A

PROJECT REPORT

ON

ARDUINO BASED

MINI CNC PLOTTER

GOVERNMENT ENGINEERING COLLEGE

KONI, BILASPUR, CHHATTISAGARH

TEAM MEMBERS

LILESH LAHRE (CSE)

PRANJAL MAHAPATRA (CSE)

SWIKRIT SHUKLA (CSE)

SPARSH VERMA (CSE)

ASHWIN AGRWAL (CSE)

KUNAL SAHU (IT)

K YASHWANT RAO (CSE)

VIKAS SINGH RAJPUT (IT)

ASHUTOSH YADAV (IT)

A PROJECT BY

CS & IT STUDENTS

1ST YEAR

CERTIFICATE

This is to certify that the project work report entitled

(ARDUINO BASED MINI CNC PLOTTER)

Has been successfully completed by following students to present in

Vishwakarma Pooja Exhibition (17 SEPT 2022)

Offered By

Government Engineering College, Koni, BILASPUR, (C.G)

SUBMITTED BY

LILESH LAHRE (CSE)

SWIKRIT SHUKLA (CSE)

SPARSH VERMA (CSE)

ASHWIN AGRWAL (CSE)

K YASHWANT RAO (CSE)

KUNAL SAHU (IT)

VIKAS SINGH RAJPUT (IT)

ASHUTOSH YADAV (IT)

ORGANISED BY

Prof. Indranee Borkar

Dr. B S Chawla

Principal

Government Engineering College

Koni , Bilaspur , Chhattisgarh

Acknowledgment

We find ourselves delighted on successful completion of our project. We express our heartfelt thanks and pay gratitude for the opportunity given by GEC Bilaspur as well as respected Dr. B S Chawla sir (Principal), Respected Sourav Yadav Sir (HoD Computer Science Engineering), Respected G.K Agrawal Sir (HoD Mechanical Department), Respected G.S Singh Sir (HoD Electrical

We were encouraged by the keen enthusiasm by Miss Sonia Wadhwa, Asst. Prof. CSE Department, for out completion of the project and we are great full to her.

We would like to Thanks our parents. Without their support and inspiration this project would never had been possible.

We also extend our thanks all those persons and well-wishes who helped us directly or indirectly in completion of this project.

GREATFUL	BRANCH
LILESH LAHRE	**COMPUTER SCIENCE**
SWIKRIT SHUKLA	**COMPUTER SCIENCE**
SPARSH VERMA	**COMPUTER SCIENCE**
K YASHWANT RAO	**COMPUTER SCIENCE**
ASHWIN AGRWAL	**INFORMATION TECHNOLOGY**
KUNAL SAHU	**INFORMATION TECHNOLOGY**
VIKAS SINGH RAJPUT	**INFORMATION TECHNOLOGY**
ASHUTOSH YADAV	**INFORMATION TECHNOLOGY**

FUNCTIONS OF THE COMPONENT

1. Arduino Uno

The Arduino Uno is an open-source microcontroller board based on the Microchip ATmega328P microcontroller and developed by Arduino.cc and initially released in 2010. The board is equipped with sets of digital and analog input/output pins that may be interfaced to various expansion boards and other circuits.

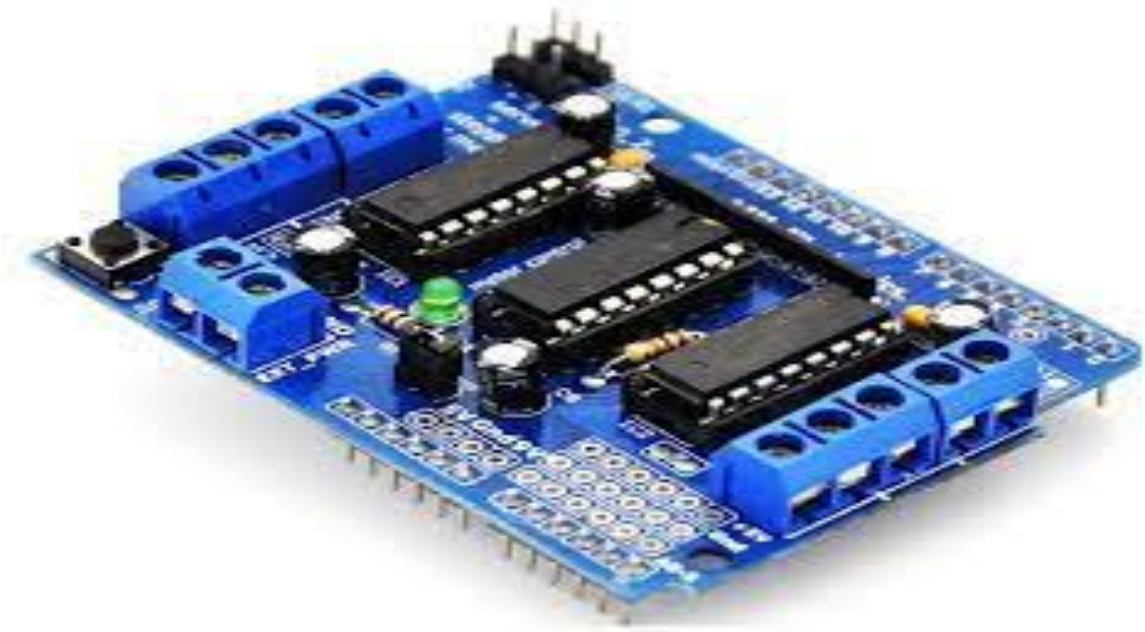
2. L293D Motor Driver Shield

The L293D is a dual-channel H-Bridge motor driver capable of driving a pair of DC motors or single stepper motor.

As the shield comes with two L293D motor driver chipsets, that means it can individually drive up to four DC motors making it ideal for building four-wheel robot platforms.

The shield offers total 4 H-Bridges and each H-bridge can deliver up to 0.6A to the motor.

The shield also comes with a 74HC595 shift register that extends 4 digital pins of the Arduino to the 8 direction control pins of two L293D chips.



