say 1, 2, 3, 4 the stepper motor completes four steps and so it is ready to start again however the counter has a capacity to go for 10 and so it goes on without interruption. If this happens the stepper motor must wait till the counter completes

its cycle of 10 which is not acceptable. This is regulated by connecting RESET to Q4 so when counter goes for five count it resets itself and starts from one, this starts the sequence of stepper.

So this is how the stepper continuous it's stepping and so the rotation happens. For a two stage the RESET pin must be connected to Q2 for the counter to resets itself in the third pulse. This way one can adjust the circuit to drive ten step stepper motor.

3.6

Project Setup

It consists of all parts i.e. CD-ROMs, Acrylic Sheet , 5v Power Supply , electrical circuit that is Arduino, CNC shield L293d ic , stepper Motor etc.

Following are the Specification of whole project setup:

Parameters	**Specification**
Size	19 x 15cm
Height	9 cm
Acrylic Sheet	mm

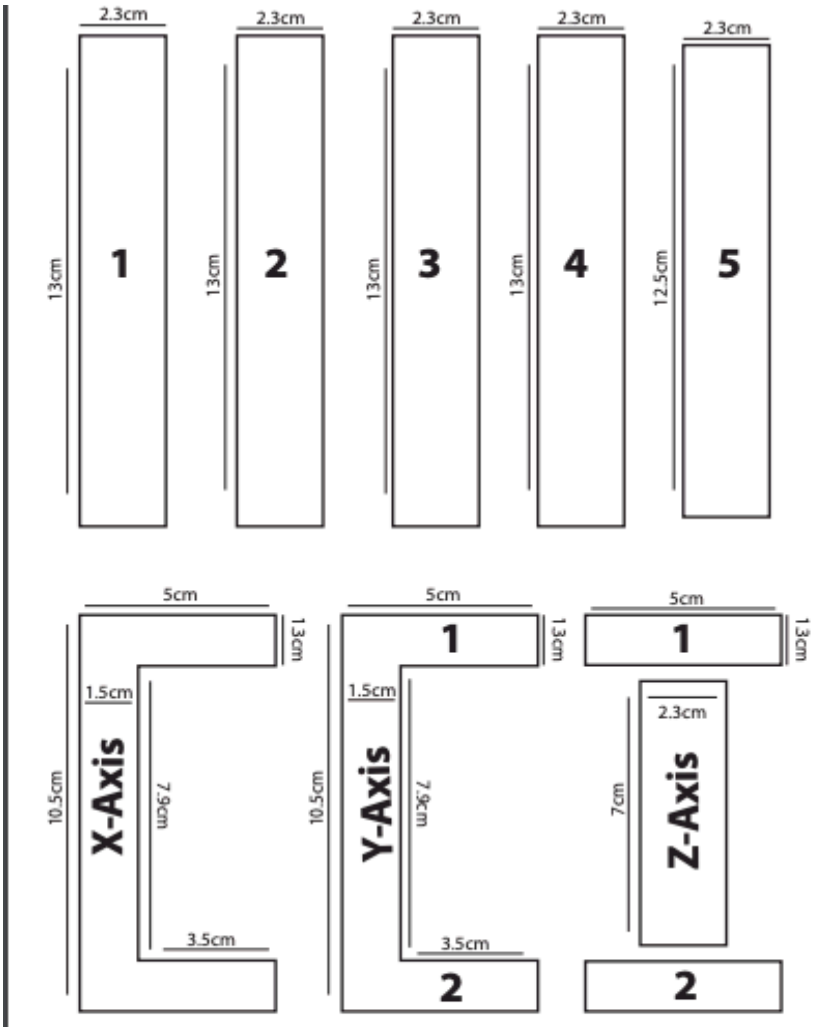
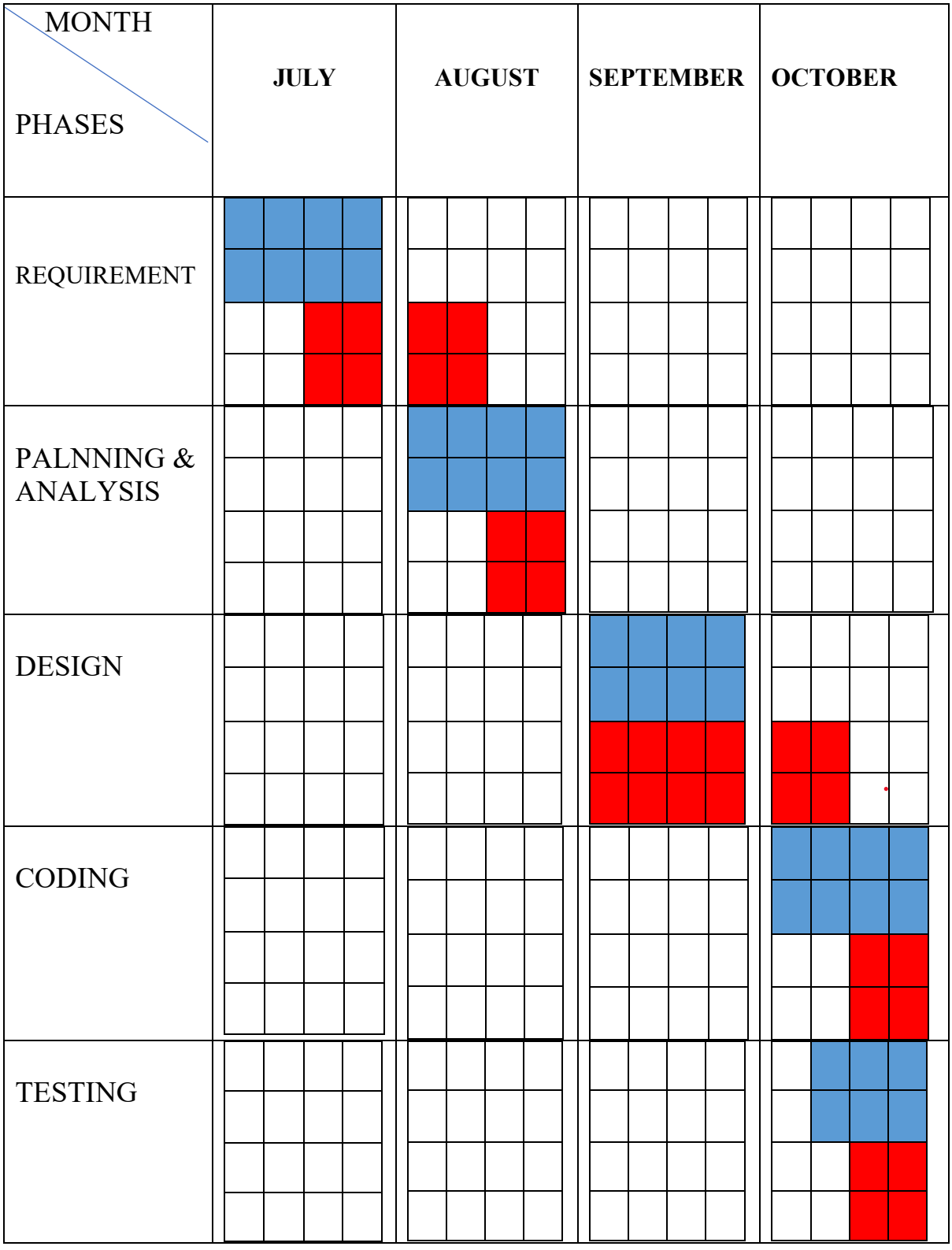


