2-D CNC MACHINE

Project report submitted in partial fulfilment of the requirement for the degree of

BACHELOR OF TECHNOLOGY
IN
ELECTRONICS AND COMMUNICATION ENGINEERING

Submitted By:
Rohit Goel (20102115)

Submitted To:
Mrs. Shradha Saxena

JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY, NOIDA

SECTOR-62

Table of Contents

Candidates' Declaration

This is to certify that the work which is being presented in B.Tech Minor Project Report entitled **2-D CNC Machine**, submitted Rohit Goel, in partial fulfilment of the requirements for the award of degree of **Bachelor of Technology** in **Electronics & Communication Engineering** and submitted to the Department of Electronics & Communication Engineering of Jaypee Institute of Information Technology, Noida is an authentic record of our own work carried out during a period from August 2022 to December 2022 under the supervision of **Mrs. Shradha Saxena**, ECE Department. The matter presented in this report has not been submitted by us for the award of any other degree elsewhere.

_____

Rohit Goel (20102115)

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

_____

Mrs. Shradha Saxena

Date:

Acknowledgement

We take this opportunity to convey our gratitude to all those who have been kind enough to offer their advice and provide assistance when needed which has led to the successful completion of the project.

We would like to express my immense gratitude to, **Prof. Shweta Srivastava**, the Head of the ECE department for their constant support and motivation that has encouraged us to come up with this project and also for providing the right ambience for carrying out the work and the facilities provided to us.

We express our warm thanks to our project supervisor, **Mrs. Shradha Saxena,** Department of **Electronics and Communication Engineering**, **Jaypee Institute of Information Technology** for her skilful guidance, constant supervision, timely suggestion and constructive criticism in successful completion of our project in time.

We wish to thank all the staff of Electronics and Communication Department for providing me all support whenever needed.

We would like to thank my parents for supporting and helping in the Completion of the project. Last but not the least I would like to thank all my friends without whose support and co-operation the completion of project would not have been possible.

List of Figures

CHAPTER 1

Introduction

1.1 Background

A 2D CNC machine is a versatile tool that can be used for a wide range of applications, from cutting and shaping metal parts to engraving and etching designs into wood or acrylic. The CNC machine is controlled by a computer, which reads a digital file containing the design or pattern to be cut. This file is typically created using specialized software, which allows the user to define the shapes and dimensions of the cut, as well as the type of tool to be used.

Once the design file has been loaded into the CNC machine, the operator sets up the workpiece on the machine bed and aligns it with the cutting tool. The machine is then started, and the cutting tool moves along the X and Y axes, cutting the material according to the design specifications. The depth of the cut, the speed of the tool, and the direction of the cut can all be programmed and adjusted as needed, allowing for precise control over the machining process.

One of the advantages of a 2D CNC machine is its ability to produce multiple identical parts with high accuracy and consistency. This is particularly useful in manufacturing and production environments, where large quantities of parts are needed for assembly. The CNC machine can also be used to produce custom parts and prototypes, allowing designers and engineers to quickly test and refine their designs.

Another advantage of a 2D CNC machine is its ability to work with a wide range of materials, including metals, plastics, and wood. This makes it a valuable tool for a variety of industries, from aerospace and automotive to jewellery making and woodworking.

Overall, a 2D CNC machine is a powerful and versatile tool that offers high precision and repeatability, making it a valuable asset in many manufacturing and production environments. With advances in technology, CNC machines continue to evolve, offering even greater capabilities and efficiency for a wide range of applications.

1.2 Statement of Problems

The main purpose of this project is to develop a low cost 2D CNC Machine capable of printing texts and images.

1.3 Objectives

i. To design a low cost 2D plotter.

ii. To study about working of CNC Mechanism.

iii. To understand the working of different modules and motor drivers.

1.4 Significances

i. Precision: The 2D CNC machine is capable of producing high-precision cuts with accuracy down to a fraction of a millimetre. This allows for the production of intricate and complex designs with a high degree of accuracy and consistency.

ii. Efficiency: The 2D CNC machine is a highly efficient tool, capable of producing multiple identical parts quickly and accurately. This reduces the time and labor required for manufacturing and production, leading to cost savings and increased productivity.

iii. Scalability: The 2D CNC machine is capable of producing large quantities of parts with high accuracy and consistency. This makes it a valuable tool for large-scale manufacturing and production, as well as for smaller-scale production runs and custom orders.

1.5 Limitations

i. 2D CNC plotters are typically designed to work with softer materials, such as wood, plastics, and thin metals. They may not be suitable for working with harder or denser materials, which can be difficult to cut or may damage the cutting tools.

ii. The programming required to operate a 2D CNC plotter can be complex and time-consuming. This can be a significant barrier to entry for those who are new to CNC machining or may not have extensive programming experience.

CHAPTER 2

Components & Study Design

2.1 Specification of Hardware

2.1.1 Arduino UNO

Arduino Uno is a popular microcontroller board that is widely used in the development of DIY electronics projects, robotics, and automation systems. It was introduced in 2010 by the Italian company, Arduino SRL, and has since become one of the most widely used microcontroller boards in the world.

The Arduino Uno is based on the Atmel ATmega328P microcontroller, which runs at a clock speed of 16 MHz. It has 32KB of Flash memory, 2KB of SRAM, and 1KB of EEPROM memory. The board features 14 digital input/output pins, 6 analog inputs, and a USB interface for programming and serial communication.