3.Servo Motor

A **servomotor** (or **servo motor**) is a rotary actuator or linear actuator that allows for precise control of angular or linear position, velocity and acceleration.[1] It consists of a suitable motor coupled to a sensor for position feedback. It also requires a relatively sophisticated controller, often a dedicated module designed specifically for use with servomotors.

Servomotors are not a specific class of motor, although the term *servomotor* is often used to refer to a motor suitable for use in a closed-loop control system.

Servomotors are used in applications such as robotics, CNC machinery, and automated manufacturing.

4. Multimeter

A **multimeter** is a measuring instrument that can measure multiple electrical properties. A typical multimeter can measure voltage, resistance, and current, in which case it is also known as a **volt-ohm-milliammeter** (**VOM**), as the unit is equipped

with voltmeter, ammeter, and ohmmeter functionality. Some feature the measurement of additional properties such as temperature and capacitance.

5. Power Adaptor 5v 1A

The power adapter serves the purpose of **converting AC voltage to a single DC voltage for your computer**. It operates as an external battery for our plotter so that it works efficiently in a constant dc supply.

6. DVD Drive Slider mechanism(for x and y direction movement)

CD writers sliding mechanism is used in our project for motion in X and Y axis direction. The pen is fitted in upper portion performing X Axis direction movement and Paper is fitted in lower portion performing Y axis direction movement.

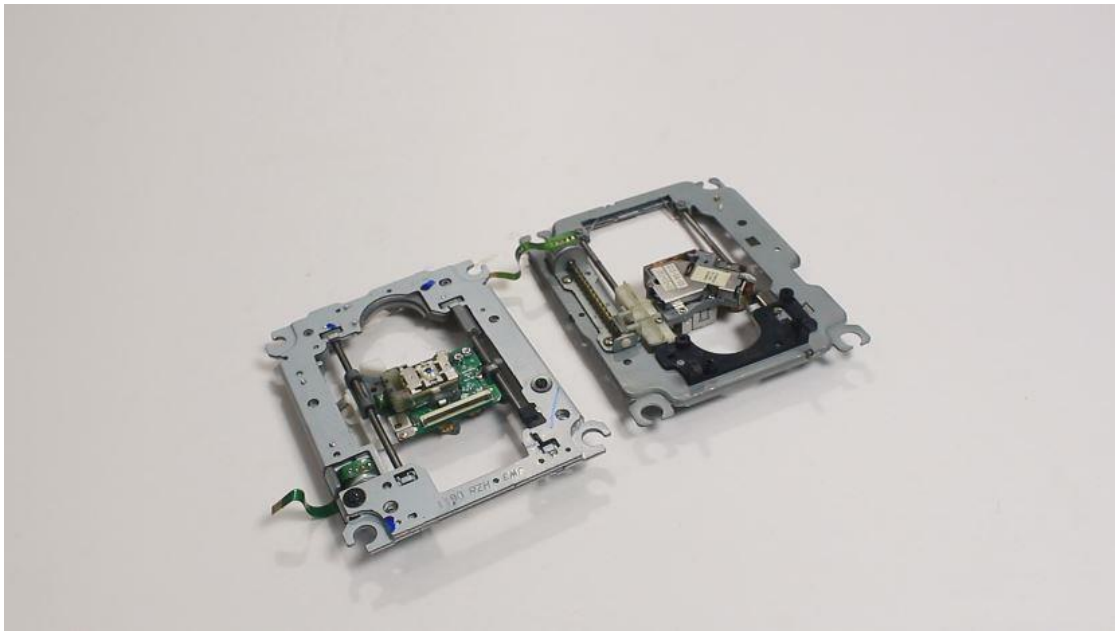
MACHINE ASSEMBLY

Step 1.

To make Arduino based mini CNC plotter machine obviously we need two scrap DVD drive.
Wepurchased this drive from local computer repairing shop .
we'll going to use its stepper motor along with sliding mechanism
here note that not all DVD drives have stepper motor in it. if the motor have 4 wires it means it is a stepper motor.
if you not found any 4 wire motor in DVD drive then it is use less.

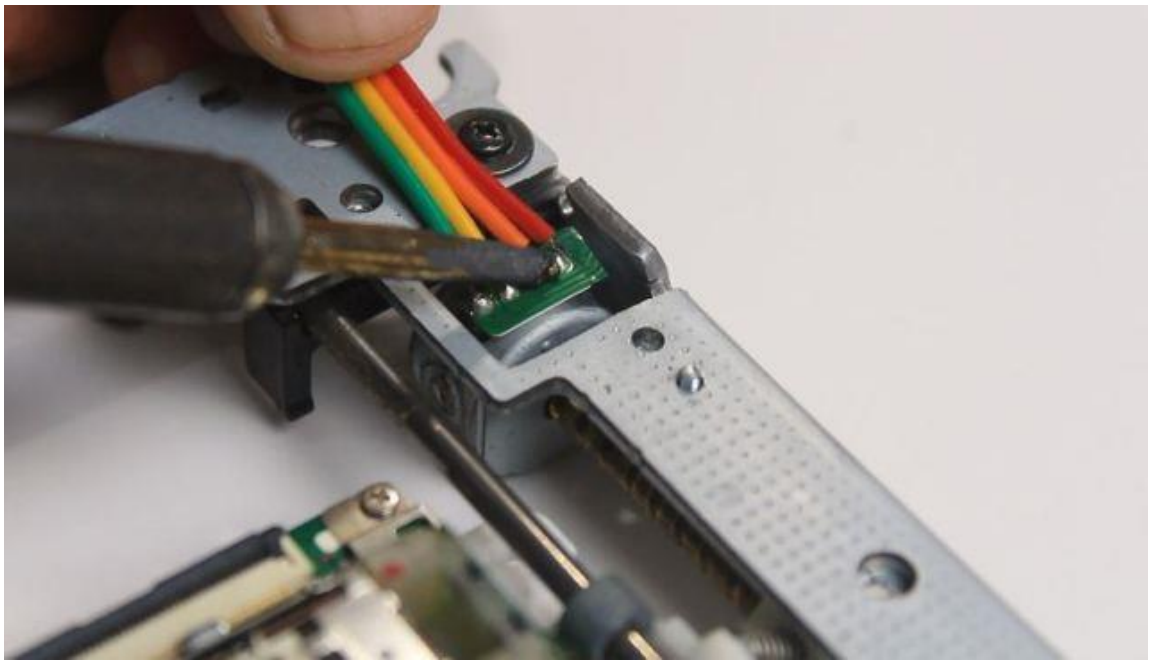
Step 2.