Fig 3.5 Size mark for cutting acrylic sheet

4

Gantt Chart



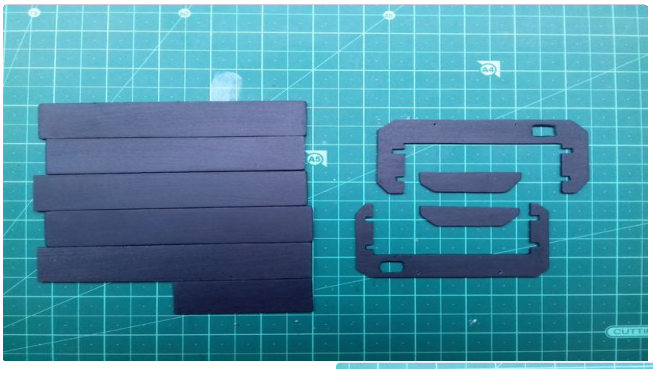
5

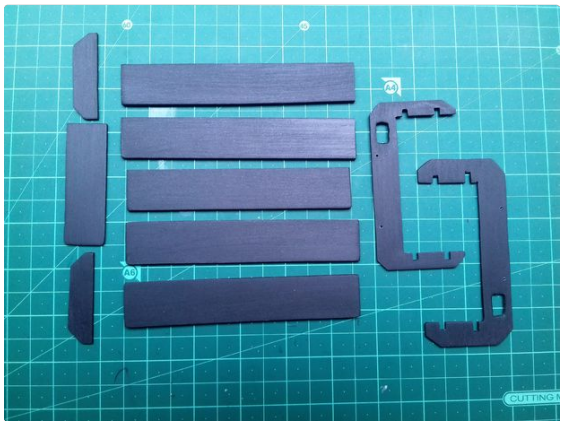
Design

5.1 Making the Pieces for Frame

Here I'm using the acrylic sheet to make the frame for the CNC plotter. It is a very good choice for making the enclosure for every Project. Personally, I frequently use this sheet to make the enclosure for my Project. It is easy to cut, bend and sand. It is also so affordable. I use the following process to make pieces for the frame.

1. Taking the measurement of the acrylic sheet for the enclosure to my need and mark it with a marker pen.
2. Then use your handsaw to cut the pieces.
3. Use sandpaper to smooth the pieces.

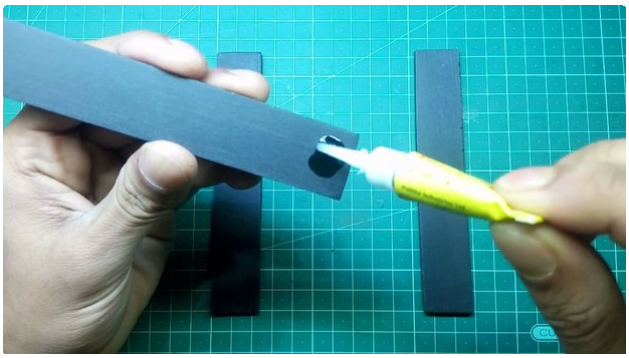


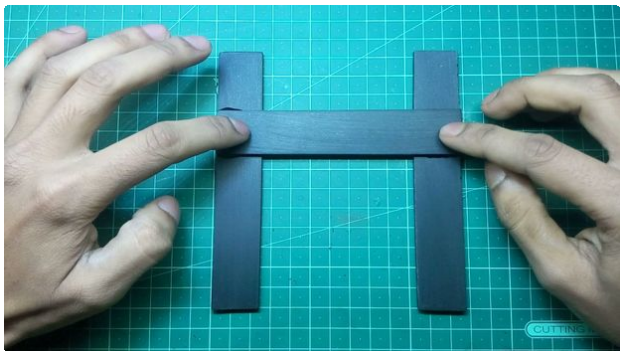


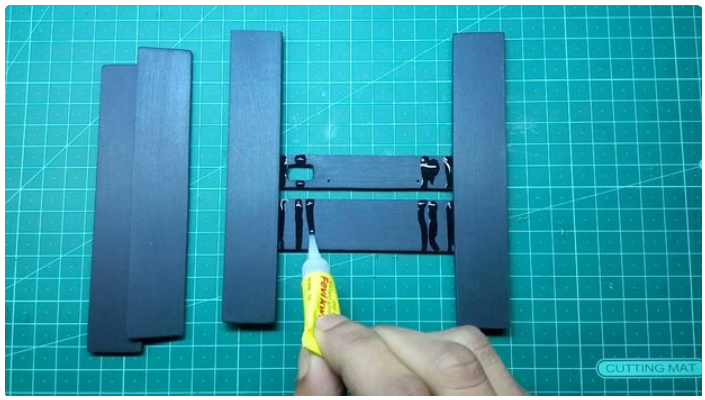
5.2 Making the Base

In this step, I'm going to make the base for my CNC plotter. So follow the step carefully.

1. First, apply super glue on one piece of acrylic then attach this piece with another two pieces.
2. Again use the super glue to attach the X-axis frame.
3. Then attach another two pieces for better support.





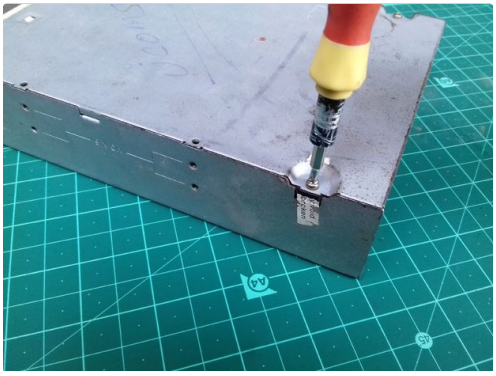


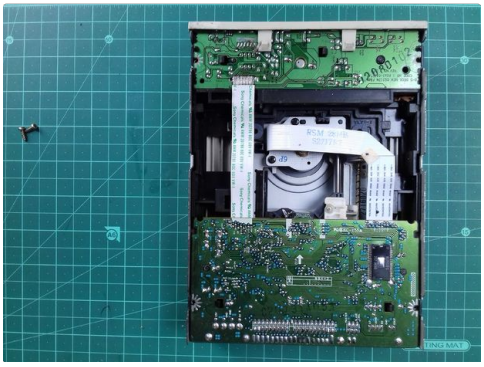
5.2

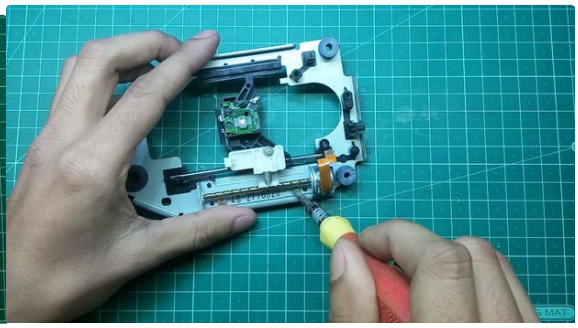
Removing the Stepper Motor

We need two old DVD writers for this project. I am finding old and broken DVD writer on local computer hardware store. Usually, they do not use this DVD writer so they can give me it at the very low price. I got one for free and I had another one laying around. So I decided to use it for my CNC project.

