An-Najah National University

Faculty of Engineering and Information Technology

Computer Engineering Department

Hardware Graduation Project

The Artist

Students:

Mahmoud Shouli
Nabeel Jamous

Supervisors:

Manar Qamhieh, Ph.D.

Omar Tamimi, MA.

Presented in partial fulfillment of the requirements for Bachelor degree in Computer Engineering.

Jan 31, 2025

Acknowledgement

This project wouldn't be possible without the huge support of our families who were the backbone behind us, the supervisors who kept watching and assessing our progress, giving us periodic feedback and comments, whether they were positive or negative, and our instructors who taught us valuable experiences and lessons throughout these past five years.

I would also love to give a huge shout out to the technical crew at the faculty for their kindness and tolerance. They were always there when we needed. And lastly our friends and colleagues that made this journey easier and funnier.

Disclaimer

This report was written by student(s) at the Computer Engineering Department, Faculty of Engineering, An-Najah National University. It has not been altered or corrected, other than editorial corrections, as a result of assessment and it may contain language as well as content errors. The views expressed in it together with any outcomes and recommendations are solely those of the student(s). An-Najah National University accepts no responsibility or liability for the consequences of this report being used for a purpose other than the purpose for which it was commissioned.

Table of Contents

List of Figures

Abstract

The Artist is a 2D CNC plotting machine that draws a sketch of real time photos. The user initiates the photo capturing using an app, the photo is then processed using image processing techniques to determine the outlines and the details of the photo, and the camera is controlled by a Raspberry Pi controller. Then the photo is drawn on paper using different colours. The pens move on the CNC machine in X and Y directions using stepper motors controlled by an Arduino controller. After the drawing is completed the paper is then taken on a production line in order to stamp it, and then hand it to the user in a fashionable way.

The significance of this project lies in its accuracy, flexibility and the ability to plot on different paper sizes, efficiency, and the automation of taking a photo, drawing it, and then converting it into something that the user will love to take. The Artist can draw any photo of any object, but our main objective is to draw a sketch of a human face to create an exciting experience for the user.

The development of this project consists of three stages:

- Developing an app where the user will choose the paper size and the colours of the outline and the details.
- Assembling the CNC machine and writing the software for it to draw the photo.
- Manufacturing the short production line to produce the final output.

There have been several similar projects that have been done before where they have the core functionality of plotting via CNC, but The Artist stands out amongst them with its extra features.

Chapter 1 Introduction

Computer Numerical Controls (CNC) are motorized machines that are computer-controlled, they accept instructions that dictate their movement along the different axes. This movement can then be associated with some defined functionality by mounting different devices (such as a Laser Engraver, a 3D printing nozzle or a drawing device) on the moving parts.

A Pen Plotter (also called an XY Plotter) is a "specimen" of CNC machines. It plots against the X and Y axes carrying a pen that moves up and down in discrete steps, drawing different shapes on a canvas.

G-Code (which stands for Geometric Code) is one of the most common CNC instruction formats out there. It consists of different operations that represent movements along the axes, each instruction starts with the word G followed by the number that distinguishes each instruction [1], and parameters are then passed to the instruction to create the desired movement.

Generating shapes to use against a CNC machine is a process that can be simplified by a Computer-Aided Design (CAD) program, which are used to "create, manipulate, and optimize objects, pictures, or other designs" [2]. These program output usually need to go through another step to convert them to proper G-Code that the CNC machine would be able to understand, a popular example of such processors are Slicer programs, which are used in 3D printing to convert 3D models (usually in STL formats) into printing instructions that create and represent the shape with its printing specification (amount of filling, speed of the machine and precision).

This paper discusses The Artist system, which plots a human face captured through a camera on a paper. The system has a UI that lets you shoot a picture, reshoot it if you don't like it, see the enhanced picture using some simple image processing techniques, provides you with the ability to choose the preferred page size and pen color. The picture is then generated into a line art sketch using AI through a mobile app like Picsart or ArtistA, this sketch is then generated into a G-code file using a specialized software named Inkscape, after that the chosen pen is grabbed and the chosen paper is rolled, and lastly the sketch is drawn.

Chapter 2 discusses the constraints of The Artist design and earlier courses that are relevant to this project. Chapter 3 looks at similar literature work done in the same field. The implementation and architecture are covered in chapter 4.Chapter 5 discusses the communication between the Raspberry Pi and the Arduinos.

Chapter 6 discusses the mechanical aspects of the system. Chapter 7 looks at the results of tests run on the system and analyzes them. Chapter 8 concludes the paper and talks about possible future work.

Chapter 2 Constraints and Earlier Course Work

2.1 Constraints

- **Inability to automate sketch generation**: the apps that convert human faces into line art sketches use AI and train a Machine Learning model with a huge data set, and this is out of the scope of work of our project, so we manually put the photo onto the app.

- **Inability to automate G-code generation:** unfortunately the extension that we use on Inkscape to generate g-codes that best fits with our machine is not available via Inkscape CLI, so we had to manually put the photo on Inkscape GUI and customize the settings.

- **GRBL supports only one servo motor:** the GRBL framework only supports M3 commands for one servo for holding the bottom of the pen, leaving the top of it hanging which causes small circles like the eyes to be inaccurate especially in small scales.

2.2 Earlier Course Work

- **Microcontrollers Course:** This course provided the basics of dealing with microcontrollers, and how to interface them with different hardware components like motors.

- **Digital Electronic Course:** This course helped with understanding electronic circuits and dealing with different modules.

- **Microcontrollers Lab**: This lab introduced us to Arduinos and how to use them and interface them with components such as LCD, Stepper Motor, and Servo Motor.

- **Digital Image Processing Course:** This course provided an understanding of basic image processing techniques which are used in the machine pipeline to remove the background and enhance the image.

Chapter 3 Literature Review

CNC plotters have been common for decades now, the community is huge and there is a variety of different implementations and architectures.

For us, we had to watch a lot of YouTube videos and read multiple articles to understand the mechanism and science behind it, here is some of the topics we looked at:

3.1 GRBL Software: Everything you need to know

Gained basic understanding of the GRBL software and how it actually works and how to use it in our project. [3]

3.2 AxiDraw V3

AxiDraw V3 is a commercial Pen Plotter, it offers the basic functionality of a Pen Plotter, with the ability to mount different drawing devices on it. It has its own software to operate it, which can take SVG formatted files and draw them. [4]

The Artist offers a very similar functionality to the AirDraw V3, however, instead of focusing on printing SVG files, the Artist offers the ability to draw actual images that can be captured by a camera, with non-vector extensions like PNG and JPG, with an acceptable accuracy.

Chapter 4 System Architecture

4.1 General Design

This is what the project looks like from side view, we built everything from scratch, and no 3D printing was used.

The project is divided into two sections, the plotter which is handled by the Arduino Uno, and the paper feeders and roll handled by the Arduino Mega.