We quickly unscrew the DVD drive case with the help of screw driver and by applying some force I take out the stepper motor mechanism from the DVD drive case.
In this way I have two stepper driver mechanism and two empty case of DVD drive.



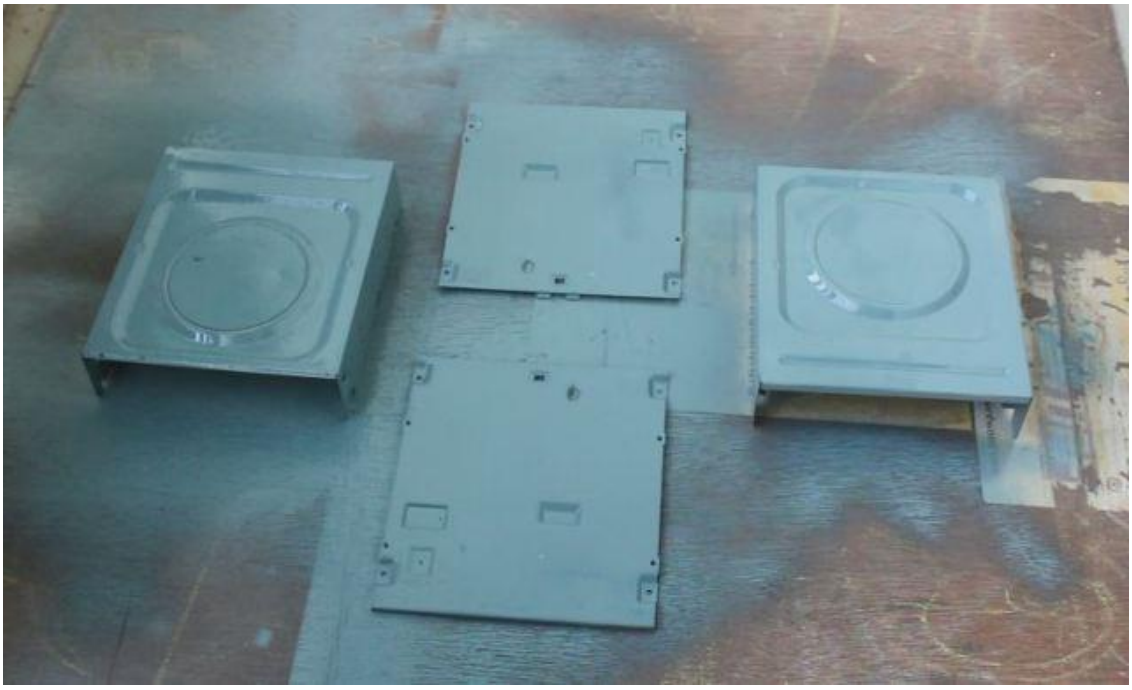
Step3.

After taking out stepper motor mechanism we cut the default motor connector strip with the help of scissor.
then we bring some DuPont 4 wire of around 40 cm and cut it into 2 pieces one for each stepper motor connection.
Then we strip the wire carefully without damaging the copper strain of wire.
and solder it to the expose terminals of stepper motor.



Step 4.

Here we have painted the the empty case of DVD drive using gray shade spray paint, this step is not compulsory its ok if you don't want to paint them.

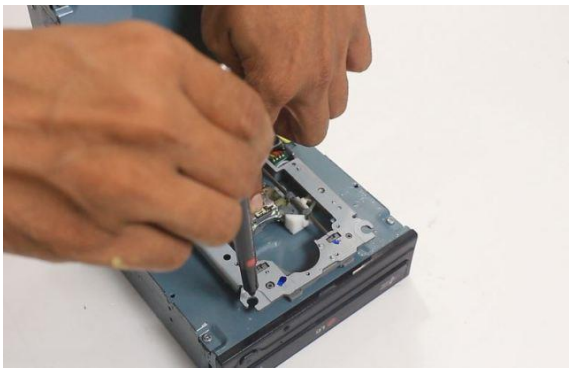


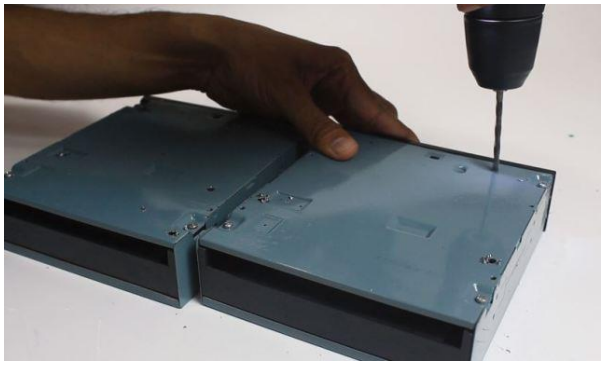
Step 5.

Then We used a piece of 20 x 20 mm aluminium angle to make holder for X-axis and Y-axis .
I drill the 5mm hole on the aluminium piece and cut it into two piece of clamp, further I use this clamp to fix the both axis with the help of M5X10 nut and bolt.

Step 6.