1. **First, disassemble two DVD driver and take off the stepper motors.**
2. **Then use your screwdriver to open the stepper motor and rails from the frame.**



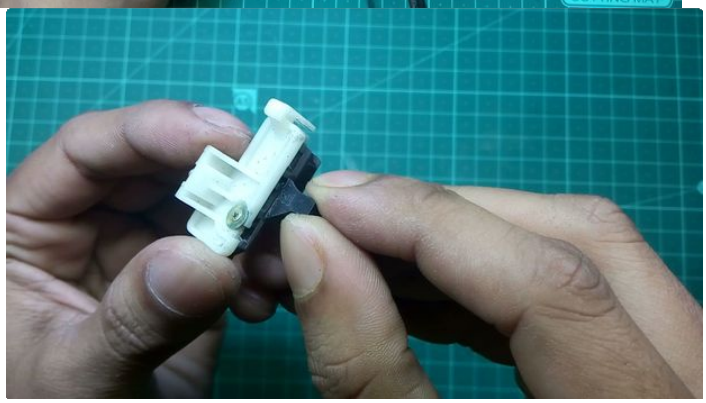


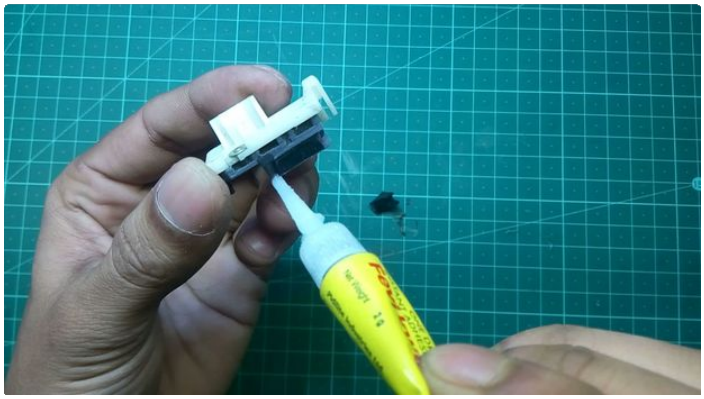


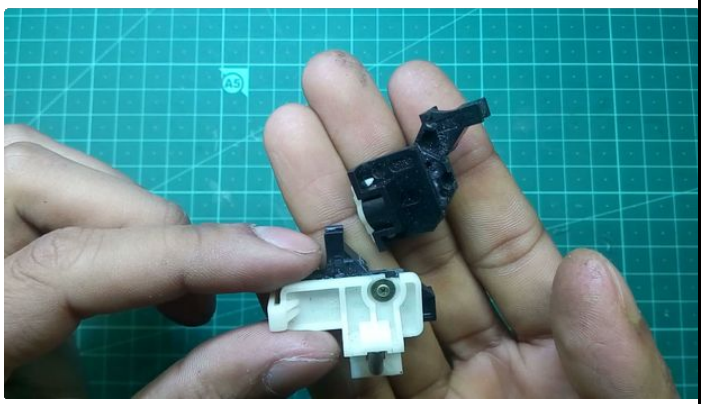
5.3

Making the Slider for X and Y Axis

1. **First, taking off the two sliders from their rails.**
2. **Use handsaw to cut the two little pieces from the slider**
3. **Then attach the two pieces using super glue.**





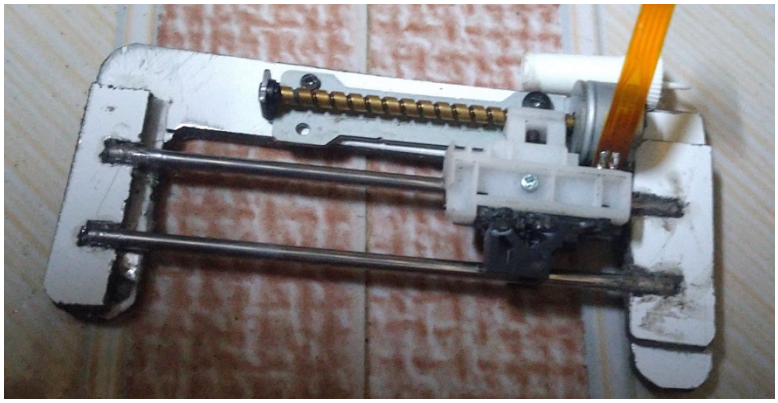


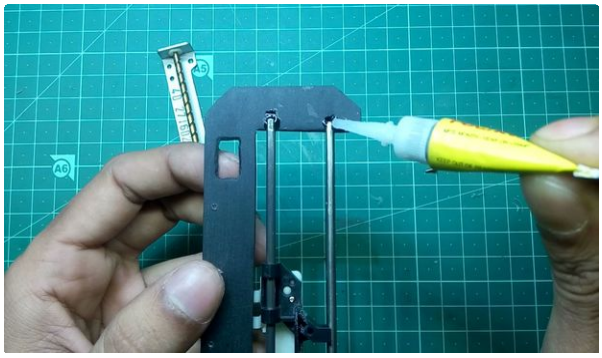
5.4

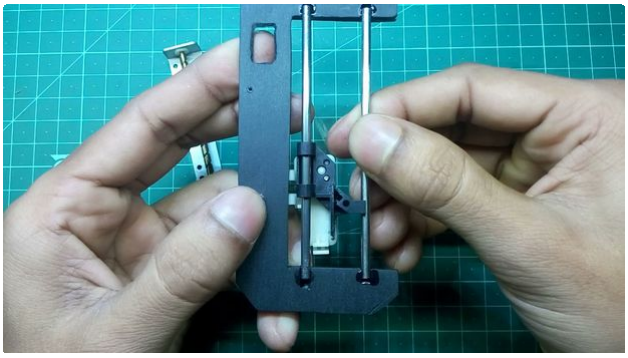
Attaching the Sliding Rods

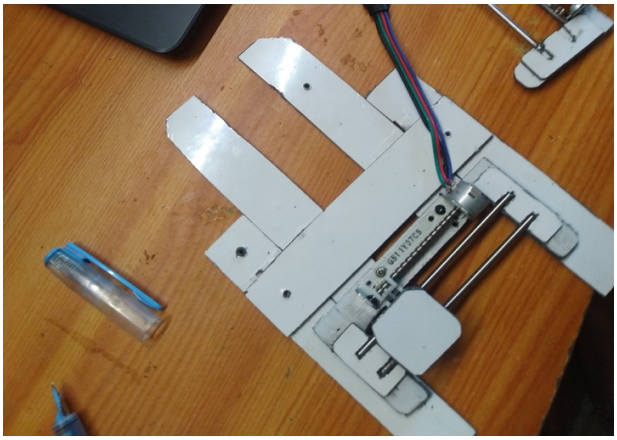
In this step, l attach the sliding rods to its frame. So I follow the step below.