One of the key advantages of the Arduino Uno is its ease of use. The board can be programmed using the Arduino Integrated Development Environment (IDE), which provides a simple and intuitive interface for writing, compiling, and uploading code to the board. The IDE is compatible with Windows, Mac, and Linux operating systems, making it accessible to a wide range of users.

The Arduino Uno is also highly customizable, with a range of shields and modules that can be added to expand its functionality. Shields are plug-in modules that can be attached to the board to add additional sensors, actuators, or communication interfaces. These shields can be purchased from a wide range of suppliers, or users can design and build their own shields using the open-source specifications provided by Arduino.

The versatility of the Arduino Uno has led to its use in a wide range of applications. It is commonly used in robotics projects, where it can be used to control motors, sensors, and other components. It is also used in home automation systems, where it can be used to control lighting, temperature, and security systems. Other applications include data logging, environmental monitoring, and machine control systems.

In addition to its technical capabilities, the Arduino Uno has also become popular due to its community of users and developers. There is a vast amount of resources available online, including tutorials, code examples, and forums where users can ask questions and share ideas. This community has helped to make the Arduino Uno accessible to a wide range of users, from hobbyists to professionals.

In conclusion, the Arduino Uno is a versatile and powerful microcontroller board that has become popular due to its ease of use, customizability, and vibrant community of users. It is widely used in a range of applications, from robotics and automation to home automation and environmental monitoring. With its open-source design and user-friendly interface, the Arduino Uno is likely to continue to play an important role in the development of DIY electronics projects for years to come.

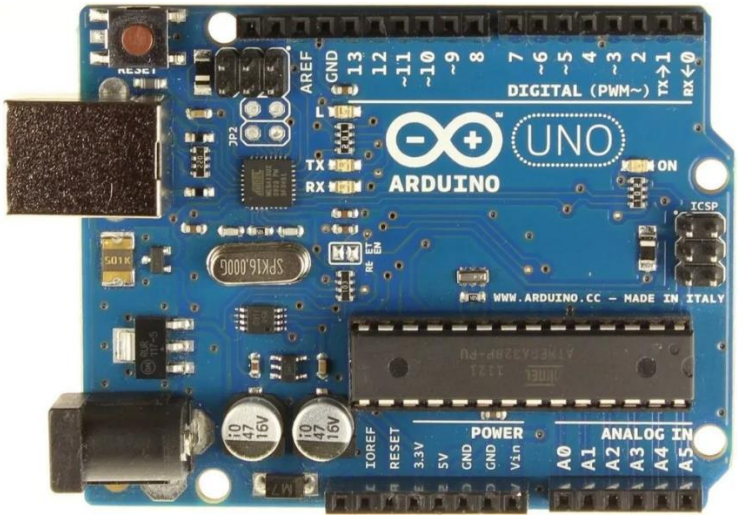

Figure 1: Arduino UNO [1]

2.1.2 NEMA 17 Stepper Motor

NEMA 17 stepper motor is a popular type of stepper motor used in a wide range of applications, including 3D printers, CNC machines, robotics, and automation systems. The name "Nema 17" comes from the standard created by the National Electrical Manufacturers Association (NEMA) that specifies the size and mounting requirements of the motor.

The Nema 17 stepper motor has a step angle of 1.8 degrees, which means it requires 200 steps to complete a full rotation. It has a holding torque of around 4-5 kg-cm, which makes it suitable for applications that require precise positioning or high torque output. The motor is also capable of running at high speeds, with a maximum speed of around 3,000 RPM.

The Nema 17 stepper motor is a bipolar stepper motor, which means it has two coils that can be energized in a specific sequence to make the motor shaft rotate. This makes it easy to control

the motor using a microcontroller or a stepper motor driver, which can generate the required signals to control the motor's movement.

One of the key advantages of the Nema 17 stepper motor is its versatility. It is available in a range of configurations, including different shaft lengths and types, as well as different wiring options. This makes it easy to find a motor that meets the specific requirements of a particular application.

The Nema 17 stepper motor is also relatively easy to use, with a simple four-wire connection for the bipolar configuration. The motor can be controlled using a variety of microcontrollers and stepper motor drivers, including popular platforms such as Arduino and Raspberry Pi.

Despite its advantages, the Nema 17 stepper motor does have some limitations. One of the main limitations is its low torque output compared to larger stepper motors, which can limit its use in high-load applications. Additionally, the motor may require additional cooling or protection in applications that require extended use or operate in harsh environments.

In conclusion, the Nema 17 stepper motor is a versatile and widely used stepper motor that offers precise positioning, high speed, and a simple control interface. Its popularity and availability make it a popular choice for hobbyists and professionals alike, with a range of applications in 3D printing, CNC machines, robotics, and automation systems. However, its low torque output and potential limitations in high-load or harsh environment applications should be carefully considered when selecting a motor for a particular application.

Figure 2: NEMA 17 Stepper Motor [2]

2.1.3 L298 Motor Driver

The L298 motor driver is a popular integrated circuit that is widely used in robotics and automation applications to control the speed and direction of DC motors and stepper motors. It is a dual H-bridge driver, which means it can control two motors independently, or one stepper motor with four coils.