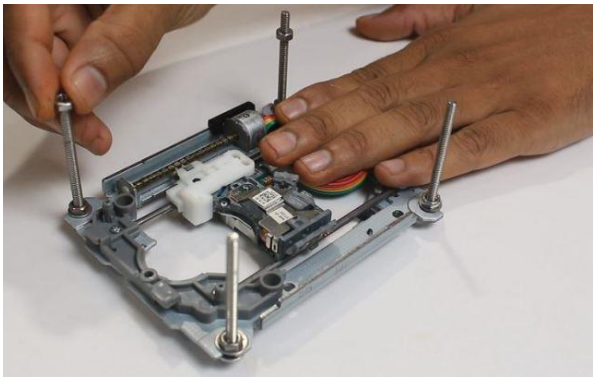
Now I am marking for hole on DVD drive case to make arrangement for mounting for both stepper motor mechanism.
I carefully drill the hole of 5mm with the help of drill machine

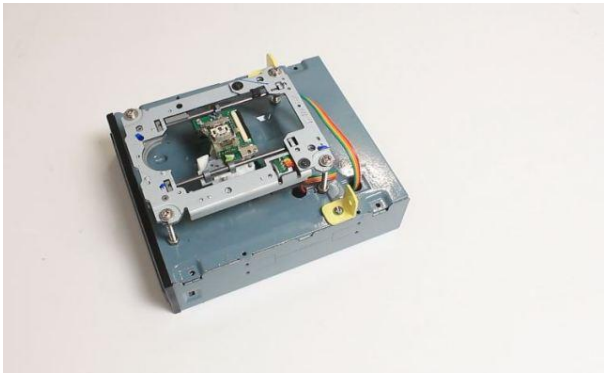


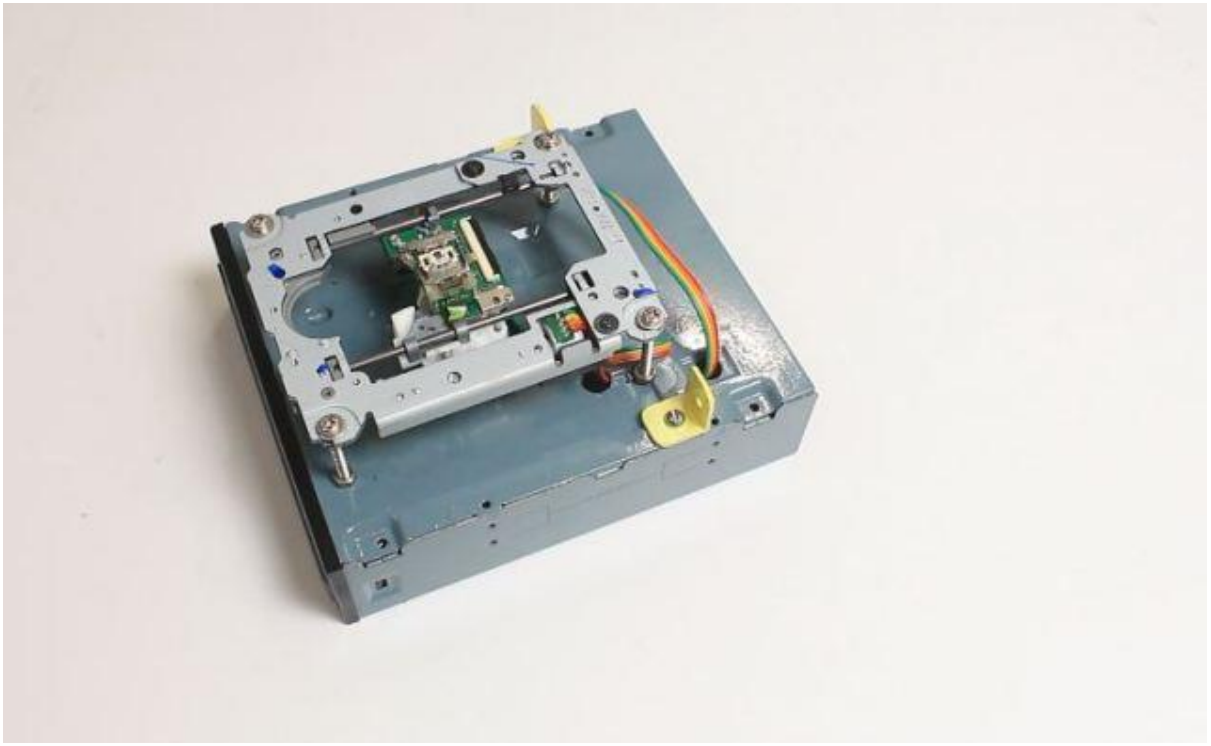


Step 7.

After drilling the hole in DVD drive case I fix the four M4 X 60 nut bolts at the four corner of stepper motor mechanism.
Now I placed the stepper motor mechanism to its place and secure all four bolts with M4 nuts.



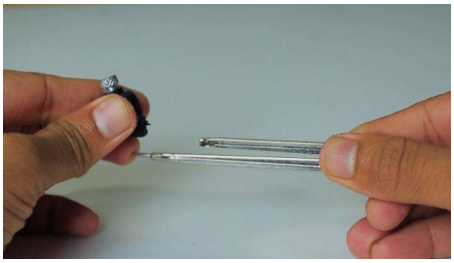
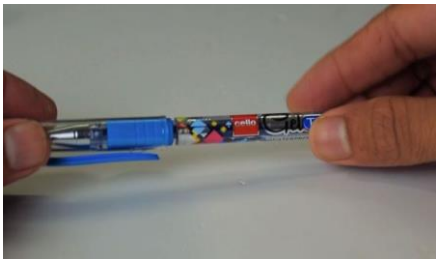
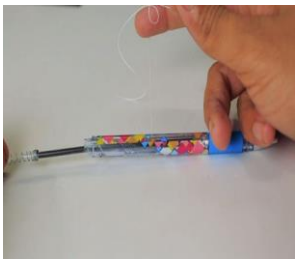
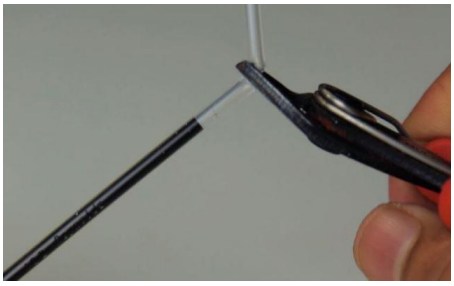
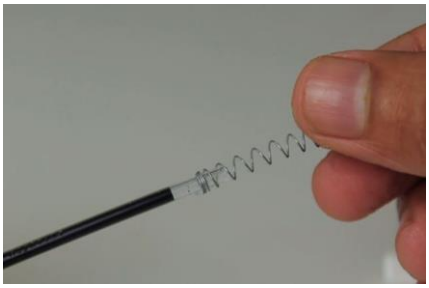
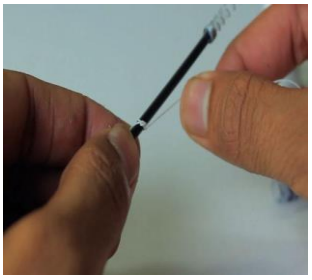