1. First, take one sliding rod and insert it into the slider.
2. Then apply super glue to attach the sliding rod with its frame and hold it still for some time.
3. Use the above method to attach another rod.
4. Repeat same procedure to attach sliding rods with X-axis frame





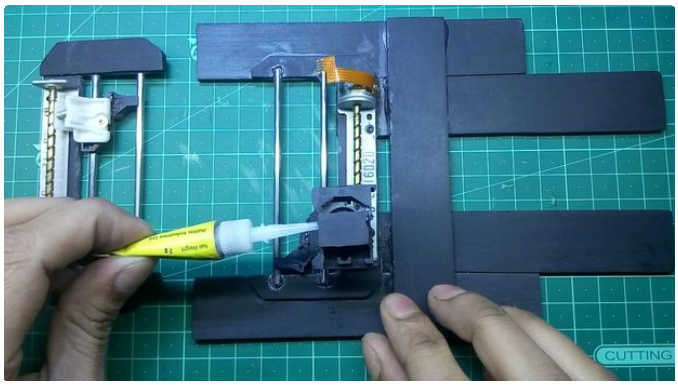


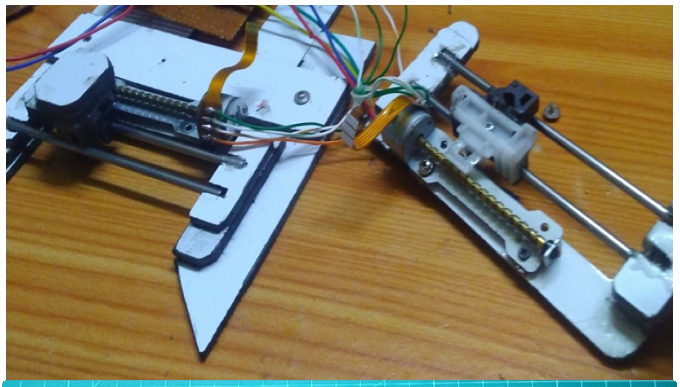


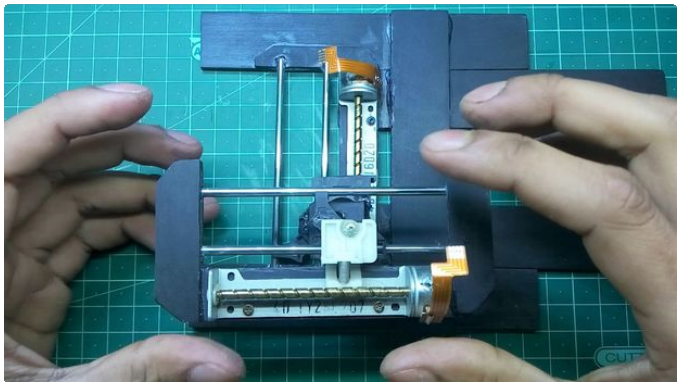
5.5

Attaching the X and Y Axis

In this step, we'll attach the Y-Axis on the top of X-Axis. Follow the above picture to attach both axes.







5.6

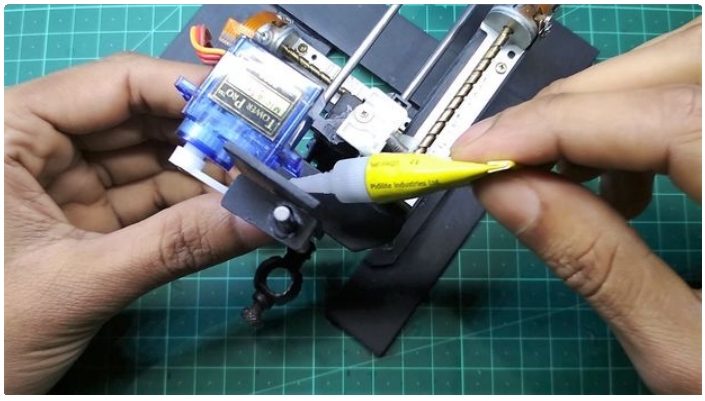
Making the Z- Axis

but I'm going to show you how to make a perfect Z-Axis. So follow steps below.

1. Take the required parts for the Z axis like Sliding rod, pen spring, compass pencil holder, some acrylic pieces.etc
2. first, take your sliding rod and insert it into its holder and insert pen spring after that use a small piece of the plastic holder to hold the pen spring.
3. Then apply super glue on the sliding rod holder and attach it to the acrylic frame.
4. After that attach a small piece of acrylic on the top of sliding rod, it also works to move the pen up and down.
5. Then attach the pen holder at the end of sliding rod.
6. Finally, mount the servo motor using the super glue.

5.7
Attach the Z-Axis

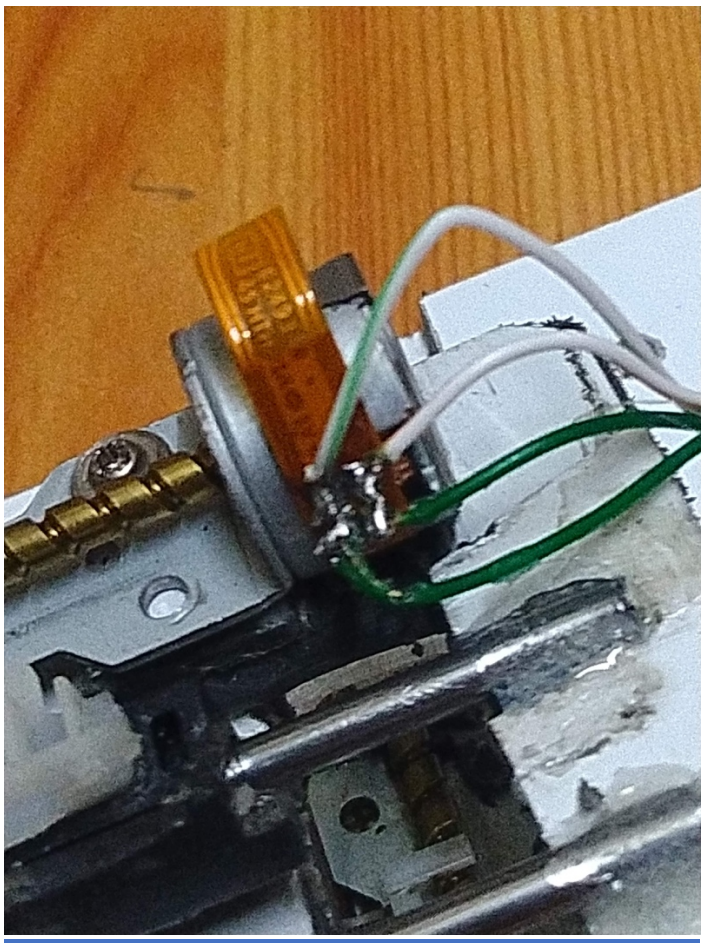
Attaching the Z-Axis to Y-axis using superglue as the below pictures showing.