The L298 motor driver is capable of driving motors with a voltage range of 7 to 46 volts and can supply up to 2 amps of current per channel. This makes it suitable for a wide range of applications, from small hobby projects to larger industrial automation systems.

The driver has several input pins that can be used to control the motor speed and direction. The input pins include two enable pins, which are used to turn the driver on and off, and four digital input pins, which are used to control the direction of the motor. By applying a combination of high and low signals to the input pins, the motor can be driven forward or backward at different speeds.

One of the key advantages of the L298 motor driver is its ease of use. The driver can be controlled using a microcontroller, such as an Arduino or Raspberry Pi, which can generate the required signals to control the motor. The driver also has built-in protection circuits, including thermal shutdown and overcurrent protection, which help to prevent damage to the driver or motor in the event of a fault.

The L298 motor driver can be used with a wide range of DC motors and stepper motors, making it a versatile choice for many different applications. It is commonly used in robotics and automation systems, such as mobile robots and 3D printers, as well as in automotive applications, such as electric vehicles and electric bicycles.

However, there are also some limitations to the L298 motor driver. One of the main limitations is its relatively low current output compared to other motor drivers, which can limit its use in high-power applications. Additionally, the driver can generate a significant amount of heat, particularly when driving high current motors, which may require additional cooling or heat dissipation measures.

In conclusion, the L298 motor driver is a popular and versatile motor driver that offers simple control, built-in protection, and a wide range of compatible motors. Its ease of use and availability make it a popular choice for hobbyists and professionals alike, with applications in robotics, automation, and automotive systems. However, its limitations in high-power applications and potential heat generation should be carefully considered when selecting a motor driver for a particular application.

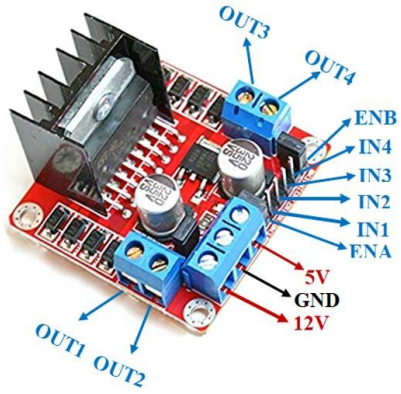

Figure 3: L298 Motor Driver [3]

2.1.4 LCD Display 16*2

A 16x2 LCD display is a type of alphanumeric display that consists of two rows of 16 characters each, making a total of 32 characters. These displays are widely used in various electronic devices, such as calculators, digital clocks, and other devices that require simple text output.

The 16x2 LCD display typically uses a parallel interface to communicate with the microcontroller or other electronic device. The display has a controller chip that controls the output of the display and receives commands and data from the microcontroller. The display also has a backlight, which provides illumination for the display in low-light conditions.

The 16x2 LCD display can display a wide range of characters and symbols, including letters, numbers, and special characters. The display can be used to display simple text messages, such as status messages, prompts, or instructions.

To use a 16x2 LCD display, the microcontroller sends commands and data to the display controller chip. The commands include instructions such as clearing the display, setting the cursor position, and turning the backlight on or off. The data is the text or symbols that are to be displayed on the screen.

One of the main advantages of the 16x2 LCD display is its simplicity and ease of use. The display is relatively low cost and requires only a few control signals to operate, making it an ideal choice for simple electronic devices. It is also highly customizable, with a wide range of fonts, sizes, and styles available.

However, there are some limitations to the 16x2 LCD display. One limitation is its relatively small size, which can limit the amount of information that can be displayed on the screen. The display is also not very suitable for displaying complex graphics or images, as it is primarily designed for displaying text.

In conclusion, the 16x2 LCD display is a simple and versatile display that is widely used in various electronic devices. Its ease of use and low cost make it a popular choice for hobbyists and professionals alike. While it has some limitations in terms of its size and ability to display complex graphics, it remains a useful tool for displaying simple text messages and status information.

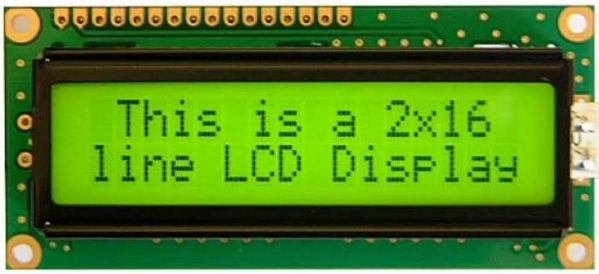

Figure 4: 16*2 LCD Display[4]

2.1.5 7805 Voltage Regulator

The 7805-voltage regulator is a linear voltage regulator that is widely used in electronic circuits to regulate a constant output voltage. It is a three-terminal device that can accept an input voltage in the range of 7V to 35V and outputs a regulated voltage of 5V at up to 1A.

The 7805-voltage regulator is a popular choice for powering low-power electronic circuits, such as microcontrollers, sensors, and other digital circuits. It is a simple and reliable solution that requires few external components to operate.

The regulator is designed to provide a stable output voltage, even when the input voltage varies or the load changes. It uses a series-pass transistor to control the output voltage, and a feedback circuit to adjust the transistor's bias current to maintain a constant output voltage.

The 7805-voltage regulator has several key advantages over other types of voltage regulators. It is a linear regulator, which means it provides a stable output voltage without any switching noise. It is also relatively low-cost and easy to use, requiring only a few external components to operate.