Step 8

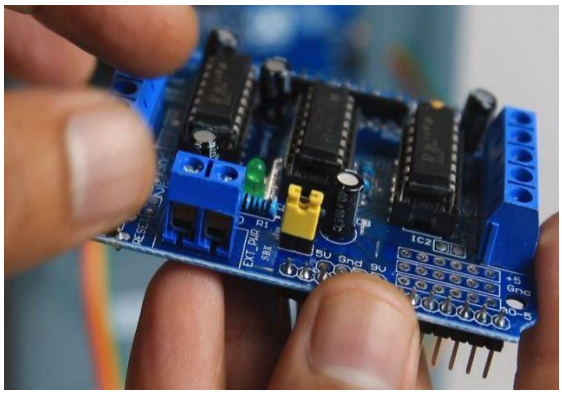
This the is most important step in making mini Arduino CNC plotter machine here we are making pen up and down mechanism.
First I take a compass and carefully remove its pen holder part.
then I used a simple pen with top and bottom openable.
first take out the refill of pen and cut about 2 cm part from the top of the refill.
now I place a spring at the top of the refill which I have salvage from other trigger type pen.
then I used a strong thread and tied it to the center of the refill and secure it with super glue on to its place.
now I make a small hole just above from the center of pen body.

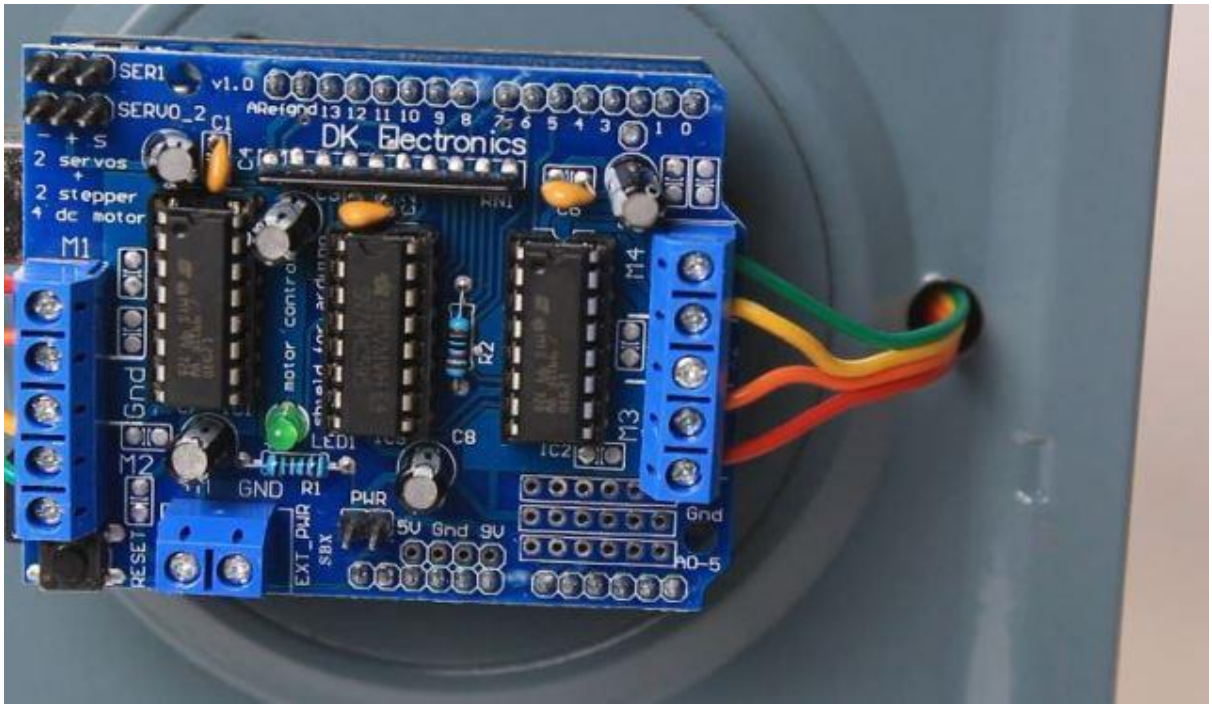
Now I carefully place the refill inside the pen and passed thread outside from the hole.
In this way I made my pen up down mechanism, when I pull the thread pen refill push upward and when I release the thread refill went down.
and due to the spring attached at top of the refill pen tip will maintain a good friction with paper..
Pen is now placed in pen holder and glue it with super glue on the X-axis

I attached a mini servo at X-axis and tie the thread with the knob of mini servo motor