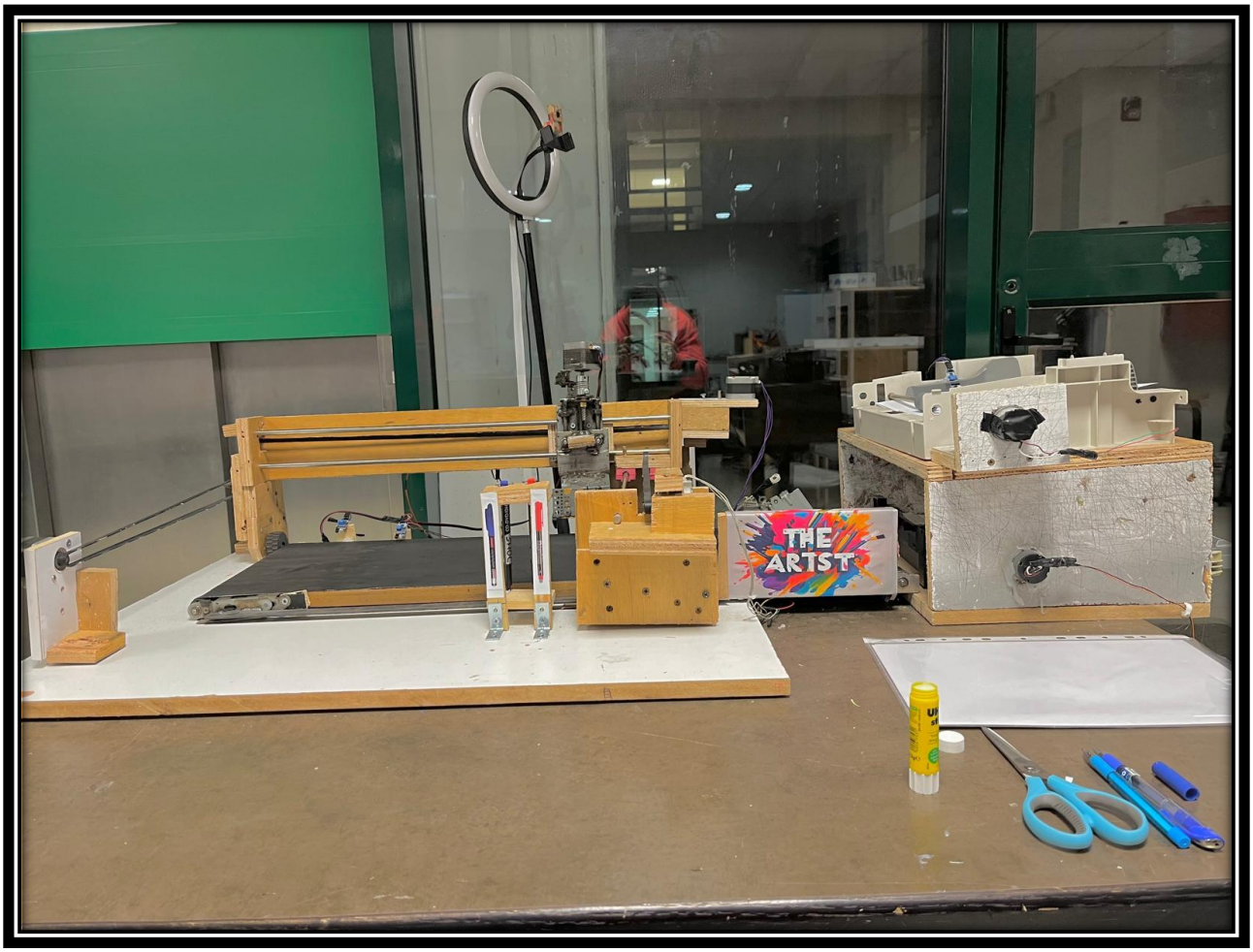

Figure 4-1: general design

4.2 Pipeline

This is the full processing pipeline in our system, there are 6 main stages as seen, but I will go through some secondary stages in addition with the mains.

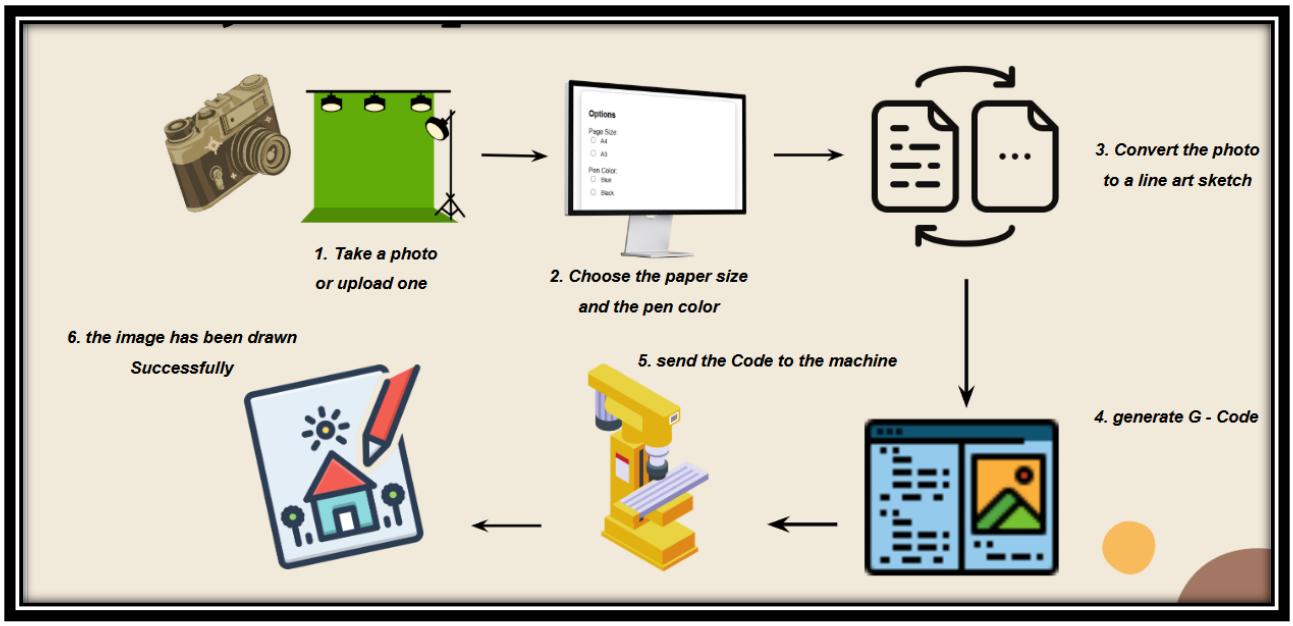

Figure 4-2: Pipeline

4.2.1 Web App

This is a simple website hosted on the Raspberry Pi, made by simple HTML and CSS. It lets you shoot a picture, upload a G-code file, choose some options and then start the process.

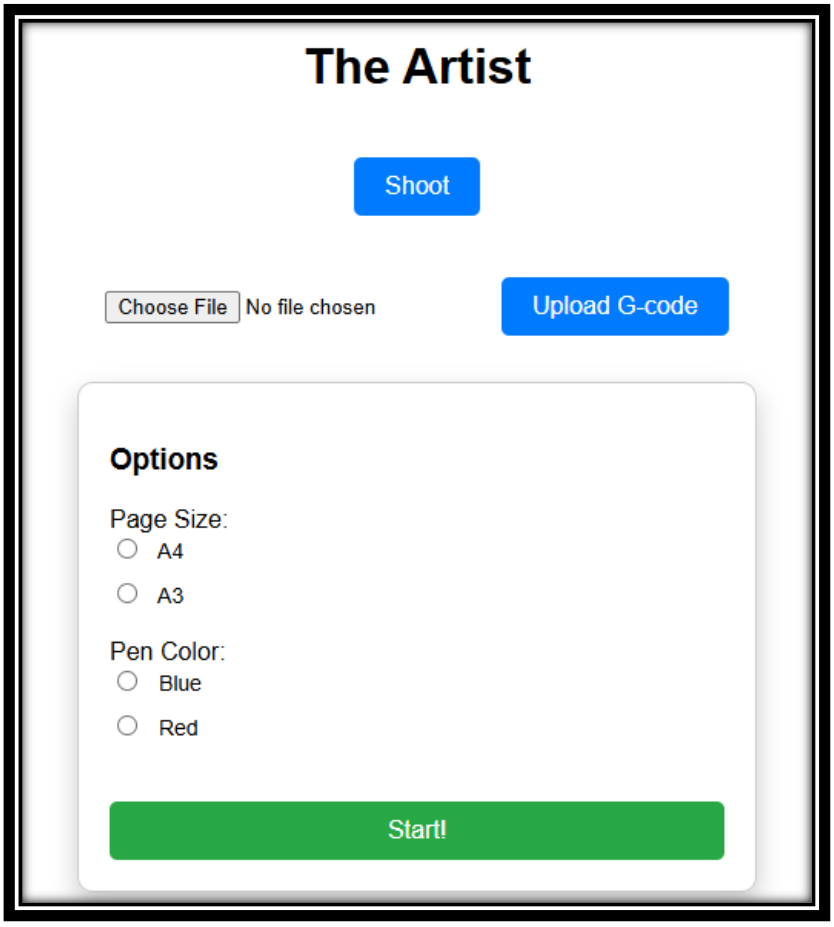

Figure 4-3: website

4.2.2 Image Processing and Converting to Line Art

Line art or line drawing is any image that consists of distinct straight lines or curved lines placed against a background.