However, there are also some limitations to the 7805 voltage regulator. One limitation is its relatively low efficiency, as it dissipates excess power as heat. This means that it is not suitable for high-power applications or where energy efficiency is critical. Additionally, it has a limited input voltage range, which can limit its use in some applications.

In conclusion, the 7805-voltage regulator is a simple and reliable voltage regulator that is widely used in low-power electronic circuits. Its ease of use and low cost make it a popular choice for hobbyists and professionals alike. While it has some limitations in terms of efficiency and input voltage range, it remains a useful tool for regulating voltage in a wide range of applications.

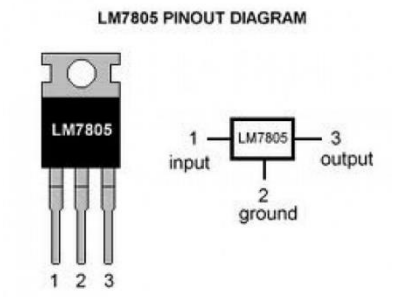

Figure 5: 7805 Voltage Regulator [5]

2.1.6 Electrolytic Capacitor

Electrolytic capacitors are a type of capacitor that are widely used in electronic circuits for their ability to store and release electrical energy. They are made of two conductive plates, called electrodes, that are separated by an electrolyte, a substance that conducts electricity.

The most common type of electrolytic capacitor is the aluminum electrolytic capacitor. It consists of a cylindrical aluminum foil, which acts as the anode, and a cathode, which is a thin layer of conductive material. The anode and cathode are separated by an electrolyte, typically a liquid or gel-like substance, which allows the capacitor to store and release electrical energy.

Electrolytic capacitors are often used in electronic circuits where high capacitance values are required. They are known for their high capacitance per unit volume, making them ideal for

applications where space is limited. They are also used in circuits that require high voltage ratings, as they can withstand high voltage levels without breaking down.

One of the limitations of electrolytic capacitors is their sensitivity to temperature and aging. The electrolyte inside the capacitor can dry out over time, causing the capacitance to decrease and the internal resistance to increase. This can lead to a decrease in the performance of the capacitor over time, especially at high temperatures.

Another limitation is that electrolytic capacitors are polarized, meaning that they must be connected to the circuit with the correct polarity. If they are connected incorrectly, they can be damaged or even explode.

In conclusion, electrolytic capacitors are a useful component in electronic circuits, particularly for applications that require high capacitance values and high voltage ratings. However, they do have some limitations, such as their sensitivity to temperature and aging, and their polarity requirements. It is important to select the correct type of electrolytic capacitor for the application and to handle them with care to avoid damage.

Figure 6: Electrolytic Capacitor

2.1.7 LED

A 3V LED is a light-emitting diode that operates at a voltage of 3 volts. LED stands for Light Emitting Diode, which is a semiconductor device that emits light when an electric current flows through it. LEDs are widely used in electronic circuits for their energy efficiency, long lifespan, and low heat output.

The 3V LED typically operates at a low current, making it ideal for battery-powered devices. It can also be used in a wide range of electronic circuits, such as indicator lights, displays, and backlighting. LEDs are available in a range of colors, including red, green, blue, white, and amber.

One of the advantages of using a 3V LED is its energy efficiency. LEDs consume less power than traditional incandescent bulbs, making them ideal for battery-powered devices. They also have a longer lifespan, with some LEDs capable of lasting up to 50,000 hours or more.

Another advantage of using a 3V LED is its low heat output. LEDs do not produce as much heat as incandescent bulbs, making them ideal for use in electronic circuits where heat can be a problem.

One limitation of using a 3V LED is that it requires a constant voltage source to operate. If the voltage is too high, the LED can be damaged, and if it is too low, it may not emit light at all. To ensure that the LED operates correctly, a resistor is typically used in series with the LED to limit the current and prevent damage.

In conclusion, the 3V LED is a versatile and energy-efficient component that is widely used in electronic circuits. Its low heat output and long lifespan make it ideal for a range of applications, including battery-powered devices and backlighting. While it does require a constant voltage source and a resistor to operate correctly, the advantages of using a 3V LED make it a popular choice for hobbyists and professionals alike.

Figure 7: 3V LED

2.1.8 Servo motors

A servo motor is an electrical device which can push or rotate an object with great precision. If you want to rotate and object at some specific angles or distance, then you use servo motor. It is just made up of simple motor which run through servo mechanism. If motor is used is DC powered then it is called DC servo motor, and if it is AC powered motor then it is called AC

servo motor. We can get a very high torque servo motor in a small and light weight package. Doe to these features they are being used in many applications like toy car, RC helicopters and planes, Robotics, Machine etc.