5.8

Solder Wire With Motors

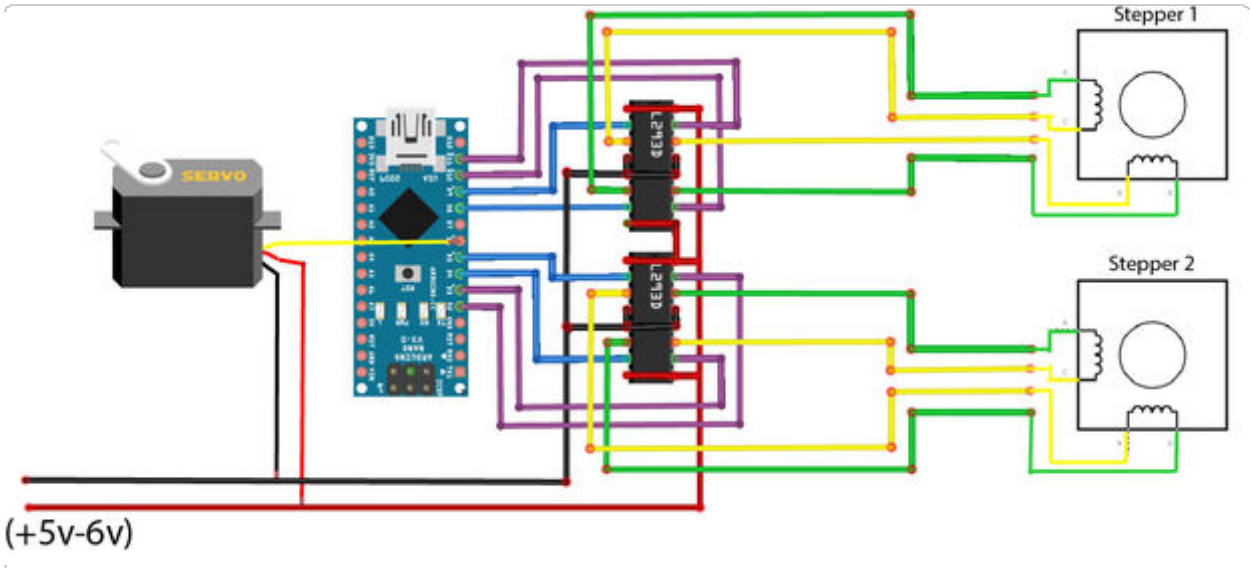
First, cutting the flexible PCB of the motors, then solder wire with motors. I'm using small pieces of a marker pen to hold my circuit board little higher.



5.9

The Circuit Diagram

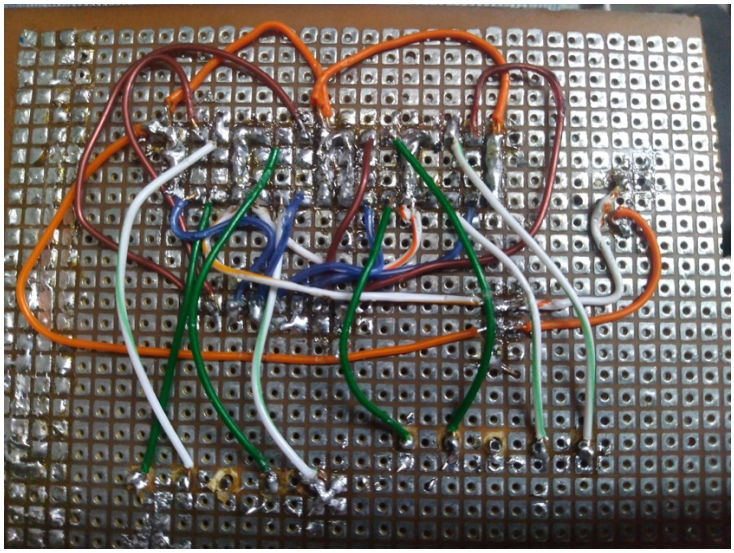
Here's the complete circuit diagram, connect everything according to the circuit diagram. If your steppers don't work properly then you have to find correct working combination by changing the pins of the L293D IC. You can also adjust the IC pins in code

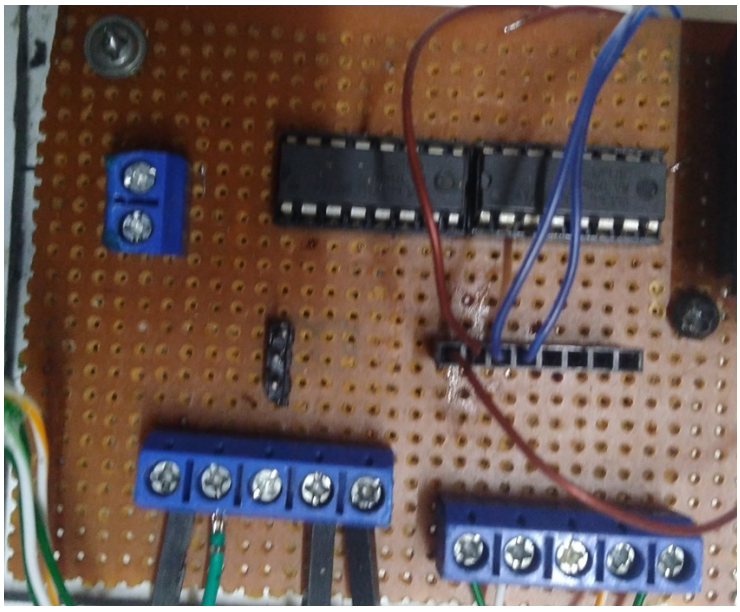


5.10

Making the Circuit Board

In this step, I'm going to show you how I made my Circuit board. First, insert the IC bases and all the header pins, then solder its pins, after that make every connection according to its circuit diagram. You have to do it very carefully and precisely