In the beginning we tried to apply edge detection techniques like Canny edge detection to get the outlines of the face, the image is first converted into gray scale, then Gaussian blur is applied to reduce noise, get the edges using Canny edge detection, and finally invert the edges to make the image look like a sketch. [5]

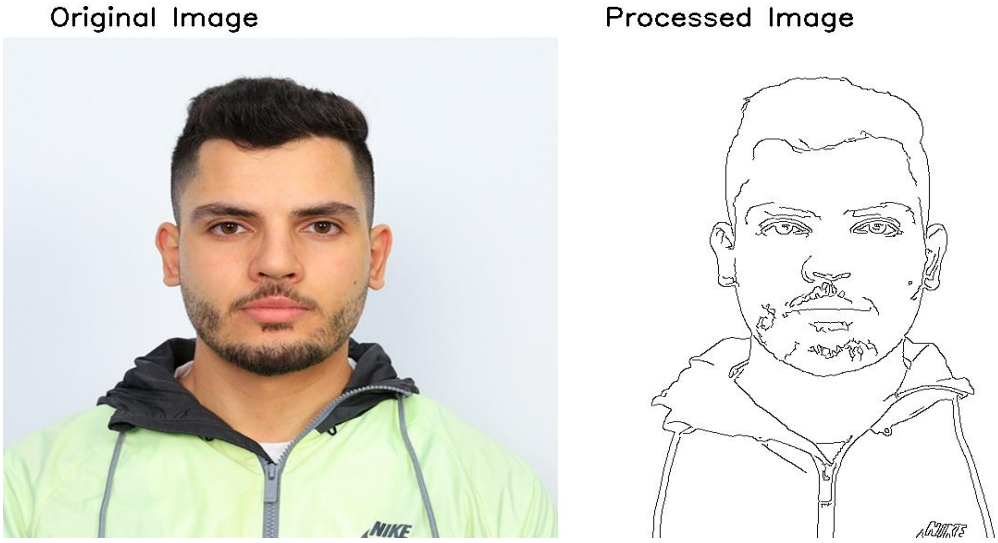

Figure 4-4: Canny edge detection

It seems good yes, realistic yes, but for our machine it's not suitable at all, our aim is to generate a simple smooth outline sketch with no details inside the features, basically a line art sketch, which is beyond simple image processing, it needs AI and Machine Learning, so we decided to simply use a mobile apps like Picsart and ArtistA that use AI to generate sketches from human faces.

So what we did is we used image processing to remove the green background by creating a mask for green color and then applying bitwise AND, and then use a sharpening filter to enhance the outlines.

This is an example of what an AI generated sketch looks like:

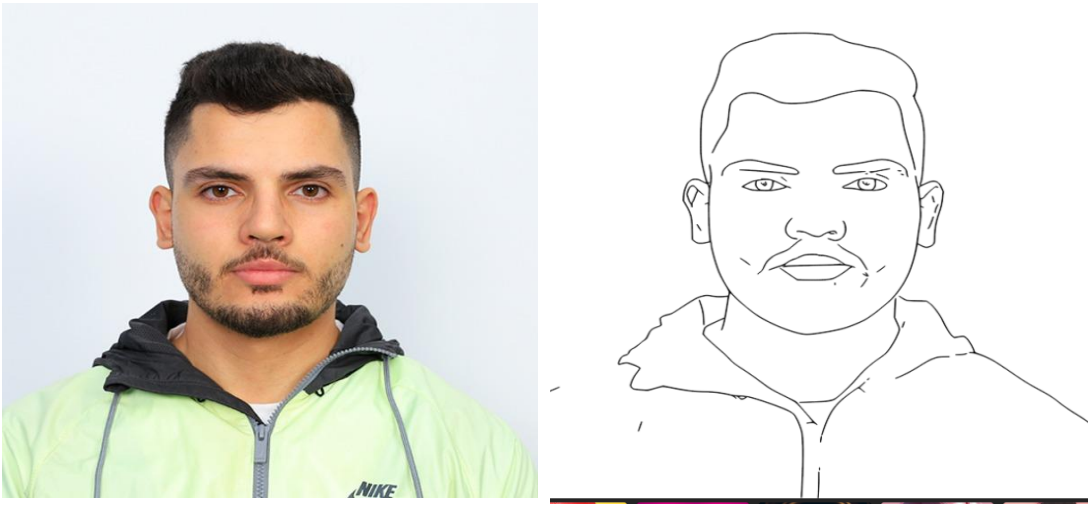

Figure 4-5: AI sketch

The lines are smoother and more connected as desired.

Yes it removes some features like the beard but we had to sacrifice it in order for the machine to work at its best.

4.2.3 Generating G-code

G-code is an instruction that tells the GRBL controlled machine how and where to move along the XYZ axes.

In our project we needed these type of commands:

1) **G10 L20 P1 X0 Y0 Z0**: The Reset Zero command, it sets the current position as the origin point.

2) **G90:** this command activates Absolute Positioning relative to the origin point, meaning after that if we moved the machine to X50 and Y50 for example, it will move to that exact position, and if we then repeated the command, it would not move another 50 in the X and Y but it will stay at (50,50)

3) **G00 X… Y..:** Rapid Positioning, it moves the machine at its maximum traversal speed to the given coordinates. It's used for movements with the pen up (no drawing).

4) **G01 X… Y… F…:** Linear Interpolation (Controlled Motion), this command moves the machine with a given feed rate (F) or speed to the given coordinates. It's used for precise movements like drawing with the pen down.

5) **M3 S…:** This command is for controlling the servo motor with the given angle. We used M3 S150 for dropping the pen, and M3 S0 to hold it.

To generate the G-code from an image, we used Inkscape with the gcodeplotutils extension. Inkscape gives you the ability to choose the paper size, then position the image on the canvas, apply centerline tracing, and save the file as .gcode using the extension.

The extension provides you with customizable settings like the width and height, the G00 feed rate and the G01 feed rate, and lastly the pen up and pen down command.

After tweaking a lot with these values, we settled for movement speed (G00 feed rate) = 25 mm/sec which translates to F1500, and for drawing speed (G01 feed rate) = 10 mm/sec which translates to F600.

Pen up command is G00 Z0 and pen down command is G00 Z2.4.