Servo motors are rated in kg/cm (kilogram per centimetre) most hobby servo motors are rated at 3kg/cm or 6kg/cm or 12kg/cm. This kg/cm tells you how much weight your servo motor can lift at a particular distance. For example: A 6kg/cm Servo motor should be able to lift 6kg if the load is suspended 1cm away from the motors shaft, the greater the distance the lesser the weight carrying capacity. The position of a servo motor is decided by electrical pulse and its circuitry is placed beside the motor. A servo consists of a Motor (DC or AC), a potentiometer, gear assembly and a controlling circuit. First of all we use gear assembly to reduce RPM and to increase torque of motor. Say at initial position of servo motor shaft, the position of the potentiometer knob is such that there is no electrical signal generated at the output port of the potentiometer. Now an electrical signal is given to another input terminal of the error detector amplifier. Now difference between these two signals, one comes from potentiometer and another comes from other source, will be processed in feedback mechanism and output will be provided in term of error signal. This error signal acts as the input for motor and motor starts rotating. Now motor shaft is connected with potentiometer and as motor rotates so the potentiometer and it will generate a signal. So as the potentiometer's angular position changes, its output feedback signal changes. After sometime the position of potentiometer reaches at a position that the output of potentiometer is same as external signal provided. At this condition, there will be no output signal from the amplifier to the motor input as there is no difference between external applied signal and the signal generated at potentiometer, and in this situation motor stops rotating.

Figure 8: Servo motor

CHAPTER 3

Methodology

3.1 Theoretical Framework

We have supply the current in Arduino with USB DATA cable to transfer Data from Computer to Arduino Board [1], Here we have used 2 Stepper Drivers to supply the G codes in Sequence to the stepper motors. Arduino will be mounted on base. L298 motor drivers will be distributing the Current in the command of Arduino. Motor driver will be converting the command of G codes in digital pulse by Stepper motor. In X direction Stepper motor will be move left and Right, Y direction stepper motor will be move in front and back direction, Z direction Stepper motor will be move in Up and down [2]. We have made much difficult design via using this machine. The accuracy of this machines result is very high. So we have used in industry to reduce the cost of design printing and maintain accuracy level. Drafting and Scaling of CNC Plotter machine is very precious.

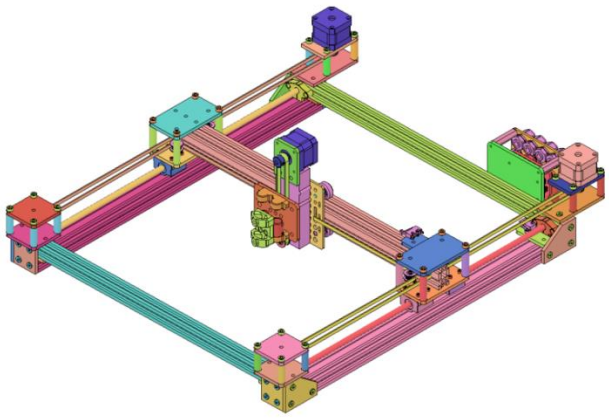

Figure 9: 2D CNC Plotter

3.2 Literature Review

3.2.1 Application of 2D CNC Plotters

1. **Signage and graphics:** 2D CNC plotters are commonly used in the signage and graphics industry to create high-quality signs, logos, and banners. They are capable of cutting and engraving various materials, including vinyl, wood, plastic, and metal.Active solar water heating systems**.**

2. **Electronics manufacturing:** In the electronics industry, 2D CNC plotters are used for prototyping and production of printed circuit boards (PCBs). They can accurately drill and mill PCBs, allowing for the precise placement of electronic components.

3. **Textile and fashion:** 2D CNC plotters are also used in the textile and fashion industry for cutting fabrics and creating patterns. They are capable of cutting various materials, including leather, denim, and synthetic fabrics.

4. **Art and design:** 2D CNC plotters are used by artists and designers to create intricate and precise works of art. They can cut and engrave various materials, including paper, wood, and plastic.

5. **Architecture and model making:** Architects and model makers use 2D CNC plotters to create precise models of buildings and other structures. They are capable of cutting and engraving various materials, including foam, wood, and acrylic.

6. **Education and research:** 2D CNC plotters are commonly used in educational settings for teaching and research purposes. They provide students and researchers with hands-on experience in computer-aided design (CAD) and manufacturing (CAM).

CHAPTER 4 Algorithm & Block Diagram of Circuit

4.1. Algorithm of The Circuit

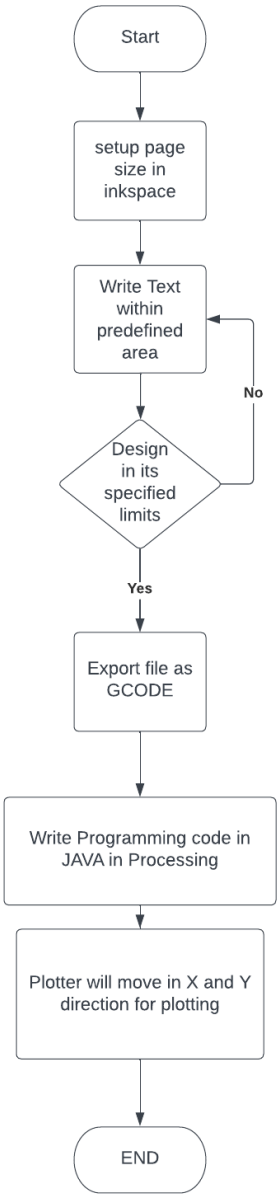

Figure 10: Flowchart of the Circuit

4.2. Block Diagram of Circuit

Figure 11: Block Diagram of the circuit

CHAPTER 5
Design & Construction

5.1 Mechanical Design

On the basis of availability of motors, a simple plotter prototype has been designed.