Step 9

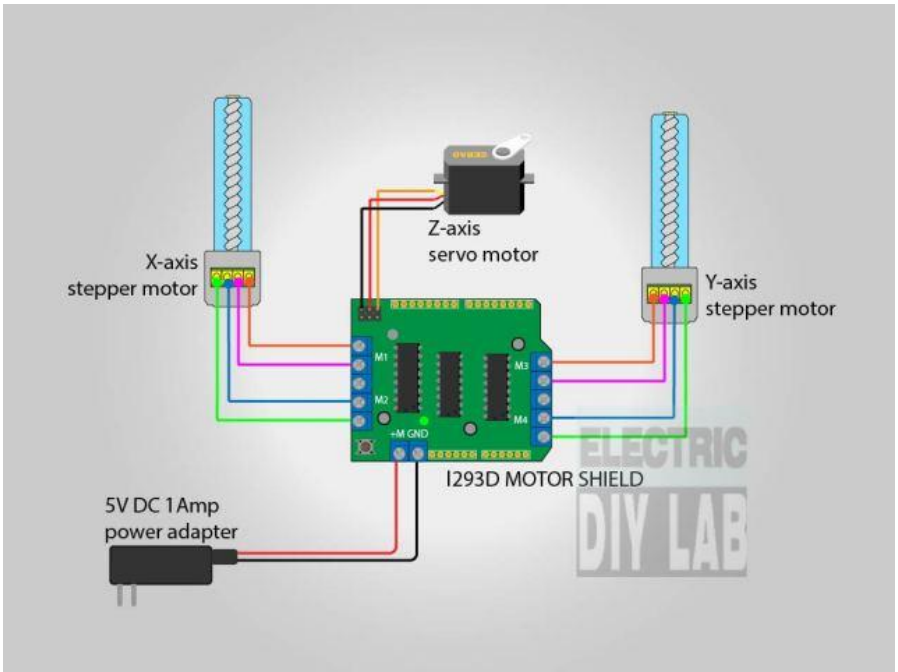




I drill four hole on the back side of machine and screw four 15mm spacer to mount arduino UNO onto it.
L293D motor shield is mounted on Arduino UNO.
In this way assembly of Arduino CNC plotter machine is competed now we will see the wiring

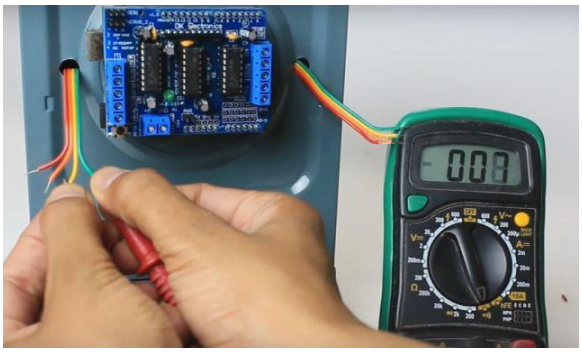
WIRING

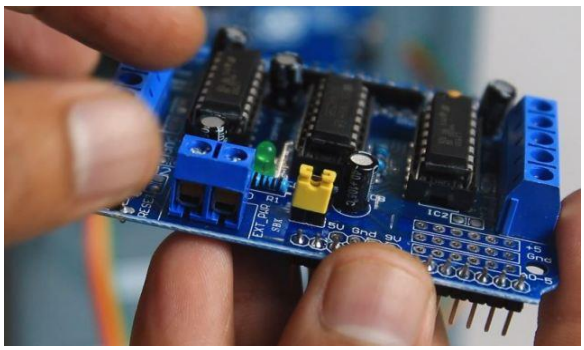
We are using Arduino UNO as a brain of CNC machine, as we know there is stepper motors used in CNC machine.

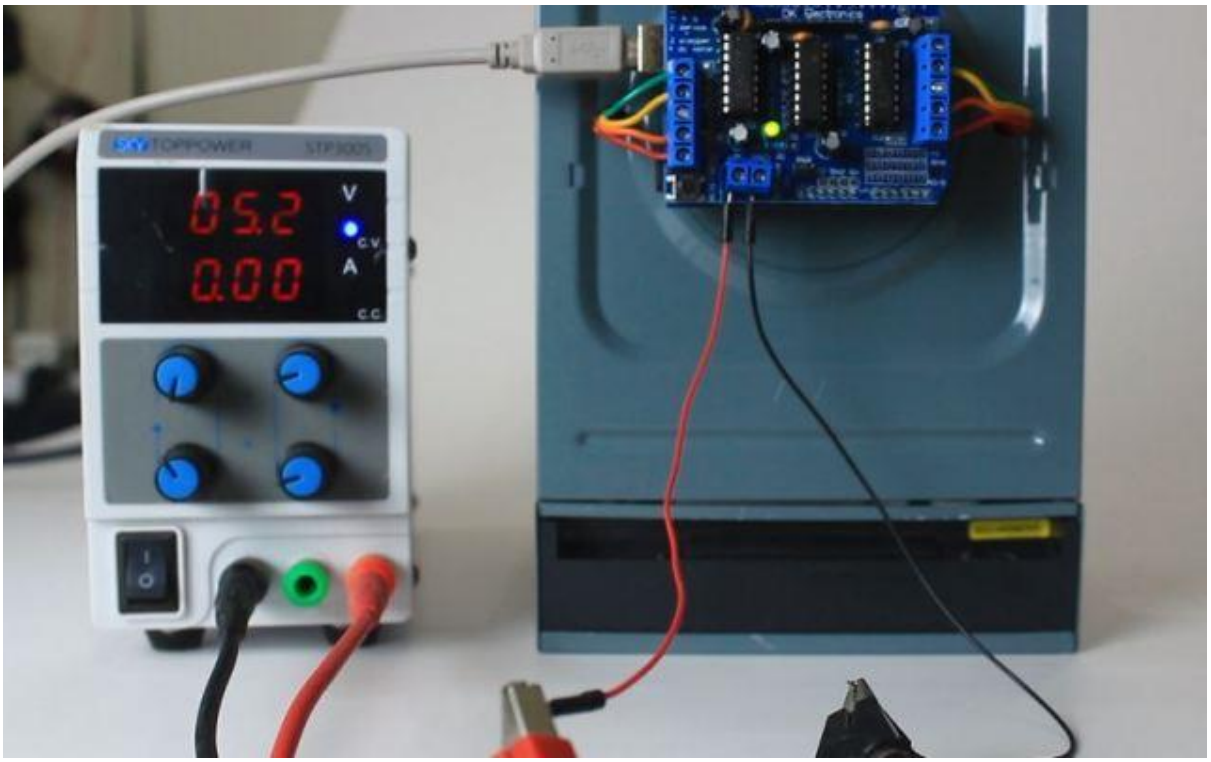


Stepper motors are not easy to control so here we are using a L293D motor shield to control our stepper motors and one servo motor is used for pen up down movement.