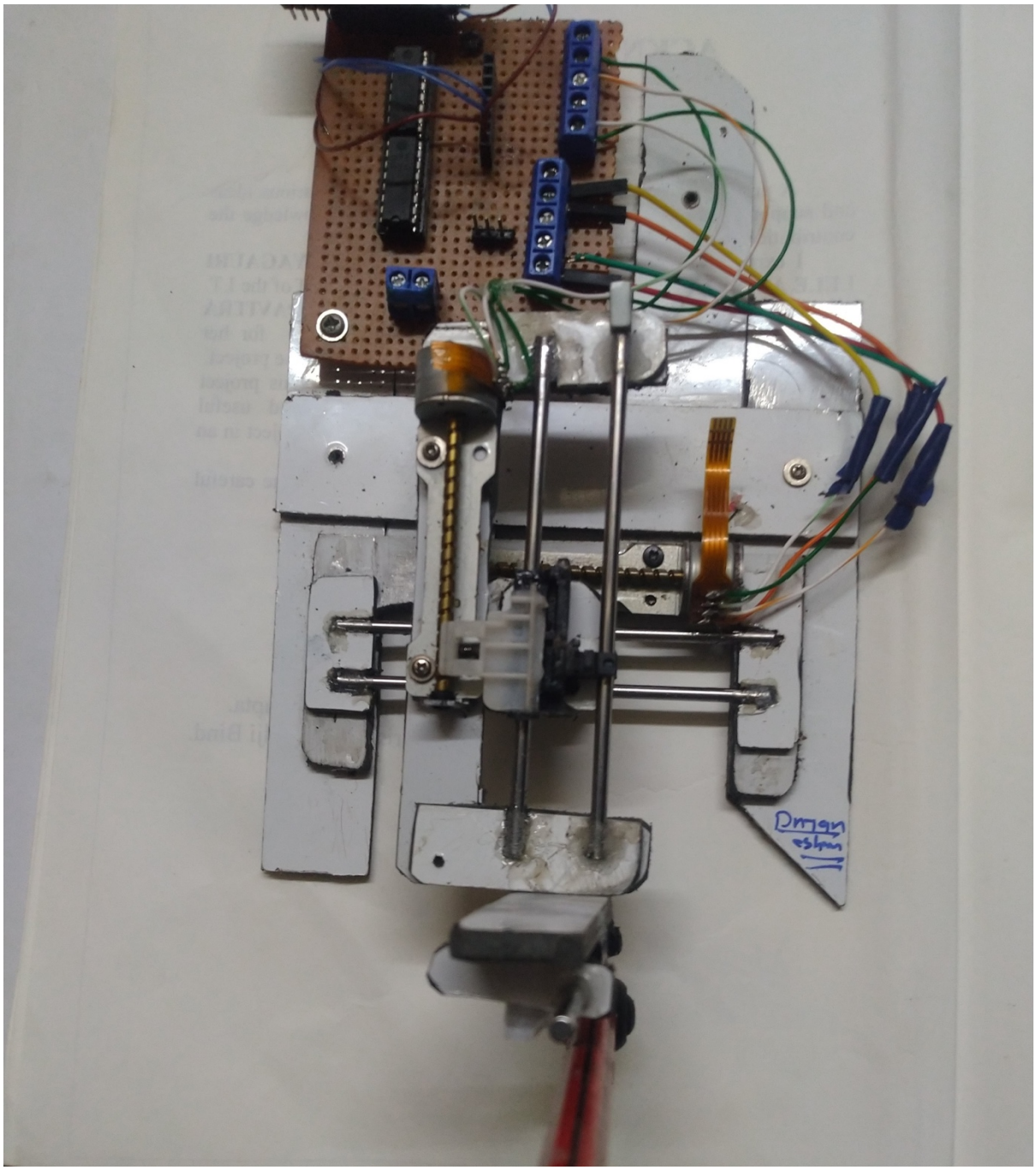




4.11

Assembling the Circuit Board

First Mount the both IC, then the Arduino nano and at the end connect the wires. Mount circuit Board to the CNC Machine.



5.12

Writing the CNC Code

In this step, we have write the CNC code to Arduino.

1. First, connect the Arduino board with PC through USB cable.
2. Select correct port and board.
3. Then Write code in Arduino ide And then upload code which we write for Arduino to control Stepper Motor.
4. Install processing 3 Software for executing gcode.
5. Write Java Code for Gcode file Execution.

6

Project Code

6.1 Arduino_CNC_plotter.ino

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

#define LINE_BUFFER_LENGTH 512

// Servo position for Up and Down
const int penZUp = 40;
const int penZDown = 85;

// Servo on PWM pin 6
const int penServoPin = 6;

// Should be right for DVD steppers, but is not too important here
const int stepsPerRevolution = 20;

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

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

// Current position of plothead
struct point actuatorPos;

//  Drawing settings, should be OK
float StepInc = 1;
int StepDelay = 0;
int LineDelay = 50;
int penDelay = 50;

// Motor steps to go 1 millimeter.
// Use test sketch to go 100 steps. Measure the length of line.
// Calculate steps per mm. Enter here.
float StepsPerMillimeterX = 6.0;
float StepsPerMillimeterY = 6.0;

// Drawing robot limits, in mm
// OK to start with. Could go up to 50 mm if calibrated well.
float Xmin = 0;
float Xmax = 40;
float Ymin = 0;
```

```
51 float Ymax = 40;
52 float Zmin = 0;
53 float Zmax = 1;
54
55 float Xpos = Xmin;
56 float Ypos = Ymin;
57 float Zpos = Zmax;
58
59 // Set to true to get debug output.
60 boolean verbose = false;
61
62 //  Needs to interpret
63 //  G1 for moving
64 //  G4 P300 (wait 150ms)
65 //  M300 S30 (pen down)
66 //  M300 S50 (pen up)
67 //  Discard anything with a (
68 //  Discard any other command!
69
70 /**********************
71  * void setup() - Initialisations
72  ***********************/
73 void setup() {
74   //  Setup
75   Serial.begin( 9600 );
76
77   penServo.attach(penServoPin);
78   penServo.write(penZUp);
79   delay(200);
80
81   // Decrease if necessary
82   myStepperX.setSpeed(250);
83   myStepperY.setSpeed(250);
84
85   //  Set & move to initial default position
86   // TBD
87
88   //  Notifications!!!
89   Serial.println("Mini CNC Plotter alive and kicking!");
90   Serial.print("X range is from ");
91   Serial.print(Xmin);
92   Serial.print(" to ");
93   Serial.print(Xmax);
94   Serial.println(" mm.");
95   Serial.print("Y range is from ");
96   Serial.print(Ymin);
97   Serial.print(" to ");
98   Serial.print(Ymax);
99   Serial.println(" mm.");
100 }
101
102 /**********************
103  * void loop() - Main loop
104  ***********************/
105 void loop()
106 {
```