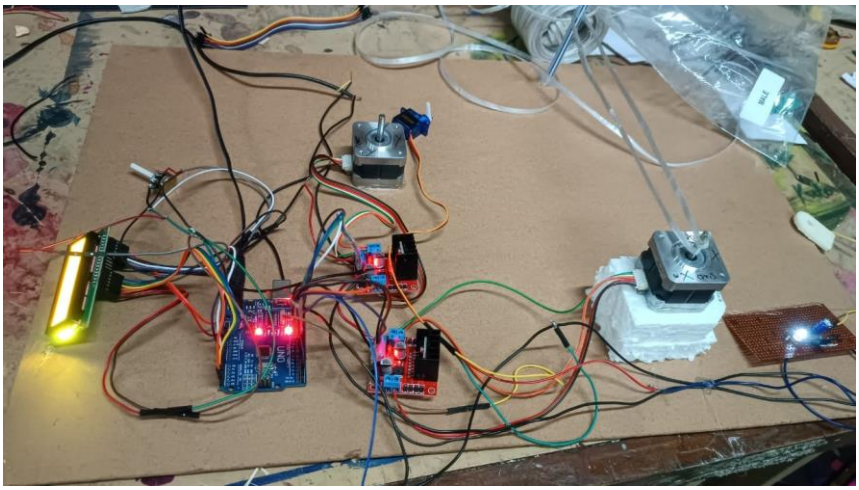

Figure 12: Mechanical Design

5.2 Electrical Design

Under electrical portion, design includes the connection of LDRs, resistors, Capacitor, Variable Resistor, 4148 Diode LM 358 IC, 555 Timer IC, LDR Sensor, L293D Motor Driver, dc to dc boost converter LM2587, LiPo Battery Charger Module Mini TP4056 IC and servo motors together in the bread board. Beside this an additional connection of battery, invertor, charge controller and load are also made. The circuit diagram is shown in the figure 14.

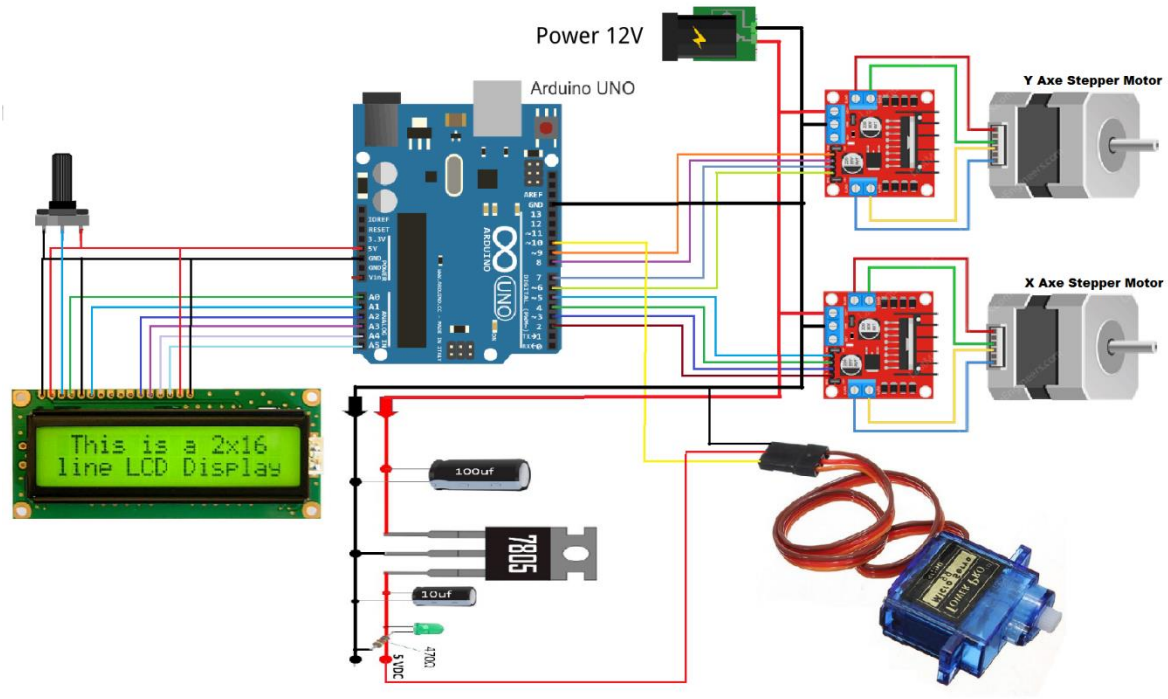

Figure 13: Circuit Diagram

CHAPTER 6
CODES

6.1 Arduino Uno Code Snippets:

Arduino_CNC_Plotter

```
#include <Servo.h>
#include <Stepper.h>
#include <LiquidCrystal.h>
LiquidCrystal lcd(A0, A1, A2, A3, A4, A5);
#define LINE_BUFFER_LENGTH 512

// Servo position for Up and Down
const int penZUp = 60;
const int penZDown = 80;

// Servo on PWM pin 6
const int penServoPin = 10;

// Should be right for DVD steppers, but is not too important here
const int stepsPerRevolution = 50;

// 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 myStepperX(stepsPerRevolution, 2,3,4,5);
Stepper myStepperY(stepsPerRevolution, 9,8,7,6);

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

// Current position of plothead
struct point actuatorPos;
```

Figure 14: a) Code Snippet

```
//  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 = 300;
float Ymin = 0;
float Ymax = 300;
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 (
```