4.2.4 Grabbing the pen

We have a pen stand that holds two pens, a red one and a blue one as shown in the website, this is what it looks like:

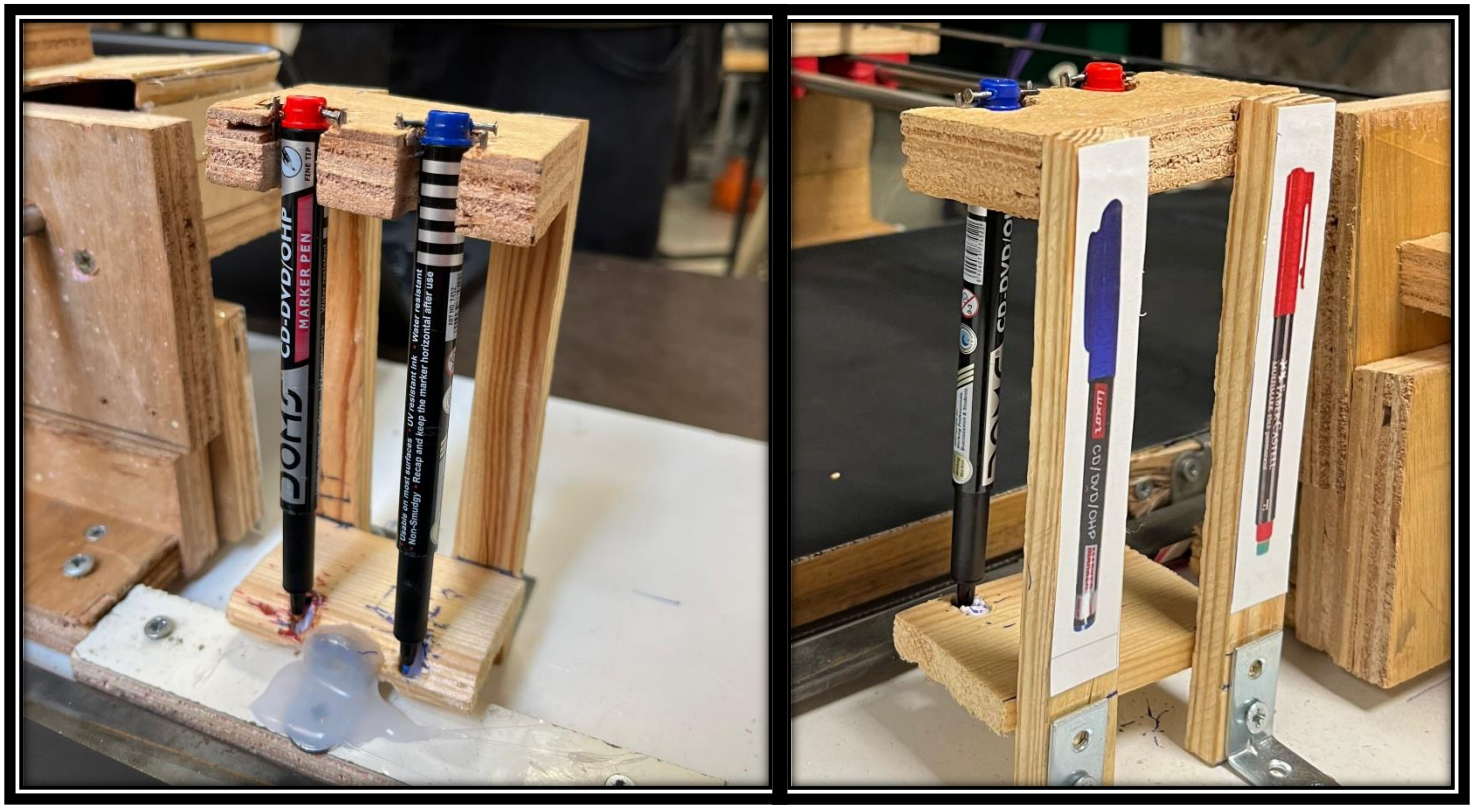

Figure 4-6: pen stand

We figured out the set of G-code commands to grab each color by manually testing, it took a lot of trial and error as it's a very precise process.

This the order of execution:

- Moves in Y direction to exactly meet the Y position of the pen.
- Moves forward in X direction to come close to the pen.
- Moves a little bit down in Z direction.
- Moves a bit more forward in X direction to meet the pen.
- Closes the servo motor to grab the pen.
- Goes back to the origin point.

4.2.5 Rolling the paper

We have two paper sizes, hence the two paper feeders, a feeder for the A3 and a feeder for the A4.

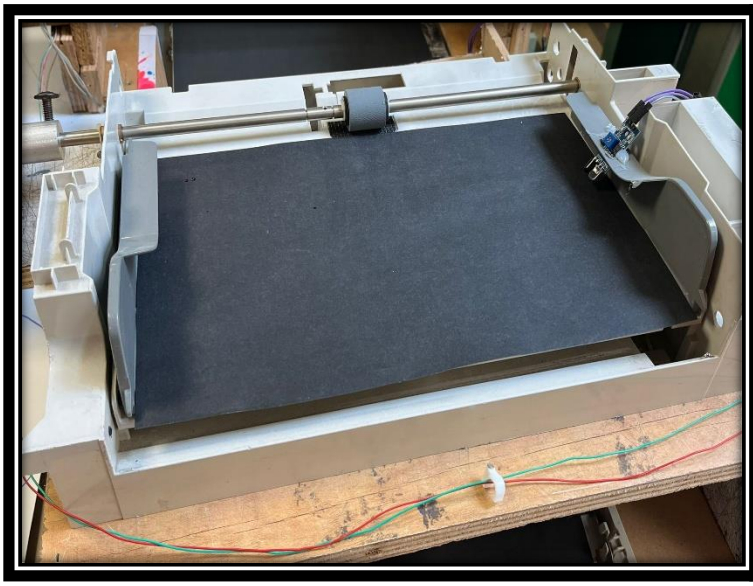

Figure 4-7: A3 feeder

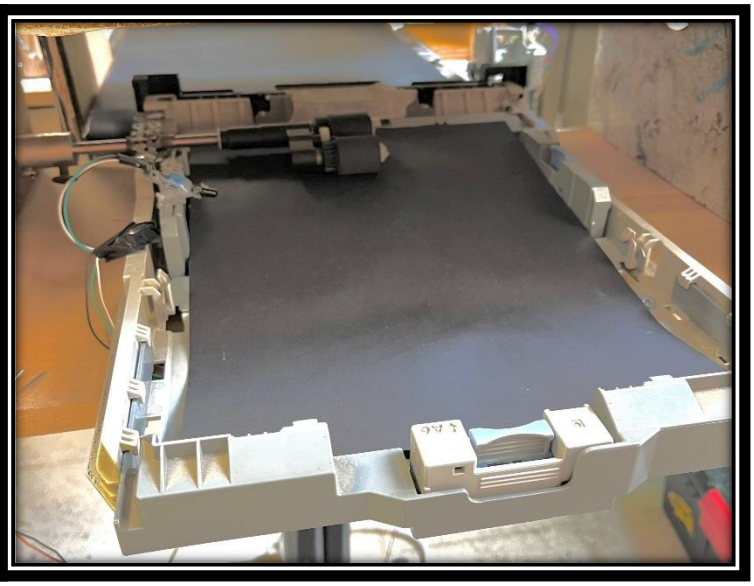

Figure 4-8: A4 feeder

The feeders have an IR sensor that detects if a paper exists or not, we put a black paper to differ from white paper.

DC motors with gearbox are used to roll the page, the motors run at maximum speed and for 17 seconds, after that the main roll which is controlled by a stepper motor starts rolling till the paper arrives at the desired IR.

These motors and IR's and connected to the Arduino Mega.

This is the main roll with the two IR sensors, the first one is the A4 IR sensor to detect that this is where it should stop, and the other one is for the A3.