Before start wiring first we nee to the know the correct wire of stepper motor.
Our stepper motor have 4 wire and stepper motor have two coils means the a set of two wire is form one coil.
so we need to find out which two wires are from one coil, so here I am using multimeter keeping multimeter on continuity test.
I connect the mutimeter prob to the wires one by one, if I get continuity(few ohms) between any two wires means that both wires belong to single coil and rest two are from other coil.







Don't forget to remove that yellow jumper, and connect the stepper motor wires as shown in drawing above.
and also connect the servo motor at servo 1 terminal of L293D Motor shield.
You need a power adapter to feed power to the machine, you can use 5VDC 1amps power adapter.
In this way wiring is completed now we can move towards arduino code uploading process.

ARDUINO CODE

First of all we need to install AFMotor library in arduino IDE if you dont know how to add library just google it.

Now simple upload the code without changing anything

Here we are explaining some important part of code which may useful for you

Followings are the servo up down values Increase or decrease if required. if servo works in opposite direction switch the value of punZUp & penZDown values.

```
// Servo position for Up and Down
const int penZUp = 120;
const int penZDown = 50;
float StepInc = 1;
int StepDelay = 1;
int LineDelay =0;
int penDelay = 50; at Xmin = 0;

float Xmax = 40;
float Ymin = 0;
float Ymax = 40;
float Zmin = 0;
float Zmax = 1;
#include <Servo.h>
#include <AFMotor.h>
#define LINE_BUFFER_LENGTH 512
char STEP = MICROSTEP ;
// Servo position for Up and Down
const int penZUp = 120;
const int penZDown = 50;
// Servo on PWM pin 10
const int penServoPin =10 ;
// Should be right for DVD steppers, but is not too important here
const int stepsPerRevolution = 48;
// create servo object to control a servo
Servo penServo;
// Initialize steppers for X- and Y-axis using this Arduino pins for the L293D H-bridge
AF_Stepper myStepperY(stepsPerRevolution,1);
AF_Stepper myStepperX(stepsPerRevolution,2);
```

```
/* Structures, global variables */
struct point {
float x;
float y;
float z;
};
// Current position of plothead
struct point actuatorPos;
// Drawing settings, should be OK
float StepInc = 1;
int StepDelay = 1;
int LineDelay =0;
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 = 100.0;
float StepsPerMillimeterY = 100.0;
// 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;
float Ymax = 40;
float Zmin = 0;
float Zmax = 1;
float Xpos = Xmin;
float Ypos = Ymin;
float Zpos = Zmax;
// Set to true to get debug output.
boolean verbose = false;
// Needs to interpret
// G1 for moving
// G4 P300 (wait 150ms)
// M300 S30 (pen down)
// M300 S50 (pen up)
// Discard anything with a (
// Discard any other command!
/**********************
* void setup() - Initialisations
```

```
**********/
void setup() {
// Setup
Serial.begin( 9600 );
penServo.attach(penServoPin);
penServo.write(penZUp);
delay(100);
// Decrease if necessary
myStepperX.setSpeed(600);
myStepperY.setSpeed(600);
// Set & move to initial default position
// TBD
// Notifications!!!
Serial.println("Mini CNC Plotter alive and kicking!");
Serial.print("X range is from ");
Serial.print(Xmin);
Serial.print(" to ");
Serial.print(Xmax);
Serial.println(" mm.");
Serial.print("Y range is from ");
Serial.print(Ymin);
Serial.print(" to ");
Serial.print(Ymax);
Serial.println(" mm.");
}
/**********************
* void loop() - Main loop
**********/
void loop()
{
delay(100);
char line[ LINE_BUFFER_LENGTH ];
char c;
int lineIndex;
bool lineIsComment, lineSemiColon;
lineIndex = 0;
lineSemiColon = false;
lineIsComment = false;
while (1) {
// Serial reception - Mostly from Grbl, added semicolon support
```

```
while ( Serial.available()>0 ) {
c = Serial.read();
if (( c == '\n') || (c == '\r') ) { // End of line reached
if ( lineIndex > 0 ) { // Line is complete. Then execute!
line[ lineIndex ] = '\0'; // Terminate string
if (verbose) {
Serial.print( "Received : ");
Serial.println( line );
}
processIncomingLine( line, lineIndex );
lineIndex = 0;
}
else {
// Empty or comment line. Skip block.
}
lineIsComment = false;
lineSemiColon = false;
Serial.println("ok");
}
else {
if ( (lineIsComment) || (lineSemiColon) ) { // Throw away all comment characters
if ( c == ')' ) lineIsComment = false; // End of comment. Resume line.
}
else {
if ( c <= ' ' ) { // Throw away whitepace and control characters
}
else if ( c == '/' ) { // Block delete not supported. Ignore character.
}
else if ( c == '(' ) { // Enable comments flag and ignore all characters until ')' or EOL.
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 lowercase
```

```
line[ lineIndex++ ] = c-'a'+'A';
}
else {
line[ lineIndex++ ] = c;
}
}
}
}
}
}
void processIncomingLine( char* line, int charNB ) {
int currentIndex = 0;
char buffer[ 64 ]; // Hope that 64 is enough for 1 parameter
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++ ] ) { // Select command, if any
case 'U':
penUp();
break;
case 'D':
penDown();
break;
case 'G':
buffer[0] = line[ currentIndex++ ]; // /!\ Dirty - Only works with 2 digit commands
// buffer[1] = line[ currentIndex++ ];
// buffer[2] = '\0';
buffer[1] = '\0';
switch ( atoi( buffer ) ){ // Select G command
case 0: // G00 & G01 - Movement or fast movement. Same here
```

```
case 1:
// /!\ Dirty - Suppose that X is before Y
char* indexX = strchr( line+currentIndex, 'X' ); // Get X/Y position in the string (if any)
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++ ]; // /!\ Dirty - Only works with 3 digit commands
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 - Repport position
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 );
}
}
}
}
/*********************************
* Draw a line from (x0;y0) to (x1;y1).
* int (x1;y1) : Starting coordinates
* int (x2;y2) : Ending coordinates
*********************************/
void drawLine(float x1, float y1) {
if (verbose)
{
Serial.print("fx1, fy1: ");
Serial.print(x1);
Serial.print(",");
Serial.print(y1);
Serial.println("");
}
// Bring instructions within limits
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;
// Let's find out the change for the coordinates
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.onestep(sx,STEP);
over+=dy;
if (over>=dx) {
over-=dx;
myStepperY.onestep(sy,STEP);
}
delay(StepDelay);
```

```
}
}
else {
for (i=0; i<dy; ++i) {
myStepperY.onestep(sy,STEP);
over+=dx;
if (over>=dy) {
over-=dy;
myStepperX.onestep(sx,STEP);
}
delay(StepDelay);
}
}
if (verbose)
{
Serial.print("dx, dy:");
Serial.print(dx);
Serial.print(",");
Serial.print(dy);
Serial.println("");
}
if (verbose)
{
Serial.print("Going to (");
Serial.print(x0);
Serial.print(",");
Serial.print(y0);
Serial.println(")");
}
// Delay before any next lines are submitted
delay(LineDelay);
// Update the positions
Xpos = x1;
Ypos = y1;
}
// Raises pen
void penUp() {
penServo.write(penZUp);
delay(penDelay);
Zpos=Zmax;
```

```
digitalWrite(15, LOW);
digitalWrite(16, HIGH);
if (verbose) {
Serial.println("Pen up!");
}
}
// Lowers pen
void penDown() {
penServo.write(penZDown);
delay(penDelay);
Zpos=Zmin;
digitalWrite(15, HIGH);
digitalWrite(16, LOW);
if (verbose) {
Serial.println("Pen down.");
```