Figure 11: b) Code Snippet

```
void setup() {//Setup
Serial.begin( 9600 );
lcd.begin(16, 2);
lcd.clear();
lcd.setCursor(5,0);
lcd.print("WELCOME");
penServo.attach(penServoPin);
penServo.write(penZUp);
delay(1000);
lcd.clear();

// Decrease if necessary
myStepperX.setSpeed(250);
myStepperY.setSpeed(250);

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()
{
  delay(200);
  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");
      }
```

Figure 14: c) Code Snippet

```
            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)
```

Figure 14: a) Code Snippet

6.2 Java Processing code:

gctrl

```
import java.awt.Frame;
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 = "COM3"; // 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');
}
```

Figure 15: a) Java code Snippet

```
  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(500, 250);
  openSerialPort();
}

void draw()
{
  background(0);
  fill(255);
  int y = 24, dy = 12;
```

Figure 15: b) Java Code Snippet

```
  fill(255);
  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");
```

Figure 15:c) Java Code Snippet

```
  if (!streaming && key == 'g') {
    gcode = null; i = 0;
    File file = null;
    println("Loading file...");
    selectInput("Select a file to process:", "fileSelected", file);
  }

  if (key == 'x') streaming = false;
}

void fileSelected(File selection) {
  if (selection == null) {
    println("Window was closed or the user hit cancel.");
  } else {
    println("User selected " + selection.getAbsolutePath());
    gcode = loadStrings(selection.getAbsolutePath());
    if (gcode == null) return;
    streaming = true;
    stream();
  }
}

void stream()
{
  if (!streaming) return;

  while (true) {
    if (i == gcode.length) {
```

Figure 15: d) Java Code Snippet

CHAPTER 7
Marketing And Material

7.1 Marketing

Marketing is done for the purchases of IC's, motors, bulb, veroboard and wires. While for the frame initially aluminium is preferred but due to unavailability of required thickness and difficult in fabrication, wood was used as its successor.

7.2 Fabrication

Fabrication includes the frame, fixing of motors and stand. This were carried out as per the design and material selection.

7.3 Testing and modification

After the completion of fabrication, project was tested for different text.

7.4 Material selection

The material selected can be wooden, plastic, iron etc. but we have used wood as the material because it was suitable to our fund. Use of plastic can be a better option because plastic can resist itself in drastic weather changes like extreme heat and rain.

CHAPTER 8
Result

8.1. Software

8.1.1 Description of the Software:

In this project we use **INKSPACE 1.2.2, Arduino IDE, Processing. pde software.** Arduino Ide is used to program the Arduino uno, inkspace is used for writng text and converting it to GCODE for coordinates of the text, Processing software is used to program microcontroller for the motor drivers to work.

8.2 Hardware

When the power is supplied to the Arduino board and the circuit, the welcome screen gets displayed on the LCD screen. After that we create a design in inkspace software, the inkspace software then process it to convert it in to a GCODE file so that we can get the coordinates of the design that we have created, after we get the coordinates, the Arduino is programmed using processing software that can read the GCODE file from provided location, and after reading the coordinates from GCODE file, the motor driver directs the motors to move accordingly in the specified X-Y plane to plot the points.