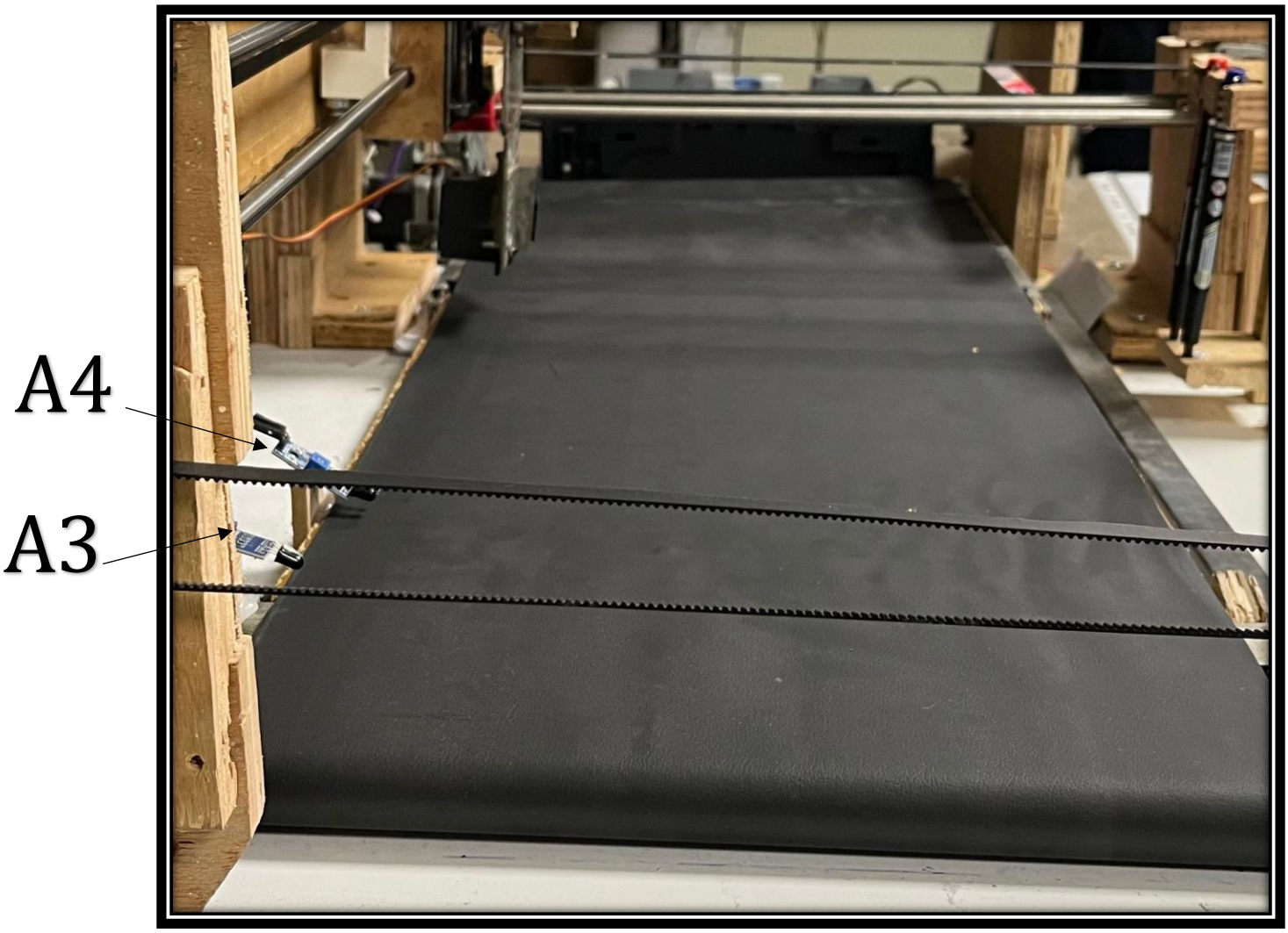

Figure 4-9: main roll

4.2.6 Drawing the photo

In this stage the uploaded G-code file is converted to an array and then this array of G commands is sent serially from the Raspberry Pi to the Arduino Uno which has GRBL flashed on.

These commands are executed one by one, but before it proceeds to the next instruction, it waits for an 'ok' response from the machine, that's how GRBL works.

During the pen grabbing and pen returning, we configure the Uno with the following values for speed settings:

$110 = 1100.00 (X-axis maximum rate, mm/min)

$111= 1100.00 (Y-axis maximum rate, mm/min)

But during the drawing, we slow it down a little bit because we aim for accuracy.

$110 = 800.00 (X-axis maximum rate, mm/min)

$111 = 800.00 (Y-axis maximum rate, mm/min)

The difference between these rates and the ones generated in the G-code, that these values are the maximum, and the ones in the G-code are the actual speed that the machine runs on. If the feed rate in the G-command exceeded the max, it will automatically reset it to max speed.

4.2.7 Signing

After the machine is done drawing the main sketch, it goes back to the origin point and then it starts drawing this simple signature at the bottom left corner. (A stands for Artist)

Figure 4-10: signature

4.2.8 Returning Pen

Just like grabbing the pen, the pen is returned to its place on the stand after its done drawing.

We concluded the instructions manually after a lot of trying and testing to get the precise right commands.

Order of execution:

- Moves in Y direction to exactly meet the Y position of the pen.
- Moves forward in X direction.
- Moves a little bit down in Z direction.
- Open the servo to drop the pen.
- Goes back to the origin point.

4.3 GRBL

"GRBL is a free, open source, high performance software for controlling the motion of machines that move, that make things, or that make things move"[7]

GRBL firmware offers great extensibility and configuration, it supports movement on the three axes (X, Y and Z). The plotter used GRBL firmware to convert G-Code instructions into stepper motor steps.

In our case, the GRBL firmware is uploaded on the Arduino Uno, so whenever a G command is sent to it, it stores it into a buffer and intercepts it as stepper motor steps.

Chapter 5 Raspberry Pi and Arduino Communication

First, I will talk about how I opened the Raspberry Pi desktop after I installed the OS on the Micro SD card inserted in the board.

5.1 Remote Connection

In the beginning I used to connect the Raspberry to a screen, mouse, and keyboard to access the desktop directly, but it was inconvenient and exhausting for the board.

I then started connecting to it remotely from my laptop through VNC Viewer which is a virtual machine that lets me access the Pi through its IP address (laptop and Pi must be both connected to the same Wi-Fi network). It was way better than the first way but it was a little slow as it's still a VM.

Lastly I found out about SSH (Secure Shell) which allows you to remotely access and control the Raspberry Pi from another device.

I simply now access the Pi from my laptop using CMD by typing ssh mahmoud@<pi-ip-address>.

5.2 Serial Communication

I have two Arduinos, a Mega that controls the roll stepper motor, the feeder DC motors, and the 4 IR's, and an LCD display. The Raspberry Pi sends commands to it serially and the Mega acts upon these commands.

The Uno controls the stepper motors that control the plotter X and Y movement, and the servo that holds the pen. It has GRBL flashed and it only keeps waiting for G commands.

Baud rate between Pi and Mega is 9600 bps.

Baud rate between Pi and Uno is 115200 bps.

Chapter 6 Components

6.1 The Shooting Stage

In this stage we used these following components:

6.1.1 Raspberry Pi and Raspberry Pi Camera

We used Raspberry Pi 4 Model B as the microcontroller, it hosts the website and has a code written in Python with Flask web framework for defining REST APIs. It has three end points. [6]

1) **/shoot**: captures the image and applies the image processing techniques.
2) **/upload**: uploads the G-code file and converts it to an array.
3) **/start**: starts the process with the given user inputs.

We chose the commercial Raspberry Pi Camera for its easy interface with the board.