G-CODE GENERATION

To draw something with CNC plotter machine we obviously needs G-code,
G-code is the language of CNC machine.
In this project we are using Inkscape software and makerboat G-code library to generate G-code of image.
First off all download the Inkscape 0.48.5 version and download the makerboat gcode extension.
Install the Inkscape software and add the extension as per below

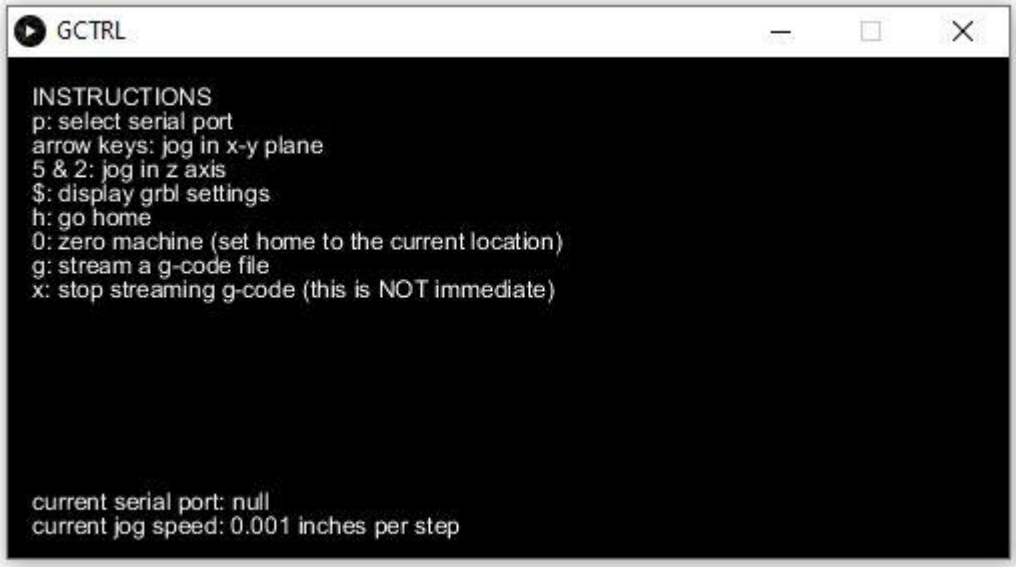
To understand the complete process of G-code generation I request you to watch the below video.
If you further have any issue you are welcome in comment section.
Only you have to note that you have to download inkscape 0.48.5 version only

GCTRL

At this point our machine is ready to plot anything just waiting for command,
We also have generated G-code but how to send this G-code to machine?

for this job We are using GCTRL a G-code sender GUI for processing.
We are streaming G-code to machine with the help of processing and GCTRL
Hope you have downloaded GCTRL from above if not download it from here

Simple open the GCTRL.pde by double clicking on it
Now click on play button visible on the top right corner of processing window,
this type of window will appears.



All the details regarding how to use this GUI is clearly written there itself.
By pressing the key “p” you can select the COM port from the drop down list.
after selecting the port you can jog X & Y axis with the help of arrow key.
use numerical key “5” & “2” to jog pen up and down.
To send G-code to arduino press key “g” and select the G-code file from the brows window as soon you select the fill and hit enter machine start to plot the drawing.

In this way the process of making Arduio based Mini CNC plotter machine is competed hope I have covered major points if you need to ask any question just simply comment in the comment section below. I’ll try my best to answer you.

BIBLIOGRAPHY

1. https://electricdiylab.com/how-to-make-arduino-mini-cnc-plotter-machine/#MATERIAL_LIST
2. https://youtu.be/Gm6bH3p6cNQ
3. Wikipedia