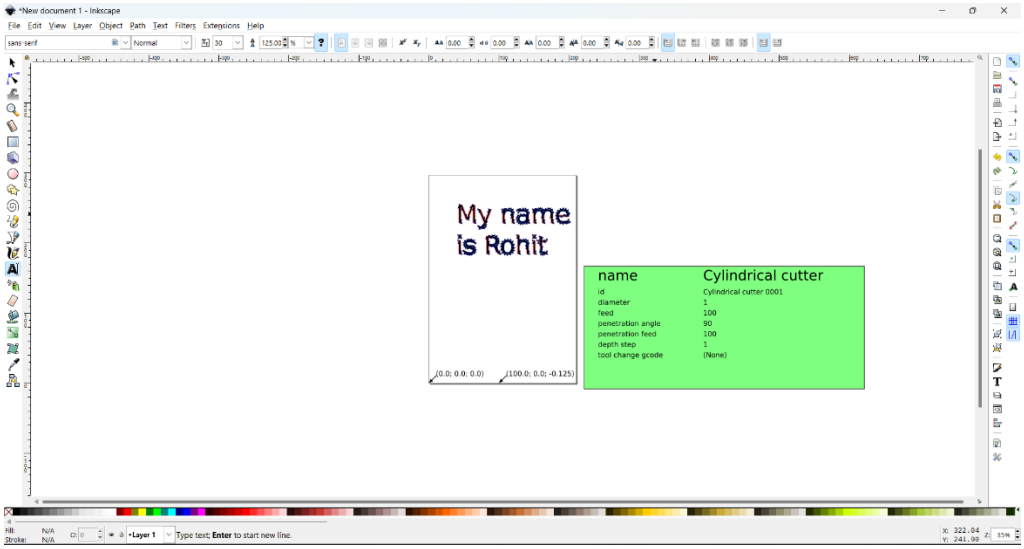

Figure 16: inkspace design

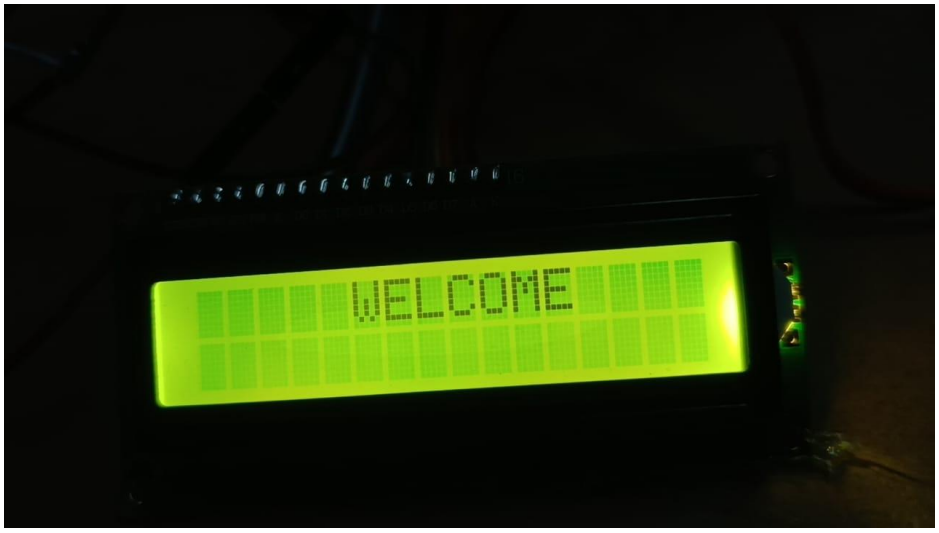

Figure 17: Welcome Screen on LCD

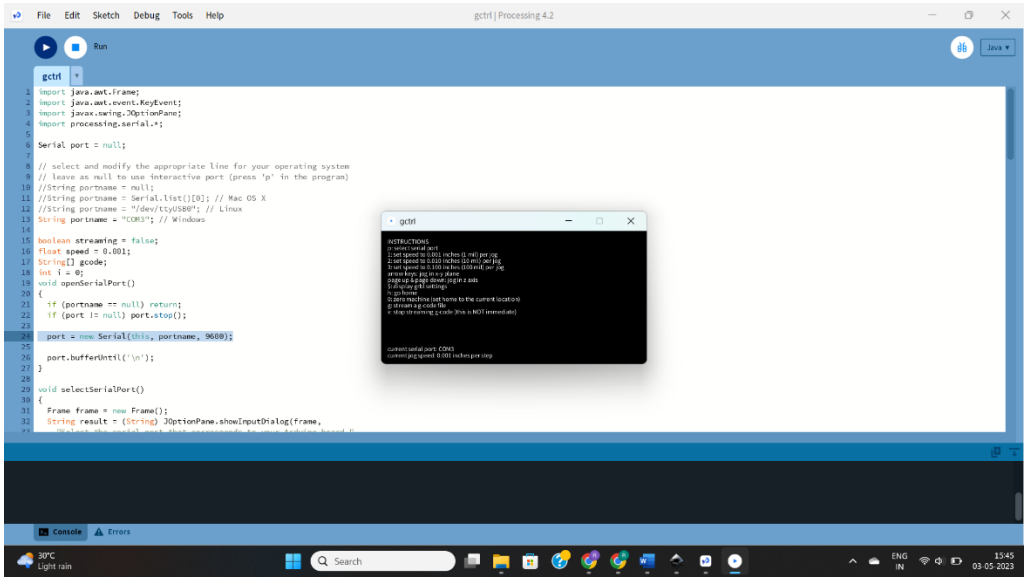

Figure 18: Processing Software

CHAPTER: 9
Conclusion And Recommendation

9.1 Conclusion

In this project, we used the concept of a low-cost plotter machine, which is easily controlled by the computer. It is easy to operate and the work also can be done accurate without any errors. This small machine can be easily transported and assembled everywhere as required. The board size of this device is 40X40 cm. Stepper Motor will run on this standard for board size. If we have an increase in the size or length of the lead coil, it will be free to make the large size of the design on paper. Due to small size it is restricted to a certain area width and length.

9.2 Recommendation

Though we have performed our work in much efficient way. There is still room for improvement for this system and it is hoped that further study can be carried out to further develop the system.

- Use higher motors with large torque value for complex designs.
- It will be better to use higher page size for clear plotting.

CHAPTER 10

Reference

[1] Team, T.A. (no date) Uno R3, Arduino Documentation. Available at: https://docs.arduino.cc/hardware/uno-rev3 (Accessed: Mar 3, 2023).

[2] Stepper Motor (2023) Wikipedia. Wikimedia Foundation. Available at: https://en.wikipedia.org/wiki/Stepper_motor (Accessed: Mar 3, 2023).

[3] Light-emitting diode (2023) Wikipedia. Wikimedia Foundation. Available at: https://en.wikipedia.org/wiki/Light-emitting_diode (Accessed: Mar 5, 2023).

[4] What is a servo motor? - understanding the basics of servo motor working (no date) Servo Motor Basics, Working Principle & Theory. Available at: https://circuitdigest.com/article/servo-motor-working-and-basics (Accessed: May 3, 2023).

[5] Welcome to processing! (no date) Processing. Available at: https://processing.org/ (Accessed: 03 May 2023).

[6] (No date) Design and implementation of low-cost 2D plotter computer numeric ... Available at: https://www.researchgate.net/publication/336584840_Design_and_Implementation_of_low-cost_2D_plotter_Computer_Numeric_Control_CNC_Machine (Accessed: 03 May 2023).