```
107   delay(200);
108   char line[ LINE_BUFFER_LENGTH ];
109   char c;
110   int lineIndex;
111   bool lineIsComment, lineSemiColon;
112
113   lineIndex = 0;
114   lineSemiColon = false;
115   lineIsComment = false;
116
117   while (1) {
118
119     // Serial reception - Mostly from Grbl, added semicolon support
120     while ( Serial.available()>0 ) {
121       c = Serial.read();
122       if (( c == '\n') || (c == '\r') ) {              // End of line reached
123         if ( lineIndex > 0 ) {                  // Line is complete. Then execute!
124           line[ lineIndex ] = '\0';                    // Terminate string
125           if (verbose) {
126             Serial.print( "Received : ");
127             Serial.println( line );
128           }
129           processIncomingLine( line, lineIndex );
130           lineIndex = 0;
131         }
132         else {
133           // Empty or comment line. Skip block.
134         }
135         lineIsComment = false;
136         lineSemiColon = false;
137         Serial.println("ok");
138       }
139       else {
140         if ( (lineIsComment) || (lineSemiColon)
141           if ( c == ')' )  lineIsComment = false;
142         }
143         else {
144           if ( c <= ' ' ) {       // Throw away whitepace and control characters
145           }
146           else if ( c == '/' ) {    // Block delete not supported. Ignore char
147           }
148           else if ( c == '(' ) {
149             lineIsComment = true;
150           }
151           else if ( c == ';' ) {
152             lineSemiColon = true;
153           }
154           else if ( lineIndex >= LINE_BUFFER_LENGTH-1 ) {
155             Serial.println( "ERROR - lineBuffer overflow" );
156             lineIsComment = false;
157             lineSemiColon = false;
158           }
159           else if ( c >= 'a' && c <= 'z' ) {        // Upcase lowercase
160             line[ lineIndex++ ] = c-'a'+'A';
161           }
162           else {
```

```
163             line[ lineIndex++ ] = c;
164           }
165         }
166       }
167     }
168   }
169 }
170
171 void processIncomingLine( char* line, int charNB ) {
172   int currentIndex = 0;
173   char buffer[ 64 ];                 // Hope that 64 is enough for 1 parameter
174   struct point newPos;
175
176   newPos.x = 0.0;
177   newPos.y = 0.0;
178
179   //  Needs to interpret
180   //  G1 for moving
181   //  G4 P300 (wait 150ms)
182   //  G1 X60 Y30
183   //  G1 X30 Y50
184   //  M300 S30 (pen down)
185   //  M300 S50 (pen up)
186   //  Discard anything with a (
187   //  Discard any other command!
188
189   while( currentIndex < charNB ) {
190     switch ( line[ currentIndex++ ] ) {              // Select command, if any
191     case 'U':
192       penUp();
193       break;
194     case 'D':
195       penDown();
196       break;
197     case 'G':
198       buffer[0] = line[ currentIndex++ ];          // /!\ Dirty - Only works with
199 2 digit commands
200       //      buffer[1] = line[ currentIndex++ ];
201       //      buffer[2] = '\0';
202       buffer[1] = '\0';
203
204       switch ( atoi( buffer ) ){                   // Select G command
205       case 0:                                   // G00 & G01 - Movement or fast
206 movement. Same here
207       case 1:
208         // /!\ Dirty - Suppose that X is before Y
209         char* indexX = strchr( line+currentIndex, 'X' );  // Get X/Y position in
210         char* indexY = strchr( line+currentIndex, 'Y' );
211         if ( indexY <= 0 ) {
212           newPos.x = atof( indexX + 1);
213           newPos.y = actuatorPos.y;
214         }
215         else if ( indexX <= 0 ) {
216           newPos.y = atof( indexY + 1);
217           newPos.x = actuatorPos.x;
218         }
```

```
219         else {
220           newPos.y = atof( indexY + 1);
221           indexY = '\0';
222           newPos.x = atof( indexX + 1);
223         }
224         drawLine(newPos.x, newPos.y );
225         //        Serial.println("ok");
226         actuatorPos.x = newPos.x;
227         actuatorPos.y = newPos.y;
228         break;
229       }
230       break;
231     case 'M':
232       buffer[0] = line[ currentIndex++ ];        // /!\ Dirty - Only works with 3
233 digit commands
234       buffer[1] = line[ currentIndex++ ];
235       buffer[2] = line[ currentIndex++ ];
236       buffer[3] = '\0';
237       switch ( atoi( buffer ) ){
238       case 300:
239         {
240           char* indexS = strchr( line+currentIndex, 'S' );
241           float Spos = atof( indexS + 1);
242           //          Serial.println("ok");
243           if (Spos == 30) {
244             penDown();
245           }
246           if (Spos == 50) {
247             penUp();
248           }
249           break;
250         }
251       case 114:                                  // M114 - Repport position
252         Serial.print( "Absolute position : X = " );
253         Serial.print( actuatorPos.x );
254         Serial.print( "  -  Y = " );
255         Serial.println( actuatorPos.y );
256         break;
257       default:
258         Serial.print( "Command not recognized : M");
259         Serial.println( buffer );
260       }
261     }
262   }
263
264
265
266 }
267
268
269 /*********************************
270
271 void drawLine(float x1, float y1) {
272
273   if (verbose)
274   {
```

```
275     Serial.print("fx1, fy1: ");
276     Serial.print(x1);
277     Serial.print(",");
278     Serial.print(y1);
279     Serial.println("");
280   }
281
282   //  Bring instructions within limits
283   if (x1 >= Xmax) {
284     x1 = Xmax;
285   }
286   if (x1 <= Xmin) {
287     x1 = Xmin;
288   }
289   if (y1 >= Ymax) {
290     y1 = Ymax;
291   }
292   if (y1 <= Ymin) {
293     y1 = Ymin;
294   }
295
296   if (verbose)
297   {
298     Serial.print("Xpos, Ypos: ");
299     Serial.print(Xpos);
300     Serial.print(",");
301     Serial.print(Ypos);
302     Serial.println("");
303   }
304
305   if (verbose)
306   {
307     Serial.print("x1, y1: ");
308     Serial.print(x1);
309     Serial.print(",");
310     Serial.print(y1);
311     Serial.println("");
312   }
313
314   //  Convert coordinates to steps
315   x1 = (int)(x1*StepsPerMillimeterX);
316   y1 = (int)(y1*StepsPerMillimeterY);
317   float x0 = Xpos;
318   float y0 = Ypos;
319
320   //  Let's find out the change for the coordinates
321   long dx = abs(x1-x0);
322   long dy = abs(y1-y0);
323   int sx = x0<x1 ? StepInc : -StepInc;
324   int sy = y0<y1 ? StepInc : -StepInc;
325
326   long i;
327   long over = 0;
328
329   if (dx > dy) {
330     for (i=0; i<dx; ++i) {
```