Figure 6-1: Raspberry Pi

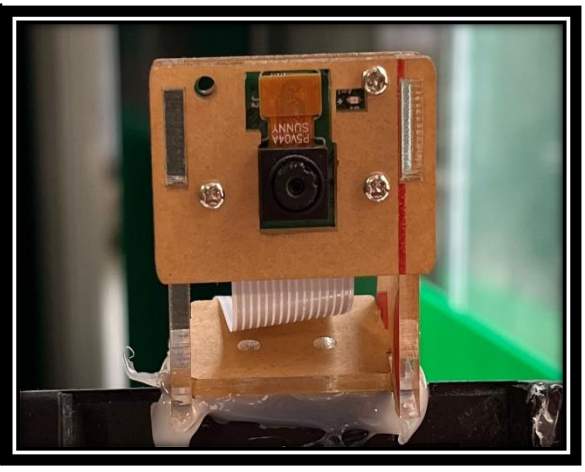

Figure 6-2: Camera

6.1.2 Green Screen and Light Ring

We used a green screen as a background to make it easier to remove it during the processing and isolate the face. A light ring for better lightning was needed also as the brightness plays a huge role in detecting facial features.

Figure 6-3: green screen and light ring

6.2 Plotter

This is the main design of our plotter, it is made from wood, and its 33.2 X 46.8 cm.

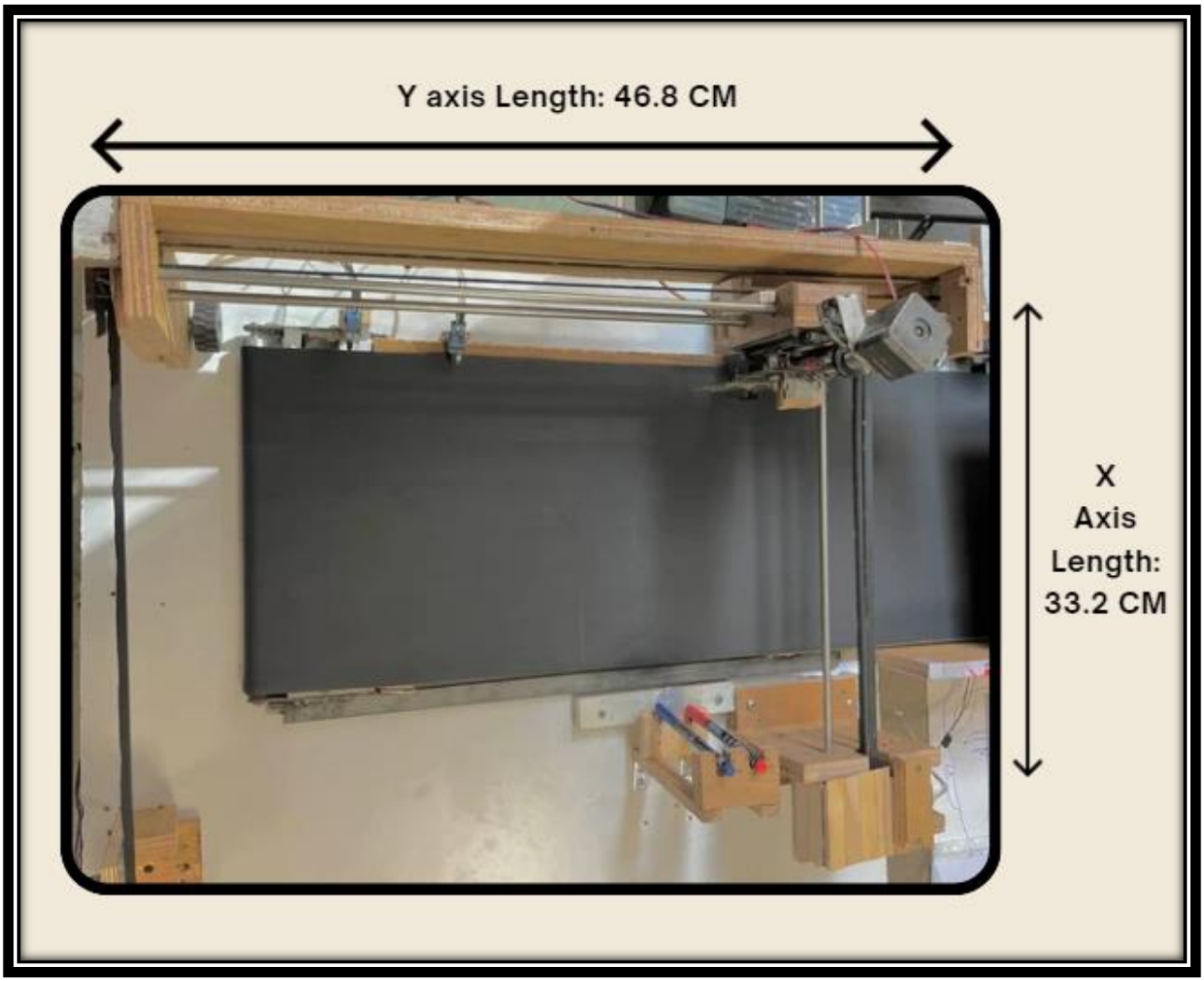

Figure 6-4: plotter top view

6.2.1 XY movements

We used one stepper motor for the Y axis, and two stepper motors for the X axis for better stability and accuracy. A timing belt wrapped around a pulley on top of each stepper motor. And liner ball bearing that enables the pen holder to move smoothly on stainless steel rods.

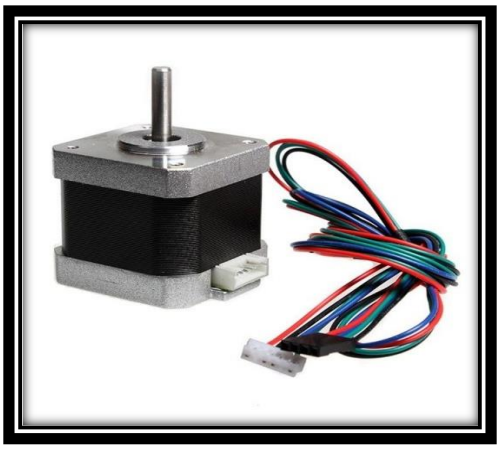

Figure 6-5: Stepper Motor Nema 17

Figure 6-6: Linear Ball Bearing

Figure 6-7: Belt

Figure 6-8: Stainless steel rods

Figure 6-9: Timing Belt Pulley

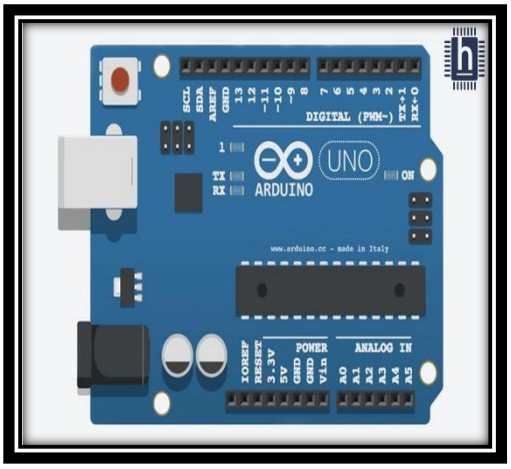

Figure 6-10: Arduino Uno

6.2.2 Pen holder

The pen holder is a metallic component that consists of a stepper motor that rotates a lead screw, and a servo motor for holding the pen.

This module is one of the most crucial modules of our project, we faced many design challenges because of its heavy weight, but managed to solve them in the end.