```
331       myStepperX.step(sx);
332       over+=dy;
333       if (over>=dx) {
334         over-=dx;
335         myStepperY.step(sy);
336       }
337       delay(StepDelay);
338     }
339   }
340   else {
341     for (i=0; i<dy; ++i) {
342       myStepperY.step(sy);
343       over+=dx;
344       if (over>=dy) {
345         over-=dy;
346         myStepperX.step(sx);
347       }
348       delay(StepDelay);
349     }
350   }
351
352   if (verbose)
353   {
354     Serial.print("dx, dy:");
355     Serial.print(dx);
356     Serial.print(",");
357     Serial.print(dy);
358     Serial.println("");
359   }
360
361   if (verbose)
362   {
363     Serial.print("Going to (");
364     Serial.print(x0);
365     Serial.print(",");
366     Serial.print(y0);
367     Serial.println(")");
368   }
369
370   //  Delay before any next lines are submitted
371   delay(LineDelay);
372   //  Update the positions
373   Xpos = x1;
374   Ypos = y1;
375 }
376
377 //  Raises pen
378 void penUp() {
379   penServo.write(penZUp);
380   delay(LineDelay);
381   Zpos=Zmax;
382   if (verbose) {
383     Serial.println("Pen up!");
384   }
385 }
386 //  Lowers pen
```

```
387 void penDown() {
388   penServo.write(penZDown);
389   delay(LineDelay);
390   Zpos=Zmin;
391   if (verbose) {
392     Serial.println("Pen down.");
393   }
394 }
```

6.2. gcode_excuter.pde

```
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()
{
  size(600, 400);
  openSerialPort();
}

void draw()
{
```

```
  background(155);
  fill(0);
  int y = 24, dy = 12;
  text("INSTRUCTIONS", 12, y); y += dy;
  text("p: select serial port", 12, y); y += dy;
  text("1: set speed to 0.001 inches (1 mil) per jog", 12, y); y +=
dy;
  text("2: set speed to 0.010 inches (10 mil) per jog", 12, y); y +=
dy;
  text("3: set speed to 0.100 inches (100 mil) per jog", 12, y); y +=
dy;
  text("arrow keys: jog in x-y plane", 12, y); y += dy;
  text("page up & page down: jog in z axis", 12, y); y += dy;
  text("$: display grbl settings", 12, y); y+= dy;
  text("h: go home", 12, y); y += dy;
  text("0: zero machine (set home to the current location)", 12, y);
y += dy;
  text("g: stream a g-code file", 12, y); y += dy;
  text("x: stop streaming g-code (this is NOT immediate)", 12, y); y
+= dy;
  y = height - dy;
  text("current jog speed: " + speed + " inches per step", 12, y); y
-= dy;
  text("current serial port: " + portname, 12, y); y -= dy;
}

void keyPressed()
{
  if (key == '1') speed = 0.001;
  if (key == '2') speed = 0.01;
  if (key == '3') speed = 0.1;

  if (!streaming) {
    if (keyCode == LEFT) port.write("G91\nG20\nG00 X-" + speed + "
Y0.000 Z0.000\n");
    if (keyCode == RIGHT) port.write("G91\nG20\nG00 X" + speed + "
Y0.000 Z0.000\n");
    if (keyCode == UP) port.write("G91\nG20\nG00 X0.000 Y" + speed +
" Z0.000\n");
    if (keyCode == DOWN) port.write("G91\nG20\nG00 X0.000 Y-" + speed
+ " Z0.000\n");
    if (keyCode == KeyEvent.VK_PAGE_UP) port.write("G91\nG20\nG00
X0.000 Y0.000 Z" + speed + "\n");
    if (keyCode == KeyEvent.VK_PAGE_DOWN) port.write("G91\nG20\nG00
X0.000 Y0.000 Z-" + speed + "\n");
    if (key == 'h') port.write("G90\nG20\nG00 X0.000 Y0.000
Z0.000\n");
    if (key == 'v') port.write("$0=75\n$1=74\n$2=75\n");
    //if (key == 'v') port.write("$0=100\n$1=74\n$2=75\n");
    if (key == 's') port.write("$3=10\n");
    if (key == 'e') port.write("$16=1\n");
    if (key == 'd') port.write("$16=0\n");
    if (key == '0') openSerialPort();
    if (key == 'p') selectSerialPort();
    if (key == '$') port.write("$$\n");
  }
```

```
109
110   if (!streaming && key == 'g') {
111     gcode = null; i = 0;
112     File file = null;
113     println("Loading file...");
114     selectInput("Select a file to process:", "fileSelected", file);
115   }
116
117   if (key == 'x') streaming = false;
118 }
119
120 void fileSelected(File selection) {
121   if (selection == null) {
122     println("Window was closed or the user hit cancel.");
123   } else {
124     println("User selected " + selection.getAbsolutePath());
125     gcode = loadStrings(selection.getAbsolutePath());
126     if (gcode == null) return;
127     streaming = true;
128     stream();
129   }
130 }
131
132 void stream()
133 {
134   if (!streaming) return;
135
136   while (true) {
137     if (i == gcode.length) {
138       streaming = false;
139       return;
140     }
141
142     if (gcode[i].trim().length() == 0) i++;
143     else break;
144   }
145
146   println(gcode[i]);
147   port.write(gcode[i] + '\n');
148   i++;
149 }
150
151 void serialEvent(Serial p)
152 {
153   String s = p.readStringUntil('\n');
154   println(s.trim());
155
156   if (s.trim().startsWith("ok")) stream();
157   if (s.trim().startsWith("error")) stream(); // XXX: really?
158 }
159
160
161
```

7

Conclusion

In modern CNC systems, end-to-end component design are highly automated using **computer aided design (CAD)** and **computer-aided manufacturing (CAM)** programs. The programs produce a computer file that is interpreted to extract the commands needed to operate a particular machine by use of a post processor, and then loaded into the CNC machines for production. Since any particular component might require the use of a number of different tools - drills, saws, etc.

With the increasing demand for small scale high precision parts in various industries, the market for small scale machine tools has grown substantially. Using small machine tools to fabricate small scale parts can provide both flexibility and efficiency in manufacturing approaches and reduce capital cost, which is beneficial for small business owners. In this project, a small scale three axis **"COMPUTER NUMERICL CONTROL MACHINE "** is designed and analyzed under very limited budget.

8

Future scope:

1. **Actual industrial CNC milling:**

   It is planned to scale up the prototype CNC machine in terms of size, use more powerful motors, strengthen the frame and worktable with materials like aluminum or cast iron, and augment the CNC control software with software for simulation ahead of actual run. The implementation of 3D printing technology to the same hardware abstract is going for printing of 3D models.

2. **The PCB Mill:**

   In the manufacturing of precised **PCBs (Printed Circuit Board)** i.e. etching or printing the conductive paths which connects the different electronic components to one another on a board.

9

Bibliography

Textbook Source of Information

1. Electronic devices and circuits by A.P. Godse, U.A Bakshi
2. **"The Fourth Industrial Revolution" by Klaus Schwab**

Internet Source of Information

- https://astromachineworks.com/what-is-cnc-machining/
- https://toolcraftfw.com/news/applications-of-cnc-machines-types-of-cnc-machining-center/
- https://www.circuitstoday.com/arduino-nano-tutorial-pinout-schematics