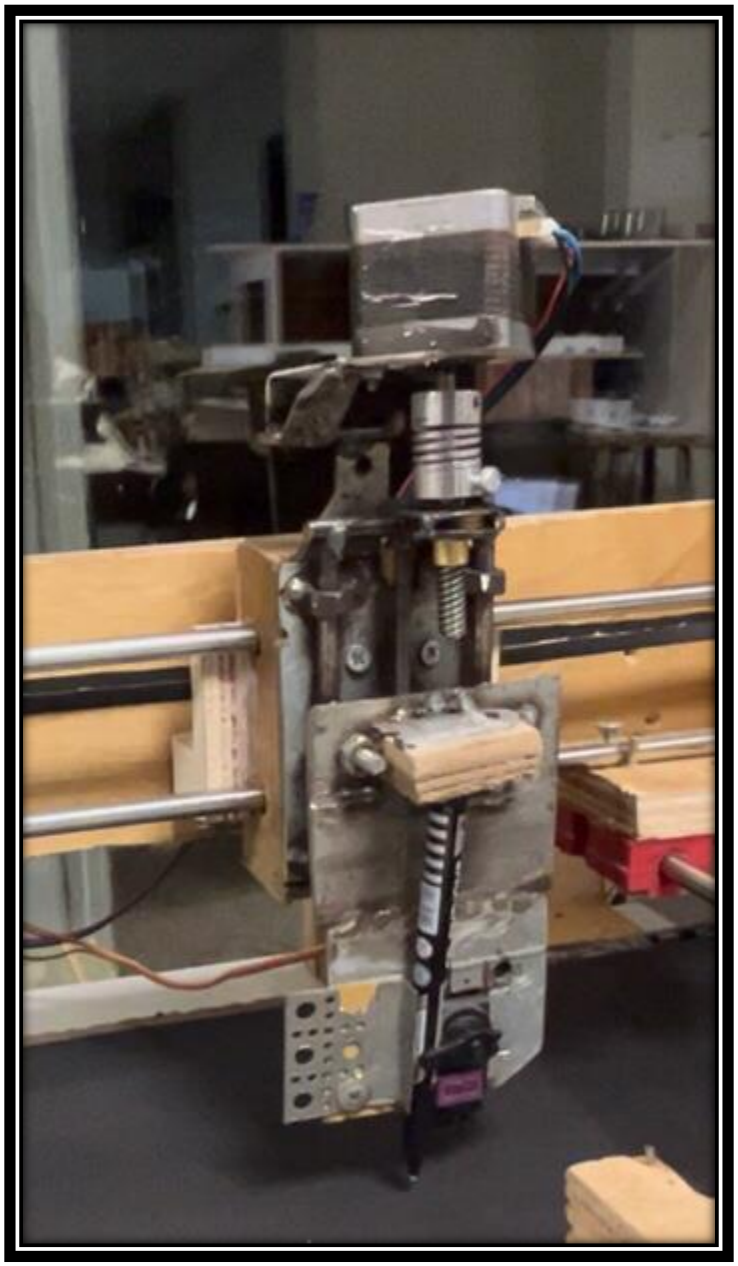

Figure 6-11: pen holder

Figure 6-12: lead screw

Figure 6-13: lead screw nut

Figure 6-14: Servo Motor

6.2.3 Paper Feeders and Paper Roll

We reused old paper feeders from old printers, you can see the feeders in the list of figures.

We used DC motors for rolling the paper, IR sensors to detect the papers, and a stepper motor to rotate the belt on the main roll.

Figure 6-15: DC Motor

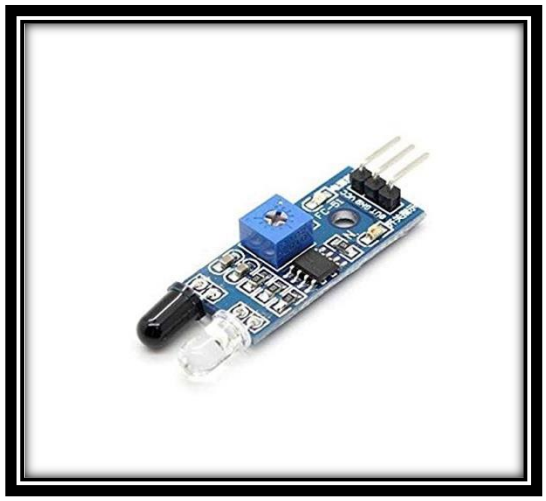

Figure 6-16: IR sensor

6.3 Drivers

For the two X axis stepper motors we used 5A drivers, and for the rest of the stepper motors we used 3.5A drivers, but in both cases we only passed 1.5A because that's how much the stepper needs.

For the DC motors we used H Bridge drivers.

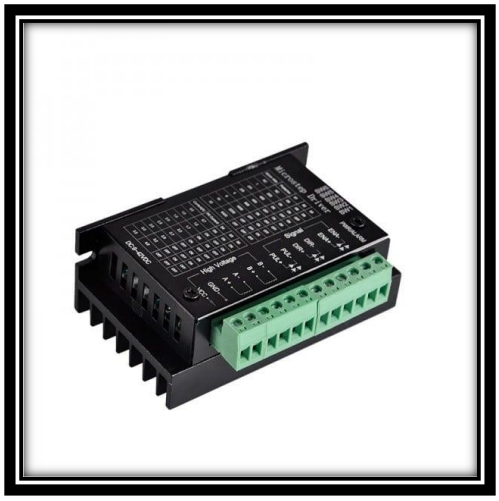

Figure 6-17: 3.5A driver

Figure 6-18: 5A driver

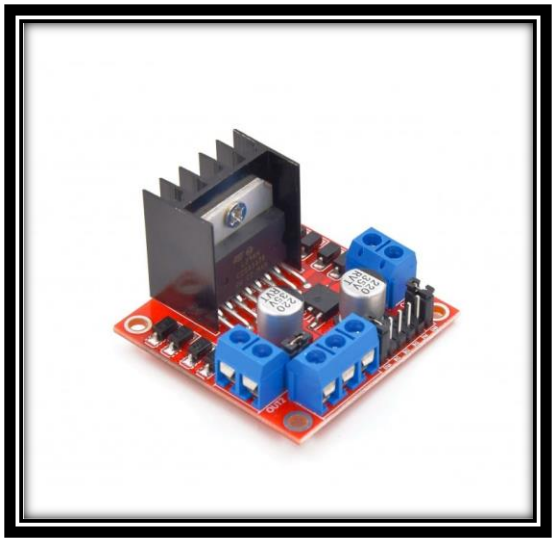

Figure 6-19: H Bridge

6.4 LCD

We used a 2x16 LCD display to show the state of the machine in each stage.

Figure 6-20: LCD

Chapter 7 Results and Analysis

There's a tradeoff between speed and accuracy in our machine, the faster it goes, and the less accurate the drawing is.

We focused more on accuracy rather than speed, it goes a little bit slow but it gives good accuracy.

Here's some of our results:

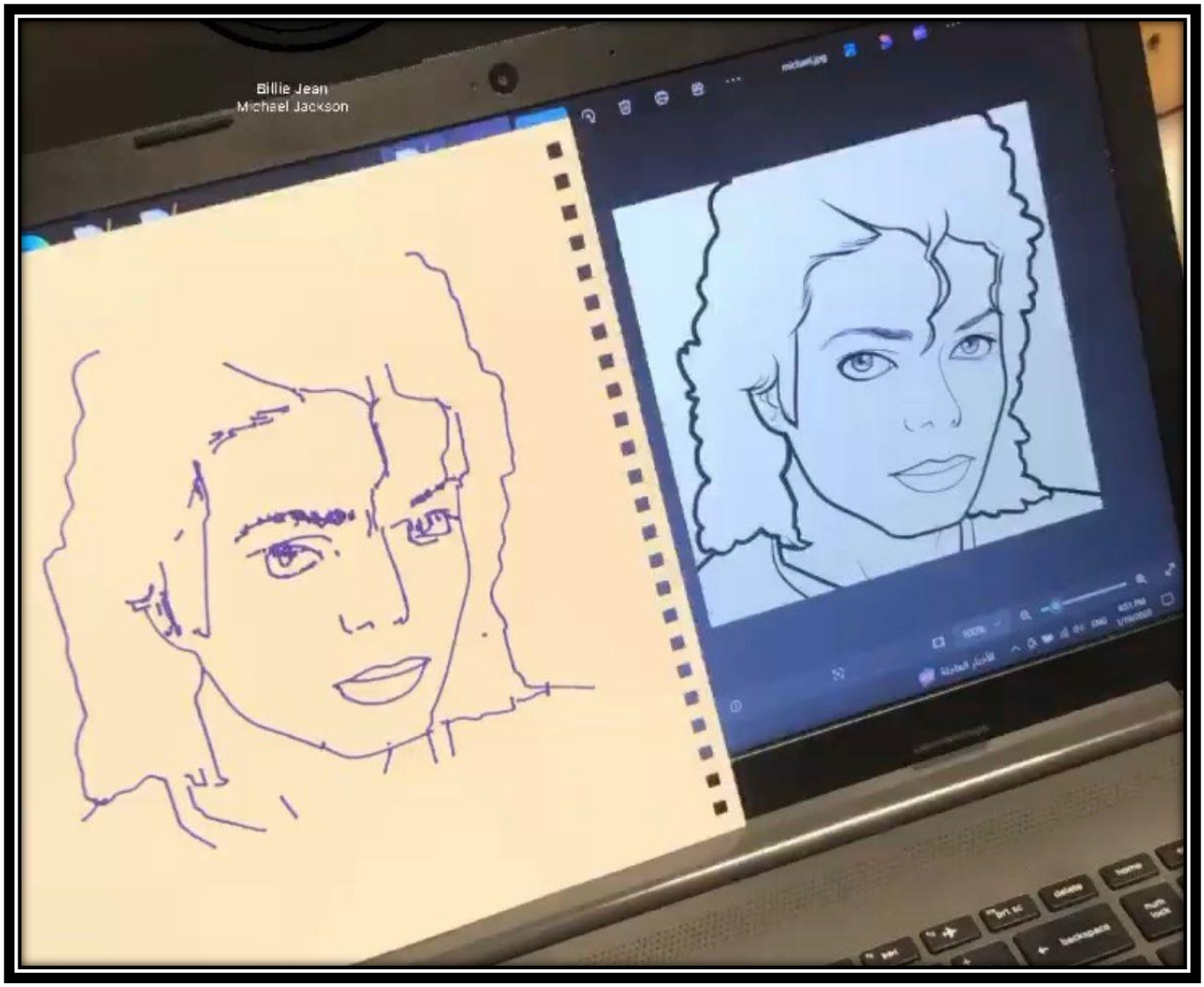

Figure 7-1: Result 1

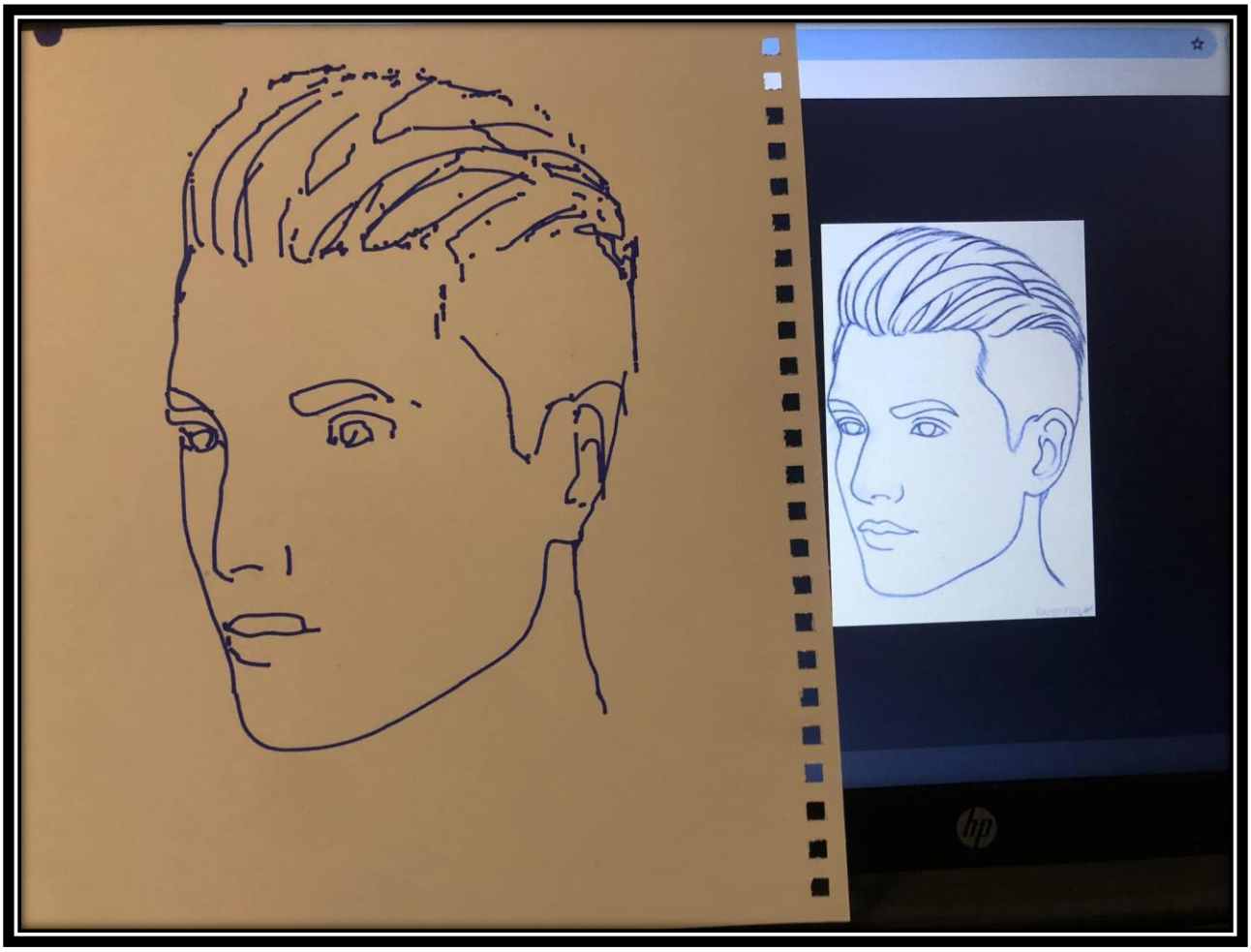

Figure 7-2: Result2

Chapter 8 Conclusion and Future Work

8.1 Conclusion

The Artist is a unique, fun, and complicated system that lets you have a memorable experience where you watch yourself being drawn through a several stages pipeline from camera to final drawing.

8.2 Future Work

- **Colorful drawings :** automatic pen color changes during the drawing process to draw colorful drawings instead of drawing with a single color
- **Framing the drawing:** frame the paper at the end of the production line.

References

[1] J. Green, D. Irkaev, B. Lee, Dejan, Alan, and Norman, Gcode explained: List of most important g-code commands, Oct. 2021. [Online]. Available: https: / / howtomechatronics. Com / tutorials/g- code- explained- list- of- most- important- gcode-commands/.

[2] K.-p. Yang, G. Alkadi, and T. Parker, “Converting SVG to g-code for 3d printers,”

[3] ALL3DB, GRBL Software: Everything you need to know (Online) Available https://all3dp.com/2/grbl-software-guide/

[4] AxiDraw Writing and Drawing Machines (Online) Available https://axidraw.com/

[5] OpenCV, Canny Edge Detection (Online) Available at https://docs.opencv.org/4.x/da/d22/tutorial_py_canny.html

[6] Raspberry Pi documentation (Online) Available https://www.raspberrypi.com/documentation/

[7] Grbl, Home · GRBL/grbl wiki. [Online]. Available: https : / / github.com/grbl/grbl/wiki.